Thesis for the Degree of Doctor of Philosophy

Development of Optimization Algorithms and Framework and Their Applications to Optimum Design of Ship Structures

by

Young-Mo Kong

Department of Mechanical Engineering,

The Graduate School

Pukyong National University

February 2006

Development of Optimization Algorithms and Framework and Their Applications to Optimum Design of Ship Structures
최적화 알고리듬과 프레임웍의 개발 및 선박구조물의 최적설계에의 적용

Advisor: Prof. Bo-Suk Yang

by

Young-Mo Kong

A thesis submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in the Department of Mechanical Engineering,
The Graduate School
Pukyong National University

February 2006

Development of Optimization Algorithms and Framework and Their Applications to Optimum Design of Ship Structures

A dissertation
By
Young-Mo Kong

Approved as to style and content by:

Chairman Prof. Dong-Jo Kim

Member Prof. Sung-Yong Bae

Member Prof. Byeong-Keun Choi

Member Dr. Su-Hyun Choi

Member Prof. Bo-Suk Yang

December 22, 2005

Contents

List of Figures

List of Tables

List of Symbols

Chapt. II Class of Optimization Problems

x	design variables
$f(x)$	objective function
$h_j(x),\ g_j(x)$	inequality and equality constraints
$\alpha_1,\ \alpha_2$	weighting factors
R^n	a set of *n*-dimensional real vector
Z^n	a set of *n*-dimensional integer vector
a	lower boundary of the inter parameter
b	upper boundary of the inter parameter
x_k^L	lower boundary of the parameter x_k
x_k^U	upper boundary of the parameter x_k
$\bar{X}$	all design variable
$\vec{X}$	original point in simplex
X_w	original simplex
X_e	extension simplex
X_c	contraction simplex
X_r	reflection point in simplex
x'	a single offspring
g_i	*i*th generation number in GA

G	total generation number
d_i	reproduced offspring number at the *i*th generation
D	population number
M_i	dynamical mutation at the *i*th generation
$\overline{P}$	population
$\tilde{P}$	single-point crossover
h	step size in R-tabu method
H	set of step size in R-tabu method
r	step number in R-tabu method
P	step ration in R-tabu method

Greek symbols

α_1, α_2	weighting factors
γ	expansion coefficient ($\gamma > 1$)
β	contraction coefficient ($0<\beta<1$)
λ_j	scalars

Chapt. III Development of NASTRAN External Calling Styled Optimization Framework (OPTSHIP)

x^*	a new trial design variable
Δx	increment of a trial design variable
MCR	maximum continuous rating
NCR	normal continuous rating

M_r	relevant moment
M_{MCR}	moment at MCR
N_r	relevant speed of the main engine or propeller
N_{MCR}	main engine speed or propeller speed at MCR
M_H	H-moment of the main engine
M_X	X-moment of the main engine
F_H	Force converted from H-moment of the main engine
F_X	Force converted from X-moment of the main engine
H	height of main engine
L	breadth of main engine
P_r	relevant pressure
P_{MCR}	pressure at MCR
V	velocity response
w	weighting factor based on the ISO 6954
N	number of peak
$V_{q,L}$	vibration velocity amplitude of qth frequency at longitudinal direction
$V_{q,V}$	vibration velocity amplitude of qth frequency at vertical direction
f	interest frequency
f_{upper}	upper limit of the frequency range
f_{lower}	lower limit of the frequency range

Chapt. IV Nonlinear Integer Programming Based on GA Parameter Optimization

N_e	number of evaluation
N_{ea}	number of all evaluation
A_{ne}	number of average evaluation
GAP	GA parameter optimization
GAF	GA for function optimization
M	number of evaluation of GAF for identical GA's parameters
P_s	population size
P_c	crossover probability
P_m	mutation probability
M_s	selection method
M_c	crossover method
$x_i^{(best)}$	best solution at each generation
$x_i^{(opt)}$	optimum solution of the design variable
Δx_i	interval of design variables
W	web size
G	girder size
L_w	web length of girder and web
W_1	weight of compass deck
R_1	maximum vibration velocity response
W_0	initial weight
R_0	basis vibration velocity response

Greek symbols

α_i, β_i	coefficient of test function
ω_n	natural frequency

α, β	weighting factors

Chapt. V RSM-based Hybrid Evolutionary Algorithm

x	coded variables
y	response
f	true response function
X	a set of N patterns
X^+, X^-	two classes of X
R^p	p-dimensional input space
R^1	single dimensional output space
$\|\| . \|\|$	Euclidean norm
d	desired response vector
w	linear weight vector

Greek symbols

ξ	controllable input variables
ε	statistical error
σ^2	variance
η	response function in terms of the coded variables
β	unknown parameter in model
φ	interpolation matrix

Development of Optimization Algorithms and Framework and Their Applications to Optimum Design of Ship Structures

Young-Mo Kong

Department of Mechanical Engineering, The Graduate School,
Pukyong National University

Abstract

Recently, the vibration problems in ships have received much attention. The vibration criteria given by the ship owners are constantly getting higher, not only for the consideration of ship durability but for the rise of crew's working environment. Especially these days, the use of higher output engines and lighter structure has caused more intense vibration problems. However, the excitation source of a ship is very complicated and vast, including the main engine, auxiliary engine, propeller, wave and fluid movement, etc. Therefore it is impossible to build a ship without vibration. However, systemization of experiences in the main causes of vibration and standardization of anti-vibration operation will help us in bringing down the cost in managing the vibration problems.

In addition, the need for optimization is especially more emphasized since the rise in personnel expenses and materials have definitely called on the importance of doing so. The optimum design of a ship is a very complicated work. Especially, large ships like commercial ships have many degrees of freedom, which make it even more complicated. Therefore researchers came up with the method of scaling down the model, and sought to handle the problem with a

variety of optimizing techniques for the validity of optimizing technology. These efforts have worked efficiently in developing new designing technologies and optimization methods. Optimum design is about pursuing the best out of using the minimum materials in making the structure while guaranteeing its safety and efficiency, deciding on the structure's geometric shape and size.

Three programs related to optimization have been developed in this thesis.

Firstly, we have developed "NASTRAN external calling styled optimization framework" that can work efficiently in optimizing large structures like ships and it can also choose various objective functions and design variables.

We call it "OPTSHIP”. The program uses NASTRAN as a solver for reaching the structure's own natural frequency, mode vector and forced response, and for figuring out global optimal solution, users can make an optimization algorithm by making a module. At present usable optimization modules are genetic algorithm (GA), random tabu search (R-tabu), simulated annealing (SA) and artificial life (AL) optimization algorithms. In addition, to check the efficiency of the proposed framework, it has been applied to the 2400 TEU containership to reduce the vibration level of the deckhouse which is placed in the rear of the ship. And to gain efficiency in analysis, sensitivity analysis of candidate design parameter with respect to objective function has been done and a design parameter with a sensitivity value of over 1.5 has been chosen for the final design parameter, in order to reduce the calculating time. In this thesis, R-tabu method is used as an external optimization module, and the result is compared with NASTRAN's optimum result to prove its excellency.

Secondly, this research developed a non-linear integer optimization method in order to directly use an optimized design parameter value in the initial stage of design process. The design parameter used in this thesis has been programmed to freely choose the thickness of the steel plates and the size of stiffeners used in shipyards. The proposed algorithm is the upgraded version of GA that can categorize design parameters into their size and express genotype. In addition, concerning the fact that differences exist in the calculating time and accuracy of optimal solution in accordance to parameters such as individual size, crossover probability, mutation probability, selection method and crossover method of GA, optimization of these parameters has been done with priority in calculating adequate values. To gain legitimacy of this optimization, the optimized parameter value of the structure in question has been tested and it was found that convergent speed to global solution was much better than those of other parameters. In addition, we have proven the validity of the developed method by applying this research's non-linear optimization algorithm in solving the vibration problem of the compass deck of a ship in danger from a vibration problem.

Lastly, to improve the convergent speed and the variety of solutions at the same time, a new hybrid evolutionary algorithm (RHEA: RSM-based hybrid evolutionary algorithm) was suggested, which combined the excellent qualities of GA, tabu list, response surface methodology (RSM), etc. The tabu list from tabu search in GA is used to secure systematic diversity in solutions. As RSM, we have used data acquired from the process of GA, made approximate function of actual objective function, found the optimum point, and generated the minor (one) individual of the next generation. Going through this process enables the shortening down of the vast convergent time of a big structure being optimized. This research tests the validity of the newly developed algorithm in comparison with the general GA algorithm by using some traditional test functions.

In addition, it applies that the algorithm suggested in designing for avoiding the resonance of the fresh water tank which is placed in the rear of the ship, showing that it efficiently searches for global optimum solution.

According to the results, the proposed new hybrid algorithm (RHEA) is a very powerful global optimization algorithm from the view point of convergent speed and global search ability.

I. Introduction

1. Background

It was during the end of 19th century when the vibration problems of ships received much attention. From the beginning of the 19th century the steam reciprocating engine and spiral propeller was put into practical use, and the structure of ships were beginning to be made of steel, ships started to get larger, faster, and the output of ships became higher during the end of 19th century.

The main causes of vibration problems in rear and deckhouse of ships, the underneath deck of machinery in engine room and various kind of tanks are; guided moment of diesel engine, fluctuation pressure due to the propeller and thrust bearing force due to the coupled vibration between torsional and axial vibration of propulsion shafting system. In essence, the causes of vibration in ships are various, including; main engine, auxiliary engine, propeller, wave, fluid movement. The structure and machineries of ships are various and complicated, so it is almost impossible to make a ship free of vibration [1].

Recently, there exist stability problems in the deck house and local area of ship due to the light-weight structure not being concerned with the vibration problems. Especially, vibration problems in the limited areas of ships are frequently happening due to the various light-weight structures such as oil tanks, fresh water tanks, the compass deck and each bulkhead. Moreover, the deckhouse of the ship has been reducing living space due to crew cutdown brought in by navigation automation, and the strengthened visibility regulations which have forced its height to go higher though it’s length and width being reduced, causing stronger vibration through resonance between the deckhouse and the main excitation source of ship because the natural frequencies of longitudinal and the transverse of the deckhouse having been reduced [2-6].

On the other hand, with the revision of ISO 6954 [7, 8] which is the regulation regarding vibration limit in the deckhouse of ships, ship owners are beginning to claim for severe vibration values for better living environments in ships. Moreover, vibration problems in the deckhouse and local structures found after it is built not only require huge expenses and time for remedy, but ship builders are losing faith in ship owners, and cause hardships in business for ship builders.

Therefore anti-vibration design at the beginning stage of design is most effective. A superstructure anti-vibration solution being applied in shipyards apply beam analogy or empirical formulation based on measurement results of sister ships in the beginning stage of design, estimate natural frequency of the superstructure by using relatively simple analysis [1, 9-11], apply avoiding resonance designing method with main excitation sources such as the main engine and propeller, and apply allowable vibration response designing method based on forced vibration analysis by a 3D finite element method. And for the solution of local vibration problems of ships, anti-vibration designing method by local vibration analysis programs based on beam theory possessed by every shipyard or 3D finite element method using the commercial program NASTRAN is used. In addition, for the anti-vibration designing method, anti-vibration standardization that uses optimization and experiences from vibration problems is efficiently used [12]. Moreover, optimization is constantly being emphasized for its importance due to the rise of personnel and material expenses in industrial fields.

Optimization is utilized to determine the size or the geometric shape of the structure to obtain the maximum performance using minimal material with safety and availability of the target structure [13]. From a mathematical point of view, the optimization is to get design variables which are maximizing or minimizing a desired objective function while satisfying the prevailing constraints. Usually, optimization needs a lot of time to get the desired information due to this repetitive process. Recently, optimization has been widely applied for decreasing the weight of structures in various industrial fields such as aerospace, civil, mechanical engineering, etc. through integrating methodology of engineering design with the technology of computer-aided engineering (CAE) and increased computer speed.

These are partly being applied to shipyards in which possess world class technology and a vast number of orders but the scope of application is quite small and is limited to the field of structure design. Therefore optimum design with consideration for dynamic factors is in definite need. And especially, the need for appropriate programs and optimum method for large scale structures like ships is definitely required.

2. Objective of Study

In general, commercial software such as NASTRAN or ANSYS is used in analyzing vibration and structural characteristics of large structures. Especially, NASTRAN offers an optimum module which is based on sensitivity analysis but it has many limitations in setting of objective functions and design variables. It is also very hard to acquire a global optimum solution, because it is a local search method. Its demerit also includes the impossibility in combining the complicated user-defined optimum technique.

Therefore, the first purpose of this research is in developing NASTRAN external calling styled optimization framework, with commonly used commercial program NASTRAN as a solver. So, it used the global optimization algorithm such as genetic algorithm (GA), random tabu search method (R-tabu method), simulated annealing (SA) method, and artificial life (AL) optimization algorithm to prevent local convergence of general optimization algorithm. This program not only enables users to choose basic objective functions such as minimization of forced response, avoidance of resonance, etc. but also enables users to set up their own complicated objective functions and choose a variety of design variables.

The second purpose of this research is in developing a non-linear integer optimization method. It is hard to directly apply real variable optimum results to fields that use standardized member just like actual shipyards. Generally, a value one step higher than that of optimized result is used in shipyards for extra safety. However, this method enables a stronger design in the structural point of view, but in view of vibration, it goes closer to resonance and can actually be quite dangerous. Therefore there is need for making usable member in the field into table, developing a non-linear integer program that chooses optimal size from the table and directly uses the optimal result values at the initial stage of design.

Optimum design requires constant repetition of work, so it takes a lot of time to gain useful information. The development of CAE and the fast processing speed of computers have enabled users to save a lot of time, but still remains as a difficult problem to solve in optimum design. To solve this problem, recently a number of researchers have suggested various hybrid genetic algorithm that is combined GA with the merits of other algorithms. Because GA holds many merits such as its ability to search for an optimum solution without any background knowledge of the search space and its characteristics of not being influenced from an initial search starting position. The third purpose of this research is to do a more intensive search on the optimum solution of multi-peak function, more rapidly and accurately. To do so, a new hybrid evolutionary algorithm (RHEA: RSM-based hybrid evolutionary_algorithm) was suggested, which combined the merits of GA, tabu list and response surface methodology (RSM). The mutation of GA offers random variety, but systematic variety can be achieved through the use of tabu list of tabu search method.

For large structure optimization it takes a lot of calculating time for one evaluation of objective function. Therefore, it is an important matter to bring down the evaluation number of objective function by using all the information attainable. From this point of view, GA's convergent speed can be improved by using RSM method which uses the information on the objective function acquired through GA process and then making response surface (approximate function) and optimizing this. Optimized solution was calculated without the evaluation of additional actual objective function, and the GA’s convergent speed could be improved.

3. Outline of Thesis

This research is comprised of 6 chapters. Except for the current introductory chapter, the rest of the chapters are summarized as follows.

Chapter Ⅱ contains overall contents of optimization. The first half introduces general information on optimization, definition of it in the engineering point of view, and its history. The second half contains a variety of optimization methods. There are several ways to classify optimization, but this research explains on the local optimization and the global optimization. Especially, specific explanations on the main algorithms such as modified method of feasible direction (MMFD) and GA are given. The former is used in NASTRAN optimization module and uses usable-feasible search direction as searching way and the latter is used as a representing global optimization algorithm. The rest are concisely summarized.

Chapter Ⅲ suggests a new optimization framework. For an optimum design of large-scale structures like ships, NASTRAN which is widely used in industries is used. However, this optimization method is limited to a local search. When the searching environment is multi-peak, accurate solution is hard to acquire. Using a global searching algorithm to analyze complicated structures is also difficult because it's hard to work on an analysis model and programming, not to mention difficult to develop to the level of commonly used programs. Therefore, this chapter uses NASTRAN for solving the problem of a structure's natural frequency, forced response and mode vector. For global optimization, an optimization framework in which the user is capable of using a module made externally is introduced. In order to prove the validity of the program suggested, this chapter has applied it to deckhouse vibration minimization of 2400 TEU containership which has a possibility of vibration problem since the deckhouse area is placed in the rear of the ship. In this thesis, R-tabu method is used as an external optimization module, and the result is compared with NASTRAN's optimum result to prove its excellency.

Chapter Ⅳ suggests a non-linear integer optimization method. Real variable optimization method cannot be directly applied in the design stage because in shipyards, the thickness of steel plates and size of stiffeners except some built-up stiffener are mostly standardized. In order to solve this problem, this chapter has extended real variables optimization problem to non-linear integer optimization algorithm and have applied it. Since the accuracy of the optimum solution and calculating time of GA which is used as an optimizer in this optimization are largely influenced by initial parameter values such as the size of individuals, crossover probability, mutation probability, selection method and crossover method, this chapter has proceeded with optimization for GA parameters. Optimized GA parameter is applied to structure in question and is illustrated as the optimum value. In addition, by using the suggested non-linear integer optimization algorithm, we have proceeded with optimization to a compass deck structure that actually is in danger of vibration problems and the problem is solved.

By doing so, we have proven the suggested method's validity and efficiency.

Chapter Ⅴ suggests a new hybrid evolutionary algorithm that combined the merits of the popular programs such as genetic algorithm, tabu search method, response surface methodology and simplex method to search for an optimum solution of multi-peak function in high accuracy and high speed. This algorithm, in order to improve the convergent speed that is thought to be the demerit of GA, uses RSM and simplex method. Though mutation of GA offers random variety, systematic variety can be secured through the use of tabu list. Especially, in the initial stages, GA's convergent speed can be improved by using RSM method which uses the information on the objective function acquired through GA process and then making response surface (approximate function) and optimizing this. Optimized solution was calculated without the evaluation of additional actual objective function, and the GA's convergent speed could be improved. Efficiency of this method has been proven by applying traditional test functions and comparing the results to GA. It also proved that the newly suggested algorithm can effectively find the global optimum solution by applying it to weight minimization of the fresh water tank that is placed in the rear of the ship designed to avoid the resonance.

Finally, it is concluded that the newly suggested algorithm (RHEA) is a very powerful global optimization algorithm from the view point of convergent speed and global search ability.

Chapter VI summaries and discusses the results obtained in this thesis.

4. Contribution of This Work

Executing optimum design utilizing the results from this thesis, the following contributions can be made.

4.1 NASTRAN external calling styled independent optimizing framework (OPTSHIP)

1) Commercial software is used for vibration analysis of large structures. PATRAN is used for both modeling and pre/post processing and NASTRAN as a solver. Although there are optimization modules for existing general-purpose programs, as used a local search method, in the case of being multi-peak function, finding the global solution is difficult. Therefore, by utilizing the OPTSHIP suggested in this thesis, with an already completed model, optimum global solution of complex structures like the ship who is normally multi-peak function can be easily found by connecting it with exterior optimization modules.
2) When using exterior optimizing modules, familiarizing the difficult internal general-purpose programming language is not necessary since it is possible for the user to comfortably apply not only global algorithms but also various optimization techniques. This also enables the user to increase the accuracy and the convergent speed of solution to all fields by using self-developed hybrid algorithm.
3) Various objective functions and design variables are adjustable.
4) OPTSHIP is not only executable in ships but can also be extended and applied on all complex structures of such matters.

4.2 Nonlinear Integer Program

1) Unlike real variables programs, it is possible to directly apply optimized results to actual designs.
2) A usable design variable is tabled and then used, enabling the user to extend or reduce this according to need.
3) Disabling of the accuracy and convergent speed of GA depending on the

initial parameters is compensated by developing a GA parameter optimization program, allowing the user to initiate parameters with ease.

4) This can be used as an exterior module for OPTSHIP.

4.3 RSM-based Hybrid Evolutionary Algorithm (RHEA)

1) Amendments have been made in the convergent speed of GA thus improving the convergent speed to reach the global solution.
2) Unlike the combination of the existing global search and local search algorithm, a new region on a new hybrid GA is attempted.
3) This can be used as an exterior module for OPTSHIP.

References

[1] K.C. Kim, B.H. Kim, S.S. Kim, Y.H. Kim, C.H. Shin, K.Y. Jeong, T.Y. Chung and S.Y. Han, 1997, *Control of Ship Vibration and Noise,* Korean Register of Shipping. (in Korean)

[2] T.Y. Chung, 2002, “The presentation of ship vibration technology,” Transactions of the Korean Society for Noise and Vibration Engineering, Vol. 12, pp.14-17. (in Korean)

[3] J.H. Park and K.Y. Chung, 2002, “Vibration characteristic and a reduction countermeasure of Containership,” Transactions of the Korean Society for Noise and Vibration Engineering, Vol. 12, pp.18-26. (in Korean)

[4] S.H. Choi, 2002, “Vibration characteristic and an anti-vibration countermeasure of VLCC and bulk carrier,” Transactions of the Korean Society for Noise and Vibration Engineering, Vol. 12, pp.27-34. (in Korean)

[5] J.K. Eom and S.Y. Han, 2002, “Vibration and an anti-vibration countermeasure of a passenger ship,” Transactions of the Korean Society for Noise and Vibration Engineering, Vol. 12, pp.35-43. (in Korean)

[6] M.S. Kim, 2002, “Vibration characteristic of LNG carrier,” Transactions of the Korean Society for Noise and Vibration Engineering, Vol. 12, pp.44-50

[7] ISO 6954, 2000, “Guidelines for the measurement, reporting and evaluation of vibration in merchant ships,” International Organization for Standardization.

[8] S. M. Lee, W. H. Kim and J. G. Bae, 2004, “Evaluation of shipboard vibration under revised ISO 6954 standard,” in Proc. of the Annual Autumn Meeting SNAK, Sancheong, 20-22 October, pp.635-639. (in Korean)

[9] VERITEC, 1985, *Vibration Control in Ships*, VERITEC Marine Technology.

[10] NIPPON KAIJI KYOKAI, 1981, *Guide to Ship Vibration*, NKK.

[11] T. Hirowatari and K.Matsumoto, 1969, “On the fore and aft vibration of

superstructure located at aftship (2nd report)," Journal of SNA Japan, Vol. 125.

[12] Y. M. Kong and S. H. Choi, 2003, "A study on standard structural design of radar mast for low vibration," Transactions of The Korea Ship and Offshore Structure Congress, Vol. 17, No.19, pp. 194-202. (in Korean)

[13] J. S. Arora, *Introduction to Optimum Design*, McGraw-Hill, Inc, 1989

II. Class of Optimization Problems

1. Introduction

Designer, in industrial fields, supplies design parameters for the product as input into the computer simulation programs which is developed by commercial vender, runs the program and then analyzes the results. If the results do not meet the design goals then the designer changes the design parameters and repeats the process. Solutions to their problems have been based mostly on judgment and experience. However, increased competition and consumer demands often require that the solutions be optimum and not just feasible solutions. The challenge to the designer is to find the best design in as short a time period as possible. It can be realized to the designer through the optimization.

Optimization is the process of maximizing or minimizing a desired objective function while satisfying the prevailing constraints. A small savings in a mass-produced part will result in substantial savings for the corporation. In ship, weight minimization can be contributed on cost reduction, ship's performance, safety, free from the repeated work, etc. Limited material or labor resources must be utilized to maximize profit [1].

In order for engineers to apply optimization at their workplace, they must have an understanding of the theory, algorithms, and techniques. This is because practical problems invariably require tuning algorithmic parameters, scaling, and even modifying existing techniques to suit the specific application. Moreover, the user may have to try out several optimization methods to find one that can be successfully applied. In operations research and industrial engineering, use of optimization techniques in manufacturing, production and scheduling has resulted in considerable savings for a wide range of business and industries. The importance of minimum weight design of structures was first recognized by the aerospace industry where aircraft structural designs are often controlled more by weight than by cost considerations. In other industries dealing with civil, mechanical and automotive engineering systems, cost may be the primary consideration although the weight of the system does affect its cost and performance. A growing realization of the scarcity of raw materials and a rapid depletion of our conventional energy sources is being translated into a demand for lightweight, efficient and low cost structures. This demand in turn emphasizes the need for engineers to be cognizant of techniques for weight and cost optimization of structures. This chapter, in its first part, introduces the general area of optimization. The definition and history of optimization is considered from the viewpoint of engineering. In the latter part, looks at various optimization methods. There are many classes into which we may partition optimization problems, and also many competing algorithms which have been developed for their solution. We outline the areas of primary interest for the present work, with some brief contrasts to classes of problems that we do not consider here. In particular, modified method of feasible direction (MMFD) and a genetic algorithm was illustrated in detail. The former is the one of the algorithms to solve the approximate optimization problem. It used the usable-feasible search direction in search direction and is used NASTRAN optimization module. The latter is simulated a heuristic probabilistic search technique that is analogous to the biological evolutionary process. This algorithm is applied to the newly developed algorithms in this study.

2. Historical Sketch [2]

The existence of optimization methods can be traced to the days of Newton, Lagrange and Cauchy. The development of differential calculus methods of optimization was possible because of the contributions of Newton and Leibnits to calculus. The use of a gradient method (requiring derivatives of the function) for minimization was first presented by Cauchy in 1847. He made the first application of the steepest descent method to solve unconstrained minimization problems. In spite of these early contributions, very little progress was made until the middle of the twentieth, when high-speed digital computers made the implementation of the optimization procedures possible and stimulated further research on new methods. Modern optimization methods were pioneered by Courant's on penalty functions in 1943, Dantzig developed the simplex method for linear programming in 1947 and Bellman stated the principle of optimal policy for system optimization for dynamic programming problems paved the way for development of the methods of constrained optimization in 1939 and Kuhn, and Tucker who derived the "KKT(Karush, Kuhn and Tucker)" optimality conditions for constrained problems laid the foundations for a great deal of later research in non-liner programming in 1951. Fletcher and Reeves of the conjugate gradient methods pioneered on unconstrained minimization. Constrained optimization methods were pioneered by Rosen's gradient projection method and Fiacco and McCormick's SUMT techniques in 1968. Geometric programming was developed by Duffin, Zener and Peterson. Gomory did pioneering work in integer programming, which is one of the most exciting and rapidly developing areas of optimization. Dantzig, Charnes and Cooper developed stochastic programming techniques and solved problems by assuming design parameters to be independent and normally distributed. In the 1960's, also, there were developments in non–gradient or 'direct' methods, principally Rosenbrock's method of orthogonal directions in 1960, the pattern search method of Hooke and Jeeves in 1961, Powell's method of conjugate directions in 1964, the simplex method of Nelder and Meade [3]. Sequential quadratic programming (SQP) methods for constrained minimization were developed in the 1970's. Development of interior methods for linear programming started with the work of Karmarkar in 1984. Most recent among direct methods are genetic algorithms (Holland [4], Goldberg [5]), Tabu search algorithm which was developed independently by Glover [6, 7] and Hansen [8] for solving combinatorial optimization problems and simulated annealing algorithms was derived from an analogy with the annealing process of material physics by Kirkpatric [9]. Special methods that exploit some particular structure of a problem were also developed. Pareto optimality was developed in the context of multi-objective optimization. The use of nonlinear optimization techniques in structural design was pioneered by Schmit in 1960. Today, applications are everywhere, from identifying structures of protein molecules to decreasing the weight of ship structures.

3. Definition of Optimization Problem

The design optimization problems are commonly found in manufacturing industries and can be represented by the following mathematically formulation.

find $\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in R^n$

which maximize or minimize $f(\boldsymbol{x})$

subject to the constraints

$g_j(\boldsymbol{x}) \leq 0$, $h_j(\boldsymbol{x}) = 0$, $j = 1$ to m

This formulation supports the specification of unconstrained and constrained problems with a single objective. Where n is the dimension of variable and m is the total number of the constraint condition(or function). $\boldsymbol{x}$ is a real or integer vector of n dimension. $f(\boldsymbol{x})$ is an objective function or a cost function. $g_j(\boldsymbol{x}) \le 0$ and $h_j(\boldsymbol{x}) = 0$ are an inequality and an equality constraints, respectively. If $\boldsymbol{x}$ satisfies $g_j(\boldsymbol{x}) \le 0$ and $h_j(\boldsymbol{x}) = 0$, $\boldsymbol{x}$ is called a feasible solution and lies in feasible. In an opposite situation, $\boldsymbol{x}$ is an infeasible solution.

In the optimization problem formulation, three elements are considered such as design variables, constraints and an objective function. Also some terminology is introduced.

3.1 Design Variables

The idea of improving or optimizing a structure implicitly presupposes some freedom to change the structure. The potential for change is typically expressed in terms of ranges of permissible changes of a group of parameters. Such parameters are usually called design variables in structural optimization terminology and denoted by a vector $\boldsymbol{x} = (x_1,\ x_2, \cdots,\ x_n) \in R^n$. Design variables can be cross-sectional dimensions or member sizes, they can be parameters controlling the geometry of the structure, its material properties, etc. Design variables may take continuous or discrete values. Continuous design variables have a range of variation, and can take any value in that range. Discrete design variables can make only isolated values, typically from a list of permissible values. Material design variables are often discrete. Design variables that are commonly treated as continuous are often made discrete due to manufacturing considerations. For example, if the beam is designed to minimize weight, then we may need to limit ourselves to commercially available cross sections. So we have to solve the problem with discrete variables. This is done by employing integer (discrete) programming. The choice of design variables can be critical to the success of the optimization process. In particular it is important to make sure that the choice of design variables is consistent with the analysis model.

3.2 Constraints

Constraints introduce the notion of limits on the design variables in the optimization problem formulation. Because of their simplicity, these upper and lower limit constraints on the values of the design variables are often treated in a special way by solution procedures, and are refereed to as side constraints. Constraints which impose upper or lower limits on quantities are by their very nature inequality constraints. Sometimes we need equality constraints. However, some strategies for the solution of nonlinear optimization problems are unable to handle equality constraints, but are limited to inequality constraints only. In such instances it is possible to replace the equality constraint with two inequality constraints that form upper and lower bound constraints with a same limiting value. However, it is usually undesirable to increase the number of constraints.

3.3 Objective Function

The objective function, when expressed as a function of the design variables, is known to the criterion with respect to which the design is optimized. The choice of objective function is governed by the nature of the problem. For structural optimization problems, weight, displacement, stresses, vibration frequencies, buckling loads, and cost or any combination of these can be used as objective functions. In some situations, there may be more than one criterion to be satisfied simultaneously. An optimization problem involving multiple objective functions is known as a multi-objective programming problem. With multiple objectives there arises a possibility of conflict, and one simple way to handle the problem is to construct an overall objective function as a linear combination of the conflicting multiple objective function. Thus if $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ denote two objective functions, construct a new (overall) objective function for optimization as

$$f(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) \tag{2.1}$$

where α_1 and α_2 are constants whose values indicate the relative importance of one objective function relative to the other.

3.4 Smoothness

Functions for which continuous derivatives of sufficiently high order exist are referred to as smooth. For continuous optimization, we are usually interested in having continuous derivatives up to and including second order. Minimization problems in which the objective and constraint functions are of such type can make use of techniques of multivariable differential calculus which are unavailable for non-smooth functions. We refer primarily to the extensive set of methods which make use of gradient and curvature information to direct an iterative search process toward a local minimum. Methods in this very broad class include the Newton or quasi-Newton methods. For functions which are not smooth, only function value information can be used to direct the search process. Such techniques are referred to generally as direct search. One early approach is the simplex method of Nelder and Mead [3].

3.5 Discrete Optimization

We discuss integer programming (both linear and nonlinear) later in this thesis, however the apparently more general problem of nonlinear optimization subject to general discrete restrictions has also received some recent attention. Such problems require a (generally nonlinear) objective to be minimized subject to nonlinear inequality constraints, with the added requirement that certain or all of the structural variables must take values from specified finite sets; the elements of these sets need not be integers. For a recent example in which the classical penalty function approach (Sequential Unconstrained Minimization Technique (SUMT) of Fiacco and McCormick [10]) is applied in order to satisfy both nonlinear constraints and the discrete requirements, see the 1990 paper by Shin, Güerdal and Griffin [11], in which applications to engineering truss design are considered.

3.6 Nonlinear Integer Programming (NIP)

Nonlinear integer programming was suggested by Reiter and Rice for solving a general quadratic programming problem, where both the objective and constraint function are quadratic. They applied a modified gradient-type method, very similar to the methods used in the continuous nonlinear programming field, to solve the problem. NLIP is an intrinsically hard problem. As with most domains of engineering, nonlinear Therefore, nonlinear problems are often solved by generating a sequence of solution to linear problems which in some sense approximate the original nonlinear problem. The NIP problem can be mathematically expressed as follows:

Maximize (or minimize) $f(\boldsymbol{x})$

subject to the constraints $\boldsymbol{x}^{\boldsymbol{L}} \le \boldsymbol{x} \le \boldsymbol{x}^{\boldsymbol{U}}$,

$\boldsymbol{x} \in Z^n$,

where, $\boldsymbol{x} = (x_1, x_{2,} \cdots, x_n)^T$ is a vector of variables or unknown in the NIP problem, Z_n is a set of n-dimensional integer vector, $\boldsymbol{x}^{\boldsymbol{L}} = (x_1^L, x_2^L, \cdots, x_n^L)^T \in Z^n$ are $\boldsymbol{x}^{\boldsymbol{U}} = (x_1^U, x_2^U, \cdots, x_n^U)^T \in Z^n$ are n-dimensional constant vectors, and $x^L \le x^U$.

Let $S = \left\{ \boldsymbol{x} : \boldsymbol{x}^{\boldsymbol{L}} \le \boldsymbol{x} \le \boldsymbol{x}^{\boldsymbol{U}}, \boldsymbol{x} \in Z^n \right\}$ denote a solution space, thus $f : S \to R$ is a cost function. Some of the NIP problems can also be viewed as integer and combinatorial optimization problem [12].

3.7 Linear Programming (LP)

Linear programming (LP) problems are linear in both objective and constraints. The special nature of this class of problems makes possible a very elegant solution algorithm known as the revised simplex method. The basic result of LP theory stems from the nature of the feasible set. The feasible set can be characterised geometrically as a convex polytope (or simplex), which can be imagined to be a n-dimensional polyhedron, and if an optimal solution exists, then there is at least

one vertex of the feasible set that is optimal. Fig. 2.1 illustrates a trivial LP in which the interior of the shaded quadrilateral OPQR represents the feasible set. The fundamental result tells us that if a finite optimal point exists, then (at least) one of the vertices O, P, Q and R (corresponding to so-called basic feasible solutions) is optimal.

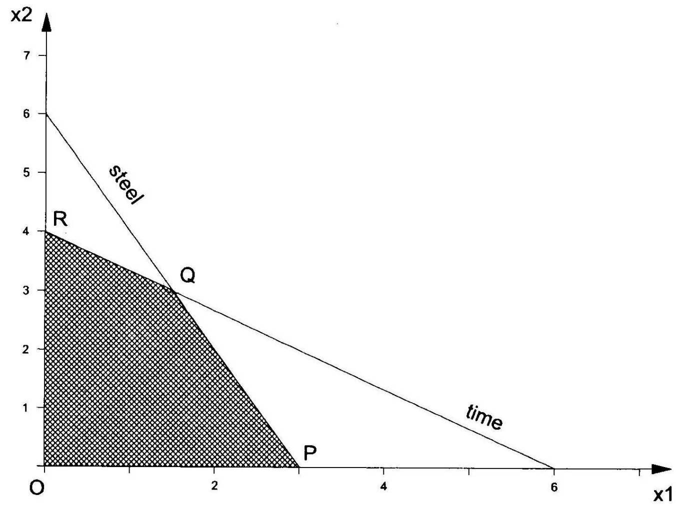


Fig. 2.1 Linear programming

3.8 Integer Linear Programming

Integer linear programming problems (ILPs) are LP problems in which extra constraints requiring some or all variables to be integer valued have been imposed.

ILP is a very common problem class where variables representing indivisible units, eg men, machines do not admit fractional solutions. Fig. 2.2 shows the combinatorial nature of such problems by an enumeration of the (finite) feasible set of lattice points, rather grossly depicted by the filled squares.

