

전체최적화를 위한 확률론적 탐색기법

Stochastic Search Techniques for Global Optimization

양 영 순*

Yang, Young Soon

김 기 화**

Kim, Gi Hwa

要 約

최적화기법의 최종 목표는 전체최적점(global optimum point)을 정확히 그리고 효율적으로 구하는 것이다. 이를 위해 확률론적인 탐색과정을 가지는 Simulated Annealing과 Genetic Algorithm에 의한 최적화과정을 살펴보고, 수학적함수와 트러스, 보 구조물에 대해 최적설계를 행하여 전체최적점에 도달한 신뢰도 및 계산 시간을 기준으로 기존의 확정론적 최적화기법에 의한 결과와 그 유용성을 비교 검토하였다.

Abstract

The final objective of optimization methods is to find global optimum accurately and efficiently. The optimization processes by simulated annealing and genetic algorithm which have stochastic search process are examined and are applied to several mathematical models and truss, beam structures. Then the robustnesses of these two methods are studied and compared with the results of deterministic optimization methods from the viewpoints of reliability and running time in obtaining the global optimum.

1. 서론

최적화기법의 최종 목표는 전체최적점(global optimum point)을 정확히 그리고 효율적으로 구하는 것이다. 많은 국부최적점(local optimum point)이 존재하는 경우 기존의 확정론적(deterministic) 최적화기법에서는 전체최적점을 구하기 위해 초기점을 달리하여 여러번계산을 수

행하여 그 중 가장 좋은 결과를 취하는 방법을 사용하였다. 이렇게 구한 결과의 신뢰도를 높이기 위해서는 다수의 초기점에 대한 반복계산을 통해 설계가능영역 전구간에 대해 검토가 요구되나, 설계변수가 많고 제한조건이 복잡한 경우 설계가능영역에 대한 인식이 매우 어려워 초기점 결정에 어려움이 따른다.

따라서 보다 효율적으로 전체최적점에 도달하

* 정회원, 서울대학교 조선해양공학과 부교수

** 서울대학교 조선해양 공학과 박사과정

이 논문에 대한 토론을 1992년 9월 30일까지 본 학회에 보내주시면 1993년 3월호에 그 결과를 게재하겠습니다.

기 위한 최적화기법으로 기존의 방법에서는 현재 탐색중인 점 근처에서 가장 좋은 정보만을 이용하는 국부적인 탐색방법만을 사용하나, 본 연구에서는 사용할 Simulated Annealing, Genetic Algorithm은 자연계의 원리를 기초로 하여 확률론적(stochastic)탐색방법을 적용하고 있어 많은 경우 국부적인 탐색범위를 벗어나 전체적인 최적화 방향으로 수렴해가는 특징이 있다.

Simulated Annealing은 금속의 소냉(annealing)과정에 기초한 방법으로 매 탐색과정시 제한적으로 목적함수 증가의 방향으로 이동을 허용하여 국소최적점을 벗어날 수 있는 가능성을 높여준 방법으로, Kirkpatrick 등에 의해 제안된 후 많은 최소화문제에 적용되어 왔다[1,2]. Genetic Algorithm은 생명체의 자연유전법칙에 기초한 방법으로 초기에 원하는 만큼의 문자열(string) 형태에 설계점을 무작위 발생시켜 그 중 적합성이 좋은 것 주위의 점들이 다음단계에 발생할 확률을 크게 가지며 교배(crossover)와 변종(mutation)의 변화과정을 거쳐 보다 좋은 방향으로 탐색을 진행하는 방법으로, Goldberg 등에 의해 많은 연구가 이루어져 왔다[3,4].

본 논문에서는 위에서 언급한 확률론적 탐색 최적화기법의 기본 계산과정에 대해 정식화 하였고, 전체최적점에 도달한 신뢰도 및 계산시간을 기준으로 수학적함수와 트러스, 보 구조물에 대해 최적설계를 행하여 그 유용성을 기존의 확정론적 최적화기법과 비교 검토하였다.

2. 정식화 과정

다음과 같이 구성된 일반 최적화(최소화) 문제에 대해 확률론적 탐색과정을 가지는 Simulated Annealing(SA), Genetic Algorithm(GA)과 확정론적인 탐색을 행하는 Gradient Projection Method(GPM) 및 비선형 최적화기법인 Simplex Method(SUMT/NM)의 기본 최적설계 계산 과정을 살펴보자.

$$\begin{aligned} & \text{Minimize } F(x) \\ & \text{subject to } g_i(x) \leq 0, i=1,2, \dots, m \end{aligned} \quad (1)$$

2.1. Simulated Annealing(SA) [1,2]

기존의 확정론적 최적화 알고리즘에서는 국소 최적점이 존재하는 골짜기로 설계점이 들어간 경우 그 영역을 벗어날 수 있는 방법이 없어서 대부분 시행착오법에 의한 새로운 초기치 선정방법을 채택했으나, Simulated Annealing에서는 목적함수의 증가를 제한적으로 허용하여(uphill movement) 국소최적점 부근에서 벗어날 수 있는 가능성을 부여함으로써 보다 좋은 방향으로 탐색이 계속 되어 결국 전체최적점에 도달할 확률을 높여주고 있다.

