
 論 文

大韓造船學會論文集
 第 31 卷 第 3 號 1994年 8月
 Transactions of the Society of
 Naval Architects of Korea
 Vol. 31, No. 3, August 1994

유전적 알고리즘과 직접탐색법의 결합에 의한 효율적인 최적화방법에 관한 연구

이동곤*, 정성재**, 김수영***

A Study on the Efficient Optimization Method by Coupling Genetic Algorithm and Direct Search Method

by

D. K. Lee*, S.J. Jeong** and S.Y. Kim***

요 약

공학설계에 있어서 최적해를 얻기 위한 방법중의 하나로 최적화방법이 많이 사용되어 왔으나, 기존의 최적화방법에서는 설계점이 국부 최적점으로 빠져 들어갈 경우 그 영역을 벗어날 수 있는 방법이 없기 때문에, 최적화의 초기점을 달리하여 반복계산을 수행하여야 하는 불편한 점이 있었다.

유전적 알고리즘은 기존의 최적화방법에 비하여 다수의 설계점을 동시에 탐색하는 특성이 있어 국부 최적점에 빠질 가능성이 적은 반면, 계산시간이 많이 소요되고 전체 최적점 근처까지는 잘 수렴하나 정확한 최적점을 잘 찾지 못하는 한계가 있다.

본 연구에서는 유전적 알고리즘과 직접탐색법을 결합하여 이들의 단점을 보완한 즉, 전체 최적점을 보다 효율적으로 찾고 계산시간을 줄일 수 있는 방법을 제시하였다. 이 방법은 유전적 알고리즘을 이용하여 최적점이 존재하는 영역을 찾은 후에, 그 영역에서 직접탐색법을 이용하여 보다 정확한 최적점을 찾는 것으로, 예제를 통하여 제안된 방법의 유용성을 보였다.

Abstract

Optimization in the engineering design is to select the best of many possible design alternatives in a complex design space. In order to optimize, various optimization methods have been used. One major problem of traditional optimization methods is that they often result in local optima. Recently genetic algorithm based on the mechanics of natural selection and natural genetics is used in many application fields for optimization. Genetic algorithm is more powerful to local optima, but it

발표일자 : 1994년도 대한조선학회 춘계연구발표회('94.4.23)

접수일자 : 1994년 4월 15일, 재접수일자: 1994년 6월 17일

* 정회원, 선박해양공학연구센터

** 정회원, 부산대학교 대학원 조선해양공학과

*** 정회원, 부산대학교 조선해양공학과 기계기술연구소

requires more calculation time and has difficulties in finding exact optimum point in design variable with real data type generally.

In this paper, hybrid method was developed by coupling genetic algorithm and traditional direct search method. The developed method finds out a region for global optimum using genetic algorithm, and is to search global optimum using direct search method based on results obtained from genetic algorithm. By using hybrid method, calculation time is reduced and search efficient for optimum point is increased.

1. 서론

공학설계에 있어서 최적해를 얻기 위한 방법중의 하나로 최적화방법이 많이 사용되어 왔다. 최적해를 효과적으로 얻기 위하여 기존 최적화방법의 효율을 향상시키기 위한 여러가지 노력이 있어 왔으나, 아직까지 모든 문제에 효과적으로 적용할 수 있는 알고리즘은 개발되지 않고 있다. 최적화방법들의 알고리즘 그 자체가 가지고 있는 근본적인 문제점은, 설계점이 국부 최적점으로 빠져 들어갈 경우에 그 영역을 벗어나서 전체 최적점을 효과적으로 구할 수 있는 방법이 없다. 이러한 문제점을 극복하기 위하여, 최적화의 출발점을 바꾸어 가면서 여러번 반복계산을 하는 방법을 사용하여 왔다.

최근에 많은 연구와 응용사례들이 발표[1,2,3]되고 있는 유전적 알고리즘(Genetic algorithm)은, 다수의 설계점들을 동시에 탐색하는 특성이 있어 국부 최적점에 빠질 가능성이 적다. 그러나 유전적 알고리즘은 계산시간이 많이 소요되고, 설계변수가 실수형일 경우에는 전체 최적점 근처까지는 잘 수렴하나 정확한 최적점을 잘 찾지 못하는 한계가 있다[4].