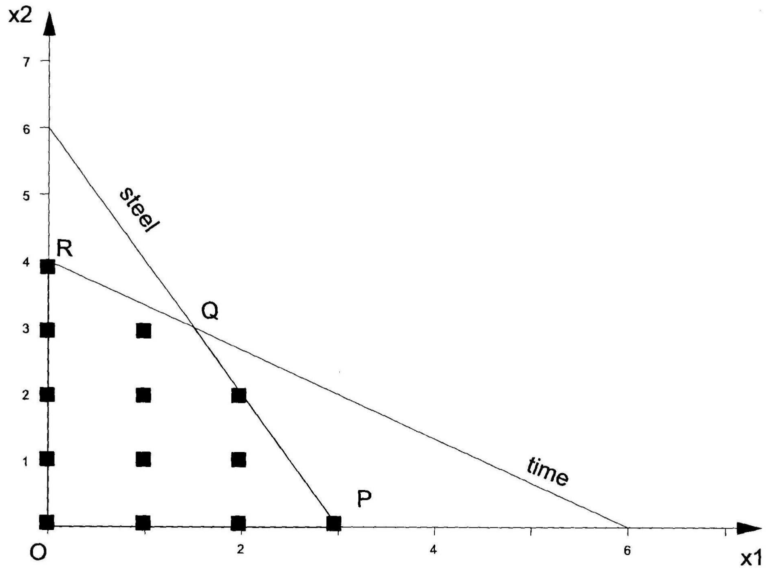


Fig. 2.2 Integer linear programming

3.9 Local and Global Optima

An unconstrained local minimum is a point $\boldsymbol{x} \in R^n$ such that there exists a neighborhood in which the objective at each other point is no better. For a smooth function, it can be pictured geometrically as being at the bottom of a through at which the gradient vector is zero and the Hessian matrix is necessarily positive semi-definite. Such points are normally not too hard to find using methods that make use of first and second derivative information, typically methods of the Newton class. In a constrained problem, a local minimum may occur at a point where the gradient is not zero, since a constraint boundary may have been reached. In general there may be many local minima, and it is also of interest to find which of the local minima is the “best”. Such is global minimization, for which a number of alternative methods exist. In general, the task of finding a global minimum is a much harder problem than the task of finding a local minimum, primarily because it is much harder to verify that the claimed global minimum is actually that.

For a large class of practical problems, global minimization is, in general, an impossible task, although in a number of practical cases, such problems have been solved in a satisfactory manner. Normally, "real-world" optimization problems are global, constrained, mixture of discrete and continuous, nonlinear, multivariate and nonconvex.

Interestingly, some of the more imaginative of recent attempts at optimization methods try to mimic perceived processes of nature. One such approach is that of simulated annealing; another is evolution via the class of so-called genetic algorithms.

The application of any nonlinear optimization method can only ensure the attainment of a local optimum which in the case of nonconvex objective functions (often the case with practical problems) is not necessarily the global optimum. Most practical engineering problems can be formulated in the light of global optimization, i.e. optimization problems in which the objective function is nonconvex and possesses many local optima in the resign of interest. In case the objective function is multimodal, i.e. has several optima, the aim of the global optimization method is to find the smallest local minima or the largest local maxima depending upon the problem.

3.10 Global Optimality

In the optimum design of system, global optimum must be satisfied under the following conditions.

1) If the cost function $f(\boldsymbol{x})$ is continuous on a closed and bounded feasible region, then Weierstrass Theorem guarantees the existence of a global minimum. For this situation, if we can calculate all the optimum points, and then select a solution that gives the least value to the cost function.
2) By showing the optimization problem to be convex because in that case any local minimum is also a global minimum.

4. Optimization Methods

The optimum seeking methods are also known as mathematical programming technique and are generally studied as a part of operations research. Operations research is a branch of mathematics concerned with the application of scientific methods and techniques to decision making problems and with establishing the best or optimal solutions. There are many classes into which we may partition optimization problems, and also many completing algorithms which have been developed for their solution. We outline the areas of primary interest for the present work, with some brief contrasts to classes of problems that we do not consider here.

4.1 NASTRAN Optimization [13]

The optimization algorithms in MSC NASTRAN belong to the family to methods generally referred to as "gradient-based", since, in addition to function values, they use function gradients to assist in the numerical search for an optimum. NASTRAN Optimization can be effectively solved design optimization in the big model with many design variables. Because it has used the function of approximation model, design variable linking and screening of constraints. Also, it can solve the structural optimization problem considering static analysis, normal mode analysis, buckling analysis, transient response analysis, frequency response analysis, aeroelastic analysis, flutter analysis. The optimizer in MSC NASTRAN are MMFD (modified method of feasible directions), SLP(sequential linear programming) and SQP(sequential quadratic programming). MMFD here is default in MSC NASTRAN optimization.

4.1.1 Modified Method of Feasible Directions (MMFD)

After the objective function and constraints are approximated and their gradients with respect to the design variables are calculated based on the approximation, we are able to solve the approximate optimization problem. MMFD is one of the algorithms used in the optimizer. The general formulation of optimization is as follows:

Find the set of design variables x_i, $i = 1, 2, \cdot \cdot \cdot, n$

Minimize $f(\boldsymbol{x})$

Subject to $g_j(\boldsymbol{x}) \le 0 \quad j = 1, 2, \cdot \cdot \cdot, n_g$

$x_i^L \le x_i \le x_i^U \quad i = 1, 2, \cdot \cdot \cdot, n$

Given an initial $\boldsymbol{x}$-vector $\boldsymbol{x}^0$, the design will be updated according to Eq. (2.2)

$$\boldsymbol{x}^q = \boldsymbol{x}^{q-1} + \alpha^* S^q \tag{2.2}$$

The overall optimization process now proceeds in the following steps:

Step 1: Start, $q = 0$, $\boldsymbol{x}^q = \boldsymbol{x}^m$.

Step 2: $q = q + 1$

Step 3: Evaluate objective function $f(\boldsymbol{x})$ and constraints $g_j(\boldsymbol{x})$
where j =1, 2, …, n_g

Step 4: Identify the set of critical and near critical constraints J

Step 5: Calculate gradient of objective function $\nabla f(\boldsymbol{x})$ and $\nabla g_j(\boldsymbol{x})$ for all $j \in J$

Step 6: Find a usable-feasible search direction $\boldsymbol{S}^q$

Step 7: Perform a one-dimensional search to find α^*

Step 8: Set $\boldsymbol{x}^q = \boldsymbol{x}^{q-1} + \alpha^* \boldsymbol{S}^q$

Step 9: Check for convergence to the optimum. If satisfied, go to step 10 otherwise, go to step 2.

Step 10: $\boldsymbol{x}^{m+1} = \boldsymbol{x}^q$

1) Search direction

In order to make further improvement in an optimization loop, a new search direction must be found that continues to reduce the objective function but keeps the design feasible. We seek a usable-feasible search direction, in which:

A usable direction is the one that reduces the objective function, and a feasible direction is the one that a small move in this direction will not violate the constraints.

This situation is shown in Fig. 2.3

2) Convergence to the optimum

Since numerical optimization is an iterative process and one of the most critical and difficult tasks is determining when to stop. The optimizer uses several criteria to decide when to end the iterative search process. This process is applied to the only solution of the approximate optimization problem.

(1) Maximum iteration

The maximum number of iterations (search directions) is included. The default for this is 40 iterations. Usually, an optimum is found sooner than this; therefore, the maximum is mainly intended to avoid excessive computations.

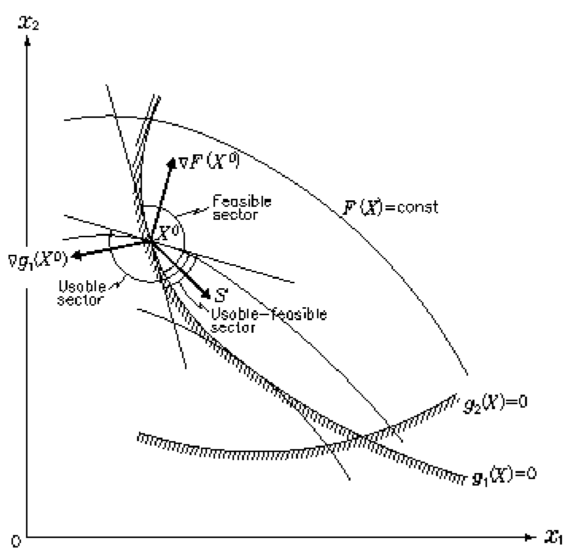


Fig. 2.3 Usable-feasible search directions (Vanderplaats, 1984)

(2) No feasible solution

If the initial design is infeasible (constraints are violated), the first priority is to overcome these violations and find a feasible solution. However, if there are conflicting constraints, a feasible solution may not exist. Therefore, if a feasible design is not achieved in 20 iterations, the optimization process is terminated.

(3) Changes of objective function

To measure the progress made in the successive iterations, this is particularly true because this program is solving an approximate problem that is to be updated on the next design cycle. Here, two criteria are used. The first criteria require that the relative change in the objective between iterations be less than a specified tolerance δ_{obj} Thus, the criteria are satisfied if:

$$\frac{\left| f(\boldsymbol{x}^{q}) - f(\boldsymbol{x}^{q-1}) \right|}{\left| f(\boldsymbol{x}^{q-1}) \right|} \le \delta_{obj} \qquad (2.3)$$

The default value for δ_{obj} is 0.001.

The second criterion is that the absolute change in the objective between the iterations is less than a specified tolerance δ_{obj}. This criteria is satisfied if

$$f(\boldsymbol{x}^{q}) - f(\boldsymbol{x}^{q-1}) \le \delta_{obj} \qquad (2.4)$$

The default value for δ_{obj} is the maximum of $0.001 \times f(\boldsymbol{x}^{0})$ and 1.0E-20.

The first criterion, relative change, is an indication of convergence if the objective function is large. However, the convergence is controlled by the second criterion, absolute change, if the objective function is small.

3) Satisfaction of Kuhn-Tucker conditions

In the case of an unconstrained problem, the conditions where the gradient of the objective function vanishes as follow:

$\nabla f(\boldsymbol{x}) = 0$

Fig. 2.4 shows the relative and global minima in the design space.

In the case of the constrained problem, the conditions of optimality are more complex. By using the Lagrangian multiplier method, we define the Largangian function as the following:

$$\nabla f(\boldsymbol{x}) + \sum \lambda_j \nabla g_j(\boldsymbol{x}) = 0, \lambda_j \ge 0 \qquad (2.5)$$

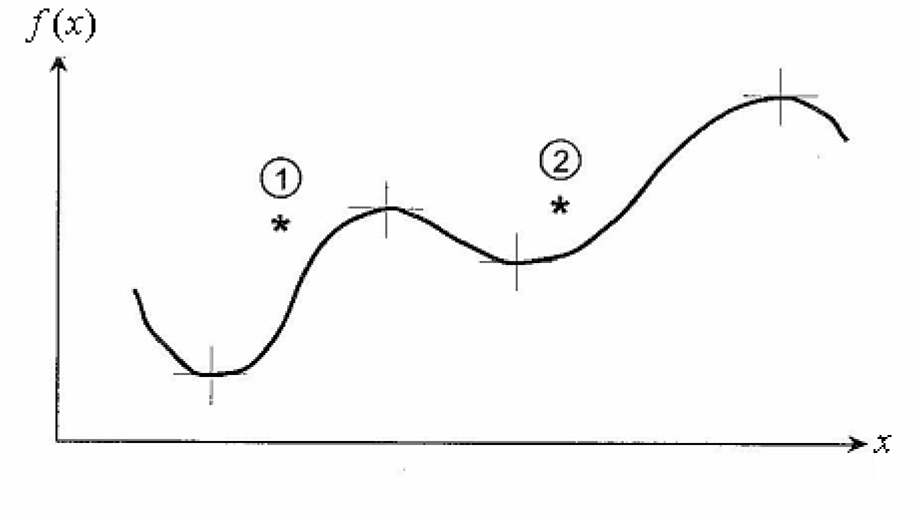


Fig. 2.4 Relative and global minima in the design space

Differentiating the Lagranigian function with respect to all variables we obtain the Kuhn-Tucker conditions which are summarized as follows:

$$\Gamma(\bar{X}, t, \lambda) = f(\bar{X}) + \sum \lambda_j (g_j + t_j^{\,2}) \qquad (2.6)$$

The corresponding λ_j is zero if a constraint is not active.

The physical interpretation of these conditions is that the sum of the gradient of the objective function and the scalars λ_j times the associated gradients of the active constraints must vectorally add to zero shown in Fig 2.5.

Fig. 2.7 Kuhn-Tucker condition at a constrained optimum. The Kuhn-Tucker condition is also sufficient for optimality when the number of active constraints is equal to the number of design variables. Otherwise, sufficient conditions require the second derivatives of the objective function and all of the constraints are convex, the Kuhn-Tucker conditions are also sufficient for global optimality.

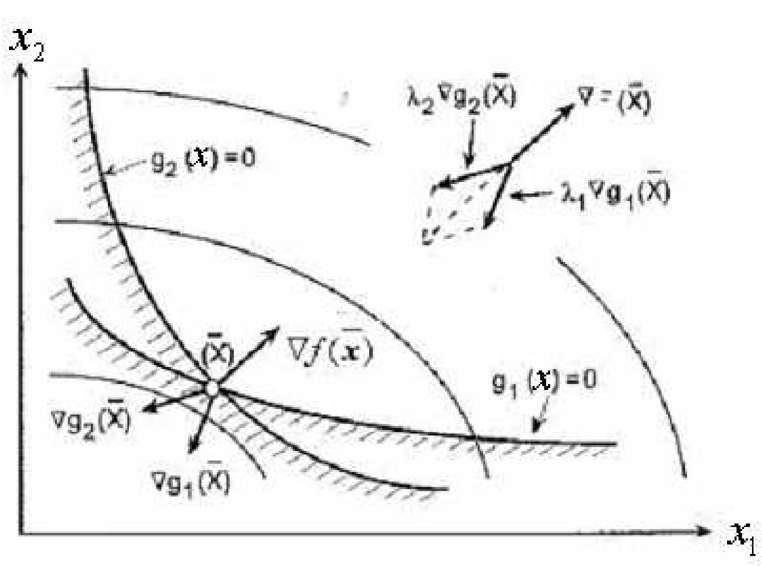


Fig. 2.5 Kucker-Tucker condition at a constrained optimum

4.1.2 Sequential Linear Programming (SLP)

SLP can obtain the solution of nonlinear problem to linear approximation using the linear programming methods. This approach can be linearized about this point, repeating the process until a precise solution is achieved. SLP linearize the nonlinear programming via a first-order Taylor series expansion as Eqs. (2.7) to (2.11)

Objective function: $f(\boldsymbol{x}) \cong f(\boldsymbol{x}^0) + \nabla f(\boldsymbol{x}^0) \cdot \delta \boldsymbol{x}$ (2.7)

Constraints: $g_j(\boldsymbol{x}) \cong g_j(\boldsymbol{x}^0) + g_j(\boldsymbol{x}^0) \cdot \delta \boldsymbol{x} \leq 0, \quad j = 1, \; m$ (2.8)

$$h_k(\boldsymbol{x}) \cong h_k(\boldsymbol{x}^0) + h_k(\boldsymbol{x}^0) \cdot \delta \boldsymbol{x} = 0, \quad k = 1, \; m \tag{2.9}$$

$$\boldsymbol{x}_i^L \leq \boldsymbol{x}_i + \delta \boldsymbol{x} \leq x_i^U, \quad i = 1, \; m \tag{2.10}$$

$$\delta \boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^0 \tag{2.11}$$

where, zero superscript identifies the point about which this Taylor series expansion is performed.

This represents a linear programming problem where the design variables are contained in the vector $\delta \boldsymbol{x}$, and the function and gradients at $\boldsymbol{x}^0$ are constants and coefficients, respectively. Fig. 2.6 shows a geometric interpretation of the SLP method. At the initial design $\boldsymbol{x}^0$, the objective and constraints are linearized to give the straight-line representations of the functions. The optimum of this linear problem is found and is seen to be near the nonlinear optimum, but it is infeasible. However, if we relinearize at this point and repeat the process, we would expect to approach the precise optimum in a few iterations. For fully constrained problems, SLP often converges rapidly to the solution. However, for under-constrained problems, those where there are fewer active constraints at the optimum than there are design variables, the method often performs poorly.

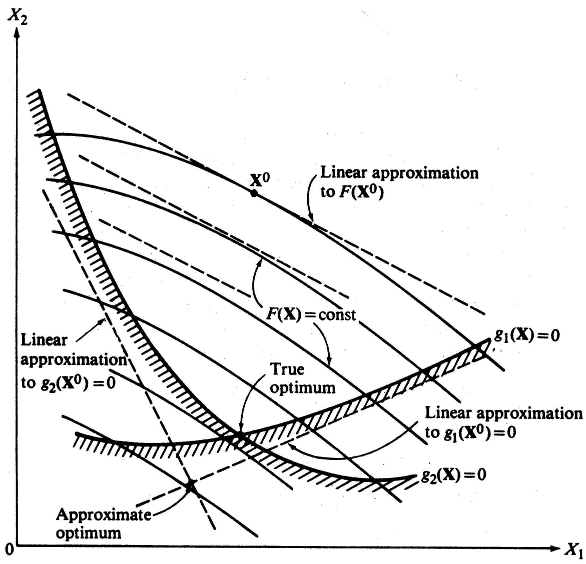


Fig. 2.6 The linearized problem

4.1.3 Sequential Quadratic Programming (SQP)

The basic concept is very similar to sequential linear programming. First the objective and constraint function are approximated using Taylor series approximations. However, a quadratic, rather than a linear approximation of the objective function is used. Linearized constraints are used with this to create a direction finding problem of the form;

$$\text{Minimize : } Q(S) = f^0 + \nabla f^T S + \frac{1}{2} S^T B S \tag{2.12}$$

$$\text{subject to } (\nabla g_j)^T S + g_j^0 \leq 0 \quad j = 1, \ m \tag{2.13}$$

where the design variables are the components of *S*. The matrix *B* is a positive definite approximation of the Hessian of the Lagrangian that is updated using the BFGS (Broyden, Fletcher, Goldfarb and Shanno) formula with ∇L replacing ∇f. However, in the linearization method, *B* = *I* = Identity matrix. This method is considered to be an excellent method by many theoreticians.

4.2 Modified Simplex Method

The simplex method is a local search technique that uses the evaluation of the current set of data to determine a promising search direction. A simplex is defined by a number of points equal to one more than the number of dimensions of the search space. For an optimization problem involving *N* variables, the simplex method searches for an optimal solution by evaluating a set of *N*+1 points, denoted as $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ⋯, $\boldsymbol{x}_{N+1}$. The method continually forms a new simplex by replacing the worst point in the simplex, denoted as $\boldsymbol{x}_w$, with a new point $\boldsymbol{x}_r$ generated by reflecting $\boldsymbol{x}_w$ over the centroid $\vec{\boldsymbol{x}}$ of the remaining points:

$$\boldsymbol{x}_r = \vec{\boldsymbol{x}} + (\vec{\boldsymbol{x}} - \boldsymbol{x}_w) \tag{2.14}$$

where

$$\vec{\boldsymbol{x}} = \frac{1}{n} \sum_{i=1}^{n+1} \boldsymbol{x}_i \ , \quad i \neq w$$

The new simplex is then defined as $x_1 + x_2 + \cdots + x_{w+1} + \cdots + x_{N+1}, \ x_r^2$. This cycle of evaluation and reflection iterates until the step size (i.e., $\boldsymbol{x}_r - \boldsymbol{x}_w$)

becomes less than a predetermined value or the simplex circles around an optimum.

Nelder and Mead [3] developed a modification to the basic simplex method that allows the procedure to adjust its search step according to the evaluation result of the new point generated. This is achieved through three ways. Firstly, if the reflected point is very promising (i.e., better than the best point in the current simplex), a new point, further along the reflection direction, is generated using the equation:

$$\boldsymbol{x}_e = \vec{\boldsymbol{x}} + \gamma(\vec{\boldsymbol{x}} - \boldsymbol{x}_w) \tag{2.15}$$

where γ is the called the expansion coefficient (γ > 1), because the resulted simplex is expanded.

Secondly, if the reflected point $\boldsymbol{x}_e$ is worse than the worst point in the original simplex (i.e., $\boldsymbol{x}_w$), a new point, close to the centroid on the same side of $\boldsymbol{x}_w$, is generated using the following equation:

$$\boldsymbol{x}_c = \vec{\boldsymbol{x}} - \beta(\vec{\boldsymbol{x}} - \boldsymbol{x}_w) \tag{2.16}$$

where β is called the contraction coefficient ($0 < \beta < 1$) because the resulted simplex is contracted.

Finally, if the reflected point $\boldsymbol{x}_e$ is not worse than $\boldsymbol{x}_w$, but is worse than the second worst point in the original simplex, a new point, close to the centroid on the opposite side of $\boldsymbol{x}_{w}$, is generated using the contraction coefficient β.

$$\boldsymbol{x}_c = \vec{\boldsymbol{x}} - \beta(\vec{\boldsymbol{x}} - \boldsymbol{x}_w) \tag{2.17}$$

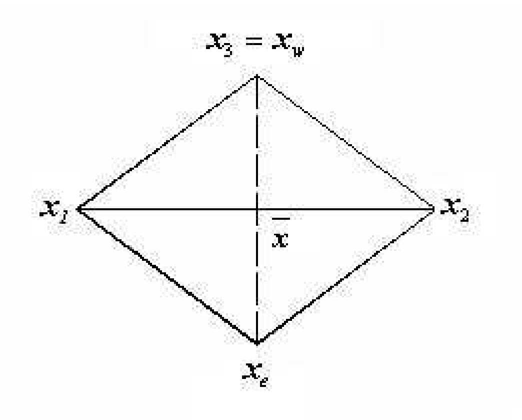


Fig. 2.7 Example of two-dimensional simplex

Fig. 2.7 illustrates how this method applies to an optimization problem involving two variables. Supposed points x_1, x_2 and x_3 are from an original simplex, and point x_3 (= x_w) has the worst evaluation. Point $\vec{x}$ represents the centroid of x_1 and x_2. The reflecting point x_3 across $\vec{x}$ generates x_e, which together with points x_1 and x_2, forms the new simplex.

However, general objective function is subjected to constraint conditions and parameters setting a minimum and a maximum value ($x_{\min} \le x \le x_{\max}$). That is to say, if the optimum solution can't satisfy constraint conditions or parameter range, even if the solution is the optimum solution, it cannot be selected. Therefore, in order to satisfy the above restriction, Nelder's simplex algorithm is modified as follows:

1) If $x_{\max} \le x_r$ or $x_{\min} \ge x_r$, then the value of x_r is replaced by $x_{\min}$ or $x_{\max}$. When x_r is created by expansion, contraction or reflection, this process will guarantee that there is x_r in the parameter range.
2) If x_r can't satisfy constraint conditions, then x_w is replaced by the second worst point in the original simple. This process is similar to the contraction because they are almost equal in the resulted simplex.

4.3 Genetic Algorithms (GA)

Genetic algorithm simulates a probabilistic search technique that is analogous to the biological evolutionary process. The GA consists of three main strategies, reproduction, crossover and mutation. The performance of the GA depends on the operating parameters, namely crossover, mutation and reproduction.

The GA consists of three main strategies (reproduction, crossover and mutation). Using reproduction in the GA, individuals are selected from the population and recombined, producing offspring, which will comprise the next generation. Two individuals are selected and their chromosomes are recombined. Crossover is the operation when two individuals are taken and their chromosomes are cut at some randomly chosen position, to produce two head and tail segments. These segments are swapped to reproduce two new full-length chromosomes. The offspring inherits some genes from each parent. Mutation is the technique used to randomly alter the genes with a small probability and is typically applied after crossover. Crossover is more important for rapidly exploring a search space. Usually, mutation provides a small amount of random search.

4.3.1 Composition of GA [4,5]

A GA for a particular problem must have the following five components:

1) A genetic representation for potential solutions to the problem.
2) A way to create an initial population of potential solutions.
3) An evaluation function that plays the role of the environment, rating solution in terms of their "fitness".
4) Genetic operator that alters the composition of children from parents.
5) Values for various parameters that the GA uses (population size, probabilities of applying genetic operators, etc.).

The parameters to be optimized are usually represented in a string (or chromosome) from since genetic operators are suitable for this type of representation. The method of representation has a major impact on the performance of the genetic algorithms. Different representation schemes might cause different performances in term of accuracy and calculating time. There are two common representation methods for numerical optimization problems [14, 15].

4.3.2 Genetic Operators

It seems that there are two important issues in the evolution process of the GA. The one is population diversity and the other is selective pressure. These factors are strongly related: an increase in the selective pressure decreases the diversity of the population, and vice versa. In other words, strong selective pressure supports the premature convergence of the GA and a weak selective pressure can make the search ineffective. Thus it is important to keep a balance between these two factors.

There are three common genetic operators: selection, crossover and mutation. An additional reproduction operator, inversion, is sometime also applied.

1) Selection

The aim of the selection procedure is to reproduce more copies of individuals whose fitness values are higher than those whose fitness values are low. The selection procedure has a significant influence on driving the search towards a promising area and finding food solutions in a short time. However, the diversity of the population must be maintained to avoid premature convergence and to reach the global optimal solution. Selection determines the reproduction provability of each individual in selection pool. This probability depends on the own objective value and the objective value of all other individuals. Three kind of selection method is usually used in application.

(1) Roulette wheel selection [4]

The mechanism of this selection is reminiscent of the operation of a roulette wheel. Fitness values of individuals represent the widths of slots on the wheel. After a random spinning of the wheel to select an individual for the next generation, individuals in slots with large widths representing high fitness values will have a higher chance to be selected.

(2) Rank-based selection [16]

According to this procedure, each individual generates an expected number of offspring, which is based on the rank of its fitness value and not on actual evaluation values. This strategy is similar to roulette wheel selection, excluding the application of uniform region and control better the selective pressure than that of the roulette wheel strategy.

(3) Tournament selection [17]

This method selects randomly a group, *k*, of individuals from a beginning population, and from this group, the most fitness individual is chosen to move on to the next population. This process is repeated population-size number of times. It is clear, that large value of *k* increases selective pressure of this procedure.

2) Crossover

This operator is considered the one that makes the genetic algorithm different from other algorithms, such as dynamic programming. It is used to create two new individuals (children or offspring) from two existing individuals (parent) picked from the current population by the selection operation. There are several ways of doing this.

(1) Simple crossover

Simple crossover is two kinds of crossover, single or one point crossover and multi-point crossover. First, two individuals are randomly selected as parents from the pool of individuals formed by the selection procedures. Second, they are cut at a randomly chosen point. Finally, the tails, which are the parts after the cutting point, are swapped, and then two new individuals (offspring) are produced.

(2) Uniform crossover

Uniform crossover is proposed to overcome the problem that the process of simple crossover may be lost a sequence of string, which is schema. This is important in that an individual solution is coded as a string. In the concept, uniform crossover is similar to multi-point crossover. The difference between simple crossover and uniform crossover is in the way that a swapping point is selected.

3) Mutation

The part of mutation is that the initial individuals are widely distributed in the search space and prevented the initial local convergence. In this procedure, all individuals in the population are checked bit by bit and the bit values are randomly reversed according to a specified rate. Unlike crossover, this is a monadic operation. That is, a child string is produced from a single parent string. The mutation operator forced the algorithm to search new areas. Eventually, it helps the GA avoid premature convergence and find the global optimal solution. In the binary coding, this simply means changing a 1 to a 0 and vice versa, and is the occasional random alteration of the value of a string position.

(1) Classical mutation

Goldberg [5] proposed this strategy in a basic GA, which was modeled by Holland [4]. A genotype of selected parent is exchanged by mutation rate, which is similarly small (or smaller) in natural population.

(2) Uniform mutation

It is similar to the definition of the classical version, which searches a new point with a uniform probability distribution. This operator requires a single parent $\boldsymbol{x}$ and produces a single offspring $\boldsymbol{x}^{'}$. The operator selects a random component $k \in (1, \cdots, q)$ of the vector $\boldsymbol{x} = (x_1, \cdots, x_k, \cdots, x_q)$ and

Produces $\boldsymbol{x}^{'} = (x_1, \cdots, x_k^{'}, \cdots, x_q)$.

$$\boldsymbol{x}_k^{'} = \boldsymbol{x}_k^L + n(\boldsymbol{x}_k^U - \boldsymbol{x}_k^L),\ n \in R\ [0\ 1] \tag{2.18}$$

where x_k^L and x_k^U are the lower and upper boundaries of the parameter x_k, respectively, and n is the real value selected randomly from 0 to 1.

(3) Dynamical mutation

If a high mutation rate is applied to all stages, we may loose the good searched candidates for optimum solutions from the previous generation. In order to avoid this problem, the elite preservation strategy and the dynamical mutation are applied. One conserves the individuals that have higher fitness with a certain proportion rate and the other guarantees that the search point (initial candidates) is wildly distributed in the search space.

Eq. (2.19) shows the dynamical mutation, which considered in the global search steps.

$$M_i = \exp\left(-\frac{D}{5(d_i + 1)} - \frac{4g_i}{G} \right) \quad (2.19)$$

where g_i is the ith generation number, G is the total generation number, d_i is the reproduced offspring number at the ith generation and D is the population number.

The feature of the dynamical mutation decreases exponentially at once with the generation increasing and is fluctuated by the reproduction rate, which is a total population number to a generation number.

4.3.3 Differences from Other Traditional Methods

Goldberg had summarized the characteristic of GA in comparison with conventional optimizations as follows [5]:

- GA is a multi-point search algorithm using a population, which is a set of random solutions, not using a potential solution.
- GA works with a coding of candidate set, not solutions themselves.
- GA uses only fitness function, not derivative or other auxiliary knowledge.
- GA is a stochastic search algorithm based on the mechanism of the natural world and natural genetics. GA starts with an initial set of random solutions called population.
- GA uses probabilistic transition rules, not deterministic rules.

GA do not have much mathematical requirements about the optimization problems. Due to their evolutionary nature, GAs will search for solutions without regard to the specific inner workings of the problem. GAs can handle any kind of objective functions and any kind of constraint (i.e., linear or nonlinear) defined on discrete, continuous, of mixed search spaces. GAs do not associate with an initial point problem. To be precise, because GAs compose randomly a group of potential solutions, GAs do not have the notion of an initial point problem. That provides us with the great belief that GAs can find out global optimum solutions and a flexibility to hybridize with domain-dependent heuristics to make an efficient implementation for a specific problem.

4.3.4 Limitations of Algorithm

However, GAs have also the following drawbacks or limitations they are:

- A binary code is not free to make a genotype of individuals
- The fittest individual may be lost during the selection process due to its stochastic nature.
- Fit individuals may be copied several times and a fit individual may quickly dominate the population at an early stage, especially, if the population size is small.
- The selection operation alone explores no new points in a search space. In other words, if cannot create new schemata.
- Different genetic parameters such as population size, crossover probability, mutation probability, etc. greatly affect the accuracy and calculation time of optimum solution.

4.3.5 Simple Genetic Algorithm [15, 16, 18]

Fig. 2.8 shows the flowchart of simple GA. A simple GA randomly generates an initial population. The GA proceeds for a fixed number of generations or until it satisfy some stopping criterion. During each generation, the GA performs fitness proportionate selection, followed by single-point crossover and mutation. Fig. 2.9 illustrates closely a process of evolution at *k* generation.

First, fitness proportionate selection assigns each individual structure in the population $\overline{P}$, according to the ratio of fitness and the probability of selection. Second, using the single-point crossover $\tilde{P}$ is composed. After the crossover stage has finished, the mutation stage begins. For every string that advances to the mutation stage, each of its bits is flipped with probability (mutation rate). The population resulting from the mutation stage then overwrites the old population (the one prior to selection), completing one generation (k+1). Subsequent generations follow the same cycle of selection, crossover and mutation.

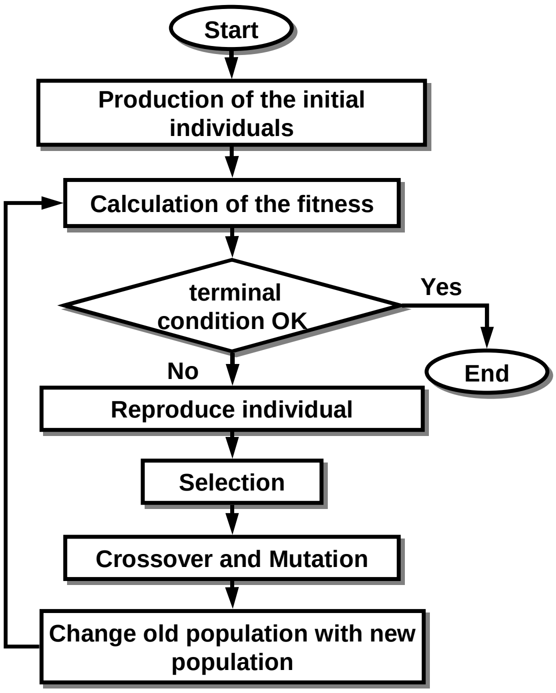


Fig. 2.8 Flowchart of the simple genetic algorithm

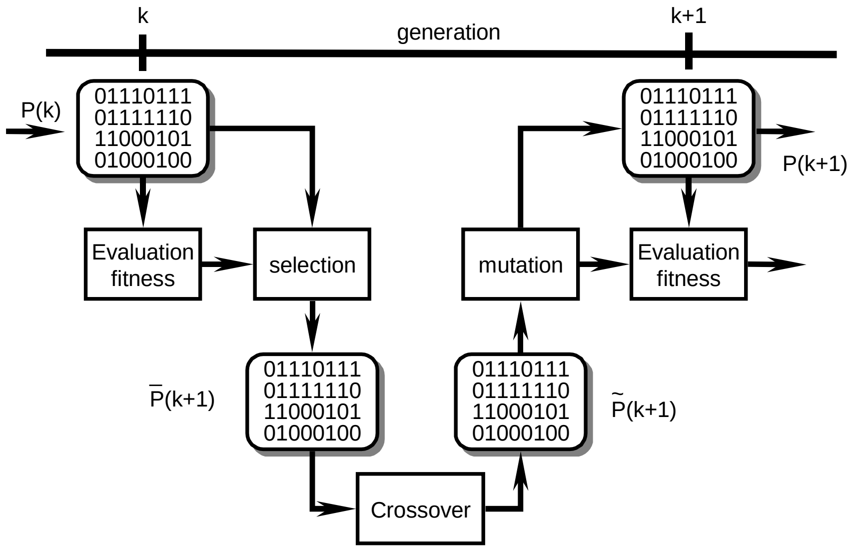


Fig. 2.9 Structure of the simple genetic algorithm

4.4 Random Tabu Search Method (R-tabu Method)

Hu [19] had improved tabu search method proposed by Glover [7] and had applied to constrained optimum. In the case of minimizing an objective function "$f(\boldsymbol{x})$" with constraint range [a, b], new parameters, step number and count number, are introduced by random tabu search method. Step number is the number of searching the neighbors and count number is the maximum iteration number of searching to search a neighbors. The initial value which is the first approximate solution satisfying the constraint condition, symbolizes $\boldsymbol{x}_0$, and then $\boldsymbol{x}_0$ surrounding neighbors $N(\boldsymbol{x}_0, h_1)$ are

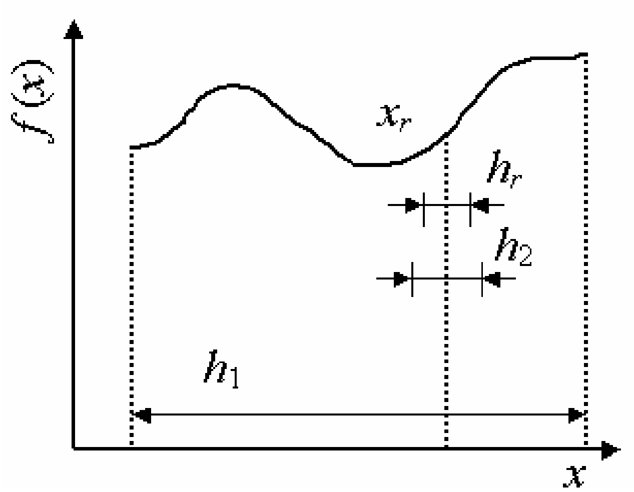
$$N(\boldsymbol{x}_0, h_1),\ N(\boldsymbol{x}_0, h_2),\ \cdot\ \cdot\ \cdot\ ,\ N(\boldsymbol{x}_0, h_r)$$

where h_i (i =1, 2, · · · , r) refers to step size, r means step number. The set of step size, H, is

$$H = \{ h_1,\ h_2,\ \cdot\ \cdot\ \cdot\ ,\ h_r \}$$

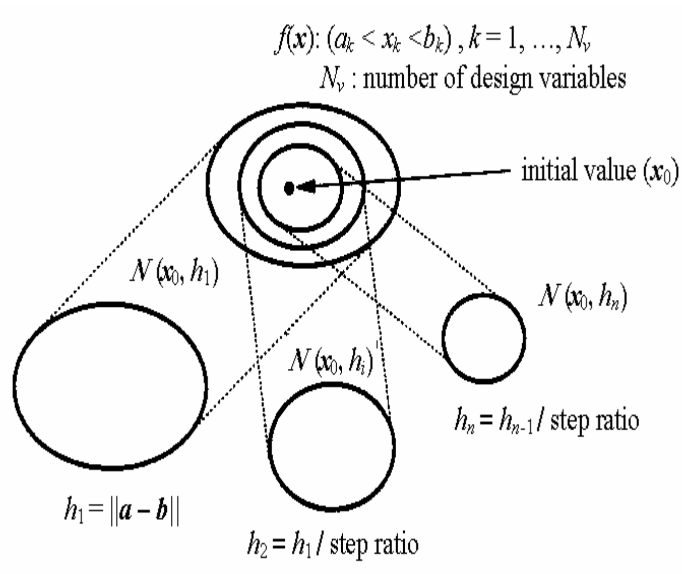
where $h_1 = b\text{-}a$, $h_2 = h_1 P$, $h_3 = h_2 P, \cdots$, $h_r = h_{r-1} P$, P is step ratio

The procedure of R-tabu method is as follows. If $f(\boldsymbol{x})$ which is led by randomly generating $\boldsymbol{x}$ in each neighbor is less than $f(\boldsymbol{x_0})$, then $\boldsymbol{x}$ is supposed to minimum value among its neighbors. The smallest $\boldsymbol{x}$ in the minimum value of each neighbor becomes the second approximate solution $\boldsymbol{x}_1$, and repeats at the each neighbor which is set up at the $\boldsymbol{x}$ surrounding. The basic principle is similar to the method combined pattern searching of the direct search method proposed by Hook-Jeeves with a lattice search method, but it is different to take a neighbor in steady of improved step size and to use several different step sizes. Features of R-tabu method are as follows. First1y, it can reduce an iteration number and promote the efficiency of searching, because each searching solution locates at different searching domain. Secondly, it is possible with this method to take a global optimum and to avoid trapping in a local optimum because of utilizing random searching. Finally, it is possible to get the optimum solution fast and accurately, if the method is combined with other optimization methods. Fig. 2.10 shows the flowchart of R-tabu method.