금속의 소냉 과정을 살펴보면, 어떠한 금속을 용융점까지 가열한후 천천히 온도를 내려주면 그 금속의 전체 에너지는 점점 낮아져 결국 최소치에 도달하게되며, 그 중간단계에서 어떠한 상태*i*에서 다음상태*j*와 에너지 차이 $\Delta E(E_j - E_i)$ 가 양수일 때는 다음의 확률한도 내에서 제한적으로 이동이 허용된다(Metropolis criterion)

$$P_{accept} = \begin{cases} \exp\left(\frac{-\Delta E}{k_B T}\right) & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases} \quad (2)$$

여기서 T : 온도

k_B : Boltzman 상수

Simulated Annealing은 위의 금속의 소냉 과정을 (1)식으로 표현되는 일반적인 최적화문제에 적용한 방법으로, 목적함수 $F(x)$ 가 에너지에 해당되며, 어떤상태 *i*에 다음상태 *j*로 변환은 각 설계변수를 임의로 증분시켜 구하고, 이 때 $\Delta F (=F(x_j) - F(x_i))$ 가 양수일 때는 $F(x_j)$ 를 허용할 확률을 (2)식의 Metropolis criterion을 따른다. 그리고 반복 계산이 진행됨에 따라 온도 T 를 조금씩 줄여 주어 목적함수의 증가상태를 허용할 확률을 점점 감소시켜 적절한 작은 확률이 될 때까지 계산을 반복한다.

(1)식의 최적설계서 P_{accept} 와 SA의 기본 계산과정은 다음과 같다. 기호 *i*는 현상태, *j*는 다음상태를 뜻한다.

--목적함수 증가시 허용확률(P_{accept})--

$$P_{accept} = \begin{cases} \exp\left(\frac{-\Delta E}{T}\right) & \text{if } \Delta F > 0 \\ 1 & \text{if } \Delta F \leq 0 \end{cases} \quad (3)$$

여기서 $\Delta F = F(x_j) - F(x_i)$

c : ΔF 의 평균값 계산을 위한 상수

T : 제어 상수

ΔF 가 양수일 때(0,1) 범위에서 임의의 수 ξ 를 발생시켜 $\xi < P_{accept}$ 면 증가를 허용하고 그렇지 않으면 취소한다. 목적함수의 증가가 허용될때마다 상수 c 와 M 는 다음과 같이 계산된다.

$$c = \frac{M \times c + \Delta F}{M + 1} \quad (4)$$

$$M = M + 1$$

여기서 c 와 M 은 초기값은 각각 1과 0이다.

(3)식의 제어상수 T 는 목적함수 증가시 허용확률을 결정하는 양으로 계산이 진행됨에 따라 조금씩 감소시켜야 하며 그 감소량(CF)은 다음과 같이 결정한다.

P_s = 목적함수 증가의 초기 허용확률

P_f = 목적함수 증가의 말기 허용확률

N = 반복계산 횟수

이라고 할 때,

초기 제어상수는

$$T_0 = \frac{-1}{\ln(P_s)} \quad (5)$$

이고, 감소량(CF)은

$$CF = \left[\frac{\ln(P_s)}{\ln(P_f)} \right]^{1/(N-1)} \quad (6)$$

이 되어, i 번째 계산에서 제어상수는 다음과 같다.

$$T_i = T_{i-1} \times CF \quad (7)$$

--Simulated Annealing 계산과정--

<step 1> P_s , N , P_f 결정하고, (5)식에서 초기 제어상수 T_0 를 계산하고, (6)식에서 제어상수 감소량 CF 를 계산한다.

<step 2> 초기 설계변수벡터 x_i 를 선택한다.

<step 3> 설계변수의 증분량을 임의로 선택한 후, x_i 의 하나의 변수에 더하여 다음 설계변수벡터

x_j 를 계산한다.

<step 4> x_j 의 허용 여부를 다음과 같이 결정한다.

- (1) x_j 가 제한조건을 위반한 경우는 x_i 가 제한조건을 만족하면 취소하고, x_i 가 제한조건을 위반한 경우는 제한조건을 위반량의 대소에 따라 허용여부를 결정한다.
- (2) x_j 가 제한조건을 만족한 경우는 x_i 가 제한조건을 위반한 경우는 허용하고, x_i 가 제한조건을 만족한 경우는 (3)식의 Metropolis 식을 따른다.

<step 5> step 4에서 목적함수의 증가가 허용된 경우에는 (4)식에서 c 와 M 을 계산하고, x_i 의 모든 변수들에 대해 증분의 허용여부가 검토될 때 까지 step 3에서 반복 계산한다.

<step 6> 설계변수를 $x_i \leftarrow x_j$ 로 대체하고, 제어상수 T 를 (7)식에 의해 변경한 후, N 번의 반복계산이 완료된 후 또는 적절한 수렴범위를 만족하면 계산을 종료한다.