본 연구에서는 전체 최적점이 존재하는 영역을 보다 잘 찾는 유전적 알고리즘의 장점과, 국부 탐색에 효율적인 직접탐색법중의 하나인 Hooke & Jeeves의 방법을 결합한 Hybrid Method를 구현하였다. 구현된 Hybrid Method는 유전적 알고리즘을 이용하여 전체 최적점이 존재하는 영역을 찾은 후에, 직접탐색법을 이용하여 보다 정확한 최적점을 찾는 방법이다. Hybrid Method를 사용함으로써 유전적 알고리즘과 직접탐색법의 단점을 보완하고, 장점을 살려 보다 효율적으로 최적화를 수행할 수 있다. 구현된 Hybrid Method를 수학적인 예제문제에 적용하여 그 결과를 보였으며, 유용성을 확인하였다.

2. 유전적 알고리즘

유전적 알고리즘은 생물들의 유전과 진화에서 착안된 방법이다. 지구상의 생물들은 지구의 환경변화에 적응하면서 진화하여 왔으며, 환경의 적응에 실패한 종(Species)들은 소멸되었다. 소멸되지 않은 종들은 나뉘대며 최적의 형태를 유지하기 위하여 계속적으로 변신하고 있다. 이런 과정이 공학문제에서 최적화의 과정과 유사하다는 개념에서 유전적 알고리즘이 출발하였으며 Fig. 1과 같이 7단계로 이루어진다.

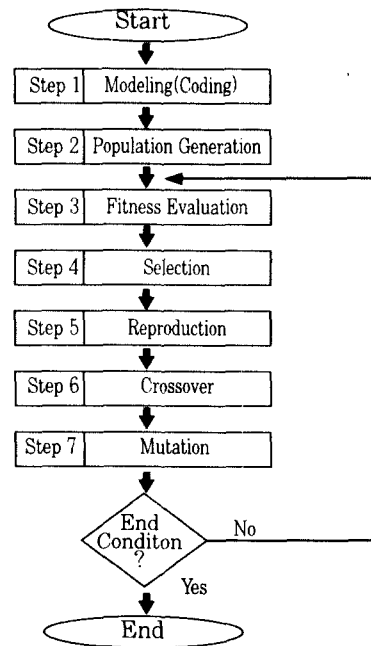


Fig. 1 Major procedure of genetic algorithm

유전적 알고리즘은 설계변수와 목적함수를 Fig. 2와 같이 컴퓨터내에 1 혹은 0의 값을 가지는 여러개의

Bit로 구성되는 문자열로 표현하고, 문자열을 조작하므로써 최적화를 수행하며 [1] 각 단계별 주요내용은 다음과 같다.

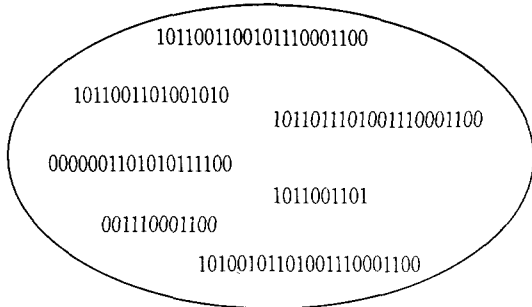


Fig. 2 Example of bit strings in genetic algorithm

- 단계 1) 유전자 형의 설정: 실세계의 최적화문제를 컴퓨터가 인식할 수 있도록 문자열로서 Coding한다. 일반적으로 0과 1의 값을 가지는 Bit열로서 표현한다.
- 단계 2) 유전자 집단의 생성: 단계 1)에서 결정한 문자열과 동일한 형태(같은 길이)를 갖는, 그러나 각 문자열의 요소가 다른 일정규모의 문자열의 집단을 무작위로 생성한다.
- 단계 3) 적합도(Fitness)평가: 단계 2)에서 생성된 문자열의 집단은 주어진 환경에 잘 적응할 수 있는것부터 그렇지 못한 것까지 포함되어 있다. 얼마나 잘 적응하는가 하는 것을 판단하기 위한 것이 적합도(목적함수)이다. 즉 각 문자열의 적합도를 계산하는 것으로, 단계 1)의 Coding에 대한 역변환으로 Decoding과정이라 볼 수 있다.
- 단계 4) 도태: 자연계에서 열등한 종들이 도태되는 개념을 도입한 것으로서, 계산된 적합도에 따라 다음 세대에서 살아 남을 개체를 선정하고 선정되지 못한 개체는 도태시킨다.
- 단계 5) 증식(Reproduction): 자연계에서 강하고 우수한 종들은 다른 종에 비하여 생존할 확률이 높으며 이런 종들의 개체수는 열등한 종들의 개체수보다 증가하는 것이 일반적이다. 이러한 개념을 유전적 알고리즘에서 도입한 것으로 단계 4)에서 도태시킨 개체수 만큼 우수한 종의 개체를 복사하여 전체 개체수를 초기의 개체수와 동일하게 만든다. 이렇게하여 적합도가 높은 것들은 즉, 목적함수의 값이 좋은 것들은 점점 숫자가 늘어

나게 되어 전반적으로 최적점으로 이동하게 된다.