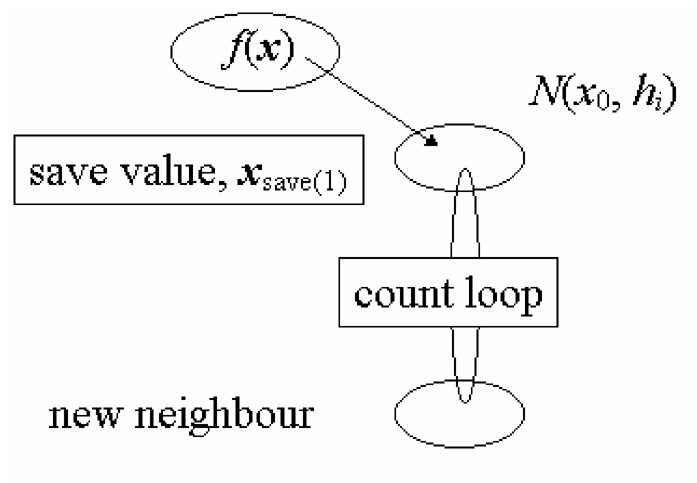


(a) Setting neighbor

Fig. 2.10 Flowchart of random tabu search method (*Continued*)

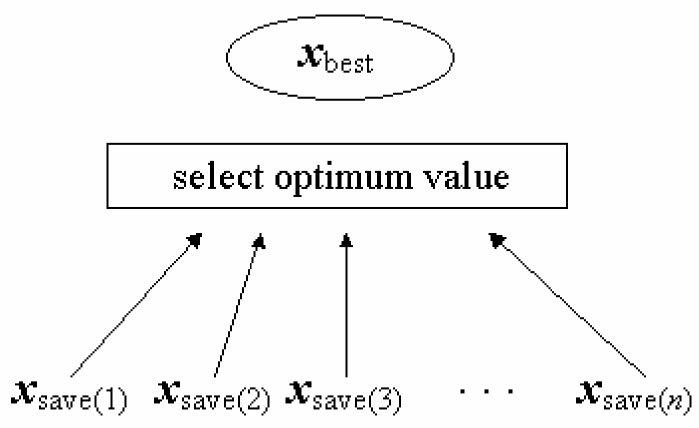


(b) Initializing neighbor



(c) Comparison

Fig. 2.10 Flowchart of random tabu search method (*Continued*)



(d) Selecting and setting new value

Fig. 2.10 Flowchart of random tabu search method

4.5. Artificial Life Algorithm

The origin of artificial life started in the late 1960s when John Conway began work in his “Game of Life”. This was simply an array of cellular automata whose two states were metaphorically dubbed by Conway “live” and “dead”. Langton who has contributed to artificial life greatly, defines artificial life as follows: “Artificial life is the study of man-made systems that exhibit behaviors characteristic of natural living systems” [20]. The most important two characteristics of artificial life are emergence and dynamic interaction with the environment. The emergence is the result of dynamic interaction among the individuals consisting of the system and is not found in an individual. The micro-interaction with each other in the artificial life’s group results in emergent colonization, the emergence, in the whole systems. The artificial world in the artificial life algorithm is defined as the domain of the given optimization problem.

Fig. 2.11 shows a circular food chain, which consists of four kinds of resources and four species of artificial organisms [21]. Artificial organisms can move about in the world consuming energy resources and producing waste. The four species of artificial organisms compose a circular food chain where one species' waste is another's food. Artificial organisms can only metabolize the resources, which they want to. The demanded resources are determined according to the four species of artificial organisms. Artificial organisms have a sensory system, which enables them to see resources as well as other artificial organisms in the world. They are also able to determine the location of the nearest resources and other artificial organisms from their present location. This nearest location of resource becomes the goal, which drives them to move forward. Artificial organisms must maintain a minimum internal energy level in order to exist. Once an artificial organism's energy level drops below the minimum energy, it is considered to be "dead" and removed from the world.

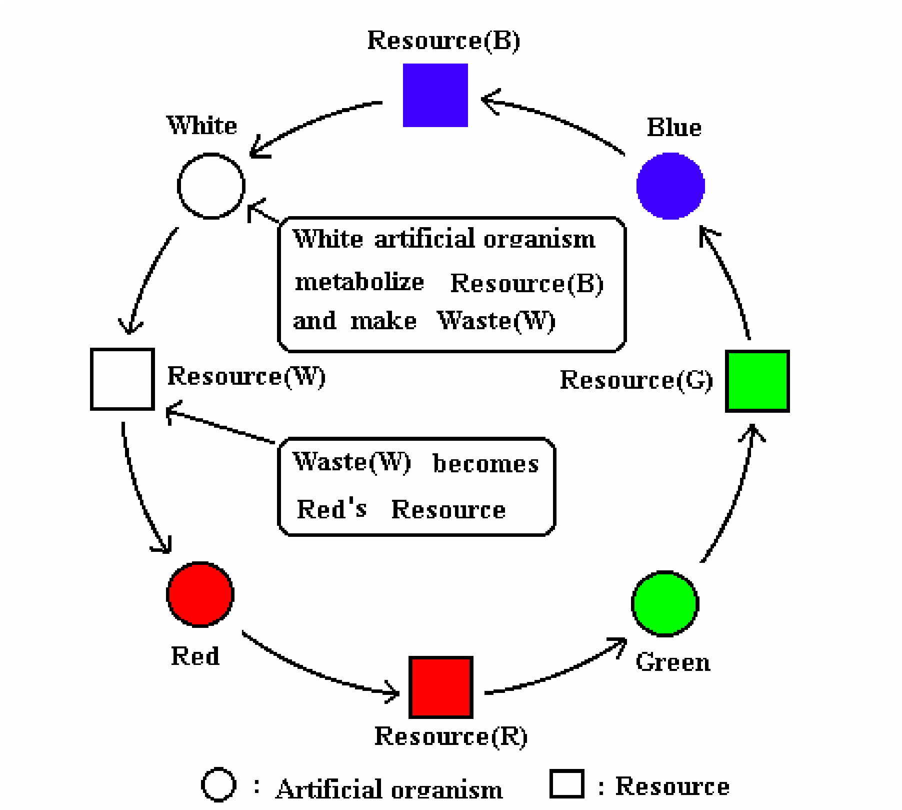


Fig. 2.11 Circular food chain of an artificial algorithm

4.6. Simulated Annealing

The simulated annealing algorithm was derived from statistical mechanics. Kirkpartick et al. [9] proposed an algorithm, which is based on the analogy between the annealing of solids and the problem of solving combinatorial optimization problems. Annealing is the physical process of heating up a solid and then cooling it down slowly until it crystallizes. The atoms in the material have high energies at high temperatures and have more freedom to arrange themselves. As the temperature is reduced, the atomic energies decrease. A crystal with regular structure is obtained at the state where the system has minimum energy. If the cooling is carried out very quickly, which is known as rapid quenching, widespread irregularities and defects are seen in the crystal structure.

This algorithm eliminates most disadvantages of the hill-climbing methods: solutions do not depend on the starting point any longer and are usually close to the optimum point. This is achieved by introducing a probability of acceptance (i.e., replacement of the current point by a new point). The probability of acceptance is a function of the values of objective function for the current point and the new point, and an additional control parameter, temperature, T. In general, the lower temperature T is, the smaller the chances for the acceptance of the system, T, is lowered in steps. This is Metropolis's criterion [22] based on Boltzman's probability.

5. Summary

In this chapter, the general items of optimization are summarized for the understanding of the theory, algorithms, and technique of it. This is because practical problems invariably require tuning algorithmic parameters, scaling, and even modifying existing techniques to suit the specific application. Also, the history of optimization is described from a view point of engineering. Finally, the optimization method is treated. There are many classes into which we may partition optimization problems, and also many competing algorithms which have been developed for their solution. We outline the areas of primary interest for the present work, with some brief contrasts to classes of problems that we do not consider here. In particular, modified method of feasible direction (MMFD) and a genetic algorithm were illustrated in detail. The former is the one of the algorithms to solve the approximate optimization problem. It used the usable-feasible search direction in a search direction and is used NASTRAN optimization module which can be effectively solved design optimization in the big model with many design variables. Because it has used a function of approximation model, design variable linking and screening of constraints. The latter is simulated a heuristic probabilistic search technique that is analogous to the biological evolutionary process. The genetic algorithm consists of three main strategies (reproduction, crossover and mutation). Using reproduction in the genetic algorithm, individuals are selected from the population and recombined, producing offspring, which will comprise the next generation. Two parents are selected and their chromosomes are recombined. Crossover is the operation when two individuals are taken and their chromosomes are cut at some randomly chosen position, to produce two head and tail segments. These segments are swapped to reproduce two new full-length chromosomes. The offspring inherit some genes from each parent. Mutation is the technique used to randomly alter the genes with a small probability and is typically applied after crossover. Crossover is more important for rapidly exploring a search space. Usually, mutation provides a small amount of random search. This algorithm is applied to the newly developed algorithms in this study.

References

[1] S. S. Rao, 1997, *OPTIMIZATION Theory and Applications*, John Wiley & Sons, New York.

[2] D. B. Ashok and R. C. Tirupathi, 1999, *Optimization Concepts and Applications in Engineering*, Upper Saddle River, New Jersey.

[3] J.A. Nelder and R. Mead, 1965, “A simplex method for function minimization,” Computer Journal, Vol.7, Issue 4, pp.308-313.

[4] J. H. Holland, 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Michigan.

[5] D. E. Goldberg, 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Co. Inc. NewYork

[6] F. Glover, 1986, “Future paths for integer programming and links to artificial intelligence,” Computers Ops Res. 5, pp. 533-549.

[7] F. Glover, 1989, “Tabu search-part I ,” ORSA Journal on Computing Vol. 1, No.3, pp. 190-206

[8] P. Hansen, 1986, “The steepest ascent mildest descent heuristic for combinatorial programming,” Numerical Methods in Combinatorial Optimization, Capri, Italy.

[9] S. Kirkpatrick, C. Gelatt and M. Vecchi, 1983, “Optimization by simulated annealing,” Science, Vol. 220, No. 4598, pp. 671-690.

[10] A. Fiacco and G. McCormick, 1968, *Nonlinear Programming: Sequential Unconstrained Minimization Technique*, Wiley.

[11] D. Shin, Z. Guerdal, and O. Griffin, 1990, “A penalty approach for nonlinear optimization with discrete design variables,” Engineering optimization, Vol. 16, pp. 29-42.

[12] G. L. Nemhauser, and L. A. Wolsey, 1988, *Integer and Combinatorial Optimization*, Wiley.

[13] G. Moore, 1994, *MSC/NASTRAN Design Sensitivity and Optimization (User's Guide)*, the MacNeal-Schwendler Corp.

[14] Z. Michalewicz, 1992, *Genetic Algorithm + Data Structures = Evolution Programs*, Springer-Verlag, New York.

[15] L. Davis, 1991, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

[16] J. E. Baker, 1985, "Adaptive selection methods for genetic algorithms, " in Proc. 1st Int. Conf. On Genetic Algorithms and their Application, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 101-111.

[17] D. E. Goldberg, 1990, "A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing," Complex Systems, Vol. 4, Complex Systems Publications, Inc, pp. 445-460.

[18] K. E., Jr. Kinnear, 1994, *Advanced in Genetic Programming*, The MIT Press, Massachusetts, USA.

[19] N. Hu, 1992, "Tabu search method with random moves for globally optimal design," International Journal Numerical Methods in Engineering, Vol. 35, pp.1055-1070.

[20] C. G. Langton, 1989, *Artificial Life*, Addison-Wesley Publishing Co. Inc., N.Y.

[21] B. S. Yang, and Y. H. Lee, 2000, "Artificial life algorithm for function optimization," ASME Design Engineering Technical Conferences, Baltimore, Sept. 10-13, DETC2000/ DAC-14524.

[22] N. Metripolis, A. W. Rosenbluth, M. N. Rosinbluth, A. H. Teller and E. Teller, 1953, "Equation of state calculations by fast computing machines," J. of Chem. Phys., Vol. 21 No. 6, pp. 1087-1092.

III. Development of NASTRAN External Calling Styled Optimization Framework (OPTSHIP)

1. Introduction

Recently, the issue of ship vibration is emerging due to the large scale, high speed and lightweight of ship. Therefore, shipbuilders are suggested to concern and applying the strict vibration criteria for low vibration levels at the deckhouse in the cabin for pleasantness (human comfort). This issue becomes an important condition for taking orders from customers, so the optimization solution to get a sound ship and to reduce the construction cost is needed.

In industrial fields, much commercial software such as NASTRAN [1], ANSYS, etc are used to analyze and predict the response of vibration and structure. The capabilities of this software cover application to the complicated and big structures like ships. Nowadays, all shipbuilders try to save the construction and design costs due to the continuously increasing costs of labor and material costs. Traditionally, the ship design is based on a sequential and iterative approach and is a very complex problem, which has been approached by optimization methods only in the last two decades. This problem is more complicated and difficult in the case of a large ship such as a merchant ship because of the large number of degrees of freedom of the structure. Many researchers have studied the reduction model to save the running time and they have attempted to solve the complicated ship design problems using different optimization techniques. These efforts effectively contributed to the development of new design and effective optimization method [2].

Optimization is to determine the size or the geometric shape of the structure to obtain the maximum performance using minimal material with safety and availability of the target structure which is a ship in this case [3]. We consider three kinds of approaches for size optimization of a ship for vibration reduction design. Firstly, the optimization module of MSC/ NASTRN [4] is considered. This module is based on the sensitivity analysis which is the sensitivity derivatives of that response with respect to the variables of the problem. The demerits of using this module are as follows,

- Many constraints to set an objective function and a design variable
- Difficulty of performing global optimization by the local search method
- The limitation of the optimization methods which cannot combine any complex user-defined optimization method.

The second approach is using the NASTRAN/DMAP [5]. This enables the user to extract useful information during the analysis. However, NASTRAN execution is accompanied by many difficulties in iterative information exchange with external programs like a user-defined optimization programs. Furthermore, using the DMAP requires considerable knowledge for a non-expert user. Due to these limitations, this approach provides limited functionality for the user who wants to optimize a structure.

Finally, this thesis presents a new optimization approach called OPTSHIP which uses NASTRAN for vibration analysis and global optimization algorithms for preventing the local convergence. The OPTSHIP employs MSC/NASTRAN and user-defined optimization methods as the analyzer and optimizer respectively. Any function optimization method can be the optimizer of the OPTSHIP. Especially, the global optimization methods are considered in this thesis, which are the genetic algorithm (GA) [6] and the random tabu search method (R-tabu) [7]. GA has an excellent ability in searching through the broad solution space and its usefulness has been demonstrated by various optimization problems. R-tabu is an iterative procedure for solving discrete combinational optimization problems. R-tabu prevents convergence to the local optimum solution and can improve the convergence level of the optimum solution. The two methods are widely used broadly and well studied by many researchers [8-14]. OPTSHIP uses forced response, mode shape vector, natural frequency, weight and combined results of those values as the objective function. Thickness of shell element, area of beam element, stiffness of spring element and thickness of shear element can be used as design variables at this stage. We applied the OPTSHIP to an actual containership, and compared the forced response of the optimized model to the original model one to verify the reliability and performance of the proposed algorithm.

2. Optimization Method Using NASTRAN As a Solver

2.1 NASTRAN Optimization Module

The function of the NASTRAN optimization module is to find a modified model to minimize (or maximize) the objective function while satisfying the constraints from the current analysis model. The design variables are categorized commonly into two groups. One is shape variables which relate to the shape optimization, and the other is sizing variables which relate to the size optimization. This module can be used in the linear static analysis, normal mode analysis and frequency response analysis for the optimization and sensitivity analysis. The objective function includes a user-defined simple expression as well as a direct result from one of the above analyses. However, the module cannot use the objective function for a design variable when the user defined expression used in the objective function is complex because this module is based on the sensitivity analysis. Since the local optimization methods such as modified feasible directions method, sequential linear programming and sequential quadratic programming are used in a NASTRAN optimization module, the module can only locally optimize a model.

The functions of NASTRAN/DMAP include constructing an objective function by extracting the meaningful information during the analysis process, modifying the sequence of the analysis process and making a user-defined program to perform a specific function. Engineering optimization requires a large number of iterative analyses. The analysis results require estimation of the objective function at each iteration. This iteration can be performed by using a specialized program called toolkit. Unfortunately the use of the toolkit asks for a highly skilled operator [5].

Therefore this thesis proposes a new methodology called OPTSHIP for an optimization using NASTRAN as an analyzer. The proposed methodology will provide a user with an easier optimization method than those of NASTRAN/ DMAP and NASTRAN/OPT modules.

2.2 OPTSHIP: NASTRAN External Calling Styled Optimization Framework

The flowchart of the proposed algorithm is described in Fig. 3.1. The OPTSHIP uses the MSC/NASTRAN [4] as a solver to estimate a user-defined objective function. Running the OPTSHIP needs a user-defined objective function, a design variable set and an analysis model file. In addition, the OPTSHIP consists of five modules: initiation module, optimization module, interface module with NASTRAN, estimation module of the objective function and base module. The term “module” is not intended as an independent execution of each module, but to emphasize functional specialization of each module. All modules are functionally related to each other and need to execute the OPTSHIP.

The base module controls the process of the OPTSHIP and manages the data which is used in the OPTSHIP by using another four properly modules according to the execution process. The information of a model, a set of design variables and an objective function are loaded by an initiation module. This module sends the analysis results which enable the estimation of the objective function and design variable, such as forced responses at specified node points, natural frequencies and mode vectors, to the interface module. The initiation module informs which optimization method is used to optimization module, and what kind of objective function that used to the estimation module of objective function. Logically, the optimization module can be constituted with most of the optimization methods. But for the present condition, this module consists of GA [6, 8, 9, 11-14], R-tabu [7, 8], simulated annealing method [15, 16] and the artificial life optimization algorithm [17]. The selected optimization module optimizes the design variable set. And the optimization module receives the revised objective function value $f(x^*)$ which is returned from the estimation module of the objective function with a new trial design variable set x^*. This module has a convergence decision with an optimized design variable set and the optimized objective function value. Simultaneously the results will be transferred into the base module when the convergence criteria are satisfied. If the estimation of the objective function module receives a new trial design variable set x^* from optimization module, it transfers the results to interface module with NASTRAN and receives the analysis results needs to estimate the objective function from the interface module. This module returns the estimated value of objective function to the optimization module. The objective function value can be a function of one or more of the forced responses at the specified node points, natural frequencies, weights and mode vectors. The interface module updates the analysis model with the trial design variable set x^* by the returning estimation module of the objective function, updates NASTRAN input file and executes NASTRAN to analyze the model. The NASTRAN analysis results concerned with estimation of objective function are transferred to the estimation module of the objective function according to the set up condition in the initiation module.

The description on the OPTSHIP is valid for readers to understand the overall structure of the OPTSHIP functionally, which explains the functions of the OPTSHIP which are divided into five modules depending on the functions.

A sequential description is also required to understand the optimization process of the OPTSHIP as follows.

Step 0: An analysis model file is made by PATRAN [18] or CAD and then the information of an objective function and a design variable set is determined and are saved into a file.

Step 1: The base module activates the initiation module.

Step 2: The analysis model, the information of a design variable set and an objective function are loaded. The analysis model and the required results to estimate the objective function are informed to the interface module.

Step 3: The base module activates the optimization module.

Step 4: The optimization module activates the estimation module and passes a trial design variable set x^* to the estimation module.

Step 5: The trial design variable set x^* is passed to the interface module. The interface module is activated.

Step 6: The analysis model is updated with consideration of the trial design variable set x^*.

Step 7: The updated analysis model is written into a NASTRAN input file.

Step 8: Interface module executes the NASTRAN.

Step 9: The analysis results by NASTRAN are loaded, which depends on the objective function.

Step 10: The selected results are returned to the estimation module.

Step 11: The objective function value is estimated by the analysis results and returned to the optimization module.

Step 12: The convergence condition is estimated.

Step 13: If the condition is satisfied, then the optimized design variable is returned to the base module and the optimization module is terminated. However, if the condition is not satisfied then a new trial design variable set is generated. The procedure is returned to step 4. The updating method of the design variable depends on the selected optimization method. However the generation of a new design variable is generalized by $x^* = x^* + \Delta x$ where Δx is the increment of a trial design variable which depends on the optimization method.

Step 14: Base module prints the optimized design variable set and the optimized analysis model, and then the OPTSHIP is terminated.

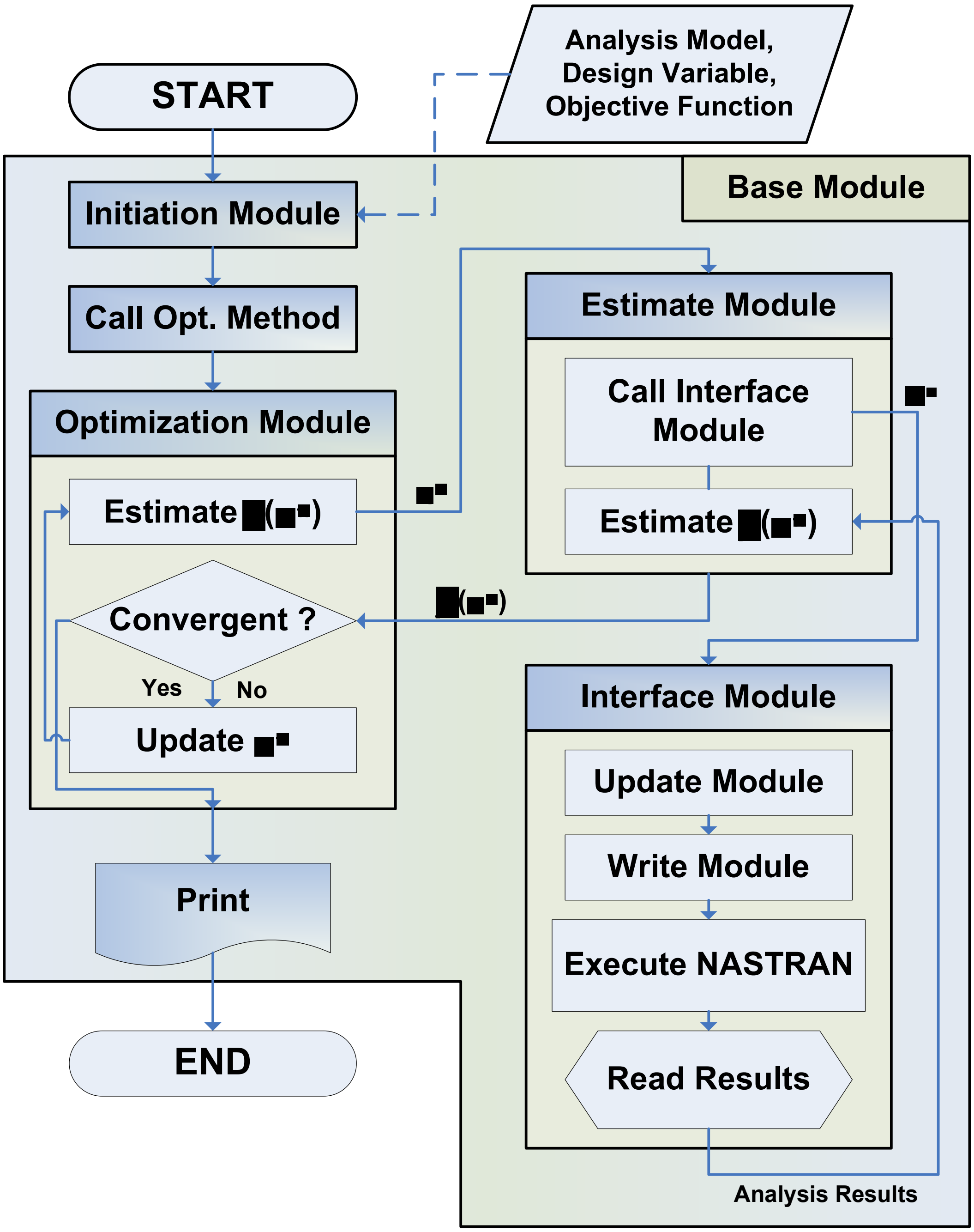


Fig. 3.1 Flowchart of OPTSHIP

3. Application of Optimum Design on an Actual Container Ship

In this study, two cases are presented as examples to show the process of searching for the optimum design of an actual containership. In these examples, the optimization process is carried out using verification and validation of the proposed optimization algorithm. In case 1, we compared and reviewed the optimized values of OPTSHIP and NASTRAN optimization. While in case 2 we are concerned about the results of optimization using R-tabu and GA as optimizers of the OPTSHIP.

The objective of this optimization is to find the design variables $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)^T$ to minimize the vibration response $f(\boldsymbol{x})$ under the constraint condition $g(\boldsymbol{x})$. This optimization problem can be stated as

$$\begin{aligned} &\text{Find design variables vector } \boldsymbol{x} \\ &\text{Minimize } \; f(\boldsymbol{x}) \\ &\text{Subject to } \; g_k(\boldsymbol{x}) \le 0, \quad k = 1, 2, \ldots, n \\ &\qquad\qquad\quad \boldsymbol{x}^L \le \boldsymbol{x} \le \boldsymbol{x}^U \end{aligned} \tag{3.1}$$

where n is the number of constraint conditions.

3.1 Analysis Model

The 2400 TEU containership is chosen as an analysis model [19] as shown in Fig. 3.2. The principal dimensions are shown in Table 3.1. The whole ship is idealized using a complete finite element (FE) model for the vibration analysis. The FE model has been constituted of a fine mesh with 3 or 4 frame spacing for the deckhouse, engine room and after body, and a relatively coarse mesh for the fore body. Table 3.2 shows the total number of nodes and elements used in this model. The main engine is also incorporated into the FE model of the whole ship to take account for realistic behavior. Furthermore, decks, bulkheads, continuous walls and side shell plates have been modeled as membrane element while girders and stiffeners are modeled as truss elements. Other descriptions for analysis model are as follows:

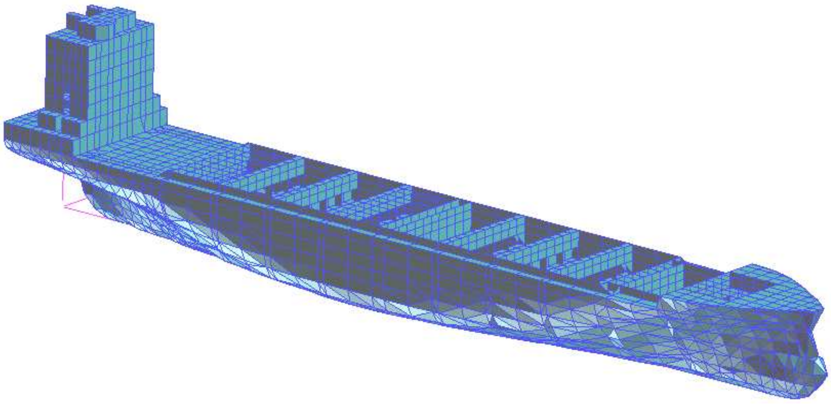


Fig. 3.2 Full model of a container ship

Table 3.1 Principal data of ship

Items	Value
Length overall	208 m
Length between perpendiculars	196 m
Breadth	29.8 m
Depth	16.4 m
Draft design	10.2 m

Table 3.2 Description of finite element model

Item	Value
Number of total nodes	4,558
Number of used elements	11,781
Plate thickness	8.0 - 32.0 mm

3.1.1 Lightship weight and cargo mass

The steel weight is automatically considered by the material properties of idealized elements while some structure not being modeled and lighter equipment components have been taken into the FE model by an appropriate adjustment of material density. Meanwhile, the mass of heavy equipments has been applied as nodal masses. The deadweight including cargo mass, ballasting liquid and the others which are documented including in the trim and stability calculation [20] has been distributed to the corresponding nodal points as concentrated masses in accordance with loading conditions.

3.1.2 Hydrodynamic mass

The virtual mass method(VMM) module in NASTRAN is used to calculate the hydrodynamic masses due to surrounding water. By this module, three dimensional hydrodynamic masses dependent upon vibration modes have been automatically applied to corresponding nodal points.

3.1.3 Damping

The vibration response are dependent upon various kinds of damping such as structural damping caused by the cargo, outfitting etc. and material damping by the surrounding water. The actual value of damping is very difficult to predict for the ship vibration analysts since it depends on the mode shapes and loading conditions. In this analysis, the modal damping coefficients linearly dependent on the frequency have been employed such as 0 % at 0 Hz and 6 % at 20.0 Hz.

3.1.4 Main excitation

The main excitation sources are the guide force moments of main engine and fluctuating pressures of propeller acting on the shell plates of the after body. External moments are given by an engine manufacturer (Sulzer) that used as the main engine excitation. Table 3.3 shows the dominant excitation sources of the main engine. The magnitude of the guide force moment linearly depends on the main engine revolution. Because the guide force is dependent upon the maximum pressure in the cylinder and thus the guide force moment is roughly proportional to the main engine revolution as the following formulation.

$$M_r = M_{MCR} \times \left(\frac{N_r}{N_{MCR}} \right)^k \tag{3.2}$$

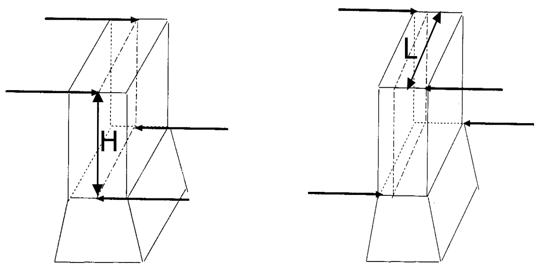
where, M_r and M_{MCR} mean relevant moment and moment at maximum continuous rating (*MCR*) speed (rpm), respectively. N_r and N_{MCR} mean the relevant speed of the main engine and the main engine speed at *MCR* condition respectively, k equals to 1 for guide force moment and internal moment. This is converted into force (F_H, F_X) using the following formulas.

$$F_H = \frac{M_H}{2H}, \quad F_X = \frac{M_X}{L} \tag{3.3}$$

where M_H and M_X mean *H*-moment and *X*-moment of the main engine respectively, H and L are height and breadth of the main engine respectively as shown in Fig. 3.3.

Table 3.3 Excitation moment of main engine at rating speed (97 rpm)

Excitation order	Lateral moment (± kN·m)	
	X-moment	*H*-moment
2nd	129	0
3rd	513	0
4th	550	0
5th	0	0
6th	0	1193



(a) *H*-moment (b) *X*-moment

Fig. 3.3 Exciting force of main engine

The propeller excitation due to fluctuating pressure have been considered at once and twice blade passing frequency. The fluctuating pressure has been calculated with the software HPUF-3A [21] developed by Massachusetts Institute of Technology for a region covering longitudinally from the transom to three times diameter of propeller by using the measured wake data of the model test. The calculated pressure has been distributed to corresponding nodal points of the outer shell. For reference, hull surface forces used for the forced vibration analysis are documented in Table 3.4. The calculated propeller pressure has been adjusted in accordance with the propeller speed by using the following formula.

$$P_r = P_{MCR} \times \left(\frac{N_r}{N_{MCR}} \right)^3 \tag{3.4}$$

where P_r and P_{MCR} mean relevant pressure and pressure at MCR speed respectively, N_r and N_{MCR} mean relevant speed of propeller and propeller speed at MCR condition.

Table 3.4 Excitation force of propeller

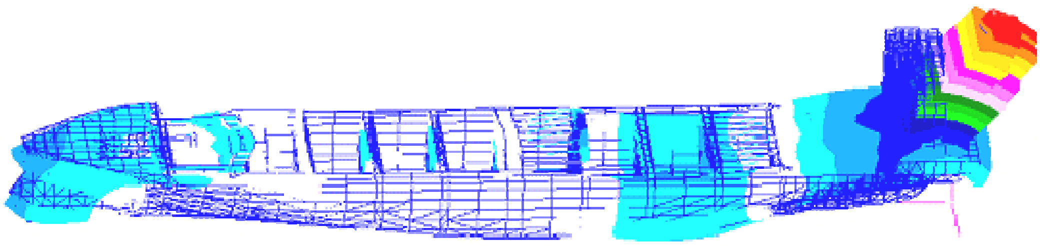
Excitation order	Excitation force (kN)		
	ΣF_x	ΣF_y	ΣF_z
1st	48.00	1.70	82.00

3.1.5. Location of forced vibration response

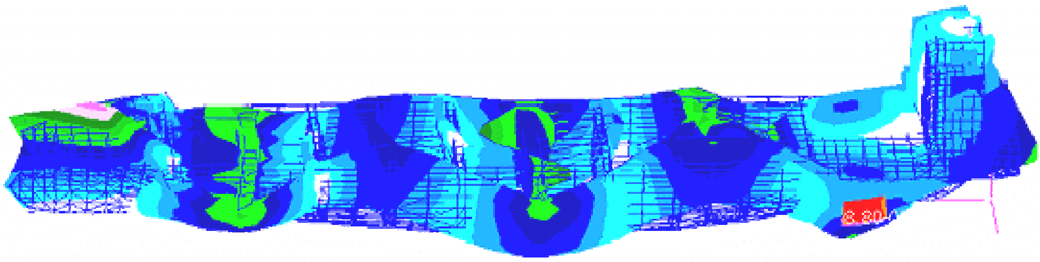
Several nodal points representing the overall vibration behavior are selected for the evaluation of the forced vibration response. The selected points come from the front wall of the deckhouse on the port side of navigation deck.