2.2. Genetic Algorithm(GA) [3,4]

기존의 최적화기법에서는 다음 단계로의 이동이 한점에서 한점으로의 이동이므로 여러개의 국소최소점이 존재하는 문제의 경우 보다 나쁜 영역으로 탐색이 이루어질 위험이 많았으나, Genetic Annealing에서는 설계영역에 대한 많은 정보를 통해 보다 좋은 영역에 가중치를 두어 여러 영역을 동시에 탐색하여 국소최소점에 빠질 가능성을 줄여 주고 있다.

생명체의 자연유전 법칙에 의하면 주위의 환경에 적합한 형질(fitness)을 가진 개체가 생존할 확률을 크게 가지며 교배(crossover)와 변종(mutation)의 과정을 겪으며 보다 좋은 방향으로 진화하고 부적합한 형질의 개체들은 진화의 과정에서 점차 도태되어 간다. 이와 같은 진화를 계속되풀이 하면 결국 주어진 환경에 가장 적합한 개체로 형성될 것이다.

Genetic Algorithm은 이와 같은 자연유전 법칙을 (1)식으로 표현하는 일반 최적화문제에 적용한 방법으로 초기에 원하는 만큼의 설계점을 문자열(string)의 형태로 발생시켜 목적함수 $F(x)$ 의 변경된 형태로 표시된 $F'(x)$ 의 값에 의해 각 설계점

에 적합성(fitness)을 부여한 후 그 적합성이 클수록 다음 계산 단계인 교배와 변종에 참여할 가능성을 크게하여(reproduction), 적합성이 좋은 설계점에 가까운 점들이 다음단계에 보다 많이 형성되어 계산이 반복될수록 좋은 점부근에서 탐색이 보다 많이 행해져 결국 좋은 방향으로 탐색이 진행된다.

이상의 Genetic Algorithm의 기본 계산과정을 설명하면 다음과 같다.

<step 1> 교배 및 변종이 발생할 확률 P_c , P_m 과 각 문자열의 길이(Lchrom), 총반복계산횟수 및 설계영역 등을 결정한다.

<step 2> 초기 문자열 발생

각 설계변수는 길이 Lchrom을 가지며 2진수로 표현된 문자열의 형태를 가지며(예, 11011; Lchrom=5), 문자열의 각 인수는 임의로 0과 1들 중 하나의 값을 가진다. 설계변수가 n개인 경우는 ($n \times Lchrom$)의 총길이를 가지는 하나의 문자열이 하나의 설계점을 형성하게 하여 원하는 수 만큼의 문자열을 임의로 발생시킨다.

<step 3> 문자열 변환 및 적합성 계산

2진수로 표현된 각 문자열을 10진수로 변환하고(decode), 주어진 설계영역 안으로 선형변환 시킨후 목적함수 $F(x)$ 값을 계산하여 좋은 문자열에는 적합성(fitness)을 크게, 나쁜 문자열에는 작게 부여한다. 하지만 (1)식과 같은 최소화문제에서는 목적함수값이 작은 설계점이 좋은 점이므로 적합성 계산시에는 목적함수의 변형된 형태인 $F'(x)$ (예; $F_{max} - F(x)$ or $1/F(x)$)를 사용하며, 그 변형된 형태는 항상 영 이상의 값을 가지도록 한다.

<step 4> 선택 (reproduction, selection)

적합성의 크기에 비례하여 다음 단계에서 선택될 확률을 부여한다. 선택에는 여러가지 방법이 있으나 여기서는 단순히 $F'(x) / \sum F'(x)$ 의 확률로 선택되도록 하였다.

<step 5> 교배 및 변종

교배는 (0,1) 사이의 임의의 수가 확률 P_c 보다 작은 경우, 선택에 의해 임의의 2개의 문자열을 임의로 짝지은 후 1과(Lchrom-1) 사이에 임의의 상수 k 를 선택하고 짝지은 2개의 문자열의 ($k+1$)과 Lchrom사이의 인수들을 서로 교체한다.

Lchrom이 5인 초기 문자열을 A_1, A_2 라 하고 변종이 일어난 후 문자열을 각각 A'_1, A'_2 라 하면 다음과 같다.

$$\begin{aligned} A_1 &= 011111 & A'_1 &= 01101 & (8) \\ A_2 &= 101101 & A'_2 &= 10111 \end{aligned}$$

mutation은 crossover에 의해 한쪽 방향으로 string들이 치우치는 것을 보완하기 위해 아주 작은 확률 (P_m)을 가지고 string의 각 인수 단위로 변종을 시켜준다. 즉 0은 1로, 1은 0으로 변환한다.

새로이 형성된 문자열들을 가지고 step 3부터 반복 계산하여 초기에 결정한 반복계산 수 또는 적절한 수렴범위를 만족하면 계산을 종료한다.

위의 과정은 제한조건이 없는 최적화문제에 대한 기본 계산 과정 이므로 (1)식과 같이 제한조건이 있는 문제는 다음의 벌칙(penalty) 함수를 사용하여 제한조건이 없는 문제로 치환 계산한다.

$$P(x, r) = F(x) + r \sum_{j=1}^m [\max(g_j(x), 0)]^2 \quad (9)$$

(9)식에 의해 제한조건을 위반한 설계점들은 적합성 매우 나쁘게 되어 다음 설계점 형성에 참여할 가능성이 매우 떨어지게 된다.