- 단계 6) 교차(Crossover): 자연계에서 생식에 의하여 새로운 개체가 생성되는 개념을 유전적 알고리즘에 도입한 것이다. 증식과정에서 적합도가 높아진 개체들을 무작위로 두개씩 선정하고, 이들 개체들의 문자열 임의부분을 잘라 문자열을 서로 교환하여 새로운 개체를 생성한다. 새롭게 생성된 개체 가운데서 우수한 유전인자를 많이 받은 개체는 적합도가 높아지게 되어 증식과정에서 자신의 세력을 확장할 수 있게 된다.
- 단계 7) 돌연변이(Mutation): 자연계의 돌연변이 과정을 이용한 것으로서, 적합도를 높이기 위하여 개체를 구성하고 있는 문자열의 요소를 무작위로 하나 택하여 값을 바꾼다. 돌연변이는 개체의 다양성을 유지하게 하는데 유효한 방법으로서, 돌발적으로 적합도가 우수한 개체가 생성될 가능성이 있고 Multimodal Objective Function에서 국부적인 최적해에 빠질 가능성을 줄일 수 있게 하여 준다.

유전적 알고리즘은 단계 3)에서 단계 7)까지의 과정을 반복하면서 최적점을 찾아가는 방법이며, Random Number Generation 과정이 깊이 관계된다. 유전적 알고리즘에서는 Coding된 문자열과 실세계 설계 변수와의 관계를 Mapping 개념을 이용하여 연결한다 (Fig. 3). 이를 위하여 입력자료로서 설계변수들의 하한값과 상한값을 입력하고, 이들 값의 범위와 Coding된 문자열을 Mapping하므로써 설계변수의 형태가 정수, 실수, 정수와 실수 등에 관계없이 사용할 수 있다 [4].

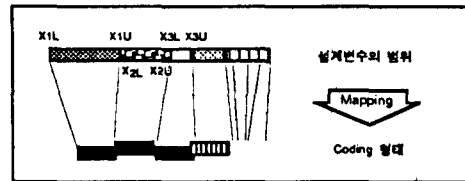


Fig.3 Relationship between design variables and codings

유전적 알고리즘이 모든 문제에 적합하다고는 할 수 없으나, 기존의 알고리즘들이 문제의 종류에 따라서 많은 제약을 받는 반면에 유전적 알고리즘은 상대적으로 문제의 종류에 크게 제한받지 않고 적용할 수 있는 장점이 있다[5]. 바꾸어 말하면, 유전적 알고리즘은 어

면 특정 종류의 문제를 해결하는데 있어서 기존의 방법에 비하여 효율은 높지 않을 수 있으나, 기존의 방법들이 적용하지 못하는 문제에도 적용이 가능하다.

3. 직접탐색법

직접탐색법(Direct Search Method)은 사용자가 입력한 Starting Point, Step Size 및 End Condition을 근간으로 설계공간의 임의 설계점을 알고리즘을 이용하여 탐색하여 가면서, 탐색된 설계점에서의 목적함수 값을 비교하여 보다 나은 방향으로 계속 탐색을 진행하는 방법이다. 본 연구에서 사용한 알고리즘은 Hooke & Jeeves의 직접탐색법[6]을 이용하였다. 이 방법은 크게 부분탐사(Local Search)와 전체이동(Global Move)으로 이루어져 있으며, 알고리즘은 다음과 같다.