3.2 Free Vibration Analysis

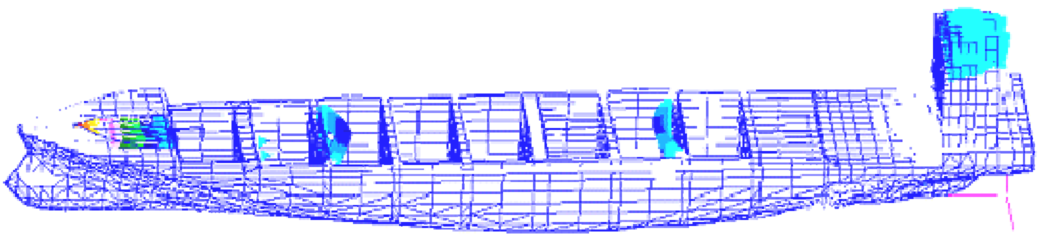
Free vibration analysis has been carried out up to the natural frequency of 18.3 Hz with the Lanczos method by means of NASTRAN. The frequency of 18.3 Hz, which corresponds to the main engine speed of 110 rpm, is far away from the *MCR* speed of 97 rpm, it is quite high to cover the highest dominant excitation frequency of propeller 2nd order (16.17 Hz). Among the free vibration analyses, several primary natural frequencies and the corresponding mode shape of the deckhouse are shown in Fig. 3.4.



(a) Longitudinal mode of deckhouse coupled with hull girder mode and double bottom (4.59 Hz)

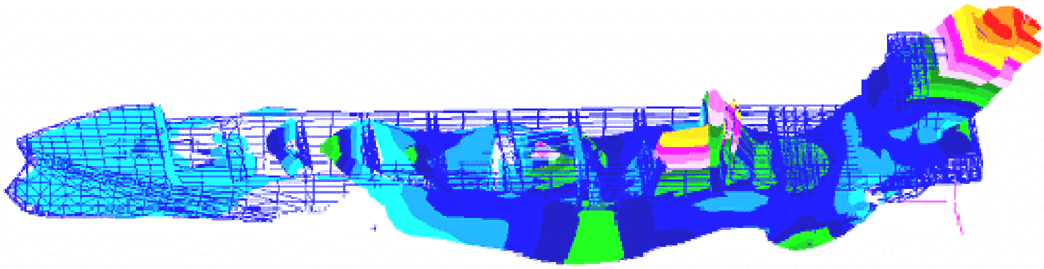


(b) Torsional mode of deckhouse coupled with hull girder (6.44 Hz)

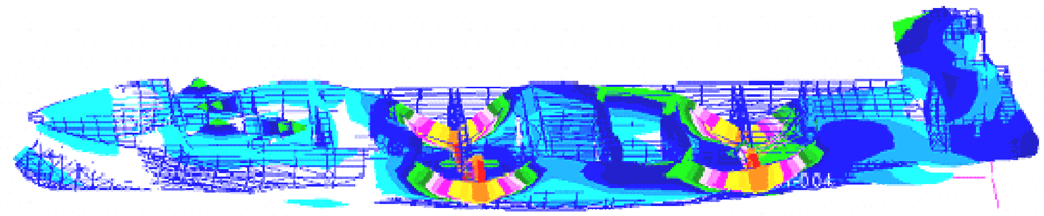


(c) Torsional mode of deckhouse coupled with hull girder (7.00 Hz)

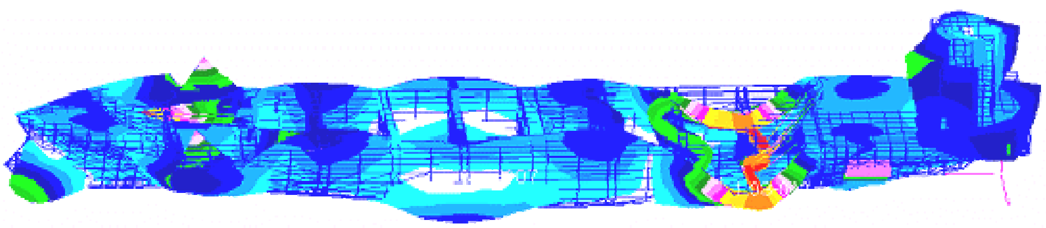
Fig. 3.4 Typical modes and natural frequencies of deckhouse area and stern of a ship (*Continued*)



(d) Longitudinal mode of deckhouse coupled with after body and double bottom (8.36 Hz)

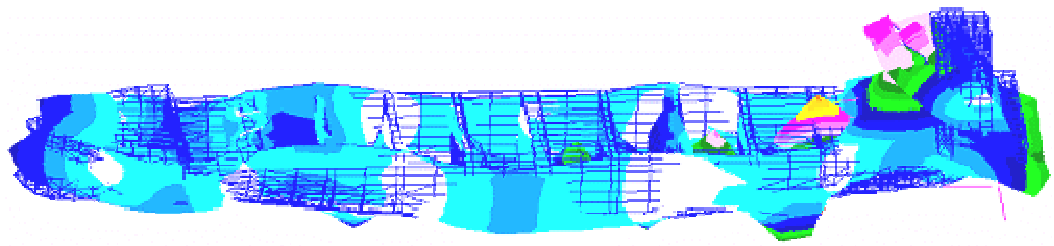


(e) Torsional mode of deckhouse coupled with after body (8.41 Hz)



(f) Torsional mode of deckhouse coupled with hull girder (9.24 Hz)

Fig. 3.4 Typical modes and natural frequencies of deckhouse area and stern of a ship (*Continued*)



(g) Longitudinal mode of deckhouse coupled with after body and double bottom (10.62 Hz)

Fig. 3.4 Typical modes and natural frequencies of deckhouse area and stern of a ship

3.3 Case 1: To verify the reliability and performance of the OPTSHIP

In the first optimization, we compared and evaluated the optimization results of NASTRAN optimization module and OPTSHIP to verify the reliability and performance of the proposed algorithm. The components of *H*-moment and *X*-moment of the main engine and the fluctuation force of the propeller were considered as the excitation source as shown in Tables 3.3 and 3.4. The method of global optimization needs more time for running than the local ones and many design variables increase the running time to converge. It is strongly recommended that the design variables should be reduced to save the running time for optimization. In this study, we divided the selection of design variables into two stages for reducing running time. In the first stage, we selected candidate design variables which are the plate thickness of deckhouse, shell expansion, engine room which have effects on the vibration response of the deckhouse directly. According to the area which has a different plate thickness and another tier, the design variables are selected differently from each other. However, if they are a symmetric structure, the same design variables are selected. We defined the interest range of design variables as the area highly affects the vibration mode of the deckhouse as shown in Fig. 3.5. Fig. 3.6 shows the typical design variables in the first stage. In the second stage, we carried out sensitivity analysis for design variables which were selected in the first stage by NASTRAN. Here, design variables which have the most sensitive values among the candidate design variables were selected as the final design variables. The number of the final design variables is reduced from 319 to 64 through sensitivity analysis.

Finally, we conducted the optimization with design variables which were decided by the second stage. The lower and upper limits of the design variables were decided as 90 ~ 140 % of the original design variables.

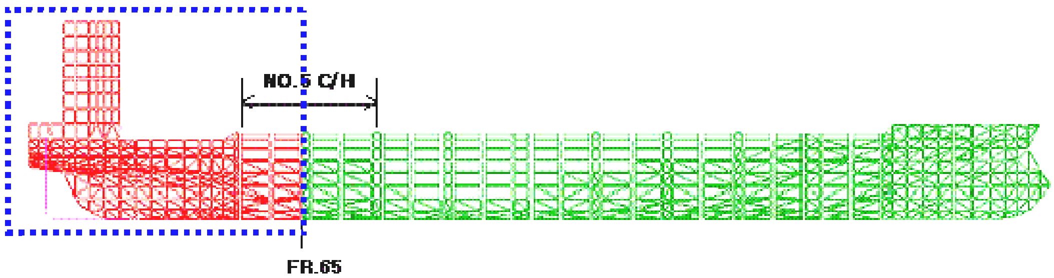
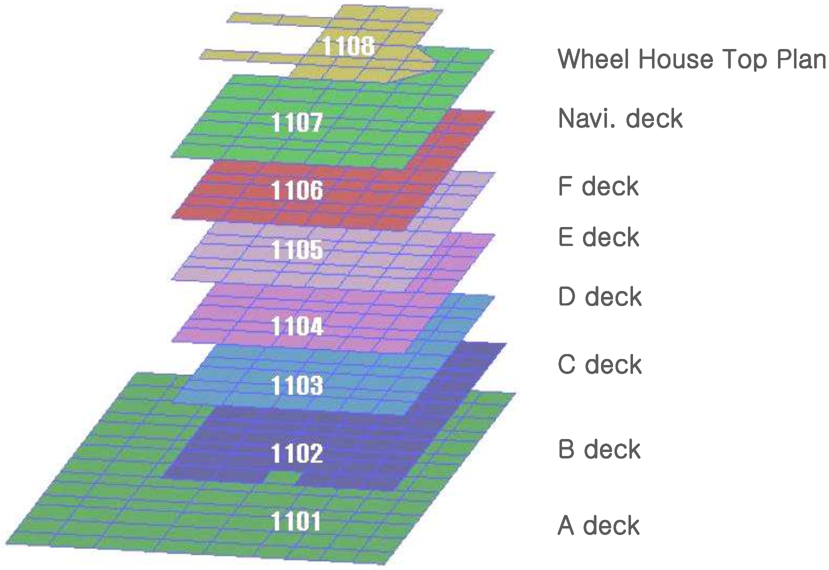
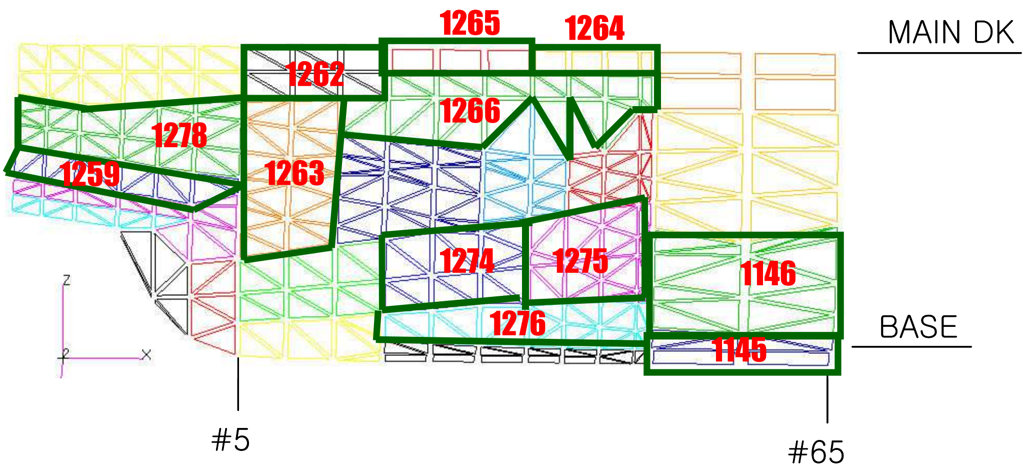


Fig. 3.5 The interest range of design variable in the containership

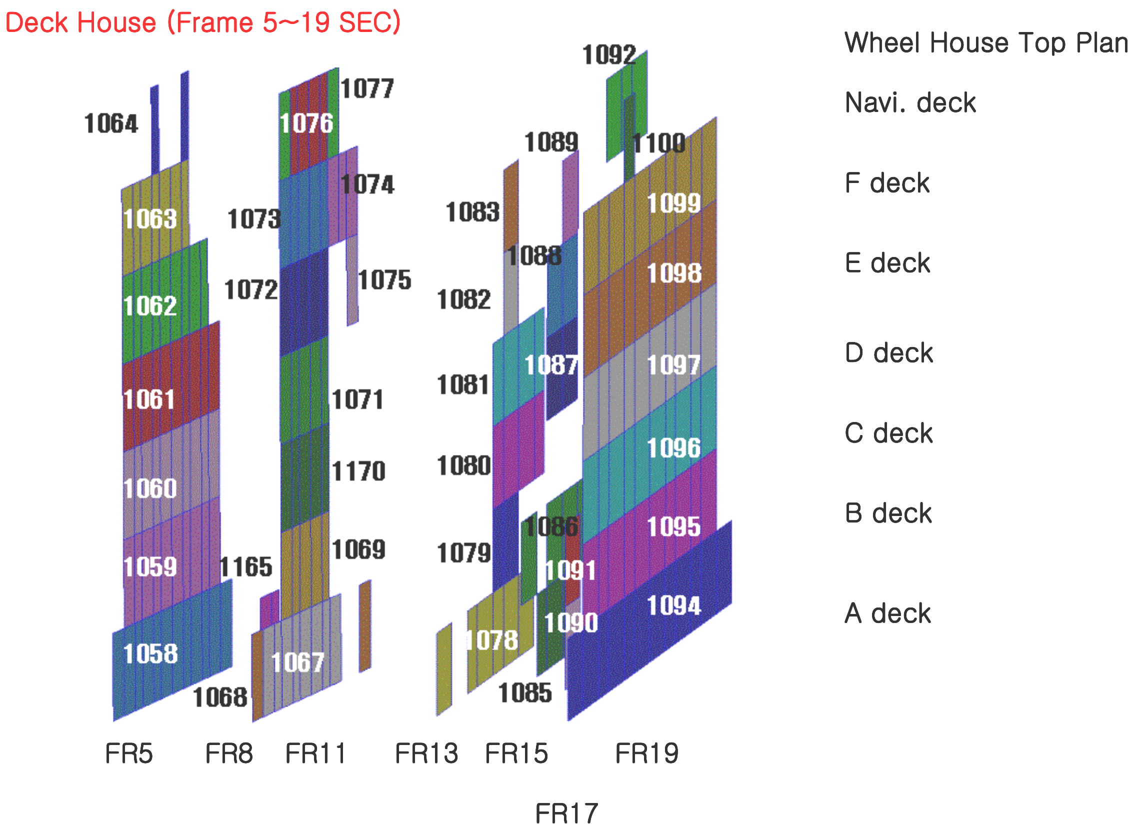


(a) Deckhouse (PLAN)



(b) Shell expansion

Fig. 3.6 An example of design variables (*Continued*)



(c) Deckhouse (Frame 5 ~ 19 SEC)

Fig. 3.6 An example of design variables

To minimize the forced vibration response, the objective function is considered as the rms value. The maximum value among the rms value at each direction is taken as the rms value of vibration velocity response. Each rms value is obtained by taking root of the average value. The average value is taken as the square of velocity response V_q which is multiplied by weighting factor w_q based on the international standard ISO 6954 [22], three directions are considerable (longitudinal, transverse and vertical) in the case of the response position within the interest frequency range.

$$f(x) = \max\left[\sqrt{\frac{1}{N}\sum_{q=1}^{N} w_q \cdot V_{q,L}^2}\,,\ \sqrt{\frac{1}{N}\sum_{q=1}^{N} w_q \cdot V_{q,T}^2}\,,\ \sqrt{\frac{1}{N}\sum_{q=1}^{N} w_q \cdot V_{q,V}^2}\right] \qquad (3.5)$$

where N is the number of peak considered, $V_{q,L}$, $V_{q,T}$ and $V_{q,V}$ are the vibration velocity amplitude of qth frequency at longitudinal, transverse and vertical directions, respectively.

This value is the realistic requirement which has been applied on the conventional ship construction. The upper and lower limits of the frequency range for forced vibration response are considered as the Rth frequency component of the main engine as follows

$$\begin{aligned} f_{\mathrm{upper}} &= R \times \frac{MCR \times 1.07}{60} \ \text{(Hz)} \\ f_{\mathrm{lower}} &= R \times \frac{MCR \times 1.07}{120} \ \text{(Hz)} \end{aligned} \qquad (3.6)$$

In this thesis, the speed of the main engine is considered as 107 % of MCR which takes into account safety margin. The interest frequency range is determined by Eq. (6) as shown in Table 3.5. The optimized results of the rms values for H-moment of the main engine which is the dominant excitation source for the deckhouse are shown in Fig. 3.7. Here, horizontal axis means rotating speed (rpm) of the main engine while vertical axis means vibration velocity response of longitudinal, transverse and vertical direction from top to bottom. The maximum responses in the longitudinal direction which is the dominant mode of the deckhouse were obtained at about 93 rpm which is close to NCR. They are 12.41 mm/s, 11.82 mm/s and 8.25 mm/s for original model, NASTRAN optimization module and OPTSHIP, respectively. It can be concluded that we can obtain the better result by using OPTSHIP than NASTRAN optimization module. Table 3.6 shows the comparison of objective function between original and optimized model considering the main excitation source of the subject ship. This table shows that the values of objective function of H-moment of the main engine represent 5.51, 5.28 and 4.28 for the original model, NASTRAN optimization module and OPTSHIP, respectively. According to the results, the variation of objective function of OPTSHIP is much improved by 79.8 % than that of the original one.

We compared the variation of final design variables considering the excitation force as *H*-moment of the main engine between OPTSHIP and NASTRAN optimization module as shown in Table 3.7. According to the results, the results of NASTRAN are convergent to the lower or upper limit of design variables which have been already defined. In the case of the OPTSHIP, however, design variables are distributed throughout the constraints ranges. This is assumed that the optimization by NASTRAN module converges to the local optima while the OPTSHIP does to the global optima.

Table 3.5 Interesting frequency range for four excitation components

Excitation components	Frequency range (Hz)
H-moment 6th	5.19 - 10.38
Propeller force 1st	4.32 - 8.65
X-moment 4th	3.46 - 6.92
X-moment 3rd	2.60 - 5.19

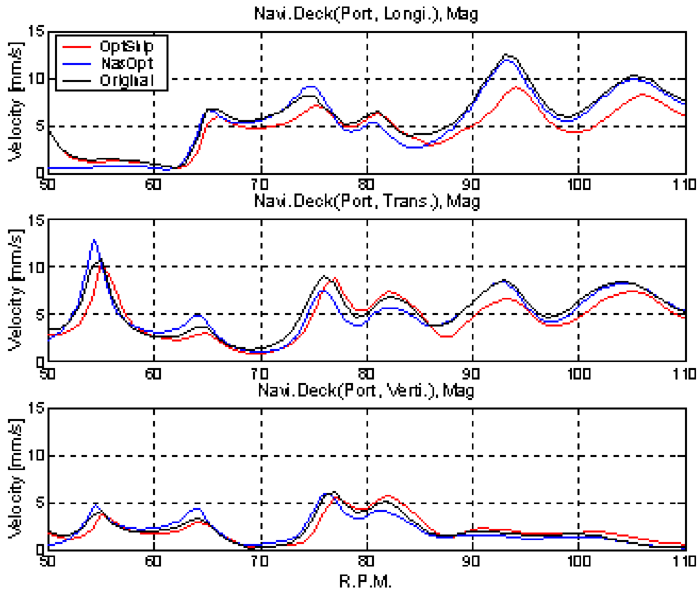


Fig. 3.7 Optimized results of rms value for *H*-moment of main engine

Table 3.6 Comparison of objective function between original and optimized model

Exciting force	Original model	NASTRAN optimum module		OPTSHIP	
		Value	Variation	Value	Variation
X-moment 3rd	7.83×10^{-5}	2.04×10^{-5}	-73.9%	1.59×10^{-5}	-79.8%
X-moment 4th	2.32×10^{-5}	2.18×10^{-5}	-6.3%	2.12×10^{-5}	-8.9%
Propeller 1st	7.40×10^{-5}	4.81×10^{-4}	-35.0%	5.10×10^{-4}	-31.1%
H-moment 6th	5.51×10^{-5}	5.28×10^{-3}	-4.3%	4.28×10^{-3}	-22.3%

Table 3.7 Comparison of optimization result (*H*-moment 6th, rms value)

SHELL ID No.	Original value (mm)	Variation of design variables (%)		SHELL ID No.	Original value (mm)	Variation of design variables (%)	
		NASTRAN	OPTSHIP			NASTRAN	OPTSHIP
1019	8.00	138	127	1201	17.00	140	112
1020	12.22	118	130	1203	30.00	140	130
1021	8.10	140	137	1204	22.33	90	137
1022	8.10	140	125	1205	15.78	90	94
1024	8.10	119	94	1206	10.80	90	106
1025	8.10	90	95	1208	10.90	114	139
1058	9.80	140	131	1220	30.00	90	116
1059	8.10	109	138	1222	14.00	90	112
1060	8.10	97	138	1248	13.00	90	133
1063	8.10	104	123	1262	62.00	90	121
1093	12.22	90	140	1266	77.00	90	93
1094	12.22	94	138	1272	62.00	90	105
1095	9.80	92	128	1280	62.00	90	99
1096	8.10	95	135	1284	15.78	90	126
1097	8.10	90	134	1287	12.00	90	100
1101	12.22	101	136	1289	13.00	90	135
1102	9.80	124	130	1290	15.78	90	123
1103	9.10	104	108	1291	23.56	140	120
1104	9.80	90	99	1295	12.00	98	125
1105	9.10	90	114	1297	14.00	90	137
1106	9.10	90	100	1305	11.67	139	125
1107	9.10	90	90	1318	19.00	90	127
1151	13.56	126	120	1321	16.56	90	126
1156	28.00	134	97	1322	16.56	90	136
1160	12.56	139	138	1324	22.00	90	129
1165	11.67	101	108	1325	17.78	90	102
1188	17.33	90	99	1333	18.00	90	107
1189	11.67	90	106	1334	18.00	90	112
1191	11.22	90	111	1335	19.00	90	107
1198	17.78	140	131	1336	21.00	90	110
1199	19.00	90	104	1342	17.78	90	134
1200	17.00	90	140	1343	11.22	93	111

3.4 Case 2: To verify the utility of the global algorithm in OPTSHIP

The second optimization was conducted to verify the utility of the global algorithm which is not only R-tabu method but also another optimization method. The optimization with R-tabu and GA method were carried out, and the results were compared with each other. In this case, the 3rd order component of *X*-moment of the main engine was applied as the excitation force. The procedure of selecting a design variable is the same as that of the first optimization method. However, the areas of all design variables were re-adjusted as close as possible in the FE model. Because the sensitivity changes according to the areas of the design variables. In the first stage, therefore the number of design variables was 279 shell elements while it was 319 elements in Case 1. And it was reduced from 279 to 51 elements in the second stage. Table 3.8 shows the variation of the number of design variable in the first and second optimizations. The objective function in this optimization is considered as the peak value. This is the maximum vibration velocity amplitude *V*, which was taken within frequency range at the three directions as follows.

$$f(x) = \max[V(\text{frequency, direction})] \tag{3.7}$$

The range of interest frequency and weighting factor are different. The upper and lower limits of interest frequencies are defined as follows:

$$\begin{aligned} f_{\text{upper}} &= R \times \frac{MCR \times 1.07}{60} \ \text{(Hz)} \\ f_{\text{lower}} &= R \times \frac{NCR - 10}{60} \ \text{(Hz)} \end{aligned} \tag{3.8}$$

where *NCR* is the normal continuous rating speed (rpm).

The lower limit frequency applied is the speed of ($NCR - 10$) rpm which is used to avoid the resonance of the structure. The weighting factor is 1.0 within the interest frequency, 0.5 for the others

$$w(f) = 1.0: \text{ if } f_{\text{lower}} < f < f_{\text{upper}} \\ \quad = 0.5: \text{ else} \tag{3.9}$$

Fig. 3.8 shows the comparison of the original model and optimization results of longitudinal, transverse and vertical directions at port side of navigation deck. In this figure, the horizontal axis represents interest frequency range (83~104 rpm) and vertical axis represents vibration velocity at a specific point. The values of objective function are reduced further about 8.65 % for GA and 6.7 % for R-tabu than that of the original one as shown in Table 3.9. Their vibration responses on the navigation deck in the longitudinal direction (4.59 Hz) are 0.1214 mm/s, 0.1133 mm/s and 0.1109 mm/s for original, R-tabu and GA, respectively. As a result, GA method gets better results than R-tabu one. However considering the evaluation number, it is not easy to decide which method is better than the others. We reached a conclusion that both algorithms can be easily applied to the complex system.

Table 3.8 The reduction of the number of design variables

Optimization	The number of design variables	
	Candidate	Final
Case 1	319	64
Case 2	279	51

Table 3.9 Comparison of objective function between original and optimized model

Exciting force	Original model	R-tabu		GA	
		Value	Variation	Value	Variation
X-moment 3rd	$1.214{\times}10^{-4}$	$1.133{\times}10^{-4}$	- 6.7%	$1.109{\times}10^{-4}$	- 8.65%

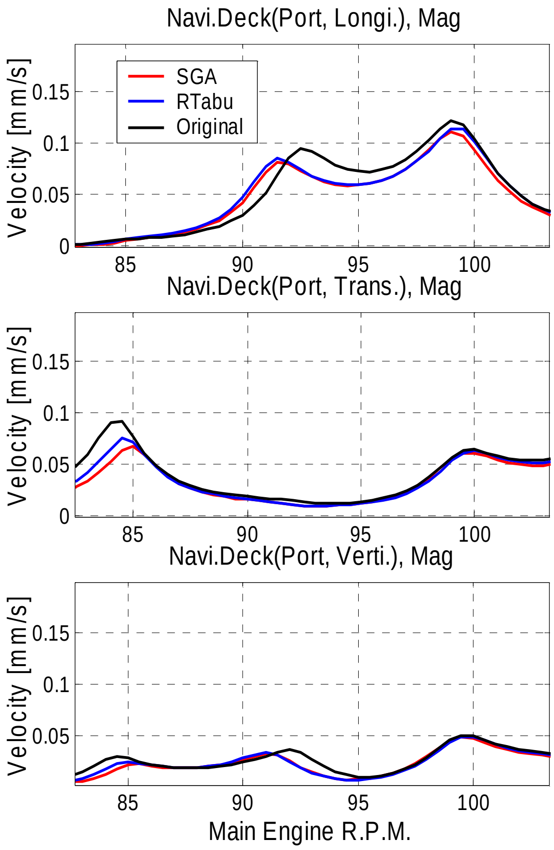


Fig. 3.8 Comparison of optimization results of peak value for *X*-moment (3rd) of main engine

4. Conclusions

This paper proposed a new optimization framework called OPTSHIP which employs NASTRAN as external analyzer for optimization method to obtain a global optimum solution of a large ship structure. The algorithm is applied to search the optimum design of an actual containership model for verification and validation purpose of the proposed algorithm. Moreover, to save running time, we carried out sensitivity analysis for design variables by NASTRAN. According to the analysis of results, we found out that the OPTSHIP have searched better solution than the NASTRAN optimum module. Furthermore, the optimization using R-tabu and GA optimization method was carried out to verify the performance of OPTSHIP as an optimizer. We confirmed that both algorithms get good results in this example. Finally, it can be concluded that the proposed optimization algorithm in this study can serve and contribute to solve the vibration problems on the ship structure.

References

[1] G. Sitton, 1993, *MSC.NASTRAN Basic Dynamic Analysis User's Guide*, The MacNeal–Schwendler Corp.

[2] T. Ray, R. P. Gokarn and O. P. Sha, 1995, "A global optimization model for ship design," Computers in Industry, Vol. 26, pp. 175-192

[3] J. S. Arora, 1989, *Introduction to Optimum Design*, McGraw-Hill, Inc.

[4] *MSC.NASTRAN Design Sensitivity and Optimization User's Guide*, 1994, The MacNeal-Schwendler Corp.

[5] B. David and R. Michael, 1994, *MSC.NASTRAN DMAP Module Dictionary*, The MacNeal-Schwendler Corp.

[6] D. E. Goldberg, 1989, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley Pub.

[7] N. Hu, 1992, "Tabu search method with random moves for globally optimal design," International Journal of Numerical Methods in Engineering, Vol. 10, pp. 299-310

[8] B. S. Yang, B. G. Choi, S. B. Jeon and D. J. Kim, 1988, "Optimum allocation of pipe support using combined optimization algorithm by genetic algorithm and random tabu search method," Korea Fuzzy Logic and Intelligent System Society, Vol. 8, No. 3, pp. 71-79 (in Korean)

[9] B. S. Yang, S. B. Jeon , B. G. Choi and Y. H. Yu, 1998, "Optimum design of damping plate using combined optimization algorithm by genetic algorithm and random tabu search method," Trans. KSME (A), Vol. 22, No. 7. pp. 1258-1266 (in Korean)

[10] M. Kitamura, H. Nobukawa and F. Yang, 2000, "Application of a genetic algorithm to the optimal structural design of a ship's engine room taking dynamic constraints into consideration," Marine Science and Technology, Vol. 5, pp. 131-146

[11] B. G. Choi and B. S. Yang, 2000, “Optimum shape design of rotor shafts using genetic algorithm,” Journal of Vibration and Control, Vol. 6, No. 2, pp. 207-222

[12] B. G. Choi and B. S. Yang, 2001, “Optimal design of rotor-bearing systems using immune-genetic algorithm,” Trans. ASME Journal of Vibration and Acoustics, Vol. 123, No. 3, pp. 398-401

[13] C. A. C. António, 2002, “A multilevel genetic algorithm for optimization of geometrically nonlinear stiffened composite structures,” Struc Multidisc Optim , Vol. 24, pp. 372-386

[14] B. S. Yang, S. P. Choi and Y. C. Kim, 2005, “Vibration reduction optimization design of a steam-turbine rotor-bearing system using a hybrid genetic algorithm,” Struct Multidisc Optim, Vol. 30, pp. 43-53

[15] F. Z. Feng, Y. H. Kim and B. S. Yang, 2005, “Applications of hybrid optimization techniques for model updating of rotor shafts,” Struct Multidisc Optim (in press)

[16] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, 1983, “Optimizations by simulated annealing,” Science, Vol. 220, pp. 671-680.

[17] J. D. Song and B. S. Yang, 2005, “Optimum design of short journal bearing by using enhanced artificial life optimization algorithm”, Tribology International, Vol. 38, No. 4, pp. 403-411

[18] *MSC.PATRAN User’s Guide (Ver. 8)*, 1998, The MacNeal–Schwendler Corp.

[19] S. H. Oh and S. H. Choi, 1996, “Technical report for vibration analysis of containership,” DW H. No.4052. Daewoo Shipbuilding & Marine Engineering Co., Ltd. (in Korean)

[20] Trim & Stability Calculation for DW H.No.4052, 1996, Daewoo Shipbuilding & Marine Engineering Co., Ltd.

[21] J. E. Kerwin, 1990, *A Numerical Method for the Analysis of Cavitating*

Propellers in Nonuniform Flow, MIT-PUF-3A User's Manual, MIT Report No.90-3, Feb.

[22] ISO 6954 (2000) Mechanical vibration - Guidelines for the measurement, reporting and evaluation of vibration with regard to habitability on passenger and merchant ships.

Ⅳ. Nonlinear Integer Programming Based on GA Parameter Optimization

1. Introduction

Optimization is utilized to determine the size or the geometric shape of the structure to obtain the maximum performance using minimal material with safety and availability of the target structure [1]. From a mathematical point of view, the optimization is a process to obtain the design variables which are the maximizing or minimizing a desired objective function while satisfying the prevailing constraints. Usually, optimization needs a lot of time to get desired information due to the repetitive process. Recently, optimization has been widely applied for decreasing the weight of structure in various industrial fields such as aerospace, civil, mechanical engineering, etc., through integrating methodology of engineering design with the technology of computer-aided engineering (CAE) and increased computer speed.

In shipbuilding, optimum design has been used in many areas. However, the applications are limited and most researches have emphasized on static optimization which does not consider dynamic factors [2-5]. Also, optimum design for ship vibration has rarely been studied. Yang et al. [6] worked on optimization of ship stiffened panel, and Kitamura et al. [7] carried out the optimal structural design of a ship’s engine room. They did the optimization considering static and dynamic constraints, and adopted a simplified analysis model to enhance computing efficiency during optimization process. Also, Yang et al. [8] and Kong et al. [9] proposed a new optimization tool called OPTSHIP

(OPTimization for SHIP) which combines NASTRAN that used as a solver with global optimization algorithm, namely random tabu search method (R-tabu) to enhance optimum design for vibration reduction of ship structure. In OPTSHIP, NASTRAN is called externally and used for calculation of the objective function. They applied it to the vibration optimal design of global and local containership using continuous variables, respectively.

In general, the final design variables that have been chosen are bigger in size than optimized results to consider the safety margin in an actual application. Of course, this choice enables the structure stronger than the optimized model. But, the natural frequency of the structure may be closer to its resonance and more dangerous than optimized design in the vibration aspects. [10, 11].

However, theses optimization results are not suitable for actual application since the selections of web and girder sizes are limited in standard shaped steel members that commercially available. Therefore, the real values programming need to be extended to non-linear integer programming (NIP) in order to apply directly the optimized result to an actual design. NIP was suggested by Reiter and Rice for solving a general quadratic programming problem in 1966, where both the objective and constraint function are quadratic. They applied a modified gradient-type method, very similar to the methods used in the continuous nonlinear programming field, to solve the problem. NIP is an intrinsically hard problem. There are rich literatures on the NIP problems [12, 13]. However, many of the NIP problems are computationally intractable and their solutions are NP complete. Thus, the optimal solutions can not be obtained in a reasonable amount of time and memory [14]. Heuristic algorithms were developed to find approximations to the optimum. Current research is on the effective approximation methods such as genetic algorithm (GA) [15], simulated annealing (SA) [16] and tabu search (TS) [17]. These methods are mainly used to solve combinatorial optimization problems. Recently, it is remarkable to apply GA to effectively solve a combinatorial problem as one of the solution methods. GA is a very powerful tool for solving a NIP problem like optimal design of system reliability and can handle any kind of objective functions and constraints.

In this thesis, we present a method for solving the NIP problem to get the best compromise solution easily while holding a nonlinear property by using the genetic algorithm for an actual design. GA is used to obtain global solutions in the proposed method. As we know, there are many parameters have to be set for GA, such as the population size, mutation probability, crossover probability, selection methods and crossover methods that greatly affect the accuracy and calculation time of optimum solution. The setting process is hard for users, and there are no rules to decide these parameters. In order to overcome these demerits, the optimization for these parameters has been also conducted using GA itself. The reliability of the proposed method has been demonstrated for solving the vibration problem on compass deck of a ship.

2. Nonlinear Integer Programming (NIP)

As with most domains of engineering, nonlinear problems are often solved by generating a sequence of solution to linear problems which in some sense approximate the original nonlinear problem. The NIP problem can be mathematically expressed as follows:

Maximize (or minimize) $f(\boldsymbol{x})$

Subject to the constraints $\boldsymbol{x}^{\boldsymbol{L}} \le \boldsymbol{x} \le \boldsymbol{x}^{\boldsymbol{U}}, \quad \boldsymbol{x} \in Z^n,$

where, $\boldsymbol{x} = (x_1,\ x_{2,}\ \cdots,\ x_n)^T$ is a vector of variables or unknown in the NIP problem, Z^n is a set of n-dimensional integer vector, $\boldsymbol{x}^{\boldsymbol{L}} = (x_1^L,\ x_2^L,\ \cdots,\ x_n^L)^T \in Z^n$ and $\boldsymbol{x}^{\boldsymbol{U}} = (x_1^U,\ x_2^U,\ \cdots,\ x_n^U)^T \in Z^n$ are n-dimensional constant vectors, and $\boldsymbol{x}^{\boldsymbol{L}} \le \boldsymbol{x}^{\boldsymbol{U}}$.