2.3 Gradient Projection Method (GPM) [5]

목적함수 $F(x)$ 의 최소화를 위해서는 주어진 제한조건을 위반함이 없이 $F(x+\delta x)$ 를 감소시키는 방향으로 δx 를 결정해야 한다. 이를 위해 복잡한 비선형으로 구성된 $F(x)$ 와 $g_i(x)$ 를 1차항 까지 전개한 선형화된 근사값을 사용한다. 즉

minimize

$$\delta F = \frac{dF}{dx} \delta x = (\Lambda^0)' \delta x \quad (10)$$

subject to

$$\begin{aligned} \bar{g}_i + \delta \bar{g}_i &\leq 0, & i=1, 2, \dots, p \\ \delta x' W \delta x &\leq \xi^2 \end{aligned} \quad (11)$$

여기서

W : 정치 행렬(positive definite matrix, 일반적으로 대각행렬)
 ξ : 매우 작은 수($\xi > 0$)

-(bar) : \in -active한 제한조건($g_i + \epsilon \geq 0, \epsilon > 0$)

P : \in -active한 제한조건의 갯수

2.4. Simplex method(SUMTNN) [6]

Simplex 방법은 설계변수 보다 한 차원 큰 다면체에 의해 reflection, expansion, contraction, reduction의 기본 탐색과정을 가지는 Nelder와 Mead방법에 설계점이 제한조건을 위반한 경우 external penalty를 준 방법으로, (1)식으로 정의된 최소화문제의 external penalty 함수(P)는 다음과 같다.

$$P(x, r) = F(x) + r \sum_{i=1}^m \max(g(x), 0) \quad (12)$$

(12)식에 의해 제한조건이 없는 문제로 치환되며, 다면체의 크기가 적절한 수렴범위 이하가 되면 계산을 종료한다.

3. 수학적 함수에의 적용 및 결과 검토

국소최소점이 다수 존재하는 수학적 함수에 2절의 4가지 최적화기법을 적용하여 각 방법의 유용성에 대해 도출된 해가 전체 최적치에 얼마나 가까운가?(effectiveness), 해의 도출까지 얼마의 계산시간이 소요되는가?(efficiency) 그리고 다양한 문제에 대해 일관되게 해를 찾아 내는가?(generality)의 3가지 관점에서 비교 검토 하였다.

3.1. 적용례

1) 시험 함수

$$\min. \quad F(x) = x_1 + 2x_2 \quad (13)$$

subject to

$$g_1(x) = 1 - \frac{x_1^2 + (x_2 - 5)^2}{25} \leq 0$$

$$g_2(x) = 1 - \frac{(x_2 - 5) + x_2^2}{25} \leq 0$$

$$g_3(x) = 1 - x_1 \leq 0$$

$$g_4(x) = 1 - x_2 \leq 0$$

국소최소점의 위치 및 목적함수값은 다음과 같

다.

(1.00, 9.90)	F=20.80
(5.00, 5.00)	F=15.0
(9.90, 1.00)	F=11.90(global)

2) Goldstein-Price 함수(GP)

$$F = [1 + (x_1 + x_2 + 1)^2(19 + 14x_1 + 3x_2^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 + 27x_2^2)] - 2 \leq x_i \leq 2, i=1,2 \quad (14)$$

4개의 국소최소점이 존재하며 그 점의 위치와 목적함수값은 다음과 같다.

(1.2, 0.8)	F=840.0
(-0.6, -0.4)	F=30.0
(1.8, 0.2)	F=84.0
(0.0, -1.0)	F=3.0(global)

3) Hartmann 함수(H3)

$$F(x) = - \sum_{i=1}^m c_i \exp(- \sum_{j=1}^n a_{ij}(x_j - p_{ij})^2) \quad (15)$$

$$0 \leq x_i \leq 1, \quad i=1,2, \dots, n$$

n=3, m=4이며 상수 a_{ij}, c_i, p_{ij} 의 값은 Table 1.에 수록하였다. 4개의 국소최적점이 존재하고 그 값은 대략 다음과 같다.

$$\{x\}_i^* \approx (P_{i1}, \dots, P_{in})$$

$$F_i^* \approx -c_i, \quad i=1, 2, 3, 4$$

4) Shekel 함수(S5)

$$F(x) = - \sum_{i=1}^m [\sum_{j=1}^n (x_j - a_{ij})^2 + c_i]^{-1} \quad (16)$$

$$0 \leq x_j \leq 10, \quad j=1,2, \dots, n$$

n=4, m=5이며 상수 a_{ij}, c_i 의 값은 Table 2.에 수록하였다. 5개의 국소최적점이 존재하고 그 값은

Table 1. Input data for Hartmann function

i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	c_i	p_{i1}	p_{i2}	p_{i3}	p_{i4}
1	3	10	30	1	0.3689	0.1170	0.2673		
2	0.1	10	35	1.2	0.4699	0.4387	0.7470		
3	3	10	30	3	0.1091	0.8732	0.5547		
4	0.1	10	35	3.2	0.03815	0.5743	0.8828		

Table 2. Input data for shekel function

i	a _i				c _i
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.2
4	6	6	6	6	0.4
5	3	7	3	7	0.4

대략 다음과 같다.

$$\{x_i^*\} \approx (a_{i1}, \dots, a_{i4})$$

$$P_i \approx -1/c_i, \quad i=1, 2, 3, 4, 5$$

3.2. 계산결과 및 비교 검토

각 방법의 특성을 살펴보기 위해 먼저 시험 함수에 대해 탐색의 과정을 검토하였다.