Local Search

- L(1):
- 기준점 B_1 과 $(B_1 + H_1 E_1)$ 에서 목적함수 값을 계산한다.
 - $(B_1 + H_1 E_1)$ 의 목적함수 값이 B_1 에서 목적함수 값보다 좋으면 B_1 을 $(B_1 + H_1 E_1)$ 으로 대체한다.
 - 만약 $(B_1 + H_1 E_1)$ 의 목적함수 값이 B_1 에서 목적함수 값보다 좋아지지 않으면 $(B_1 - H_1 E_1)$ 에서 목적함수 값을 계산한다.
 - $(B_1 - H_1 E_1)$ 의 목적함수 값이 B_1 에서 목적함수 값보다 좋으면 B_1 을 $(B_1 - H_1 E_1)$ 으로 대체한다.
 - 만약 $(B_1 - H_1 E_1)$ 의 목적함수 값이 B_1 에서 목적함수 값보다 좋아지지 않으면 원래의 B_1 으로 돌아간다.
- L(2):
- 모든 설계변수에 대하여 L(1)의 과정을 수행한 후에 새로운 기준점 B_2 에서 Global Move를 한다
- 여기서, H_1 : Step Size
 E_1 : Unit Vector

Global Move

- G(1): $G_1 = (2 * B_2 - B_1)$ 으로 G_1 을 계산한다.
 G(2): G_1 의 목적함수 값이 B_2 에서 목적함수 값보다 좋아지면 Local Search를 한다.
 · 만약 G_1 의 목적함수 값이 B_2 에서 목적함수 값보다 좋아지지 않으면 B_2 에서 Local Search를 한다.

- G(3): $G(2)$ 과정에서 목적함수 값이 좋아지면 $G(1)$ 과정으로 간다.
 · 만약 $G(2)$ 과정에서 목적함수 값이 좋아지지 않으면 종료조건을 검사하고, 종료조건을 만족하면 종료한다. 종료조건을 만족하지 않으면 Step Size를 줄이고 L(1)과정으로 간다.

유전적 알고리즘과 직접탐색법은 모두 제한조건이 없는 최적화문제에만 적용할 수 있는 방법이다. 따라서, 이 방법들을 제한조건이 있는 최적화문제에 적용하기 위하여 External Penalty Function Method [6]를 사용하였다.

4. Hybrid Method 구현

일반적으로 기존의 최적화기법들은 빠른 시간내에 최적화가 이루어지나, 얻어진 최적점이 국부적인 최적점일 가능성이 있으므로, 입력 데이터를 바꾸어 가면서 여러번 계산을 하여야하는 번거러움이 있다. 반면에 유전적 알고리즘은 설계공간내의 여러점들을 동시에 탐색하므로 국부적인 최적점에 빠질 가능성은 적으나, 탐색점의 증가로 인하여 계산시간이 급격히 증가하고 최적화문제가 실수형 변수로 구성된 경우에는 최적점 부근까지는 빨리 도달하나 그 부근에서 수렴하며, 다목적함수의 경우에는 각 목적함수의 최적점을 정확하게 찾아내지 못하는 단점이 있다[4].

따라서 탐색시간을 줄이고 전체적인 최적점을 효율적으로 구하기 위하여 유전적 알고리즘과 직접탐색법을 결합한 Hybrid Method를 개발하였다. 유전적 알고리즘과 직접탐색법의 장점을 유지하면서 단점을 보완한 Hybrid Method는 먼저, 유전적 알고리즘을 이용하여 간략하게 최적화를 수행하여 최적점이 존재할 가능성이 높은 영역을 구한다. 유전적 알고리즘을 이용하여 얻어진 최적점을 Hooke & Jeeves Method의 입력 데이터 즉 출발점으로 사용하여 최적화를 다시 수행한다. 이와 같은 방법을 사용하면 국부적인 최적점에 빠질 가능성이 줄어들고, 기존의 최적화기법은 상대적으로 좁은 영역에서 탐색을 하게 되므로 보다 효율적이다. 유전적 알고리즘과 직접탐색법을 결합한 Hybrid Method를 SUN SPARC Engineering Workstation에 구현하였으며, Fig. 4에 그 구성도를 나타내었다.

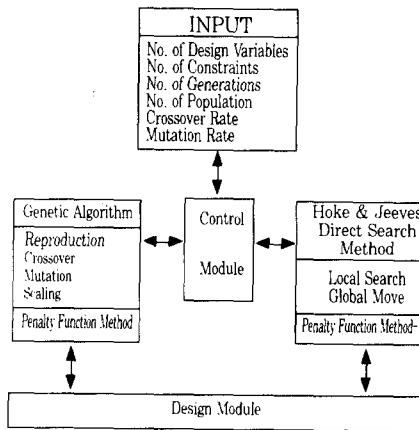


Fig. 4 System configuration of the developed hybrid method

5. 적용 예

5.1 수학적 예제

Problem 1: 설계변수가 1개이고 제한조건이 없는 문제

$$f = 3x^3 - 8x^2 - 6x + 9$$

Problem 2: 설계변수가 2개이고 제한조건이 4개인 문제

$$f = (1 + (x_1 + x_2 + 1)^2)(19 - 14x_1 + 3x_1^2 + 14x_2 + 6x_1x_2 + 3x_2^2) \\ (30 + 2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$$