Let $S = \{ \boldsymbol{x} : \boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U, \boldsymbol{x} \in Z^n \}$ denote a solution space, thus $f : S \to R$ is a cost function. Some of the NIP problems can also be viewed as integer and combinatorial optimization problem [18].

3. The Optimization for GA Parameters

As mentioned before, initial parameters setting of GA is hard for users and influences the optimization results. For example, a proper mutation probability can increase the probability for a getting a global optimum solution due to the diversity of solutions, but high mutation probability has effect on the convergent speed. Also, population size is critical to get a precise solution. If population size is too small, it may fail to reach the optimal solution, on the contrary if not, it brings out falling-off in efficiency.

In this study, the optimization for GA parameters is carried out based on GA itself using trial function. The flowchart for optimization is shown in Fig. 4.1, where, N_e, N_{ea} and A_{ne} mean the number of evaluation, all evaluation and average evaluation, respectively. GAF represents GA for function optimization, while GAP does GA parameter optimization. GAP consists of design variables with GAF's parameters, namely, population size, crossover probability, mutation probability, selection method and crossover method. When the GAF is terminated, the individual fitness of GAP is determined on the number of average evaluations of objective function in GAF. GAF will be terminated if the condition of Eq. (4.7) is satisfied. Since GA is probability search, the same processes are repeated M times (M = 5) using the same parameters, and the number of average evaluation is obtained. The objective function of GAF is defined as the trial function Eq. (4.4). Design variables and constraints are expressed as Eqs. (4.5) and (4.6). This trial function has a global solution ($f(\mathbf{x} = 0) = 0$) and 27^N local solutions. There are 10 design variables that are the same as the number of design variables for the applied structure, which is shown in Eq. (4.5).

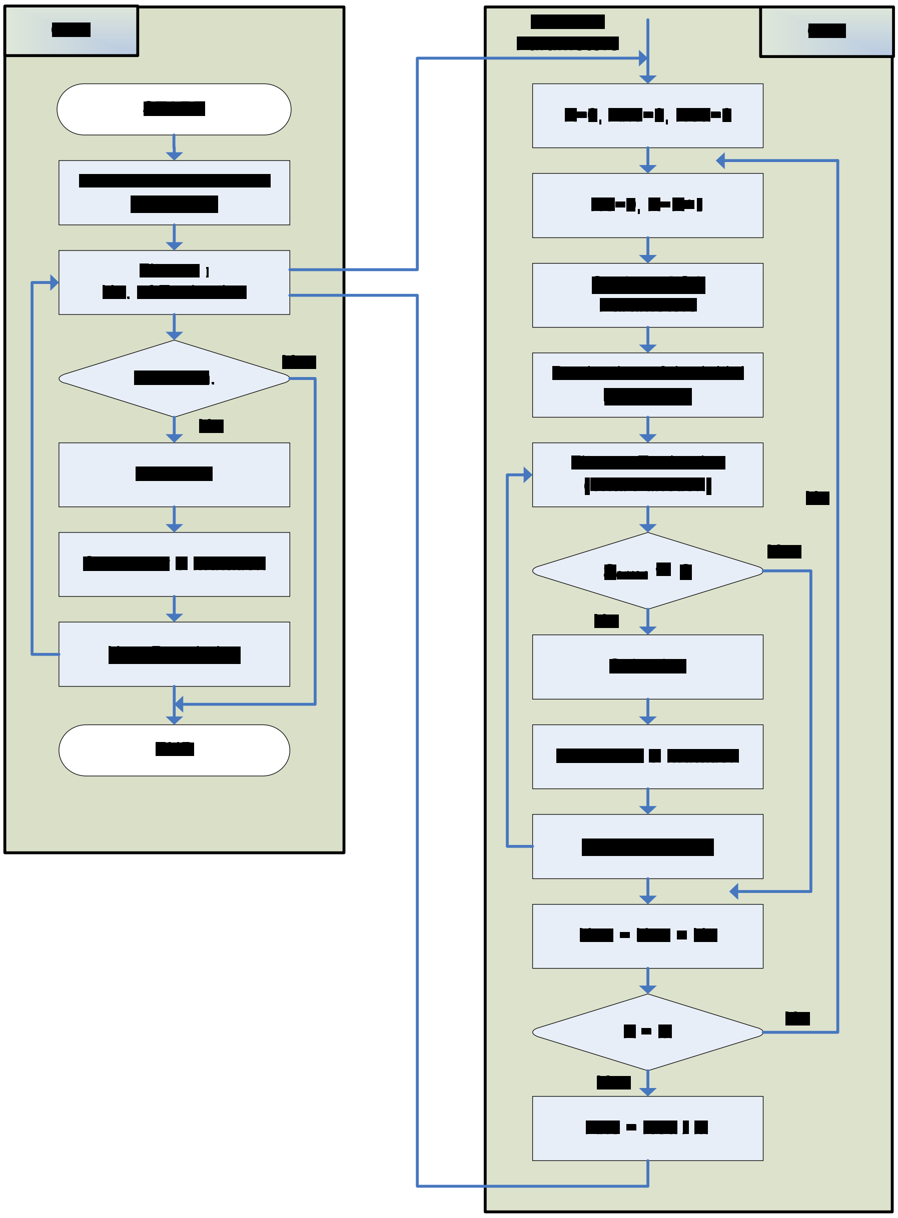


Fig. 4.1 Flowchart of GA for parameter optimization

3.1 Formulation for Optimization

In this study, five GA parameters are considered for optimization: population size, crossover probability, mutation probability, selection method and crossover method which have effect on genetic calculation, as shown in Eq. (4.2).

3.1.1 Formulation for GAP

Minimize

$$f(\boldsymbol{x}) = A_{ne} \ (= N_{ea} / M) \tag{4.1}$$

where M means the number of evaluation of GAF for identical GA parameters, here $M = 5$.

Design variables

$$\boldsymbol{x} = \{P_s \ P_c \ P_m \ M_s \ M_c\}^T \tag{4.2}$$

Subject to:

$$P_s = \{10, 20, 30, \cdots, 180, 190\} \tag{4.3}$$

$$P_c = \{0.1, 0.2, \cdots, 0.8, 0.9\}$$

$$P_m = \{0.005, 0.01, \cdots, 0.065, 0.9, 0.95\}$$

$$M_s = \{\text{Roulette wheel selection, Ranking based selection}\}$$

$$M_c = \{\text{Simple crossover, Multi-point crossover, Uniform crossover}\}$$

where, P_s, P_c and P_m are population size, crossover probability and mutation probability of GAF, respectively. M_s and M_c represent selection method and crossover method, respectively.

3.1.2 Formulation for GAF

Minimize

$$f(x) = \sum_{i=1}^{N}[x_i^2 - \alpha_i \cos(\frac{2\pi x_i}{\beta_i}) + \alpha_i] \tag{4.4}$$

Design variables:

$$\boldsymbol{x} = \{x_1 \ \ x_2 \ \ x_3 \ \ x_4 \ \ x_5 \ \ x_6 \ \ x_7 \ \ x_8 \ \ x_9 \ \ x_{10}\}^{\mathrm{T}} \tag{4.5}$$

subject to

$$-10 \le x_i \le 100, \ i = 1, \ldots, N \tag{4.6}$$

where, $\alpha_i = 1$, $\beta_i = 4$, $N = 10$.

The termination condition of GAF is as follows:

$$\varepsilon_{error} = \frac{1}{N}\sum_{i=1}^{N} \frac{\left| x_i^{(best)} - x_i^{(opt)} \right|}{\Delta x_i} \le \varepsilon \tag{4.7}$$

where, ε is predefined value, here 0.01, $x_i^{(best)}$ is the best solution at each generation, $x_i^{(opt)}$ is the optimum solution of ith design variable. Δx_i represents the interval of design variables. The optimization results are shown in Table 4.1.

Table 4.1 Comparison of GA parameters before and after optimization

Parameter	Original	Optimum
Population size	100	10
Crossover probability	0.8	0.1
Mutation probability	0.1	0.065
Selection method	Roulette	Ranking
Crossover method	Uniform	Simple

In order to confirm the validity of optimization results, the objective function is evaluated using other parameters and optimum parameters. The compared results are shown in Table 4.2 and Figs. 4.2 – 4.6. According to the results, the optimum parameters are good for the accuracy and speed of convergence in GA.

Based on the above demonstration, the optimum GA parameters can be used for the integer optimum design of a compass deck.

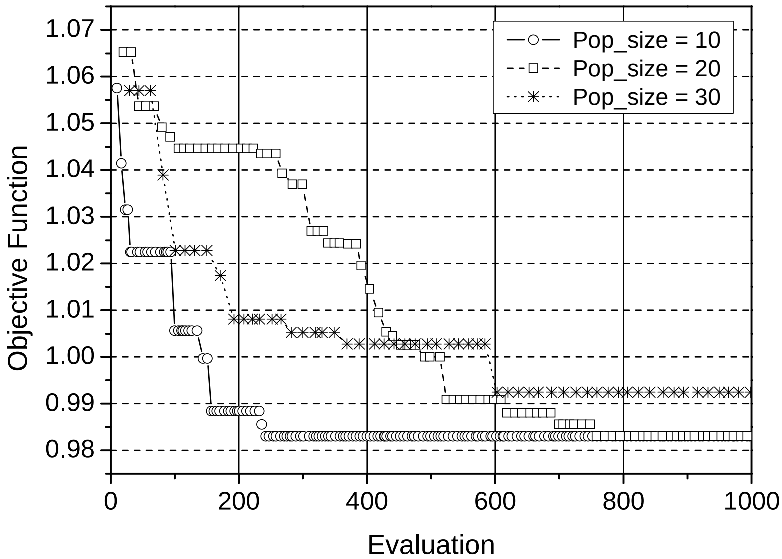


Fig. 4.2 Comparison of population size

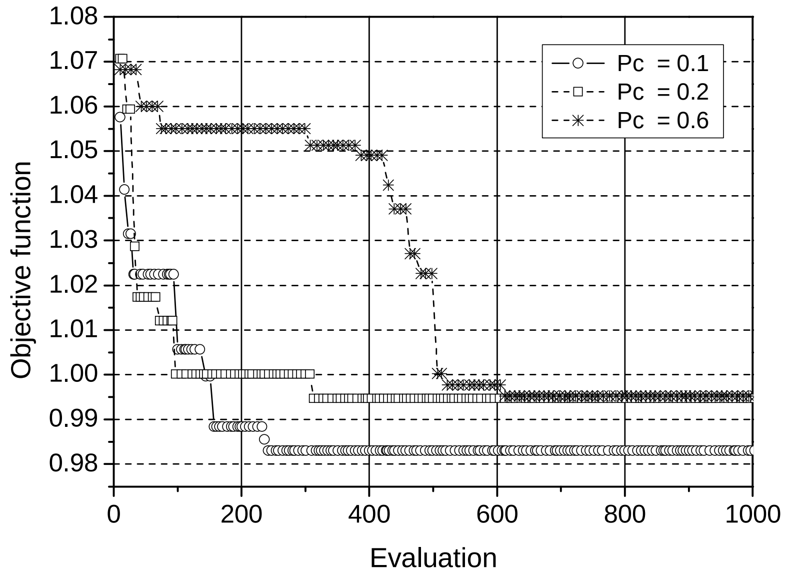


Fig. 4.3 Comparison of crossover probability

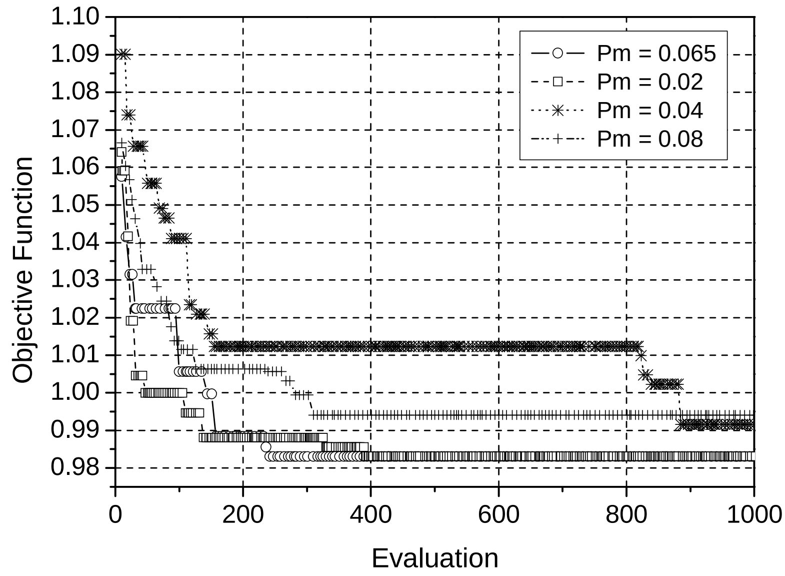


Fig. 4.4 Comparison of mutation probability

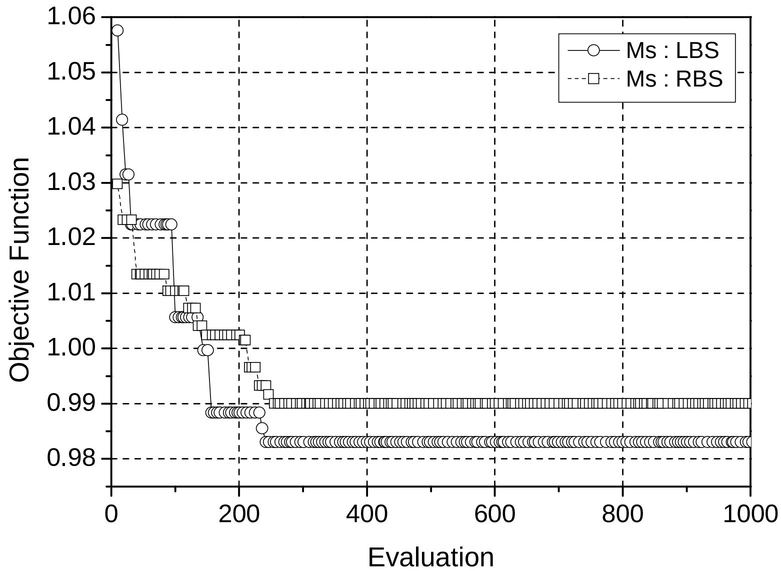


Fig. 4.5 Comparison of selection method

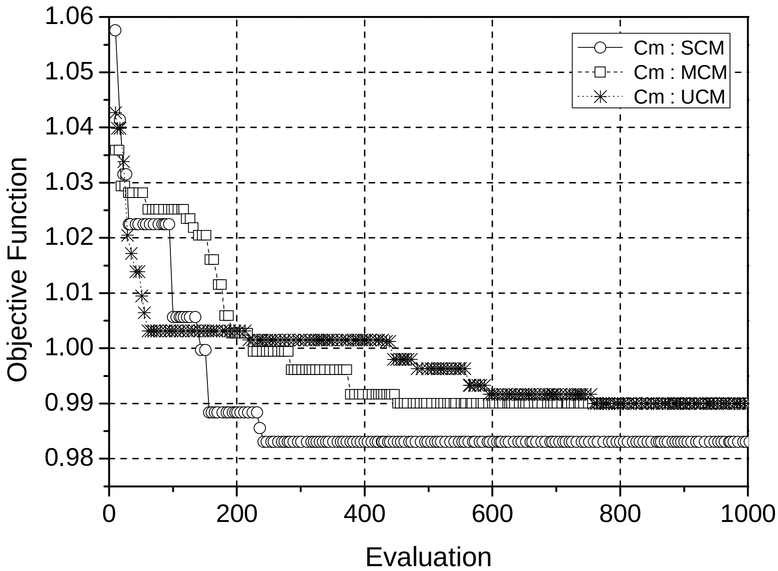


Fig. 4.6 Comparison of crossover method

Table 4.2 Comparison of the optimization results according to GA parameters

Parameters		Function value	No. of evaluation
Population Size	**10**	**0.98304**	**242**
	20	0.98304	758
	30	0.99246	1000
Crossover Probability	**0.1**	**0.98304**	**242**
	0.2	0.99469	1000
	0.6	0.99524	1000
Mutation Probability	0.02	0.98304	392
	0.04	0.99159	1000
	0.065	**0.98304**	**242**
	0.08	0.99410	1000
Selection Method	Roulette Wheel Selection	0.99004	1000
	Ranking Based Selection	**0.98304**	**242**
Corssover Method	**Simple Crossover**	**0.98304**	**242**
	Multi-point Crossover	0.99004	1000
	Uniform Crossover	0.99004	1000

4. Vibration Analysis of Compass Deck

The vibration analysis of a compass deck is carried out by using NASTRAN which is a commercial finite element program and widely used for big structures like a ship. Fig. 4.7 shows the model of compass deck and radar mast. In particular, the girder and web of a compass deck is displayed as three dimensions, which are design variables in this study. Fig. 4.8 shows the design variables and boundary conditions of a compass deck. The main dimensions of subject ship are shown in Table 4.3 and the main data of modeling of compass deck are listed in Table 4.4.

Considering the precision of analysis and time consuming modeling process, the range of modeling of a compass deck is constrained to the its deck only based on experience of analysis and impact test at the shipyard. The boundary conditions for the model are specified: the simple supports are used to the bulkheads shown as solid lines and two pillars are connected between the compass deck and the navigation deck. Fixed supports are used at the cross-points of bulkheads. We modeled the arbitrary box at the location of radar mast and considered the weight by adjusting the mass density, because the weight of radar mast on compass deck has considerable effect on the vibration behavior of the compass deck. Table 4.5 shows the specification of main excitation sources.

In general, the design for avoiding local structure resonance of a ship requires that natural frequency of a structure must be two times higher than the blade passing frequency of propeller under the maximum rpm of main engine. In this study, design target frequency is set above 18.87 Hz which is considered safety margin and twice blade passing frequency of the propeller (16.33Hz).

Fig. 4.9 shows the first three modes and natural frequencies of a compass deck structure by NASTRAN. The 1st mode (16.78 Hz), which frequently occurred on the compass deck during the voyage, is the vertical mode on front area of the radar mast as shown in Fig. 4.9(a). The lower part of compass deck could not be installed the bulkhead because of problems securing the workspace compared to the other cabins. Therefore, the corresponding weak stiffness of the structure results in low natural frequency which is close to the main excitation source of ship. In this model, the 1st natural frequency of structure is also within the resonance region where twice blade passing frequency of propeller is 16.33 Hz. The safety margin is only 2.8 %, which is usually 10 %. The 2nd and 3rd modes occurred on the sides of the compass deck. Their natural frequencies are higher than the main excitation frequency of the ship and the possibility of resonance is rare. So, in order to design a safe structure, the 1st vertical mode of a compass deck is specified as the concerned mode in this study.

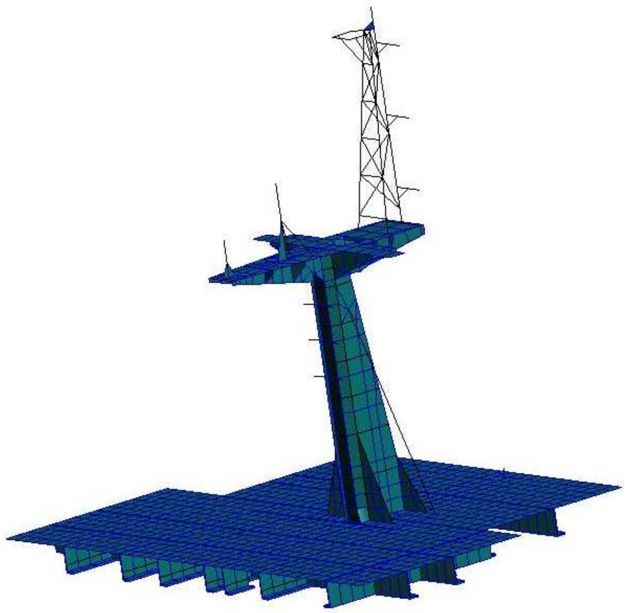


Fig. 4.7 Model of compass deck and radar mast

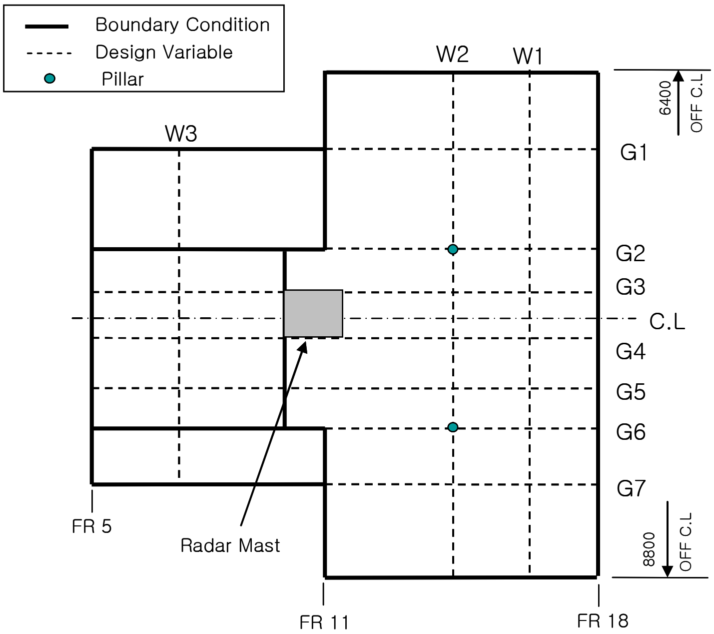


Fig. 4.8 Design variables and boundary conditions of a compass deck

Table 4.3 Principal dimensions

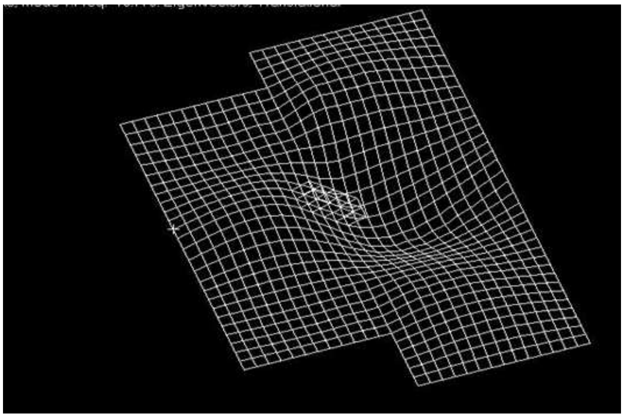
Length overall	208 m
Length between perpendiculars	196 m
Breadth moulded	29.8 m
Depth moulded	16.4 m
Draft design	10.2 m

Table 4.4 Main data of modeling

Geometry data		Material data	
Plate thickness	8.0 m	Elastic modulus	206 GN/m^2
Web & girder size	250×90 ×10/15A	Poisson ratio	0.3
Frame/ longitudinal space	800 mm	Mass density	7850 kg/m^3

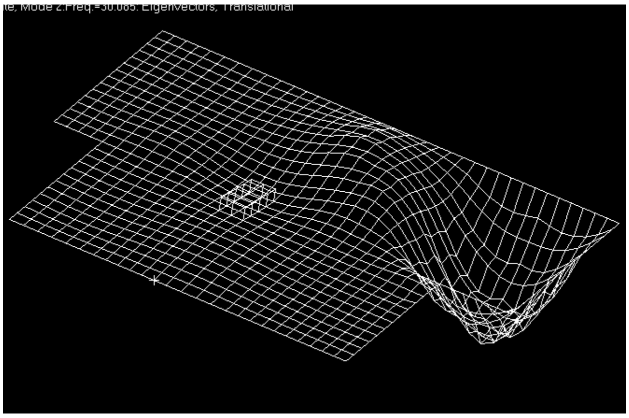
Table 4.5 Specification of main excitation sources

Excitation source	MCR	Excitation	
		Order	Frequency
Main engine (6RTA72U)	98 rpm	3rd	4.90 Hz
		4th	6.53 Hz
		6th	9.80 Hz
Propeller (Blade: 5EA)		1st	8.17 Hz
		2nd	16.33 Hz

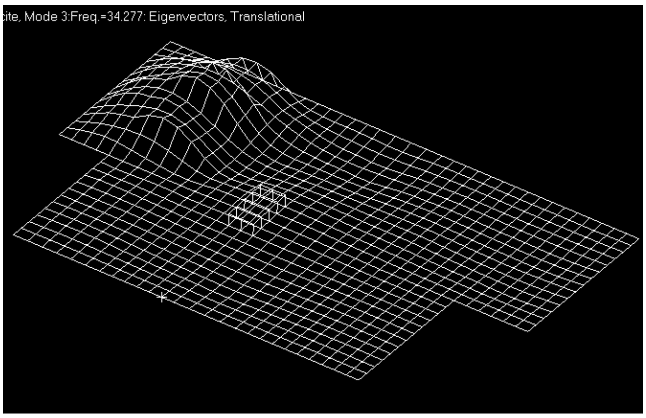


(a) 1st mode (16.78 Hz)

Fig. 4.9 Mode shapes of compass deck (*Continued*)



(b) 2nd mode (30.09 Hz)



(c) 3rd mode (34.28 Hz)

Fig. 4.9 Mode shapes of compass deck

5. Optimum Design of Compass Deck

5.1 Formulation for Optimum Design

5.1.1 Design variables

The main vibration mode on the compass deck is a global mode of girder and web in vertical direction. One of the most important factors is the stiffness of girder and web. In this study, the size of girder and web on the compass deck in Fig. 4.8 are defined as design variables in Eq. (4.8).

$$\boldsymbol{x} = \{W_1\ W_2\ W_3\ G_1\ G_2\ G_3\ G_4\ G_5\ G_6\ G_7\}^T \quad (4.8)$$

where W and G mean the size of girder and web, respectively.

5.1.2 Constraints

The web length of stiffener L_w is restricted as Eq. (4.9) due to ceiling height, namely the distance from navigation deck to compass deck, which is based on the building specification. The stiffener is also restricted to available standard sizes in the fields as shown in Table 4.6.

$$200 \leq L_w \leq 550 \text{ mm} \quad (4.9)$$

Also, the basic concept of local vibration design is the minimization of the response at each point. However, it is difficult to evaluate how much the excitation force influences on local structure. So, in this study, natural frequency of the structure is restricted as Eq. (4.10) which is considered safety margin with twice blade passing frequency of the propeller.

$$\omega_n \geq 18.87\text{Hz} \quad (4.10)$$

Fig. 4.10 shows section of stiffener and plate.

Table 4.6 Corresponding cross section of steel members

Stiffener size	$L_w \times L_f \times T_w / T_f$
200A	200 × 90 × 9/14
250A	250 × 90 × 10/15
300A	300 × 90 × 11/16
350A	350 × 100 × 12/17
400A	400 × 100 × 12/18
450A	450 × 125 × 11.5/18
500A	500 × 150 × 11.5/18
550A	550 × 150 × 12/21

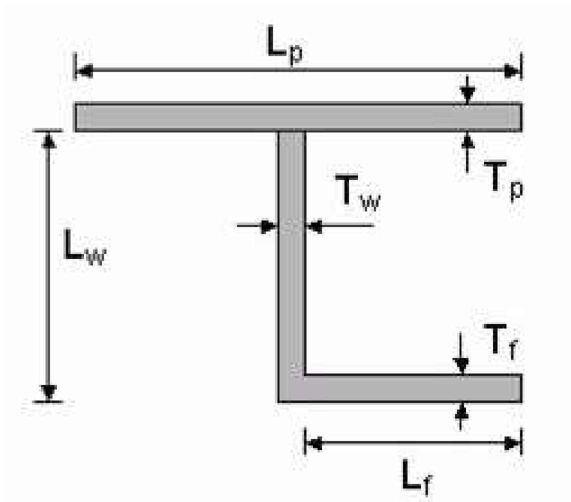


Fig. 4.10 Section of stiffener and plate.

5.1.3 Objective function

In general, the main target of the optimum design is to decrease the weight of structure or to reduce the vibration level on the specific point with avoiding the resonance between excitation source and the natural frequency of subject structure. In this thesis, we considered the objective function as two cases as follows:

1) Case 1

The objective function combines linearly the weight of compass deck, W_1 with maximum vibration velocity response, R_1 at an interest range (below MCR rpm) like Eq. (4.11).

$$\text{Minimize } f(\boldsymbol{x}) = \alpha(W_1/W_0) + \beta(R_1/R_0) \quad (4.11)$$

where, α and β are weighting factors. In this study, $\alpha = 1$, $\beta = 0$ [9]. W_0 means initial weight (including the weight of radar mast), R_0 is a basis vibration velocity response (vertical direction, the maximum amplitude at center).

2) Case 2

The objective function combines linearly the weight of compass deck, W_1 with natural frequency of structure is expressed by Eq. (4.12). The objective is to get an economic and sound structure to reduce the weight of stiffener and to increase the natural frequency.

$$\text{Minimize } f(\boldsymbol{x}) = \alpha\left(\frac{W_1}{W_0}\right) + \beta\left(\frac{\omega_t}{\omega}\right) \quad (4.12)$$

where, ω_t and ω mean target and current natural frequency, respectively. α and β are weighting factors. In this study, $\alpha = 0.5$, $\beta = 0.5$.

5.2 Optimization Results and Discussion

The optimum design was carried out to obtain an optimal size of web and girder on the compass deck to maintain the anti-vibration design of it. Nonlinear integer algorithm by GA is used as an optimal algorithm in order to apply directly the optimized result to an actual design. As stated above in section 3, the optimum GA parameters are applied to this problem.

Tables 4.7 and 4.8 show the results of the design variables before and after optimization for case1 and case2, respectively. It shows that the center girder of structure G_3 is increased 80% and the others are reduced 20% in case 1. In case 2, the center girder of structure G_4 is increased 120 % and the others are similar to case 1. These results indicate that the most reasonable modification method is to increase the stiffness of a member where the maximum amplitude exists in vibration mode. To get a higher natural frequency of structure, it is required to increase the stiffness of G_4 which is located in a wider area than that of G_3. The role of G_7 in case 2 supports the stiffness of G_4 due to limit stiffener size. Tables 4.9 and 4.10 show the natural frequency, vibration response at a MCR in a unit excitation force and the weight of compass deck before and after optimization for case 1 and case 2, respectively. According to the results, the 1st natural frequency increased 12.69% and 38.74% from 16.78Hz to 18.91Hz and 23.28Hz, and the safety margin with twice passing frequency of propeller correspondingly changed from 2.80% to 15.80% and 42.60% for case 1 and case 2, respectively. Therefore, the structure is free from the resonance. Moreover, the amplitude of vibration velocity response for case 1 and case 2 reduced 61.24% and 93.40%, respectively. The weights of stiffeners which are applied to design variables also decreased in spite of higher natural frequency and reduced the vibration response. In summary, the local vibration problems have been successfully solved by the proposed optimization method, which moves the natural frequency to a higher one without any additional weight. Fig. 4.11 shows the 1st vibration mode after optimization. Although there is a little change on the 1st vibration mode shape due to the mode of the global compass deck, the natural frequency increased based on the calculation result. And we confirmed that the vibration response at the MCR rpm has been significantly reduced as shown in Fig. 4.12 for case 1 and Fig. 4.13 for case 2, respectively.

Table 4.7 Comparison of original and optimal design variables for case 1

Design variable	Original	Optimum	Remarks
W_1	250	200	-20%
W_2	250	200	-20%
W_3	250	200	-20%
G_1	250	200	-20%
G_2	250	200	-20%
G_3	250	450	80%
G_4	250	200	-20%
G_5	250	200	-20%
G_6	250	200	-20%
G_7	250	200	-20%

Table 4.8 Comparison of original and optimal design variables for case 2

Design variable	Original	Optimum	Remarks
W_1	250	200	-20%
W_2	250	200	-20%
W_3	250	200	-20%
G_1	250	200	-20%
G_2	250	200	-20%
G_3	250	200	-20%
G_4	250	550	120%
G_5	250	200	-20%
G_6	250	200	-20%
G_7	250	250	0%

Table 4.9 Comparison of results for case 1

Item	Original	Optimum	Remarks
Natural frequency	16.78Hz	18.91Hz	12.69%
Response at MCR	10.50mm/s	4.07mm/s	-61.24%
Weight	2760kg	2537kg	-8.08%

Table 4.10 Comparison of results for case 2

Item	Original	Optimum	Remarks
Natural frequency	16.78Hz	23.28Hz	38.74%
Response at MCR	20.17mm/s	0.69mm/s	-93.48%
Weight	2760kg	2757kg	-0.11%

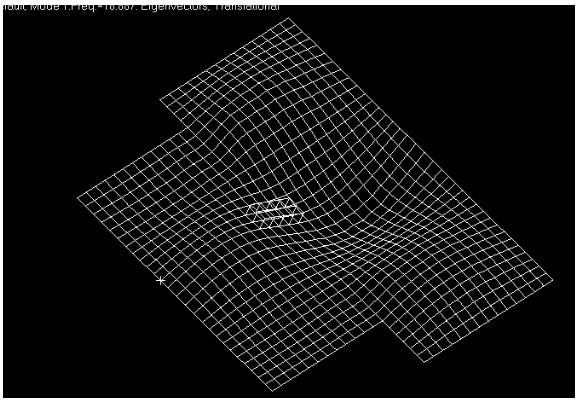


Fig. 4.11 Mode shape of compass deck after optimization

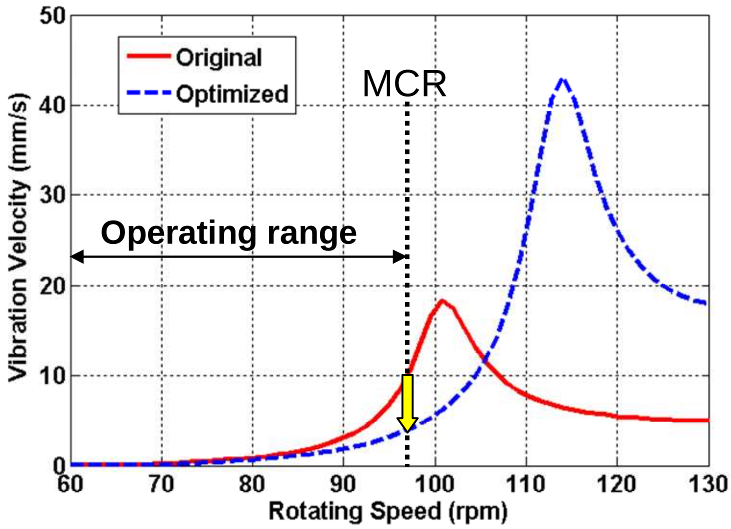


Fig. 4.12 Comparison of response between original and optimum results for case 1

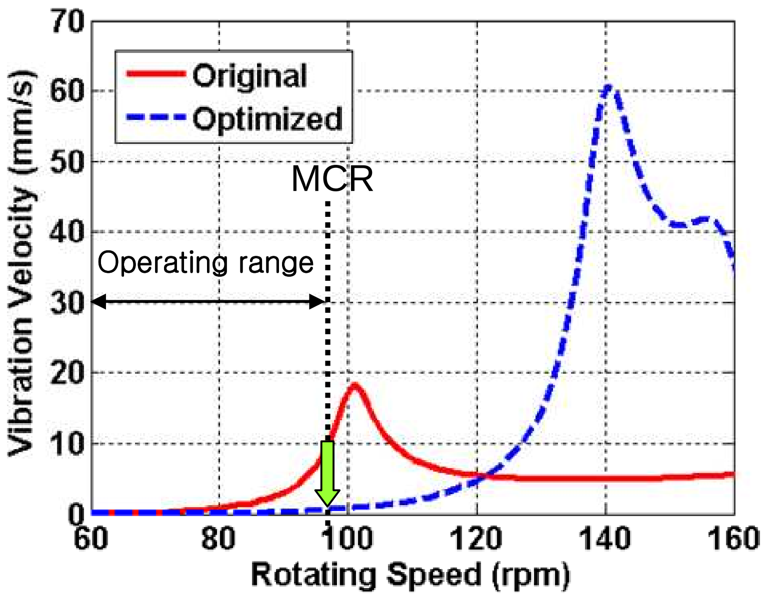


Fig. 4.13 Comparison of response between original and optimum results for case 2

6. Conclusions

In this study, we proposed non-linear integer programming to apply directly the optimized result to an actual design. GA is used to obtain global solutions in the proposed method. In order to get proper GA parameters, the optimization of GA parameters is also carried out through the trial function by GA itself. The reliability of the proposed method has been demonstrated for solving the vibration problem on compass deck of a ship. After optimization, local vibration problem has been successfully solved: the structure is free from resonance, safety margins increased, and the amplitude of vibration velocity response reduced without additional weight. The results indicated that the proposed method can be used as an optimum design tool in other structure optimization designs.