Simulated Annealing은 시험 함수에 대해 여러 초기점에서 출발한 결과 전체 최적점에 모두 잘 도달하였고 계산시간도 양호하였다. Simulated Annealing의 특징인 목적함수 증가의 허용량을 결정하는 P_s 를 0.9와 0.5인 경우에 대해 계산한 결과 그림 1에 나타난 바와같이 P_s 가 큰 경우는 계산의 초기과정에서 천천히 목적함수의 감소 방향으로 탐색이 진행되고 P_s 가 작은 경우에는 초기에는 목적함수의 감소 방향으로 탐색이 급격히 진행되다가 제한조건선에 도달한 후 그 제한조건선을 따라 국부최소점 지역을 벗어난 과정을 보여주고 있다. 그러므로 일반적인 경우 국부최소점들 주위의 골짜기가 매우 깊고 여러개 존재하는 경우에는 P_s 를 크게하는 것이 효과적일 것이며, 그렇지 않은 경우에는 너무 큰 P_s 의 사용은 초기 계산에서 시행착오가 많을 것이다. 또한 Simulated Annealing의 계산시 증분량의 크기를 (-1, 1), 또는 (-2, 2)사이의 임의의 값을 취했는데 대부분의 경우 최적점 근처에서 탐색의 진전속도가 매우 느려서 보다 자세한 탐색을 위해 최적설계가 종료한 후 계산된 최소점을 초기점으로 하고 목적함수 증가의 허용확률(P_{accept})과 총 반복계산 횟수를 줄여 주며 증분량의 범위를 매우 작게 취하여 재계산을 행하여 보다 정확하게 최소점을 찾을 수 있었다.

Genetic Algorithm은 시험 함수의 계산시 여러 초기 집단의 분포에 대해 전체 최적점에 도달하였으나 계산시간이 많이 소요되었으며 대부분의 경

우 참값에 정확히 도달하지 못했고 그 부근의 점 (near optimum pt.)에서 더 이상 진전이 이루어지지 못했다. 그 이유는 2진수로 표시된 문자열을 설계영역안으로 선형변환할 때 $1/(2^{Lchrom}-1)$ 의 간격을 가지는 이산화한 값을 취하기 때문이며, 문자열의 길이 $Lchrom$ 을 크게하여 주면 그 간격이 매우 좁아지나 문자열의 값이 가질 수 있는 경우의 수와 전체 반복계산수와의 상대적 비가 커지므로 여전히 한계가 존재한다. Genetic Algorithm은 설계점의 집단의 형태로 탐색이 이동되므로 다른 방법과의 비교를 위해 초기 설계점의 발생 영역을 인위적으로 제한하고, 그 다음 계산부터는 전체설계영역을 사용하여 주었다. 시험 함수의 경우 초기 설계영역을 $x_1; (6.5, 7.5), x_2; (14, 15)$ 사이로 그리고 GP함수로 x_1, x_2 모두 (1.95, 2.0)으로 좁혀주어 초기 설계점의 발생 영역을 좁혀 주었다. 결과는 이러한 인위적인 설계점의 발생에의 제한에도 불구하고 전체 최적치에 잘 도달하였다. 실제 계산 즉 이러한 인위적인 제한이 없을 때 전체 설계영역에서 계산이 진행될수록 설계점의 분포를 살펴보았는데, 그림 2에 나타난 바와 같이 초기에는 설계영역 전체에 걸쳐 골고루 설계점이 분포되어 있으나 계산이 진행되어 감에 따라 최소점에서 멀리 떨어진 설계점들과 제한조건을 위반한 설계점들은 점차 소멸해져서 최후에는 전체 최소