Subject to :

$$-2.0 < = x_1$$

$$-2.0 < = x_2$$

$$x_1 < = 2.0$$

$$x_2 < = 2.0$$

Problem 3: 설계변수가 4개이고 제한조건이 10개인 문제

$$f = x_1 - x_2 - x_3 - x_1x_3 + x_1x_4 + x_2x_3 - x_2x_4$$

Subject to:

$$0 < = 8 - x_1 - 2x_2$$

$$0 < = 12 - 4x_1 - x_2$$

$$0 < = 12 - 3x_1 - 4x_2$$

$$0 < = 8 - 2x_3 - x_4$$

$$0 < = 8 - x_3 - 2x_4$$

$$0 < = 5 - x_3 - x_4$$

$$0 < = x_1, x_2, x_3, x_4$$

3문제 모두 최소값을 구하는 것으로서, 각 문제에 대한 계산결과를 Table 1, 2, 3에 각각 나타내었다.

Table 1 Calculation results of problem 1

	Genetic Algorithm		Hooke & Jeeves Direct Search Method		Hybrid Method	
$f(x)$	-9.86	-9.99	17.0	-10.0	-10.0	
x	-0.937	-0.998	2.0	-1.0	-1.0	
time	1.5	1.4	0.07	0.04	0.764	
i n p u t	A	100	200		200	
	B	30	15		5	
	C			1.5	50	
	D			0.1	0.1	0.1
$f_{opt} = -10.$		crossover rate = 0.4, mutation rate = 0.1				
$x_{opt} = -1.0$		A = population size, B = no. of generations				
		C = starting points, D = step size, time = sec				

Table 2 Calculation results of problem 2

	Genetic Algorithm		Hooke & Jeeves Direct Search Method		Hybrid Method	
$f(x)$	3.52	3.0	840.	30.	2.99	
x_1	-0.01	0.00	1.19	-0.60	0.00	
x_2	-0.97	-1.00	7.99	-3.99	-1.00	
time	3.9	5.8	0.03	0.03	0.32	
i n p u t	A	200	200		100	
	B	20	30		10	
	C			0.5, 0.5	-0.5, -0.5	
	D			0.3	0.3	0.3
$f_{opt} = 3.0$		crossover rate = 0.4, mutation rate = 0.1				
$x_{opt} = 0., -1.$		A = population size, B = no. of generations				
		C = starting points, D = step size, time = sec				

5.2 결과분석

얻어진 결과를 분석하면 다음과 같다. 유전자 알고리즘만을 사용하였을 경우에는 정확한 최적해를 찾는 경우가 적었고 상대적으로 시간이 많이 소요되었다. 유전적 알고리즘이 실수형의 최적화문제에서 정확한 최적해를 구하지 못하는 것은, Bit 문자열로 표현된 설계변수가 Decoding 될때 1대 1 대응관계가 이루어지지 않기 때문으로 생각된다. 유전적 알고리즘에서 계산횟수는 Population의 크기와 Generation의 횟수의 곱으로 표현되는데, 계산횟수가 같을 경우에는 Generation 수를 증가시키는 것보다는 Population의 크기를 증가시키는 것이 보다 효율적이었다. Hooke & Jeeves의 직접탐색법을 사용하였을 경우에는 계산시

Table 3 Calculation results of problem 3

		Genetic Algorithm		Hooke & Jeeves Direct Search Method		Hybrid Method
$f(x)$		-8.20	-13.2	-14.99	-12.99	-14.99
x_1		0.03	0.10	0.00	2.99	0.00
x_2		2.39	2.81	2.99	0.00	2.99
x_3		0.03	0.10	0.00	3.99	0.00
x_4		2.49	3.92	3.99	0.00	3.99
time		3.2	3.3	0.06	0.04	1.23
i n p u t	A	100	200			200
	B	30	15			5
	C			1.0	1.0	
				1.0	-1.0	
				1.0	1.0	
D			0.3	0.3	0.3	
$f_{opt} = -15.0$ $x_{opt} = 0.0, 3.0, 0.0, 4.0$ crossover rate=0.4, mutation rate=0.1 A=population size, B=no. of generations C=starting points, D=step size, time=sec						

간이 매우 적게 소요되었으나, 계산의 출발점에 따라서 국부최소점에 빠지는 경우가 있었다.