References

[1] J. S. Arora, 1989, *Introduction to Optimum Design*, McGraw-Hill, Inc.

[2] C. D. Jang, S. I. Seo and K. K. Kim, 1996, "A study on the optimum structural design of surface effect ships." Marine Structure, Vol. 9, pp. 519-544.

[3] H. W. Leheta and A. E. Mansour, 1997, "Reliability-based method for optimal structural design of stiffened panels," Marine Structure, Vol. 10, pp. 323-352.

[4] M. Valorani, D. Peri and E. F. Campana, 2003, "Sensitivity analysis methods to design optimal ship hulls," Optimization and Engineering, Vol. 4, pp. 337-364.

[5] O. Reyhan, 2004, "A review of recent ship vibration papers," The Shock and Vibration Digest, Vol. 36, No. 3, pp. 207-214.

[6] J. M. Yang, C. N. Hwang, S. F. Lin, C. C. Wu, and C. P. Chen, 1999, "The application of genetic algorithm on optimization of ship structural elements," in Proc. of the 13th Asian Technical Exchange and Advisory Meeting on Marine Structures, pp. 431-444.

[7] M. Kitamura, H. Nobukawa, and F. Yang, 2000, "Application of a genetic algorithm to the optimal structural design of a ship's engine room taking dynamic constraints into consideration," Marine Science and Technology, Vol. 5, pp. 131-146.

[8] B. S. Yang, Y. M. Kong, S. H. Choi, S. I. Chae, J. D. Song and Y. H. Kim, 2005, "Development of NASTRAN-based Optimization Framework for Vibration Optimum Design of Ship Structure," Trans. of the Korean Society for Noise and Vibration Engineering, Vol. 15, No. 11, pp. 1223-1231.

[9] Y. M. Kong, S. H. Choi, J. D. Song, and B. S. Yang, "Optimum design for vibration reduction of compass deck structure in ship," Structural Multidisciplinary Optimization. (submitted)

[10] T. Yokota, M. Gen, and Y. X. Li, 1996, "Genetic algorithm for non-linear mixed integer programming problems and its applications." Computer in Industrial Engineering, Vol. 30, No. 4, pp. 905-917.

[11] O. B. Augusto and A. Kawano, 1998, "A mixed continuous and discrete nonlinear constrained algorithm for optimizing ship hull structural design." Ocean Engineering, Vol. 25, No. 9, pp. 793-811.

[12] G. L. Nemhauser and L. A. Wolsey, 1988, *Integer and Combinatorial Optimization*, John Wiley and Sons, New York.

[13] R. G. Parker and R. L. Rardin, 1988, *Discrete Optimization*, Academic Press Inc., New York.

[14] M. R. Garey and D. S. Johnson, 1979, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, San Francisco.

[15] J. H. Holland, 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Michigan.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, 1983, "Optimization by simulated annealing," Science, Vol. 220, pp. 671-680.

[17] F. Glover, 1986, "Future paths for integer programming and links to artificial intelligence," Computers and Operations Research, Vol. 13, No. 5, pp. 533-549.

[18] G. L. Nemhauser and L. A. Wolsey, 1988, *Inter and Combinatorial Optimization*, Wiley.

Ⅴ. RSM-based Hybrid Evolutionary Algorithm

1. Introduction

The focus of many dynamic analyses is to find the maximum response and avoid the resonance in a given structure under all excitation forces. Usually, these features provide the basis of a design limit and are thus employed to determine the dynamic characteristics of a structure and its weight. For this reason, weight minimization for reducing the response and avoiding resonance has always been a major concern of design engineers. Many classic optimization methods and practical software have been developing and most of them are very effective, especially to solve the practical problems. However they are a hard task to find a global optimum solution for the system. To overcome this disadvantage, many search algorithms have been developed for searching for a global optimum solution. One of the most popular methods is the genetic algorithm (GA) [1, 2]. The GA is a technique in the field of evolutionary computation and it is a powerful and general global optimization method, which does not require the strict continuity of classical search techniques, instead it allows non-linearity and discontinuity to appear in the solution space. Due to the evolutionary characteristics, the GA can handle all kinds of objective functions and constraints defined on discrete, continuous, or mixed search spaces. However, the global access of the GA requires a computationally random search. So, the convergent speed to the exact solution is slow. Furthermore, the coding of the chromosome for a large dimensional problem will be very long, in order to get a more accurate solution. This results in a large search space and huge memory requirements for the computation. To overcome these demerits, many researchers have studied to develop many hybrid genetic algorithms which combined genetic algorithm with other ones [3, 4]. These can save computation time and find the global solution as far as it goes. However, new algorithms are requested for better accuracy and faster convergent speed to get an optimum solution in the complicated and big structures like ship.

In this thesis, to search for the optimum solution of multi-peak function in high accuracy and high speed, a new hybrid evolutionary algorithm is suggested, which combines the merits of the popular programs such as GA, tabu search method, response surface methodology (RSM) and simplex method. This algorithm, in order to improve the convergent speed that is thought to be the demerit of GA, uses RSM and simplex method. Though mutation of GA offers random variety, systematic variety can be secured through the use of tabu list of tabu search method. Especially, in the initial stages, GA's convergent speed can be improved by using RSM which is using the information on the objective function acquired through GA process and then making response surface (approximate function) and optimizing this. The optimum solution was calculated without the evaluation of an additional actual objective function, and the GA’s convergent speed could be improved. This method has been proven to be the efficiency by applying traditional test functions and comparing the results to GA. It also confirmed that the global optimum solution is being searched efficiently by applying the proposed algorithm to weight minimization where avoiding resonance of the fresh water tank located on the rear of the ship was considered.

2. Response Surface Methodology (RSM)

2.1 Introduction

RSM [5] is an optimization tool that was introduced in the early 1950's by Box and Wilson [6]. It is a collection of statistical and mathematical techniques that is useful for developing, improving, and optimizing processes. These techniques are employed in order to estimate the optimization function and to find search directions to sub-regions of the domain with improved and hopefully optimal solutions.

The most extensive applications of RSM are in particular situations where several input variables potentially influence some performance measure or quality characteristic of the process. Thus performance measure or quality characteristic is called the response. The input variables are sometimes called independent variables, and they are subject to the control of the scientist or engineer. The RMS usually contains three stages: 1) design of experiments, 2) response surface modeling through regression, and 3) optimization. The main advantage of RSM is the reduced number of experimental trial needed to evaluate multiple parameters and their interactions. The experimental data was utilized to build mathematical models using regression methods. Once an appropriate approximating model is obtained, this model can then be analyzed using various optimization techniques to determine the optimum conditions for the process. In general, the engineer is concerned with a product, process, or system involving a response y that depends on the controllable input variables $\xi_1, \xi_2, \cdots, \xi_k$. The relationship is

$$y = f(\xi_1, \xi_2, \cdots, \xi_k) + \varepsilon \tag{5.1}$$

where the form of the true response function f is unknown and perhaps very complicated, and ε is a term that represents other sources of variability not accounted for in f. Usually ε includes effects such as measurement error on the response, background noise, the effect of other variables, and so on. Usually ε is treated as a statistical error, often assuming it to have a normal distribution with mean zero and variance σ^2. If the mean of ε is zero, then

$$E(y) = \eta = E\left[f(\xi_1, \xi_2, \cdots, \xi_k) \right] + E(\varepsilon) = f(\xi_1, \xi_2, \cdots, \xi_k) \tag{5.2}$$

The variables $\xi_1, \xi_2, \cdots, \xi_k$ in Eq. (5.2) are usually called the natural variables, because they are expressed in the natural units of measurement, such as degrees Celsius(℃), pounds per square inch(psi), etc. In much RSM work it is convenient to transform the natural variables to coded variables $x_1, x_2, \cdots, x_k$, which are usually defined to be dimensionless with mean zero and the same standard deviation. In terms of the coded variables, the response function (5.2) will be written as

$$\eta = f(x_1, x_2, \cdots, x_k) \tag{5.3}$$

Because the form of the true response function *f* is unknown, we must approximate it. In fact, successful use of RSM is critically dependent upon the experimenter's ability to develop a suitable approximation for *f*. Usually, a low-order polynomial in some relatively small region of the independent variable space is appropriate. In many cases, either a first-order or a second order model is used.

The first-order model is likely to be appropriate when the experimenter is interested in approximating the true response surface over a relatively small region of the independent variable space in a location where there is little curvature in *f*.

For the case of two independent variables, the first-order model in terms of the coded variables is

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \tag{5.4}$$

The form of the first-order model in Eq. (5.4) is sometimes called a main effects model, because it includes only the main effects of the two variables x_1 and x_2. If there is an interaction between these variables, it can be added to the model easily as follows:

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 \tag{5.5}$$

This is the first-order model with interaction. Adding the interaction term introduces curvature into the response function.

Often the curvature in the true response surface is strong enough that the first-order model (even with the interaction term included) is inadequate. A second-order model will likely be required in these situations. For the case of two variables, the second-order model is

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^{\,2} + \beta_{22} x_2^{\,2} + \beta_{12} x_1 x_2 \tag{5.6}$$

This model would likely be useful as an approximation to the true response surface in a relatively small region.

The second-order model is widely used in response surface methodology for several reasons:

1) The second-order model is very flexible. It can take on a wide variety of functional forms, so it will often work well as an approximation to the true response surface.
2) It is easy to estimate the parameters β in the second-order model. The method of least squares can be used for this purpose.
3) There is considerable practical experience indicating that second-order models work well in solving real response surface problems.

In general, the first-order model is

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \tag{5.7}$$

and the second-order model is

$$\eta = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k} \beta_{jj} x_j^{\ 2} + \sum_{i=1}^{k} \sum_{j=2}^{k} \beta_{ij} x_i x_j \qquad (5.8)$$

In some infrequent situations, approximating polynomials of order greater than two are used. The general motivation for a polynomial approximation for the true response function *f* is based on the Taylor series expansion around the point $x_{10}, x_{20}, \cdots, x_{k0}$. For example, the first-order model is developed from the first-order Taylor series expansion

$$f \cong f(x_{10}, x_{20}, \cdots, x_{k0}) + \frac{\partial f}{\partial x_1} \Big|_{x=x_0}$$

$$+ \frac{\partial f}{\partial x_2} \Big|_{x=x_0} + \cdots + \frac{\partial f}{\partial x_1} \Big|_{x=x_0} \qquad (5.9)$$

where *x* refers to the vector of independent variables and x_0 is that vector of variables at the specific point $x_{10}, x_{20}, \cdots, x_{k0}$. In Eq. (5.9) we have only included the first–order terms in the expansion, thus implying the first-order approximating model in Eq. (5.7). If we were to include second-order terms in Eq. (5.9), this would lead to the second-order approximating model in Eq. (5.8).

Finally, let’s note that there is a close connection between RSM and linear regression analysis. For example, consider the model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \qquad (5.10)$$

The β’s are a set of unknown parameters. To estimate the values of these parameters, we must collect data on the system we are studying. Because, in general, polynomial models are linear functions of the unknown β’s, we refer to the technique as linear regression analysis.

Optimization theory consists of a body of numerical methods for finding and identifying the best candidate from a collection of alternatives without having to explicitly evaluate all possible alternatives [7]. In the context of RSM, empirical (mathematical) models are built using regression techniques on the results of a selected set of experiments. A well fitted model represents, approximately, all possible experiments with their experimental factors within the preset bounds. Through the use of optimization techniques, the optimum of the model corresponding to the experiment with conditions that will presumably produce the best result can thus be found. The final step is to perform experimental verification based on the optimal, experimental conditions. Among the optimization techniques, the steepest ascent (or descent) is commonly used, but the method is relatively inefficient and is a local optimization technique capable of finding only local optima. Global optimization techniques such as GA, tabu search method, etc., although even less efficient than the steepest ascent from the viewpoint of convergent speed, are considered. In this thesis, tabu search method is used as global schemes.

2.2 Sequential Nature of the Response Surface Methodology

Most applications of RSM are sequential in nature as follows:

Phase 0: At first some ideas are generated concerning which factors or variables are likely to be important in the response surface study. It is usually called a screening experiment. The objective of this factor screening is to reduce the list of candidate variables to a relatively few so that subsequent experiments will be more efficient and require fewer runs or tests. The purpose of this phase is the identification of the important independent variables.

Phase 1: The experimenter's objective is to determine if the current settings of the independent variables result in a value of the response that is near the optimum. If the current settings or levels of the independent variables are not consistent with optimum performance, then the experimenter must determine a set of adjustments to the process variables that will move the process toward the optimum. This phase of RSM makes considerable use of the first-order model and an optimization technique called the method of steepest ascent (descent).

Phase 2: Phase 2 begins when the process is near the optimum. At this point the user usually wants a model that will accurately approximate the true response function within a relatively small region around the optimum. Because the true response surface usually exhibits curvature near the optimum, a second-order model (or perhaps some higher-order polynomial) should be used. Once an appropriate approximating model has been obtained, this model may be analyzed to determine the optimum conditions for the process.

This sequential experimental process is usually performed within some region of the independent variable space called the operability region or experimentation region or region of interest.

3. Radial Basis Function Networks

Radial basis function (RBF) networks are feed-forward networks trained using a supervised training algorithm. They are typically configured with a single hidden layer of units whose activation function is selected from a class of functions called basis functions. While similar to back propagation in many respects, radial basis function networks have several advantages. They usually train much faster than back propagation networks. They are less susceptible to problems with non-stationary inputs because of the behavior of the radial basis function hidden units.

Radial basis functions were first introduced by Powell to solve the real multivariate interpolation problem [8]. This problem is currently one of the principal fields of research in numerical analysis. In the field of neural networks, radial basis functions were first used by Broomhead and Lowe [9]. Other major contributions to the theory, design, and applications of RBF networks can be found in papers by Moody and Darken [10], RBF networks have proven to be useful neural network architecture. The design of a RBF network in its most basic form consists of three separate layers. The input layer is the set of source nodes (sensory units). The second layer is a hidden layer of high dimension. The output layer gives the response of the network to the activation patterns applied to the input layer. The transformation from the input space to the hidden-unit space is nonlinear. On the other hand, the transformation from the hidden space to the output space is linear. A mathematical justification of this can be found in the paper by Cover [11]. Cover states that a pattern classification problem cast in a high-dimensional space is more likely to be linearly separable than in a low-dimensional space. This statement is called Cover's theorem on separability of patterns. It is also the reason for making the dimension of the hidden-unit space high in an RBF network.

The major difference between RBF networks and back propagation networks (that is, multilayer perceptron trained by back propagation algorithm) is the behavior of the single hidden layer. Rather than using the sigmoidal or S-shaped activation function as in back propagation, the hidden units in RBF networks use a Gaussian or some other basis kernel function. Each hidden unit acts as a locally tuned processor that computes a score for the match between the input vector and its connection weights or centers. In effect, the basis units are highly specialized pattern detectors. The weights connecting the basis units to the outputs are used to take linear combinations of the hidden units to produce the final classification or output.

3.1 Cover's Theorem on the Separability of Patterns

Before we talk about the radial basis function networks, we need to introduce the term “separability of patterns.” The Cover’s theorem gives a detailed description of the separability of patterns. This theorem explains how a radial basis function network can perform a complex pattern classification task. Considering the Cover’s theorem, Haykin declares that a radial basis function network performs a complex pattern-classification task by transforming the problem into a high-dimensional space in a nonlinear manner [12]. He gives a detailed definition of Cover’s theorem as follows:

Consider a family of surfaces, each of which naturally divides an input space into two regions. Let X denote a set of N patterns (points) $x_1, x_2, \cdots, x_N$, each of which is assigned to one of two classes X^+ and X^-. This dichotomy (binary partition) of the points is said to be separable with respect to the family of surfaces if there exists a surface in the family that separates the points in the class X^+ from those in the class X^-. For each pattern $\boldsymbol{x} \in X$, define a vector made up of a set of real-valued functions $\{\varphi_i(\boldsymbol{x}) | i = 1, \cdots, M\}$, as shown by

$$\varphi(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}), \cdots, \varphi_M(\boldsymbol{x})]^T \tag{5.11}$$

Suppose that the pattern $\boldsymbol{x}$ is a vector in a p-dimensional input space. The vector $\varphi(\boldsymbol{x})$ then maps points in p-dimensional input space into corresponding points in a new space of dimension M. We refer to $\varphi_i(\boldsymbol{x})$ as a hidden function, because it plays a role similar to that of a hidden unit in a feed forward neural network. A dichotomy $\{X^+, X^-\}$ of X is said to be φ-separable if there exists an M-dimensional vector w such that we may write [11]

$$w^T \varphi(\boldsymbol{x}) \geq 0, \qquad \boldsymbol{x} \in X^+$$

and

$$w^T \varphi(\boldsymbol{x}) < 0, \qquad \boldsymbol{x} \in X^- \tag{5.12}$$

The hyperplane defined by the equation

$$w^T \varphi(\boldsymbol{x}) = 0 \tag{5.13}$$

describes the separating surface in the j space. The inverse image of this hyperplane, that is,

$$\left\{ \boldsymbol{x} : w^T \varphi(\boldsymbol{x}) = 0 \right\} \tag{5.14}$$

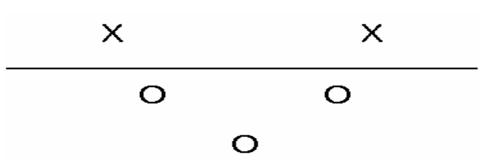
defines the separating surface in the input space.

After giving the definition of the Cover's theorem on separability of patterns, Haykin gives a mathematical explanation for the class of mapping explained above. The separating surfaces corresponding to such mappings are referred to as *r*th-order rational varieties. A rational variety of order r in a space of dimension p is defined by the *r*th-degree homogenous equation in the coordinates of the input vector $\boldsymbol{x}$, as illustrated by

$$\sum_{0 \le i_1 \le i_2 \le \cdots \le i_r \le p} a_{i_1 i_2 \cdots i_r} x_{i_1} x_{i_2} \cdots x_{i_r} = 0 \tag{5.15}$$

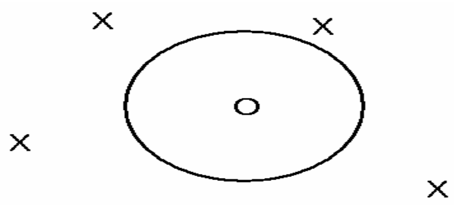
where x_i is the ith component of input vector $\boldsymbol{x}$, and x_0 is set equal to unity in order to express the equation in a homogenous form. Some examples of this type of separating surfaces are hyperplanes (first-order rational varieties), quadrics (second-order rational varieties), and hyperspheres (quadrics with certain linear constraints on the coefficients). Fig. 5.1 illustrates the examples for a configuration of five points in two dimensions

.

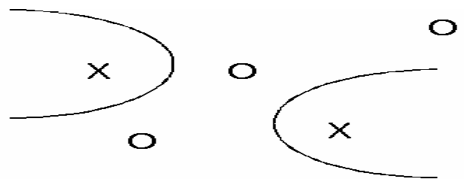


(a) Linearly separable dichotomy

Fig. 5.1 Three examples of φ-separable dichotomies of different sets of five points in two dimensions (*Continued*)



(b) Spherically separable dichotomy



(c) Quadrically separable dichotomy

Fig. 5.1 Three examples of φ-separable dichotomies of different sets of five points in two dimensions

Polynomial separability, as defined here, can be considered as a natural generalization of linear separability. Haykin provides an important point, which states that given a set of patterns x in an input space of random dimension p, a non-linear mapping $\varphi(x)$ of high enough dimension M can be found so that linear separability in the space is obtained. The next section talks about the interpolation problem which has great importance in solving the nonlinearly separable pattern classification problem.

3.2 Interpolation Problem

This section talks about the interpolation problem that allows us to solve the nonlinearly separable pattern classification problem. The interpolation plays the final role in solving the problem since it finds the linear weight vector of the network.

In solving a nonlinearly separable pattern classification problem, there is generally a practical benefit in mapping the input space into a new space of sufficiently high dimension. This is an important point that comes forth from Cover's theorem on the separability of patterns. Let us consider a feed forward network with an input layer, a single hidden layer, and an output layer having a single unit. The network can be designed to perform a nonlinear mapping from the input space to the hidden space, and a linear mapping from the hidden space to the output space.

The network represents a map from p-dimensional input space to the single dimensional output space, expressed as

$$s: R^p \rightarrow R^1 \tag{5.16}$$

The theory of multivariable interpolation in high-dimensional space has a long history starting with Davis [13]. The interpolation problem, in its strict sense can be stated as follows:

Given a set of N different points $\{x_i \in R^p \,|\, i=1, 2, \cdots, N\}$ and a corresponding set of N real numbers $\{d_i \in R^1 \,|\, i=1, 2, \cdots, N\}$, find a function $f: R^N \rightarrow R^1$ that satisfies the interpolation condition [12]:

$$f(x_i) = d_i \qquad i=1, 2, \cdots, N \tag{5.17}$$

The interpolating surface (i.e. function f) has to pass through all the training data points. The radial basis function technique consists of choosing a function that has the following form given by Powell [8].

$$f(x) = \sum_{i=1}^{N} w_i \varphi(\|x - x_i\|) \tag{5.18}$$

where $\{\varphi(\|x - x_i\|) \,|\, i=1, 2, \cdots, N)\}$ is a set of N random (usually nonlinear) functions, known as radial basis functions, and || . || represents a norm that is generally Euclidean.

The known data points $x_i \in R^p$, $i = 1, 2, \cdots, N$ are the centers of radial basis functions [8].

If the interpolation conditions Eq. (5.17) are inserted in Eq. (5.18), the following set of simultaneous linear equations can be obtained for the unknown coefficients (weights) of the expansion w_i :

$$\begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix} \tag{5.19}$$

where $\varphi_{ji} = \varphi(\|x_j - x_i\|) \qquad j, i = 1, 2, \cdots, N$ (5.20)

Let

$$\boldsymbol{d} = [d_1, d_2, \cdots, d_N]^T \tag{5.21}$$

$$\boldsymbol{w} = [w_1, w_2, \cdots, w_N]^T \tag{5.22}$$

The vectors $\boldsymbol{d}$ and $\boldsymbol{w}$ represent the desired response vector and linear weight vector, respectively. Let $\boldsymbol{\varphi}$ denote an N-by-N matrix with elements φ_{ji}:

$$\boldsymbol{\varphi} = \{\varphi_{ji} \mid j, i = 1, 2, \cdots, N\} \tag{5.23}$$

The matrix $\boldsymbol{\varphi}$ is called the interpolation matrix. Eq. (5.19) can be written in the compact form:

$$\boldsymbol{\varphi w} = \boldsymbol{x} \tag{5.24}$$

Assuming that $\boldsymbol{\varphi}$ is nonsingular and therefore that the inverse matrix $\boldsymbol{\varphi}^{-1}$ exists, we may go on to solve Eq.(5.24) for the weight vector w as shown by

$$\boldsymbol{w} = \boldsymbol{\varphi}^{-1}\boldsymbol{x} \tag{5.25}$$

Micchelli [14] gives a remarkable property for a class of radial basis functions which obtains a positive definite interpolation matrix $\boldsymbol{\varphi}$. This remarkable property can be expressed as follows:

Let $x_1, x_2, \cdots, x_N$ be distinct in R^p. Then the N-by-N interpolation matrix $\boldsymbol{\varphi}$ whose jith element is $\varphi_{ji} = \boldsymbol{\varphi}(\|x_j - x_i\|)$, is nonsingular.
The common examples of this specific class of radial basis functions are given as follows:

1) Multiquadrics:

$$\boldsymbol{\varphi}(r) = (r^2 + c^2)^{1/2} \quad \text{for some } c > 0 \text{ and } r \in R \tag{5.26}$$

2) Inverse multiquadrics:

$$\boldsymbol{\varphi}(r) = \frac{1}{(r^2 + c^2)^{1/2}} \quad \text{for some } c > 0 \text{ and } r \in R \tag{5.27}$$

3) Gaussion Functions:

$$\boldsymbol{\varphi}(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad \text{for some } \sigma > 0 \text{ and } r \in R \tag{5.28}$$

The multiquadrics and inverse multiquadrics are both due to Hardy [15]. For the radial-basis function listed in Eq. (5.26) to Eq. (5.27) to be nonsingular, the points $x_1, x_2, \cdots, x_N$ must all be different (i.e., distinct). This is all that is required for nonsingularity of the interpolation matrix $\boldsymbol{\varphi}$, whatever the values of size N of the data points or dimensionality p of the vectors (points) x_i.

The inverse multiquadrics of Eq. (5.27) and the Gaussian function of Eq. (5.28) share a common property: They are both localized functions, in the sense that $\boldsymbol{\varphi}(r) \to 0$ as $r \to \infty$. In both of these cases the interpolation matrix $\boldsymbol{\varphi}$ is positive definite. By contrast, the multiquadrics of Eq. (5.26) are nonlocal in that $\boldsymbol{\varphi}(r)$ becomes unbounded as $r \to \infty$; and the corresponding interpolation matrix $\boldsymbol{\varphi}$ has $N-1$ negative eigenvalues and only one positive eigenvalue, however, is that an interpolation matrix $\boldsymbol{\varphi}$ based on Hardy's multiquadrics is nonsingular, and therefore suitable for use in the design of RBF networks.

In this thesis, Gaussian function which is most commonly used. It can be used to approximate a smooth input-output mapping with greater accuracy than those that yield a positive-definite interpolation matrix

4. Concept of RSM-based Hybrid Evolutionary Algorithm (RHEA)

The main idea is to reduce the evaluation number of the objective function by using RSM which is one among the designed experiments to reduce the repetitive number, since it is one of the demerits of optimum design. The RHEA consists of four main categories: GA for governing the general algorithm; tabu-list for systematic variety of solution; RSM for improving convergent speed for getting a candidate solution; modified simplex method for local search. Fig. 5.2 represents the flowchart of the RHEA. The left side of the flow chart shows global search region that is similar to the flowchart of standard genetic algorithm, excluding the function assurance criterion (FAC), Sh (part A), tabu-list (part B), and RSM (part C). These parts offer candidate solutions, which are considered as initial search points in the local search region. The right side represents the local search region. This part finds out the optimum solution by the modified simplex method, which use the final solution by results of global search as initial search point.

4.1 Sh (a set of History)

Part A in Fig. 5.2 shows the Sh region which provide the well distributed points to make a response surface (refer to Fig. 5.3). The Sh is constructed the following procedures:

Step 1: Read individuals from the current population

Step 2: NSh = NSh + pop_size

where NSh and pop_size mean size of a set of history and size of population, respectively

Step 3: if NSh $\leq$ $Nsh_{\max}$, then go to Step 7

where $Nsh_{\max}$ means maximum size in Sh.

Step 4: Evaluate the dense grade (DG) for each individual

$$DG = \max(d_{ik}) + mean(d_{ik})$$

where d_{ik} is Euclidian distance between i and k;

$$\left\| x^{(i)} - x^{(k)} \right\|, \quad i = 1, \cdots, NSh; \quad i \neq k$$

Step 5: Rank the individuals for DG

Step 6: Select the higher ranked first $Nsh_{\max}$ individuals.

Step 7: Store the solutions in Sh and go out.

4.2 Tabu List

Part B in Fig. 5.2 shows that the tabu list is checked to have a diversity of solution (refer to Fig. 5.4). The one individual which is selected in GA's individuals after crossover is reviewed to secure the diversity of solution. If diversity of solution is secured, we select the individual and if not, we repeat the crossover process. That is, individual is selected when it is located far away from the dense area. So, a dense grade criterion of solution and acceptance criteria of individual are made as follows:

1) Definition

$D \subset R^N$: Normalized domain, where *N* is number of design variables

$V \subset R^N$: A domain having the equally divided by $Nsh_{\max}$ from D.

Let $|V|$ is size of *V*, then

$$|V| = \frac{l^N}{Nsh_{\max}} \tag{5.29}$$

l : One side length of D

$\delta \in R$: One side Euclidian length of hyper-polygon in V as follows:

$$\delta = \sqrt[N]{|V|} = \frac{l}{\sqrt[N]{Nsh_{\max}}} \tag{5.30}$$

2) To decide the acceptance of individual, an aspiration function for a given target design vector is represented as follows:

$$f_t = 1 - \sum_{k=1}^{Nsh} h(r) \quad = \quad 1 - \sum_{k=1}^{Nsh} h(r) = e^{-\gamma \|X_k - X_t\|} \tag{5.31}$$

Let $h(r)$ = $e^{-\gamma r}$, where r = $\|X_k - X_t\|$, X_t is target individual position

3) To set the γ, it is assumed that the following ideal conditions are satisfied.

- Sh is full
- All members of Sh are placed in the center of the $Nsh_{\max}$ sub-domains which are supposed to have same hyper-volume and not to have any cross set of each other and to fit the domain D absolutely. That is,

$$V_i \cap V_j = \Phi \quad i \neq j, \qquad V_1 \cup V_2 \cup \cdots \cup V_{Nsh_{\max}} = D \tag{5.32}$$

Then, set as $f_t = \beta$, where β means acceptance probability criterion.

4) Find γ from the following equation.

$$f_t = \beta = 1 - \sum_{k=1}^{Nsh} e^{-\gamma r} \quad = \quad 1 - \left[2N\, e^{-\gamma\delta} + R \right] \tag{5.33}$$

The first term on the right side corresponds to the closest member of Sh to the target individual. The second term, R, is the residuals. The nature of $h(r)$, which is exponentially decreasing along with distances, makes R be much smaller than the first term, namely R can be neglected.

$$f_t = \beta \approx 1 - 2N\ e^{-\gamma\delta} \tag{5.34}$$

$$\ln\left[\frac{1-\beta}{2N} = e^{-\gamma\delta}\right] \tag{5.35}$$

$$\gamma = -\frac{1}{\delta}\ln(\frac{1-\beta}{2N}) \tag{5.36}$$

In the case of considering first and second terms, we can write as follows:

$$f_t = \beta = 1 - \sum_{k=1}^{Nsh} e^{-\gamma r} = 1 - \left[2N\ e^{-\gamma\delta} + 2N(N-1)\ e^{-\gamma\delta\sqrt{2}} + R_2\right] \tag{5.37}$$

5) Aspiration criteria

- If rand > f_t then accept, where rand = [0 1]
- If trial number > maximum trial number (where, set 50)

If target individual is not satisfied with above aspiration criteria, one crossover is generated again. And the process is repeated.

Example: This example is to get γ

Set $Nsh_{\max}$ = 9, N = 2, l =1 and β = 0.5.

From Eq. (5.30) and Eq. (5.36),

$$\delta = \sqrt[N]{|V|} = \frac{l}{\sqrt[N]{Nsh_{\max}}} = \frac{1}{\sqrt[2]{9}} = 0.3333$$

$$\gamma = -\frac{1}{\delta}\ln(\frac{\beta}{2N}) = -\frac{1}{1/3}\ln(\frac{0.5}{4}) = 6.2383$$

Fig. 5.5 shows the graph of aspiration function using above results.

The procedure is summarized as follows:

Step 1: Read (*N-1)* individuals from selection process.

Step 2: Crossover (*N*-2) individuals according to the crossover probability and go to step 5.

Step 3: One individual selected for tabu-list.

Step 4: If rand > f_i, then go to step 5, otherwise return to step 3

Step 5: Add generated individuals.

4.3 RSM (Response Surface Methodology)

Part C in Fig. 5.2 represents an RSM region. It is largely divided by 3 parts.

Firstly, considering the boundary condition in the response surface for optimization, the upper and lower values of design variables can be considered in this calculation process. However, the merits of this method are diminished when addition constraints like natural frequency are considered, because it has to evaluate the objective function to get the results from external calculations. To overcome this problem, this thesis used Sh as training data and inferred the satisfaction of constraint condition using RBF network [16]. In this way, calculation of actual problems could be avoided.

Secondly, it makes a response surface from Sh by using the least square method (LSM) as shown in Fig. 5.6 C2

Finally, the optimum solution of the response surface is calculated by using tabu search method as shown in Fig. 5.6 C3. To increase optimization speed, gradient based algorithm can be used. However, the solutions satisfying constraint condition cannot be guaranteed since the constraint condition is difficult to define precisely. Also we adopt tabu search method which has an excellent initial convergent speed, because the implementation of the response surface concept is to search for the approximate candidate solution.

The generated optimum solution in Fig. 5.6 C3 is added with other existing GA's individuals according to the sequence of Fig. 5.2 and fitness calculation is performed.