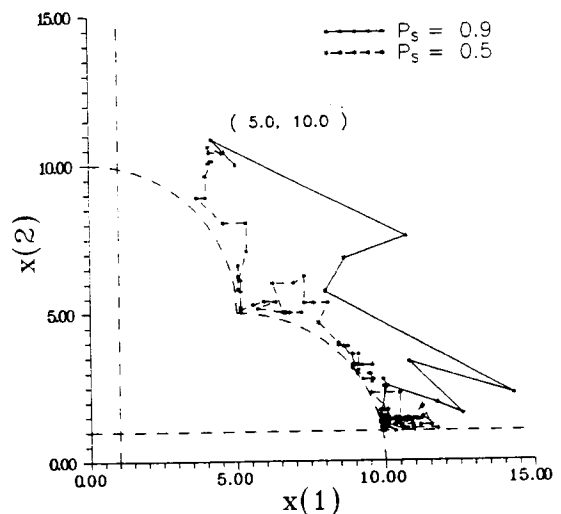


Fig. 1 Trajectory to optimal point of SA for TEST function

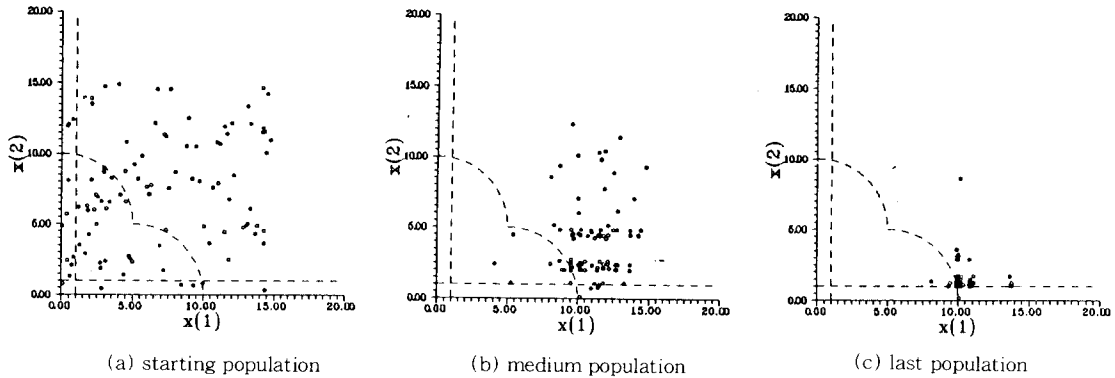


Fig. 2 String distribution of GA for TEST function

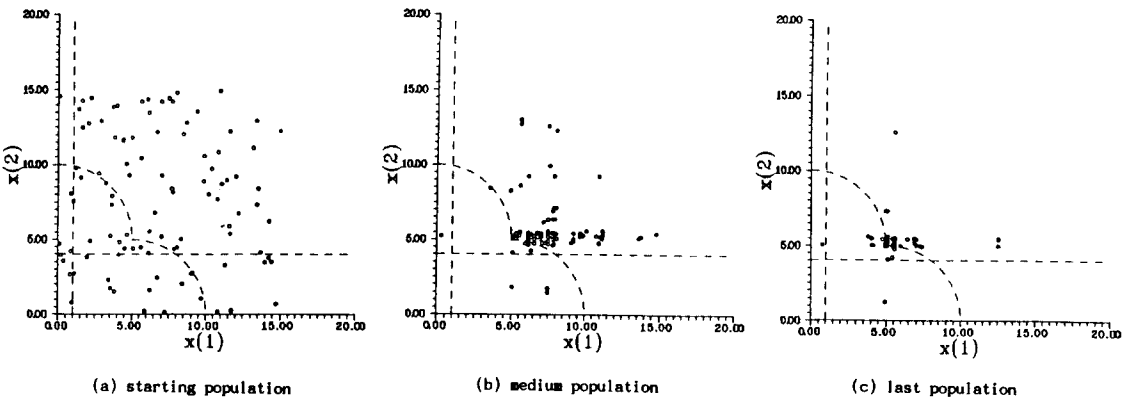


Fig. 3 String distribution of GA TEST function

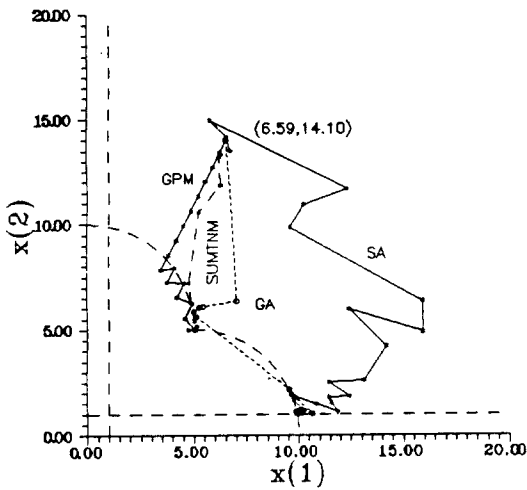
점 부근에 설계점들이 밀집하여 분포함을 알 수 있다. 이러한 설계점의 분포의 이동을 점검하기 위해 제한조건 g_4 를 $x_2 > 4$ 로 바꾸었더니 그림 3과 같이 앞의 예제의 경우 전체최소점이 제한조건을 위반했으므로 그 다음 최소점에 설계점들이 밀집되어 가는 경향을 볼 수 있다. 그리고 현 단계에서 최소점이 다음단계에서 발생하지 않을 위험이 있으므로 다음단계에서 가장 나쁜 설계점과 교체하여 주어 계산의 효율을 증가시켜 주었다.

상기 두 방법과 기존의 최적화 방법을 서로 비교하기 위해 시험 함수에 대해 계산을 하여 얻은 결과를 그림 4에 도시하였다.