본 논문에서 구현한 방법을 예제문제에 적용한 결과, Hooke & Jeeves의 직접탐색법보다 계산시간이 많이 소요되는 점이 있었으나 출발점에 따라 국부최소점에 빠지는 경우가 없었다. 또한 유전적 알고리즘에 비하여 계산시간이 줄어들고, 상대적으로 정확한 최적점을 찾을 수 있었다.

유전적 알고리즘에서 어느정도까지 최적화를 수행한 후에, 직접탐색법으로 계산을 넘기는 것이 효율적인가 하는 것은 문제의 특성에 따라 달라질 것이다. 일반적으로는 유전적 알고리즘에서 목적함수 값이 더 이상 향상되지 않을 경우가 적절한 시점이라 생각된다. 그러나 계산중에 그러한 시점을 찾는 것이 쉽지 않으므로, 어느정도 계산이 수행된 후나, 목적함수의 평균값과 최소값이 동시에 크게 변화하지 않는시점을 고려하는 것도 하나의 방법이 될 것이다.

6. 결론

최적화의 효율을 향상시키기 위하여 유전적 알고리즘과 직접탐색법을 결합한 Hybrid Method를 구현하였다. 구현된 방법을 몇개의 문제에 적용하여 본 결과,

기존의 방법만을 사용한 경우보다 전체 최적점을 효과적으로 얻을 수 있었다. 유전적 알고리즘에서 계산횟수가 같을 경우에는 Generation 수를 증가시키는 것 보다는 Population의 크기를 증가시키는 것이 보다 효율적이었다.

본 논문에서 사용한 예제문제의 경우에는 최적점을 정확하게 찾을 수 있었으나, 설계공간의 형상이 복잡하여지면 최적점을 찾는 데 어려움이 있을 것으로 생각된다. 따라서 유전적 알고리즘에서 설계변수의 하한값과 상한값의 범위를 가능한 줄여주는 것이 효율적이다. 즉, 좁은 범위에서 같은 크기의 Population이 할당되면 최적점을 얻을 가능성이 증대된다. 또한 Mutation Rate와 Crossover Rate 등의 초기값도 유전적 알고리즘에서는 중요한 인자이다. 실제 공학문제에 보다 효과적으로 이용하기 위한 방법의 하나로서, 유전적 알고리즘의 설계변수의 범위와 직접탐색법의 탐색폭의 결정 및 유전적 알고리즘에서 직접탐색법으로 계산을 넘겨주는 시점의 결정이나 Mutation Rate와 Crossover Rate 등의 초기값 결정등에 전문가의 지식을 이용하는 지식베이스 시스템과의 결합도 필요할 것으로 생각된다. 즉, 실제 공학설계에서는 설계문제에 따라, 설계전문가들이 그 범위를 어느 정도 예상할 수 있으므로 전문가의 지식을 이용하는 것이 가능할 것이다[7].

참고 문헌

- [1] D.E. Goldberg, *Genetic Algorithms in Search Optimization & Machine Learning*, Addison-Wesley Pub. Company, 1989.
- [2] T. Okada, I. Neki, "Optimization of Ship Structural Design by Genetic Algorithm", *일본조선학회논문집*, Vol. 171, June 1992.
- [3] H. Maeda, T. Yamazaki, "Decision Method of Optimal Safety Measures for Mobil Offshore Structures by Means of Genetic Algorithm", *일본조선학회논문집*, Vol. 173, June 1993.
- [4] 김기화, "Genetic Algorithm에 의한 다목적함수 최적구조 설계", 서울대학교 박사학위논문, 1994년 2월.
- [5] L. Borup, A. Parkinson, "Comparison of Four Non-Derivative Optimization Methods on Two Problems Containing Heuristic and Analytic Knowledge", *Advances in*

- Design Automation, ASME*, Vol. 1, 1992.
- [6] M. G. Parsons, "Optimization Method for use in Computer-Aided Ship Design", *S-TAR Symposium, SNAME*, 1975.
- [7] 이동근, "최적화기법과 지식기반시스템", *대한조선학회지*, 1994년 6월.