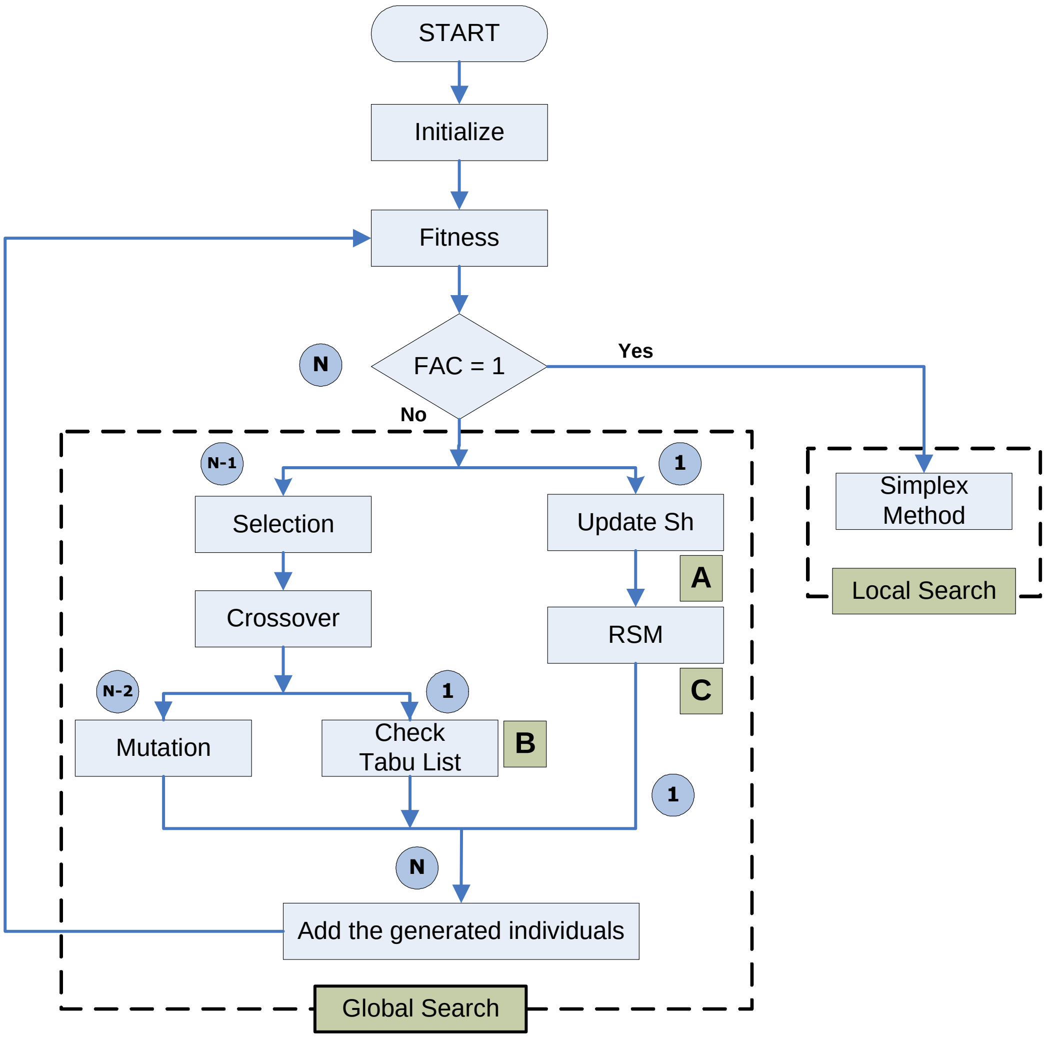


Fig. 5.2 Flowchart of RHEA

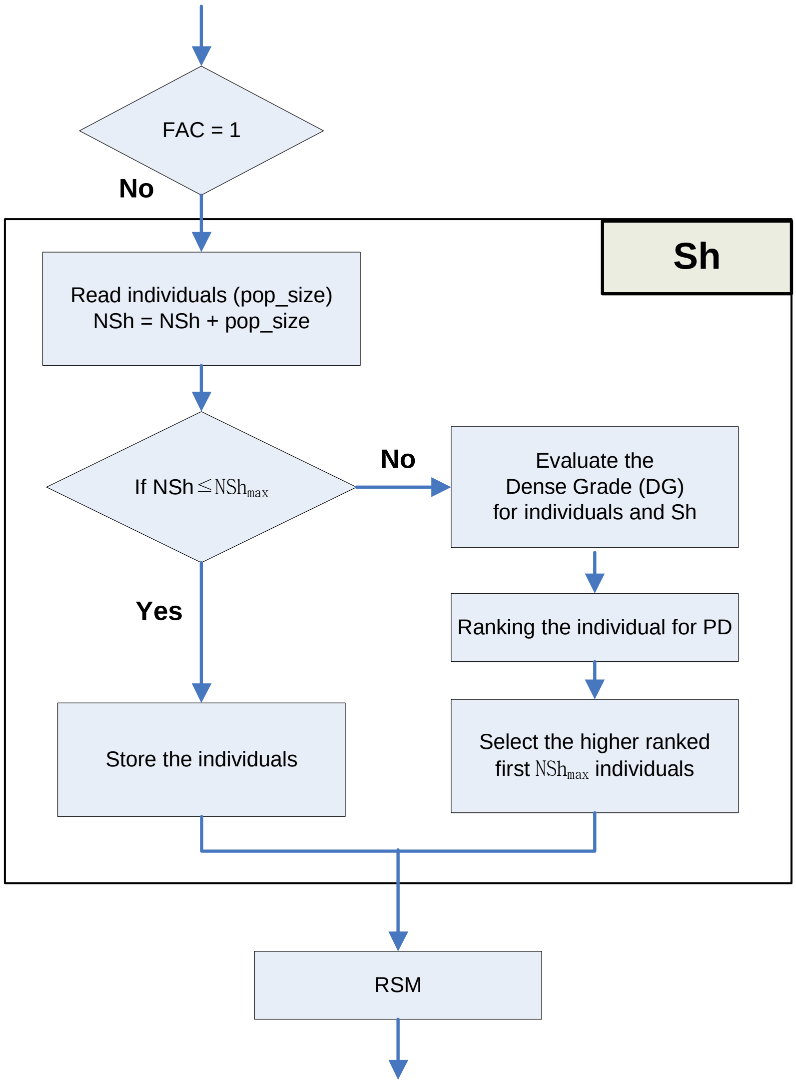


Fig. 5.3 Flowchart of update Sh

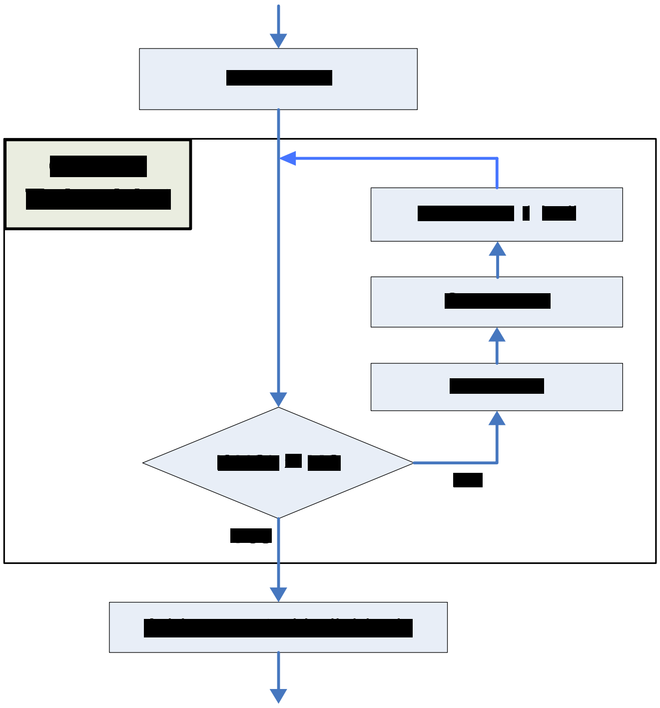


Fig. 5.4 Flowchart of the crossover + tabu list

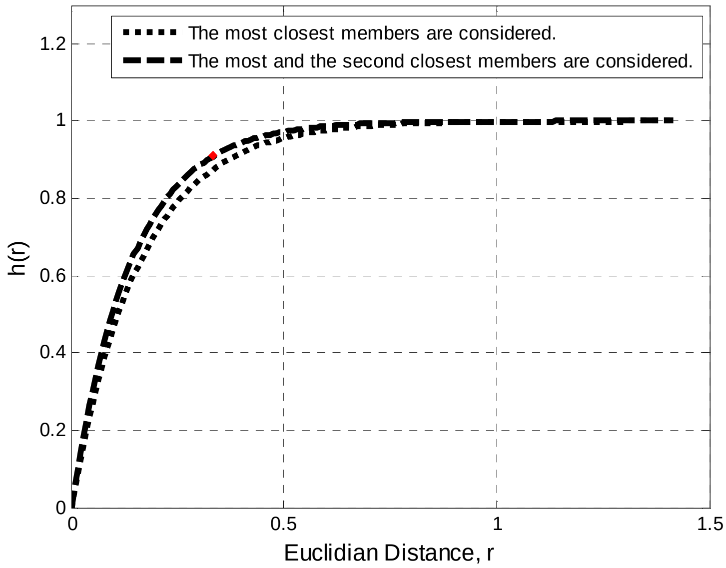


Fig. 5.5 Graph of aspiration function using above results

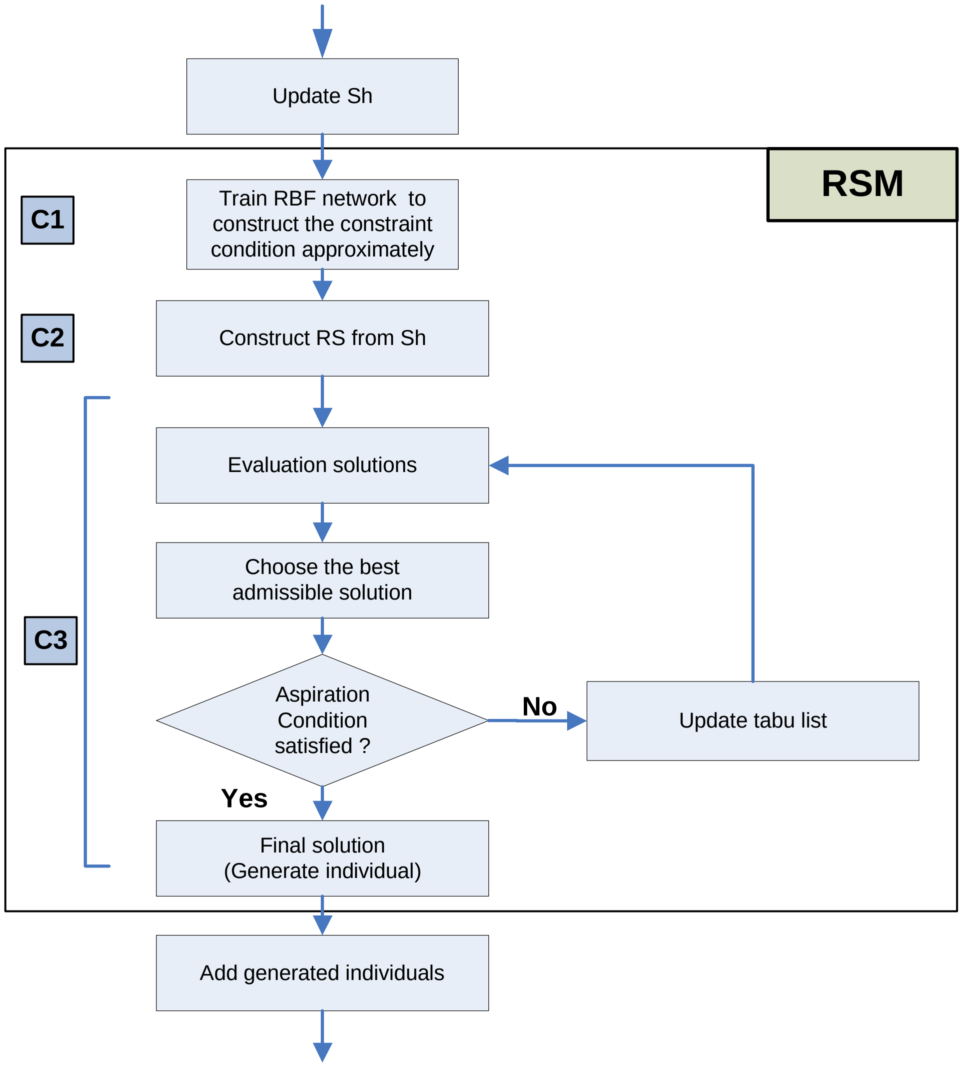


Fig. 5.6 Flowchart of RSM

4.4 Final Global Candidate Solution

Each candidate for optimum solutions is decided by the *FAC* [17]. The *FAC* defined by Eq. (5.38) is a standard value to estimate the convergence of the initial candidate.

$$FAC = \frac{\left| f_{i-1}^{T} \; f_i \right|^2}{(f_{i-1}^{T} \; f_{i-1})(f_i^{T} \; f_i)} \tag{5.38}$$

where f_i is the row vector, formed by the fitness values of the individuals at the i th generation and f^T is the transpose of f .

The row size depends on the number of optimum solutions according to a designer's requirement. Theoretically the range of *FAC* is from 0 to 1.0. When the value is equal to 1, the convergence of optimization is completed. The value, however, is difficult to converge to 1.0 considering the many candidate solution to evaluate the *FAC*. Therefore, in this thesis, the *FAC* is set to 0.9999.

5. Procedures of RSM-based Hybrid Evolutionary Algorithm (RHEA)

RSM-based hybrid evolutionary algorithm (RHEA) is introduced as follows:

Step 1: The parameters are set up as follows:

Pop_size: population size

P_c : cross probability

P_m : mutation probability

M_s : selection method

M_c : crossover method

Step 2: Generate the initial chromosome $v_k \; (k = 1,\, 2,\, \cdots,\; pop\_size)$

randomly with n elements.

$$v_k = [x_{k1}\ x_{k2}\ \cdots\ x_{kn}]$$

When generating the chromosomes, following conditions should be satisfied:
The element value range of each chromosome is satisfied as below,

$$x_{kj}{}^{L} \le x_j \le x_{kj}{}^{U}$$

Each chromosome satisfies all constraints as follows:

$$g_i(v_k) \ge 0,\ \forall i$$

When a chromosome does not satisfy the conditions, then the chromosome has the lowest fitness. So it has a low possibility of selection to the next generation after all.

Step 3: Generate the initial solutions, and estimate constraint and set up a parameter range

Step 4: Evaluate the fitness of individuals

Step 5: Evaluate the *FAC*, if it is satisfied, go to step 13 otherwise go to step 6

Step 6: Update Sh : $\text{Sh} = \left\{ (X_{sh},\ F) \middle| X_{sh} \in R^N,\ F \in R \right\}$

where $X_{sh} = [x_1,\ x_2,\ \cdots,\ x_i\ ,\ \cdots,\ x_N]$

Step 7: Selection

Step 8: Crossover and check tabu list

Step 9: Construct RS (Response Surface) from Sh:

$$f_{rs} = \alpha_0 + \sum_{i=1}^{N} \alpha_{ii} x_i + \sum_{i=1}^{N} \alpha_{ii} x_i{}^2 + \sum_{i=2}^{N} \sum_{j=1}^{i-1} \alpha_{ij} x_i x_j \qquad (5.39)$$

where $\alpha_0,\ \alpha_{ii},\ \alpha_{ij}$ are coefficients calculated by LSM

Step 10: Train RBF network by Sh to construct the constraint conditions approximately.

Step 11: Calculate the optimum design on the response surface by tabu search method and generate one individual based on *X**.

Step 12: Mutate and go to step 4

Step 13: Search the optimum solutions by the local concentration search (modified simplex method) for best candidate.

6. Numerical Examples of Several Function Optimizations

Three test functions are used to verify the efficiency of the proposed hybrid algorithm: the first one is the four-peak function, which has one global optimum with three local optima; and the second one is Rosenberk's function which is known as banana function and has just one global optimum; and the last one is the Rastrigin function which has one global minimum with 220 local minima.

6.1 Four-Peak Function [18]

$$f(x_1,\ x_2) = e^{\log_{10}(0.25)\times\left(\frac{x_1-0.2}{0.8}\right)^2} \times\cos^6(1.5\pi x_1) \ + \ e^{\log_{10}(0.25)\times\left(\frac{x_2-0.1}{0.8}\right)^2} \times\cos^6(1.5\pi x_2) \qquad (5.40)$$
$$\left(-0.4 \ \le \ x_1,\ x_2 \ \le \ 1\right)$$

This test function has a global optimum solution $f(\boldsymbol{x})$ = 1.954342 at x_1 = 0, x_2 = 0, and three local optima solutions $f(\boldsymbol{x})$ = 1.807849, 1.705973, 1.559480 as shown in Fig. 5.7. Conventional gradient based hill-climbing algorithms can be easily stuck to local optimum because of their dependency on start point, while global search algorithm finds global optimum in general.

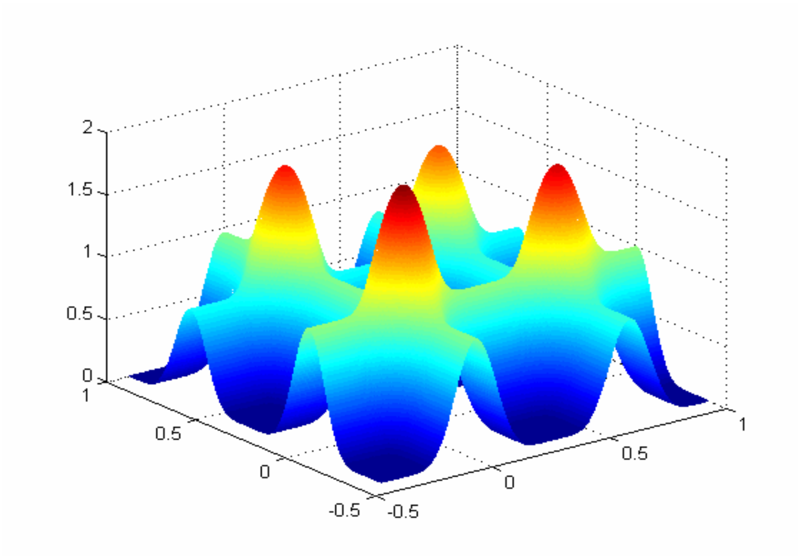


Fig. 5.7 Four-peak function

6.2 Rosenbrock Function [19]

$$f(x_1,\ x_2) = 100(x_1 - x_2)^2 + (1 - x_1)^2, \quad (-2.0 \ \leq \ x_1,\ x_2 \ \leq \ 2.0) \qquad (5.41)$$

This function is called banana function whose shape is the one like Fig. 5.8. The objective of this function is to find the variable x, which minimizes the objective function. This function has only one optimum solution $f(\boldsymbol{x}) = 0$ at $x_1 = 1.0$ and $x_2 = 1.0$. It is difficult to find an optimum solution because of a valley phenomenon [20].

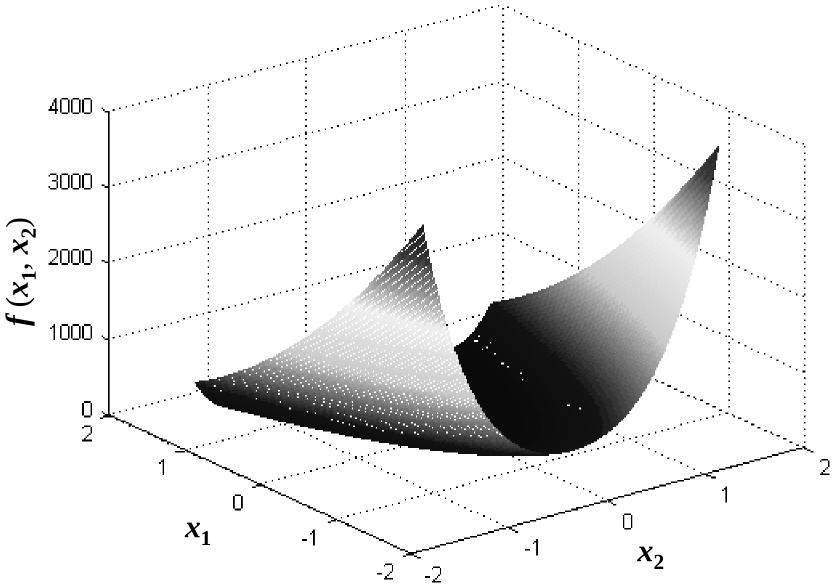


Fig. 5.8 Rosenbrock function

6.3 Rastrigin Function [21]

This function is often used to evaluate the global search ability because there are many local minima around the global minimum as shown in Fig. 5.9. It is not easy to find a global minimum within a limited function call. The objective of this example is to minimize a function defined by Eq. (5.42). This function has 220 local minima and one global minimum $f(\boldsymbol{x}) = 0$ at (0, 0).

$$f(\boldsymbol{x}) = 2\times10 + \sum_{i=1}^{2}\{x_i^2 - 10\cos(2\pi i)\} \quad (-5.0 \le x_1, x_2 \le 5.0) \tag{5.42}$$

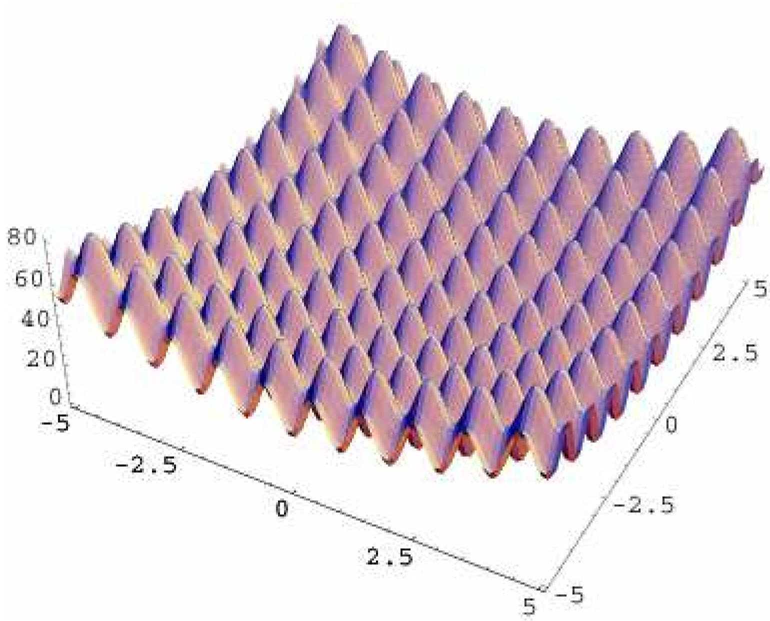


Fig. 5.9 Rastrigin function

Figs. 5.10-5.12 represent the convergent trend of objective function for each test function. According to the results, GRSM (GA+RSM) and GRSMT (GA+RSM+Tabu list) algorithms which are based on RSM have faster convergent speed and more accurate solutions than GA, which validated the efficiency of RSM on the calculation. Also tabu list enables convergence to solutions quickly on the multi-peak function due to the systematic diversity of solution. The setting parameters for each algorithm are listed in Table 5.1. Table 5.2 shows the comparison of optimization results for the above stated three test functions. The evaluation number means total evaluation number of the objective function used in optimization procedure, and it is directly proportion to the total calculation time. Fig. 5.13 shows the contour for optimum solutions obtained by RHEA. According to the results, for all test functions, RHEA can give better solutions than GA on accuracy and convergent speed. For the Rastrigin function, which is very useful to evaluate the global search ability because there are many local minima around the global minimum, RHEA found global minimum with higher accuracy and less elapsed time compared to GA. According to these results, the proposed new hybrid algorithm is a powerful global optimization algorithm from the view of convergent speed and global search ability.

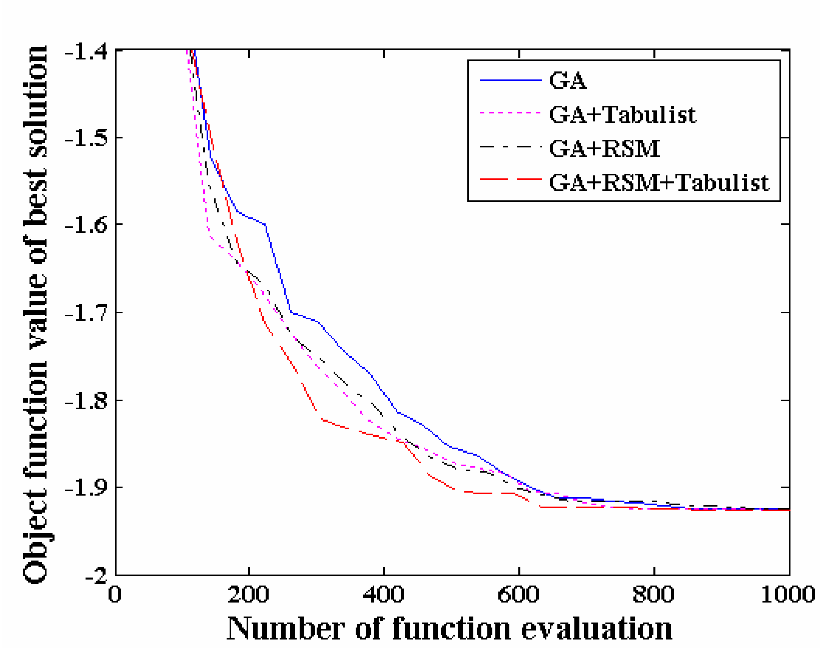


Fig. 5.10 Convergent trend of objective function (Four-peak function)

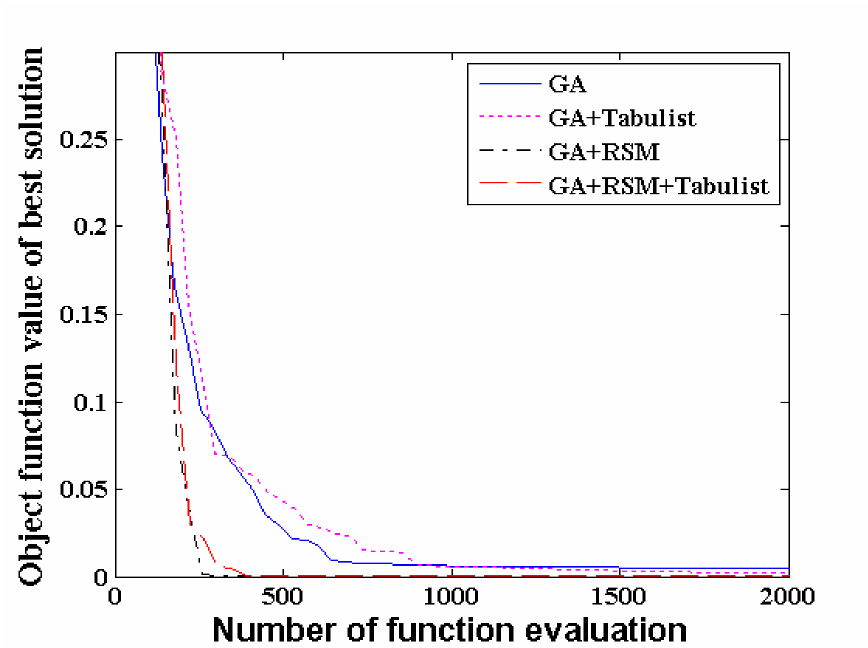


Fig. 5.11 Convergent trend of objective function (Rosenbrock function)

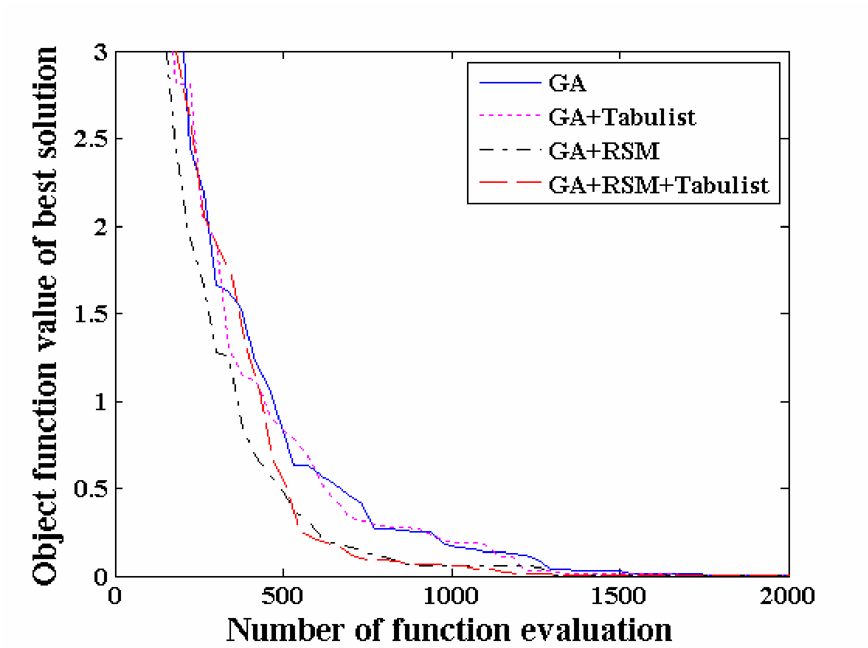


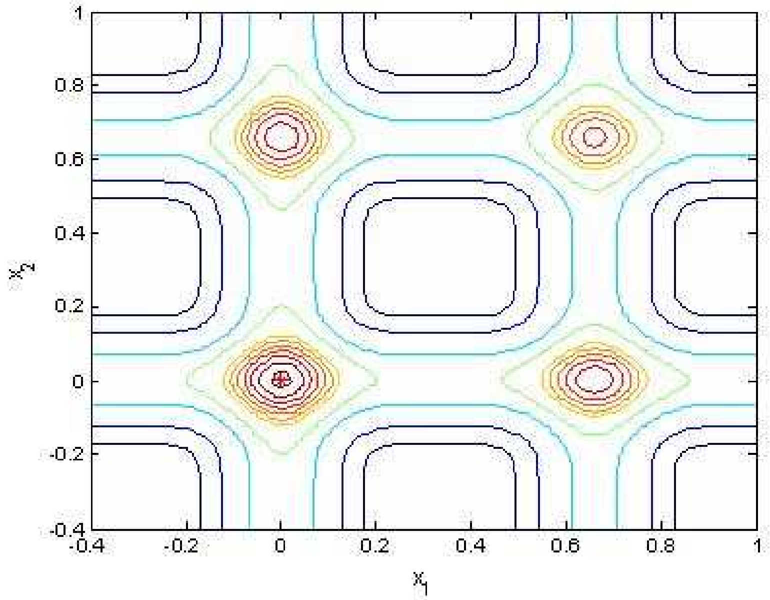
Fig. 5.12 Convergent trend of objective function (Rastrigin function)

Table 5.1 Set parameters for GA and RHEA

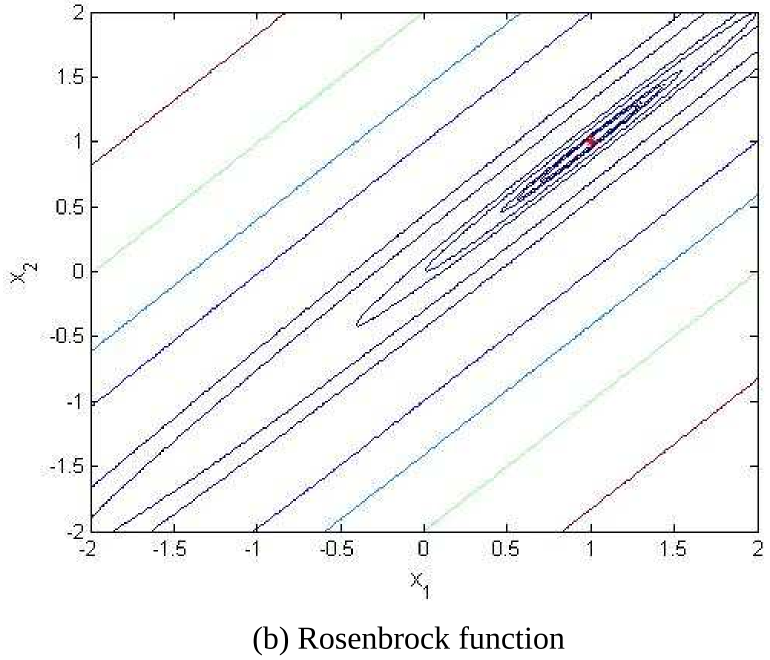
Parameters	Value	Remarks
No. of generation	100	GA & RHEA
Population size	100	
Crossover probability	0.5	
Mutation probability	0.1	
Size of Sh for RSM (Nsh_{max})	1000	RHEA only
Step size for R-tabu	10	
Count number for R-tabu	3	

Table 5.2 Comparison of optimization results

Test function	Exact solutions	Methods	Results		No. of evaluation
			$f(x)$	(x_1, x_2)	
Four-peak function	$f(x) = 1.9543$, $x_1 = x_2 = 0$	GA	1.927	2.403 e-3 2.787 e-3	2353
		RHEA	1.927	2.736 e-3 2.736 e-3	459
Banana function	$f(x) = 0$, $x_1 = x_2 = 1$	GA	1.640 e-5	9.960 e-1 9.960 e-1	1046
		RHEA	0.0	1.0 1.0	419
Rastrigin function	$f(x) = 0$, $x_1 = x_2 = 0$	GA	1.586 e-4	1.408 e-4 8.15 e-4	2109
		RHEA	0.0	-3.076 e-9 -7.747 e-10	514

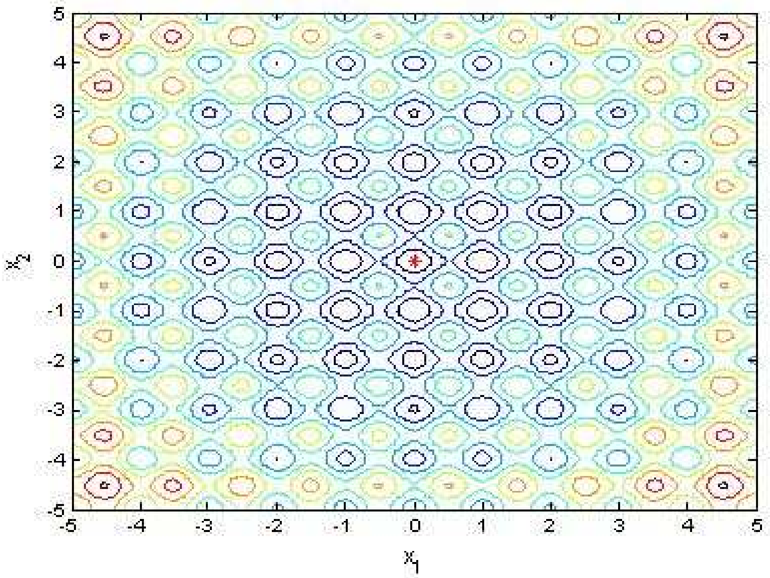


(a) Four-peak function



(b) Rosenbrock function

Fig. 5.13 Optimum solution obtained by RHEA (*Continued*)



(c) Rastrign function

Fig. 5.13 Optimum solution obtained by RHEA

7. Application for Optimum Design of Fresh Water Tank of Ship

In the engine room and the rear of the ship, there are so many tank structures contacting with fresh water, sea water or oil. Also these possibly subject to the excessive vibration during voyage because they are arranged around the main excitation sources of ship such as the main engine and propeller. If problems occur, it takes a lot of cost, time and effort to improve the situation because the reinforcement work for empting the fluid out of the tanks, additional welding and special painting and so on is required. It is therefore very important to predict the precise vibration characteristics of the tank structures at the design stage. Optimum design needs to be applied. Especially when the structure is in contact with fluid much analysis time is taken. So, a new hybrid optimization algorithm is required for getting a short analysis time and accurate solution. In this thesis, optimum design of a fresh water tank in an actual ship is carried out to verify the validity of the proposed optimization algorithm (RHEA) and the results are compared to that of GA.

7.1 Vibration Analysis of Fresh Water Tank

It is difficult to predict the vibration response of a local structure due to the complicated transfer mechanism of excitation force and the difficulty of assuming the damping ratio. Traditionally, therefore, the vibration analysis considering the design of avoiding resonance is conducted to prevent the local vibration.

In this thesis, the vibration analysis of the fresh water tank is carried out using NASTRAN which is a commercial finite element program and widely used for big structures like ships. Fig. 5.14 shows the model and arrangement of the fresh water tank. Fig. 5.15 shows the design variables and boundary condition of the fresh water tank. Considering the precision of analysis and time consuming modeling process, the range of modeling of fresh water tank is constrained to one side of the tank. The boundary conditions for the model are specified: the simple supports are used to the tank boundary area which is connected to the other bulkhead and deck. Table 5.3 shows the specification of main excitation sources.

In general, the design for avoiding local structure resonance in ships requires that the natural frequency of the structure must be two times higher than the blade passing frequency of the propeller under the maximum rpm of the main engine. In this thesis, design target frequency is set as above 14.02 Hz which considers safety margins and twice blade passing frequency of the propeller (12.13Hz).

Fig. 5.16 shows the first three modes and natural frequencies of the fresh water tank by NASTRAN. These three modes frequently occurred on the fresh water tank during voyage. Especially, the 1st mode (8.60 Hz) is a stiffener (stringer) mode which generates a strong vibration and much effect on the structure. In this model, the 1st natural frequency of the structure is also within the resonance region where twice blade passing frequency of propeller is 12.13 Hz. Therefore, the natural frequency of structure is needed to be increased up to the target frequency under the condition that the tank is fully filled. The natural frequency of structure which is contacting with fluid can be changed according to the water line of the tank. So, in order to design a safe structure, the three modes of the fresh water tank are concerned in this study.