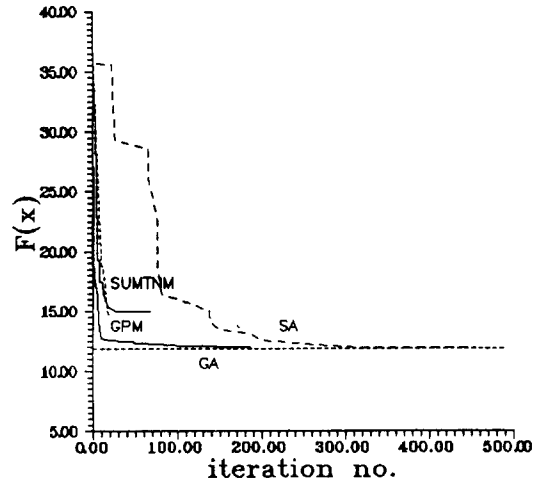
Gradient Projection 방법에 의한 계산은 설계점이 제한조건을 만족하는 경우 $-\nabla F$ 방향으로 탐색이 진행되다가 제한조건에 도달한 경우 제한조건선을 따라 지그재그의 형태로 목적함수가 감소하는 방향으로 탐색이 진행되고, Simplex방법

은 구성된 다면체의 각 점중 가장 나쁜 점의 반대 방향으로 탐색이 진행된다. 따라서 두 방법 모두 국부최소점이 존재하는 골짜기로 설계점이 들어간 경우 그영역을 벗어날 수 있는 방법이 없었다. 그러므로 여러개의 국부최소점이 존재하는 경우 초기시작점을 달리하여 여러번 시도하여 보아야 하는데 설계공간의 형태에 대한 인식이 부족한 경우 초기점 설정에 어려움이 있고 여러번의 시도 후에도 계산된 최적점에 대한 신뢰도가 떨어진다.

그리고 앞의 주어진 4문제에 대해 계산 횟수와 목적함수의 관계를 각 함수에 따라 표 3에 수록한 초기값에 대해 계산하여 그림으로 도시하였다(그림 4-그림 7). 모든 그림에서 Simulated Annealing의 곡선은 실제로는 목적함수 증가의 허용으로 인해 곡선이 위로 증가하는 현상이 많이 존재하나 그림의 단순화를 위해 감소경향만을 도시하였다. Genetic Algorithm에 대한 곡선은 매 계

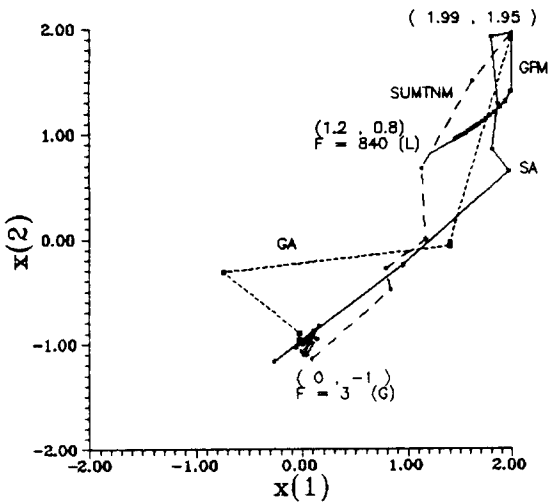


(a) trajectory in design space

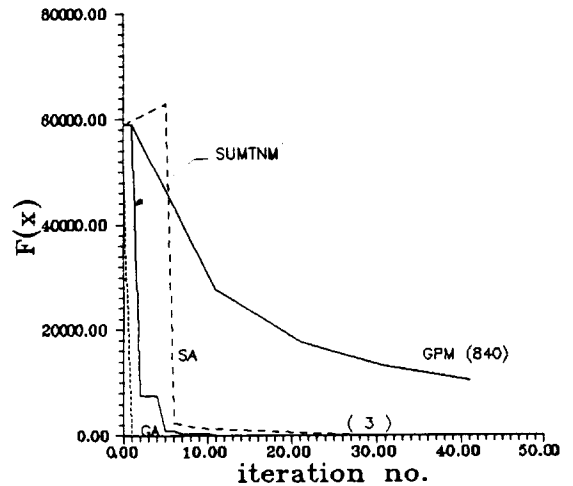


(b) object function vs. iteration

Fig. 4 Optimization of TEST function



(a) trajectory in design space



(b) object function vs. iteration

Fig. 5 Optimization of GP function

Table 3. Starting point for mathematical functions

	Starting point
Test	(6.59, 14.01)
GP	(1.99, 1.95)
H 3	(0.0, 0.0, 0.0)
S 5	(0.0, 0.0, 0.0, 0.0)

산마다 가장 좋은 설계점의 값을 도시한 결과이고, 시험 함수(그림 4(b))와 GP 함수(그림 5(b))에서는 앞에서 설명한 인위적인 초기 설계영역을