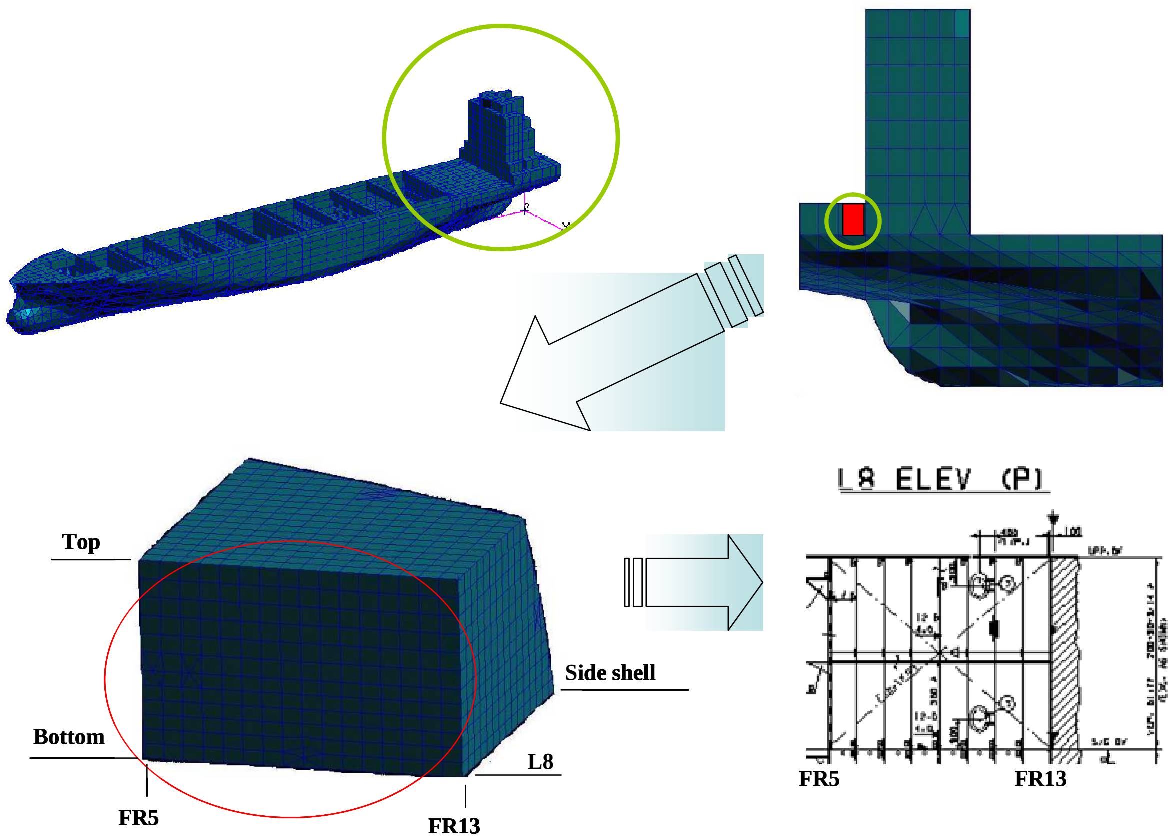
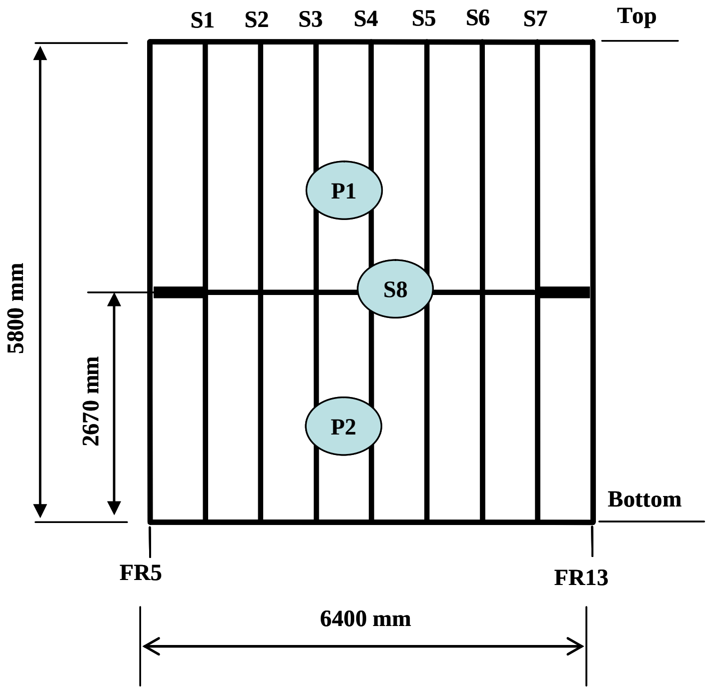


Fig. 5.14 Model and arrangement of fresh water tank

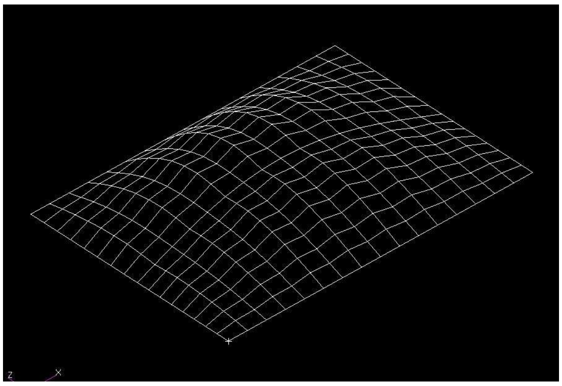


S1	Stiffener 1
S2	Stiffener 2
S3	Stiffener 3
S4	Stiffener 4
S5	Stiffener 5
S6	Stiffener 6
S7	Stiffener 7
S8	Stringer 1
P1	Plate 1
P2	Plate 2

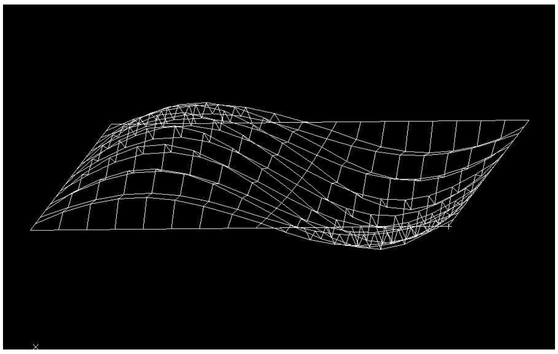
Fig. 5.15 Design variables and boundary conditions of fresh water tank

Table 5.3 Specification of main excitation sources

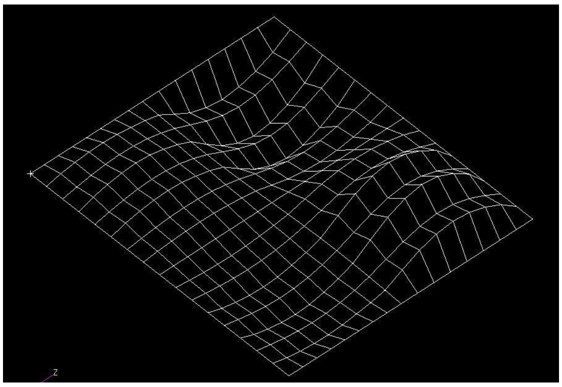
Excitation source	MCR	Excitation	
		Order	Frequency
Main engine (6S 70MC-C)	91 rpm	3rd	4.55 Hz
		4th	6.07 Hz
		6th	9.10 Hz
Propeller (Blade: 4EA)		1st	6.07 Hz
		2nd	12.13 Hz



(a) 1st mode (8.60 Hz)



(b) 2nd mode (18.82 Hz)



(c) 3rd mode (19.17 Hz)

Fig. 5.16 Mode shapes of fresh water tank

7.2 Optimum Design of Fresh Water Tank

7.2.1 Formulation for Optimum Design

a) Design variables

The main vibration modes on the fresh water tank are stiffener modes in transverse direction. One of the most important factors is the stiffness of stiffeners. In this study, the stiffener size and plate thickness of fresh water tank in Fig. 5.15 are defined as design variables in Eq. (5.43).

$$\boldsymbol{x} = \{S_1\ S_2\ S_3\ S_4\ S_5\ S_6\ S_7\ S_8\ P_1\ P_2\}^T \qquad (5.43)$$

where *S*and *P* mean stiffener size and plate thickness, respectively.

b) Constraints

The web length of stiffener L_w is restricted as two categories such as Eq. (5.44) and (5.45) according to the shipyard's practice.

$$150 \le L_w \le 450 \text{ mm for stiffeners } (S_1\text{-}S_7) \qquad (5.44)$$

$$500 \le L_w \le 1000 \text{ mm for stringer } (S_8) \qquad (5.45)$$

Also, the basic concept of local vibration design is the minimization of the response at each point. However, it is difficult to evaluate how much the excitation force influences on local structure. So, in this study, natural frequency of the structure is restricted as Eq. (5.46) which considers a safety margin of twice blade passing frequency of the propeller.

$$\omega_n \ge 14.02 \text{ Hz} \qquad (5.46)$$

c) Objective function

The objective function combines linearly the weight of fresh water tank with natural frequency of structure like Eq. (5.47). The objective is to get an economic and sound structure to reduce the weight of stiffener and to increase the natural frequency.

Minimize $f(x) = \alpha\left(\frac{W_1}{W_0}\right) + \beta\left(\frac{\omega_t}{\omega_0}\right)$ (5.47)

where, ω_t and ω_0 mean target and current natural frequency, respectively. α and β are weighting factors (α = 0.5, β = 0.5).

7.3 Optimization Results and Discussion

The optimum design was carried out to get an optimal size of stiffener and plate thickness on the fresh water tank to maintain the anti-vibration design of it. Table 5.4 shows the results of the design variables before and after optimization. It shows that the stringer S8 is increased by 72% and the others by 4.0-52%. This result indicates that the most reasonable modification method is to increase the stringer which has an effect on the decreasing the span of the vertical stiffeners. In this case, however, the plate thickness does not have any effect on the natural frequency of the structure. Table 5.5 shows the variation of natural frequency and weight of structure before and after optimization. According to the results, the 1st natural frequency increased by 163 % from 8.6Hz to 14.02Hz, and the safety margin with twice passing frequency of the propeller correspondingly changed from -29.1% to 11.56%. Therefore, the structure is free from resonance. Moreover, the weights of stiffeners which are applied to the design variables also decreased in spite of higher natural frequency. In summary, the local vibration problems which require avoidance of structure resonance through the movement of natural frequency without additional weight has been successfully solved by the proposed optimization method. Table 5.6 and Fig. 5.17 show the comparison of optimization results between GA and RHEA. The evaluation number means a total evaluation number of the objective function used in the optimization procedure, and is directly proportional to the total calculation time. According to the results, RHEA can give better solutions than GA on accuracy and convergent speed. These results lead us to draw the conclusion that the proposed new hybrid algorithm is a more powerful global optimization algorithm from the view of convergent speed and global search ability.

Table 5.4 Comparison of original and optimal design variables

Design variable	Original	Optimum		Remarks (RHEA)
		GA	RHEA	
S_1	200	214	207	4.0 %
S_2	200	320	223	12.0 %
S_3	200	253	285	43.0 %
S_4	200	325	283	42.0 %
S_5	200	328	303	52.0 %
S_6	200	277	251	26.0 %
S_7	200	281	230	15.0 %
S_8	550	893	947	72.0 %
P_1	11.0	10.7	10.3	-6.36 %
P_2	11.0	10.6	10.0	-9.09 %

Table 5.5 Comparison of results

Item	Original	Optimum	Remarks
Natural frequency	8.60 Hz	14.02 Hz	163 %
Weight	4883kg	4652 kg	-4.73 %

Table 5.6 Comparison of optimization results

Item	Natural frequency	Weight	Objective function	No. of evaluation
GA	14.04 Hz	5001 kg	0.5547	1846
RHEA	14.02 Hz	4652 kg	0.5167	1638

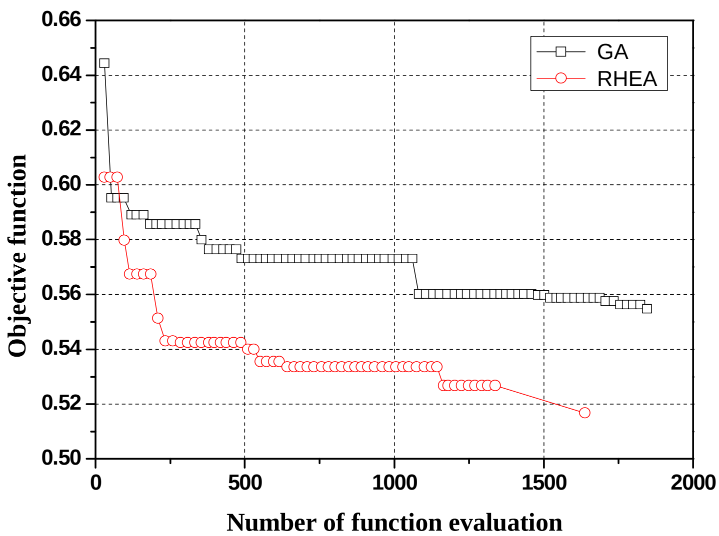


Fig. 5.17 Convergent trend of objective function

8. Conclusions

This chapter introduces an RSM-based hybrid evolutionary algorithm, as a new kind of a hybrid optimization algorithm that combined the merits of the popular programs such as genetic algorithm, tabu search method, response surface methodology. This algorithm, in order to improve the convergent speed that is thought to be the demerit of genetic algorithm, uses response surface methodology and simplex method. Though mutation of GA offers random variety, systematic variety can be secured through the use of tabu list. Especially, in initial stages, GA's convergent speed can be improved by using RSM method which use the information on objective function acquired through GA process and then making response surface (approximate function) and optimizing this. An optimized solution was calculated without the evaluation of additional actual objective function, and the GA’s convergent speed could be improved. Efficiency of this method has been proven by applying traditional test functions and comparing the results to GA. It also proved that the newly suggested algorithm can effectively find the global optimum solution by applying it to weight minimization of fresh water tank that is placed in the rear of ship designed to avoid resonance. Finally it is concluded that the proposed new hybrid algorithm (RHEA) is a very powerful global optimization algorithm from the view point of convergent speed and global search ability.

References

[1] D. E. Goldberg, 1989, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley Publishing Company, pp. 1-146.

[2] L. Davis, 1991, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, pp. 3-23.

[3] Y. C. Kim and B. S. Yang, 2002, "An enhanced genetic algorithm for global and local optimization search," Trans. KSME (A) Vol. 26, No. 6, pp. 1008-1015. (in Korean)

[4] T. Sato and M. Hagiwara, 1998, "Bee system: finding solution by a concentrated search," T. IEE Japan, Vol. 118-C No. 5, pp. 721-726.

[5] H. M. Raymond and D. C. Montgomery, 1995, *Response Surface Methodology: Process and Product Optimization Using Designed Experiment*, A Wiley-Interscience Publication, New York, pp. 1-15.

[6] G. E. P. Box and K. B. Wilson, 1951, "On the experimental attainment of optimum conditions," Journal of the Royal Statistical Society, Series B, Vol. 13, pp. 1-38, discussion: 38-45.

[7] G. V. Reklaintis, A. Ravindran and K. M. Ragsdell, 1983, *Engineering Optimization: Methods and Applications*, John Wiley and Sons, New York, USA.

[8] M. J. D. Powel, 1985, "Radial basis functions for multivariable interpolation: A review," IMA Coference on Algorithms for the approximation of Functions and Data, RMCS, pp. 143-167.

[9] D. S. Broomhead and D. Lowe, 1988, "Multivariable functional interpolation and adaptive networks," Complex Systems, Vol. 2, pp. 321-355.

[10] J. Moody and C. J. Darken, 1989, "Fast learning in networks of locally-turned processing units," Neural Computation, Vol. 1, pp. 281-294.

[11] T. M. Cover, 1965, "Geometrical and statistical properties of systems of

linear inequalities with application in pattern recognition," IEEE Trans. on Electronic Computers, Vol. EC-14, pp. 326-334.

[12] S. Haykin, 1999, *Neural Networks (A Comprehensive Foundation)*, Prentice-Hall, Inc., New Jersey, USA.

[13] P. J. Davis, 1963, *Interpolation and Approximation*, Blaisdell.

[14] C. A. Micchelli, 1986, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," Constructive Approximation, Vol. 2, pp. 350-373.

[15] R. L. Hardy, 1971, "Multiquadric equation of topography and other irregular surface," Journal of Geophysics Research, Vol. 76, pp. 1905-1915.

[16] D. Howard, B. Mark and H. Martin, 2005, *Neural Network Toolbox for Use with MATLAB*, The MathWorks.

[17] M. I. Friswell and J. E. Mottershead, 1996, *Finite Element Model Updating in Structural Dynamics*, Kluwer Academic Publishers, Vol. 38, pp. 56-59.

[18] Y. C. Kim, 2003, *Development of Enhanced Genetic Algorithm and Its Applications to Optimum Design of Rotating Machinery*, Ph.D. Dissertation, Pukyong National University, South Korea

[19] A. Homaifar, C. Qi and S. Lai, 1994, "Constrained optimization via genetic algorithm simulation," Electronics Letter, Vol. 62, No. 4, pp. 242-254.

[20] T. Kohonen, 1995, *Self-Organizing Maps*, New York: Springer-Verlag.

[21] Y. H. Kim, B. S. Yang, Y. C. Kim and S. J. Lee, 2003, "Bearing parameters identification using hybrid optimization algorithm," in Proceedings of 32 nd International Congress and Exposition on Noise Control Engineering, Jeju, Korea, pp. 4212-4219.

VI. Conclusions

The objective of this paper is to propose algorithms and a framework for optimum design in order to reduce the vibration on large structures like ships. For this purpose, two algorithms and a framework related to optimization have been developed in this study. They have been applied to known test functions to ascertain their usefulness, and their excellence was proven. To prove the practicality of the algorithms and a framework, an optimization work for global and local vibration of ships constructed in shipbuilding was executed.

Firstly, in order to optimize large structures, an optimization framework was developed. This framework is able to choose from many global optimization methods and use NASTRAN as a solver. This framework was called OPTSHIP and the merits of OPTSHIP are as follows.

- Large structures like ships can be easily optimized.
- General-purpose analysis program, NASTRAN is used as a solver.
- Various optimization algorithms can be diversely utilized as an optimizer.
- Implementation of new optimization method created by the user is easy.
- Objective functions can be diversely selected.
- Various design variables can be selected.
- Global optimum solution can be obtained.

To verify the reliability and performance of OPTSHIP, this algorithm is applied to minimize the vibration level of the deckhouse in the 2400 TEU containership. The excellence of the result is proven by comparing it with the optimization result of the existing NASTRAN optimization module which is widely used for general-purpose program.

Secondly, a non-linear integer programming (NIP) algorithm based on GA was developed. This method enables the optimized result directly applicable in the design of the stiffeners and steel plates which used in the shipbuilding. Especially, taking the difference of the accuracy of optimum solution and convergent speed according to the initial parameters due to the characteristics of GA into account, this thesis executed optimization of GA parameters simultaneously. Then, the optimized GA parameters were applied to the object structure and it is proven that parameters were an optimized value. The NIP algorithm was used to perform optimum design of the compass deck structure of a ship with potential vibrations, thus solving the vibration problems proved the efficiency of the proposed method.

Thirdly, a new hybrid evolution algorithm (RHEA: RSM-based hybrid evolutionary algorithm) was developed. This method employs the GA as its base in order to ensure tabu list of tabu search method that provides a systematic variety of solutions and to secure response surface methodology (RSM) which provides a quick convergent speed. Mutation of GA provides random diversity, but by implementing tabu list of tabu search method, a systematic variety could be obtained. By using the information of objective function obtained in the process of GA while implementing RSM, response surface (approximation function) was created. By optimizing this, optimized solution was calculated without the evaluation of additional actual objective function, and the GA's convergent speed could be improved. The efficiency of this method has been proven by applying traditional test functions and comparing the results to GA. Finally, we can conclude that the proposed new hybrid algorithm (RHEA) is a very powerful global optimization algorithm from the view point of convergent speed and global search ability.

Additionally, the outcome of this thesis is open to applications on ships as well as other complex structures. The optimization process of a ship can be applied not only to structures but also other various fields and it is useful to attain a dominant position in the competition of future shipbuilding. Furthermore, an expanded application as well as a perennial development on a more efficient optimization algorithm is necessary. Finally, by undergoing numerous experiences, I hope that the enhanced version of the algorithm mentioned in this thesis being developed.

최적화 알고리듬과 프레임웍 개발 및 선박 구조물의 최적설계에의 적용

공 영 모

부경대학교 대학원 기계공학부

국문 요약

최근 선박 진동문제에 있어서 대두되고 있는 과제는 과대 진동에 따른 구조물의 피로파괴 회피와 선원의 근무환경 여건의 고급화에 따른 선실의 쾌적함 추구이다. 이에 따라 선주가 요구하는 진동 허용치가 지속적으로 엄격해지고 있다.

그러나 본질적으로 선박 진동의 기진원은 주기관, 보조기관, 프로펠러, 파도, 유체운동 등으로 다양하며, 구조계 및 선내 기계장치가 복잡하게 설치되어 있어 진동이 없는 선박을 건조한다는 것은 사실상 불가능하다. 더구나 선박의 고출력 엔진 채용과 경량화를 지향하고 있는 최근의 동향과 맞물려 심각한 진동문제가 야기되고 있는 실정이다. 따라서 전술했던 제반 문제의 해결방안은 빈번하게 진동문제를 일으키는 부분을 경험적으로 체계화하여 선박 최적설계를 도모함으로써 방진 표준화 작업을 수행하여 진동을 최소화하는 것이다.

여기에서 말하는 ‘최적설계’란 대상구조물의 안정성과 유용성을 가지고 최소의 재료를 사용하여 최대의 효과를 얻을 수 있도록 구조물의 부재 치수 혹은 기하학적인 형상을 결정하는 것을 의미한다.

이와 같은 선박 최적설계는 근간 산업현장의 임금 인상과 선박 제조에 필요한 재료비 인상에 따라 보다 절박하게 요구되고 있는 선박 건조 비용 절감 측면에서 보건대 진동문제 해결을 위한 경비를 경감할 수 있으므로 경제적 효용가치 또한 높다.

그러나 현실적으로 선박 최적설계는 손쉽게 해결할 수 있는 과제가 아니다. 특히 상선과 같은 큰 배의 경우에는 구조물의 자유도수가 많기 때문에 더욱 복잡하고, 어렵다. 그에 따라 이 분야의 선행 연구자들은 실행 시간 단축을 위하여 모델 축소화를 시도하고, 최적화 도구의 유용성을 높이기 위하여 다양한 최적화 기술을 도입·적용함으로써 복잡한 최적설계 상의 문제를 해결해왔다.

본 연구에서는 두 가지의 최적화 알고리듬과 한 가지의 최적화 프레임웍을 제안하고, 그에 따른 관련 프로그램을 개발하였다. 그 세부 내용을 요약해 보면 다음과 같다.

첫째, 대형 구조물의 최적화에 용이하고, 다양한 목적함수를 선정할 수 있으며, 설계변수를 다양하게 선택할 수 있는 NASTRAN 외부 호출형 독립 최적화 프레임웍을 개발하고, 이를 OPTSHIP이라 명명하였다.

OPTSHIP 은 구조물의 고유진동수, 강제응답 및 모드벡터 등을 구하기 위한 Solver 로써 NASTRAN 을 사용하였고, 전역 최적해를 구하기 위한 최적화 알고리듬은 외부에서 이용자가 모듈을 만들어 사용할 수 있도록 하였다. 현재 사용 가능한 최적화 모듈은 유전알고리듬(GA), 랜덤터부탐색(R-tabu)법, 시뮬레이티드어닐링(SA)법, 인공생명(AI) 최적화 알고리듬 등이다.

제안된 프레임웍의 유용성을 검증하기 위하여 2400TEU 컨테이너선의 거주구 진동 최소화에 적용하였다. 왜냐하면 위 컨테이너선의 거주구가 선미부에

위치하여 진동 측면에서 매우 불리한 특성을 지니고 있어 본 프로그램의 효용가치를 확인하기 위한 최적의 구조물로 판단되었기 때문이다.

OPTSHIP 을 활용한 해석의 효율성을 높이기 위하여 목적함수에 대한 후보설계변수의 감도해석을 NASTRAN 을 이용하여 수행하였다. 그에 따라 감도값이 1.5 이상인 설계변수를 최종설계변수로 채택하여 계산시간을 단축할 수 있었다.

본 예제에서는 OPTSHIP 최적화 외부 모듈로써 R-Tabu 법을 사용하여 그 계산결과와 상용 프로그램인 NASTRAN 최적화 결과를 비교해 봄으로써 그 우수성을 입증하였다.

둘째, 본 논문에서는 최적화된 설계변수 값을 초기 설계 단계에서 직접적으로 사용할 수 있도록 ‘비선형 정수형 최적화 기법’을 개발하였다.

비선형 정수형 최적화 기법은 현재 조선소에서 널리 사용하고 있는 강판의 두께와 형강의 크기를 설계변수로 하여 이를 자유롭게 설정할 수 있도록 프로그래밍화한 것이다.

개발된 알고리듬은 이들 설계변수의 크기가 같은 것끼리 그룹으로 나누어 유전형을 표현하도록 GA 를 개선한 것으로써 GA 의 개체 크기, 교배 확률, 돌연변이 확률, 선택 방법 및 교배 방법 등과 같은 파라미터의 최적화를 우선 수행하여 가장 적합한 값을 구하였다. 이는 파라미터의 변화에 따라 최적해의 정도와 계산 소요시간 상의 차이가 발생하는 점을 고려한 것이다.

본 프로그램을 활용하여 산출한 최적화 파라미터 값을 대상 구조물에 적용한 결과 전역해에 수렴하는 속도가 다른 값들에 비하여 양호함을 확인할 수 있었다.

그리고 실제 진동이 발생할 우려가 있는 선박의 컴퍼스 갑판 구조물을 대상으로 최적설계를 수행하여 문제를 해결함으로써 본 기법의 유용성을 검증하였다.

마지막으로, 새로운 조합 진화 알고리듬(RHEA: RSM-based Hybrid Evolutionary Alogrithm)을 제안하였다. 이는 해의 다양성과 수렴속도를 동시에 개선할 수 있는 GA, Tabu list 와 실험계획법으로 잘 알려져 있는 반응표면법(Response Surface Methodology)의 장점들을 활용·결합하여 만든 알고리듬이다. 이는 Tabu 탐색법의 Tabu list 를 GA 에 도입하여 해의 체계적인 다양성을 확보하고, RSM 을 적용하여 실제 목적함수 평가에 많은 시간이 소요되는 대형 구조물의 최적화에 수렴하는 속도를 향상시킨 것이다.

RSM 을 활용한 목적함수 평가 과정은 다음과 같다. GA 의 과정 중에 얻어진 정보를 이용하여 목적함수의 근사함수를 만들어 최적점을 찾은 후, 그 값을 다음 세대의 집단 중 소수(1 개)의 개체를 생성시켜 최적화를 수행한 것이다. 그에 따라 추가적인 목적함수 평가 없이 보다 최적화된 값을 얻을 수 있었다.

본 기법의 유용성을 검증하기 위하여 다음과 같은 연구를 수행하였다. 즉, 전통적으로 이용되고 있는 시험함수를 사용하여 기존의 탐색알고리듬인 GA 와 RHEA 의 결과치를 비교하여 보았다. 그리고 선미부에 위치한 청수탱크의 공진회피설계를 고려한 중량 경량화에 본 알고리듬을 적용하여 보았다. 그에 따라 새롭게 제안된 조합알고리듬(RHEA)이 해의 정확성이 높고, 수렴속도가 탁월함을 확인할 수 있었다.

이상과 같이 본 연구를 통해 보다 효율적이고, 높은 해의 정확성 및 신뢰도를 지닌 OPTSHIP (NASTRAN External Calling Styled Optimization Framework), NIP(Nonlinear Integer Program), RHEA(RSM-based Hybrid Evolutionary Alogrithm)를 개발하였다. 본 연구는 제안된 신기법을 특정 선박의 일부분에만 적용하여 그 유효성을 검증한 것으로 앞으로 여러 종류의 선박에 확대·적용하여 그 경험을 체계화하고, 보다 우수한 알고리듬 개발과 선박 최적설계 기술 발전을 위해 지속적인 노력을 기울여야 할 것으로 본다.

Acknowledgments

Twenty-one years have passed since I joined to Intelligent Mechanical Laboratory. During this period, I am indebted to many persons. Without their help, I could not complete this thesis. Here I would like to express my sincere appreciation to my advisor, Professor Bo-Suk Yang who took me into the vibration analysis field and taught me with passion and love. His passion of knowledge, members and lectures will be memorized forever

I also express my thanks to Prof. Dong-Jo Kim, Prof. Sung-Yong Bae, Prof. Byeong-Keun Choi and Dr. Su-Hyun Choi. They examined my thesis and gave me many valuable comments and suggestions.

I would like to thanks to Prof. U. J. Lim, Prof. K. Y. Lee, Prof. D. M. Bae, Prof. S. J. Lee, Prof. S. B. Kim, Prof. S. J. Kim, Prof. B. T. Kim, Prof. M. C. Yoon, Prof. Y. J. Kim and Prof. J. S. Goo. Their enthusiastic lectures are based on my thesis.

I would like to express my special thanks to members of Daewoo Shipbuilding & Marine Engineering Co., Ltd including the vibration and Noise R&D team. Thanks to their help, this thesis can be completed successfully.

My special thanks must go to Jin-Dae Song for sharing of idea and useful discussions about optimization programming. I wish to thank J.Y. Ha, Widodo and B. K. Jeong for their advice and correction.

Especially I would like to thank the members of my laboratory. Seniors gave me scientific passion. And I obtained many help and support from juniors. I would like to show my appreciation to my friends and their family: J. I. Yoon, B. S. Park, J. K. Lee, J. S. Kim, B. D. Jeong, B. K. Jeong and my 438 the same class for their support and believed in me without any doubts.

Finally, I am particularly grateful to my family: my parents, mother-in-law, brothers, sisters, brothers-in-law, sisters-in-law, my sons and my wife. Without their support and help, the hard process of study would not have been completed. Especially, my sons, Min-Je, Min-seo, they are my pride and their innocent laugh gave me always boundless energy. And my honey, Mi-Gyeong, she does her best for my sons and me. Her endless love and unfailing support is priceless to the completion of my thesis.

I would like to give this thesis to all of my indebted persons as a gift.

감사의 글

학부 및 대학원 생활을 거쳐 이 논문이 완성되기까지 여러 방면에서 도움을 주신 많은 분들께 이 자리를 빌어 감사의 말씀을 전하고자 합니다.

결코 짧지 않은 기간인 나의 인생 반을 양보석 교수님과 본 연구실과의 인연을 갖게 된 것은 저에게 가장 큰 행운이었습니다. 지난 20년여 동안 부족한 저를 이끌어 주시며 많은 가르침을 주신 양 보석 지도교수님께 가슴속 깊이 감사를 드립니다. 참된 연구자로서 항상 사회에 공헌할 수 있는 사회인으로서 몸소 실천하시는 교수님을 통해 저 또한 미약 하나만 본 받고 싶습니다. 또한 교수님을 제자들한테 양보하시고 저희들을 가족처럼 편안하게 해 주신 엄정미 사모님께도 깊은 감사를 드립니다.

바쁘신 와중에서도 저의 논문 심사를 맡아주시며 부족한 저의 학위 논문이 최소한의 형태나마 갖출 수 있도록 지도해 주신 김 동조 교수님, 배 성용 교수님, 최 병근 교수님, 최 수현 박사님께 깊은 감사를 드립니다.

열정적인 강의로 본 논문의 기초를 닦아주신 임 우조 교수님, 이 규용 교수님, 이 수종 교수님, 배 동명 교수님, 김 상봉 교수님, 김 성진 교수님, 김 병탁 교수님, 윤 문철 교수님, 김 용직 교수님 그리고 구 자삼 교수님 감사합니다. 본 논문을 작성하는 동안 룸메이트 로서 많은 아이디어를 공유하고 토론을 통하여 보다 더 논문을 알차게 만들어 준 송 진대 후배에게 고마움을 전합니다. 그리고 서투른 영어를 보다 더 매끄럽게 만들어준 하 종룡, Widodo 와 병권에게 고마움을 전합니다.

회사의 바쁜 일정에도 유학의 기회를 주신 대우조선해양㈜ 가족분들에게 고마움을 전하고 싶습니다. 특히, 유학준비 과정에 많은 도움을 주신 배영수 상무님, 김 만수 부장님, 최 수현 이사님 및 연구소와 인사팀 관계 분들의 고마움을 잊을 수 가 없습니다. 또한 유학 과정 중에 계속하여 연구할 수 있도록 힘써 주시고 격려하여 주신 이 성근 상무님, 한용섭 상무님, 배 재류부장님 및

연구소와 설계실 동료 및 선후배님께 감사의 말씀을 드립니다. 한 사람, 한 사람의 역할이 절실한 상황에서 오랜 공백 기간 동안 저의 역할을 대신하여 주신 진동소음연구팀의 오상헌 부장님, 백일국 부장님 김노성 부장님 이하 팀원들에게는 고마운 마음 보다 도 미안한 마음이 더 깊었음을 고백합니다. 그리고 지금은 진동팀을 떠났지만, 오랜 친구와 선후배로 남겨 있는 손성완님, 김영찬님, 강동억님, 정윤호, 제종준, 장강석, 서상윤, 민덕기 에게도 고마움을 전합니다. 본 논문이 완성되기까지 연구실 생활 동안 여러 선배님들의 가르침과 도움이 없었다면 이 논문은 완성되기 힘들었을 것입니다. 김 원철 교수님, 전순기 교수님, 김 인수 부장님, 김 남설 사장님, 임 우섭 본부장님, 유영훈 교수님, 김진욱 박사님, 최원호 박사님, 김종완 박사님, 안영공 박사님, 이장우 박사님 그리고 석사과정 시절에 함께 연구실 생활을 하였던, 최성필 박사, 박준호, 이재무 김병수, 하종훈, 하진수, 김영찬 박사, 주호진, 장우교, 김유신, 박형섭, 이윤희, 그리고 박사과정 시절 함께 알게 된 사랑스런 후배들: 철현, 소환, 용한, Han Tian, 영천, 용수, 호일, 종룡, 광진, 원우, 상일, 형은, 동호, 종덕, 민찬, 재갑, 선화, 애희, 도균, 기태, 기용, 백석, Widodo, Niu Gang, Di Xiao, Feng 박사에게 감사를 드립니다. 같이 수업을 하면서 많은 도움을 주신 신창혁님, 이수목 부장님, 길병래 교수님 감사합니다. 이경석 부장님 이하 해석팀 멤버와 조강문 부장님 최종석 부장님 이하 거주구 식구들, 영원한 벗 삼영회 친구들, 함께 유학 생활을 시작하면서 많은 격려를 주신 유민철님에게 고마움을 전합니다.

항상 힘이 되어준 고향 친구인 배선, 재갑, 진석, 병두, 경찬, 병권, 종오 및 438 동기회에 감사하고, 객지에 나와 끈끈한 정을 나눈 친구인 정인, 영포, 신우에게 감사하며, 특히 바보성 친구들과 그 가족들에게도 감사함을 전합니다.

저에게 무한한 사랑을 가르쳐 주신 부모님, 늘 그 사랑을 베풀어 주시기만 하셨던 두 분께 조그마한 기쁨과 자랑이 되길 바라는 마음으로 이 논문을 바칩니다. 큰형님과 형수님, 영만 형님과 형수님, 영삼 형님과 형수님, 누나와 자형, 그리고 미희와 안서방에게도 감사의 마음을 전합니다. 부족한 사위지만 저를 자랑스럽게 여기시고, 격려해주신 장모님과 사위 사랑도 느껴 보지 못하시

고 하늘 나라에 계시는 장인어른께도 깊은 감사의 마음을 전하며, 처남과 처제께도 감사를 드립니다. 너무나 착하고 순수한 나의 아들 민제와 민서, “아빠, 논문 이제 몇 장 썼어요?” 하고 볼 때 마다 확인하는 큰아들 민제, “아빠, 오늘 자고 가면 안돼요?” 하고 조금이라도 아빠와 같이 있고 싶어하는 까불이 민서, 두 아들의 응원과 웃음 덕분에 논문을 잘 마무리 할 수 있었습니다. 오직 사랑으로 공가네를 이끌어 가면서 이 논문을 만들기까지 힘찬 격려를 아끼지 않으며 내조해준 나의 아내 미경씨에게 미안함과 동시에 사랑과 감사함을 이 논문에 담아 선사합니다.