준 경우이고 나머지 함수에 대해서는 초기부터 전체 설계영역을 사용하였으며, 그 곡선을 살펴보면 전체최소점 부근에 도달하는데는 작은 계산횟수가 소요됨을 알 수 있다. 모든 경우에서 Simulated Annealing과 Genetic Algorithm은 전체 최적점에 잘 도달했고 Gradient Projection 방법과 Simplex 방법은 대부분 국부최적점에 도달했음을 볼 수 있다. 여기서 주의할점은 계산횟수의 산정은 각 방법마다 개념이 다르기 때문에 계산시간의

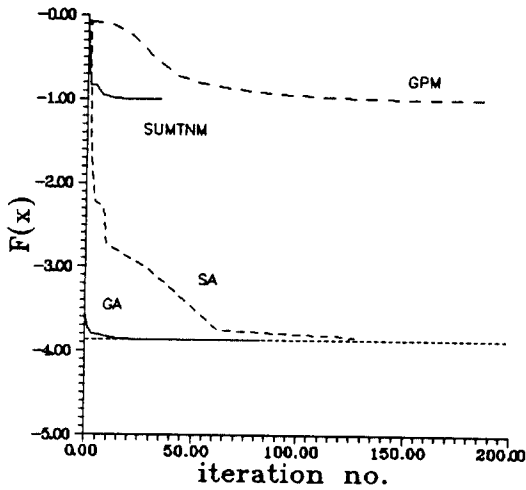


Fig. 6 Optimization of H3 function

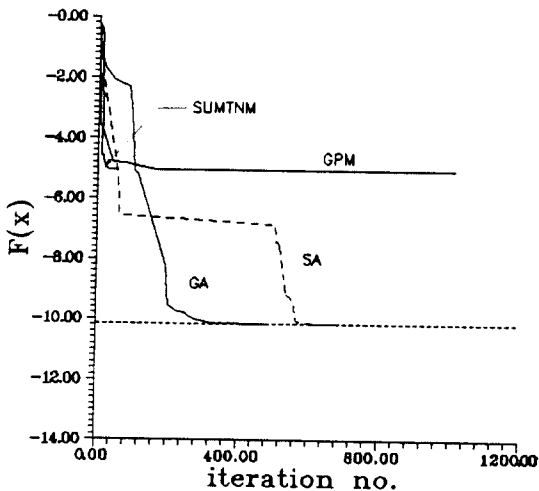


Fig. 7 Optimization of S5 function

비교는 계산횟수로 비교하면 안된다.

이상의 4가지 최적화기법에 의해 3.1. 절의 함수들의 최소값을 구한 결과 Simulated Annealing과 Genetic Algorithm은 모든 문제에 대해 전체최소점에 잘 도달하였다. 이에 반해 Gradient Projection 방법과 Simplex 방법은 전체최소점에 가까운 초기점에 대해서만 전체최소점에 도달했고 S5 함수의 경우는 두 경우 모두 국부최적점에 도달하였다(표 4참조).

전체최적치에 도달한 경우에 대해서만 계산시간을 비교한 결과 Simplex 방법과 Simulated

Table 4. The number(n) of finding global minimum for N different starting points(n/ N)

	S A	G A*	G P M	SUMTNN
Test	5/5	2/2	2/5	2/5
G P	5/5	2/2	2/5	4/5
H 3	2/2	2/2	1/2	1/2
S 5	2/2	2/2	0/2	0/2
WELD	1/1	1/1	0/1 ⁽¹⁾	0/1
TRUSS	2/2	1/1	1/2 ⁽¹⁾	1/2 ⁽¹⁾
error	0.28%	3.4%	0.0%	5.4%

* ; checking only whether finding global optimum because it runs from string set (population)

(1) occurrence of singular matrix for one case

(2) successively 3 iterations for one case

Table 5. The running time ratio to global optimum (real time: sec, IBM 486 PC)

	S A(time)	G A	G P M	SUMTNN
Test	1 (0.05)	240	0.1	0.1
G P	1 (0.11)	139	10	0.01
H 3	1 (0.44)	43	4	0.2
S 5	1 (0.44)	470	3(*)	0.2(*)
WELD	1 (1.10)	114	×	0.2(*)
TRUSS	1 (212.8)	11	0.01	0.3

(*); local optimum point

Annealing이 우수한 결과를 주었고 Genetic Algorithm이 가장 많은 시간이 소요되었다(표 5 참조).

현재까지의 결과로는 전체최적치에 도달할 신뢰도의 기준으로는 Simulated Annealing과 Genetic Algorithm 모두 믿음만 하고 계산시간의 측면까지 고려하면 SA가 더 우수하다고 말할 수 있다.

4. 실제 구조물에서의 적용 및 검토

4.1. 용접보의 최소비용설계 [7]

그림 8과 같은 길이 14인치의 용접된 보에 하중(F) 6000lb가 작용하고 있을 때 구조물의 최소비용설계를 하려한다. 소요되는 비용은 인건비, 용접비 및 재료비가 요구된다. 설계변수를 (18)식과 같이 할 때 목적함수 및 제한조건은 다음과 같다.

$$x = [x_1, x_2, x_3, x_4]^t = [h, l, t, b]^t \quad (18)$$

$$\min. F(x) = (1+c_3)x_1^2x_2 + c_4x_3x_4(L+x_2) \quad (19)$$

여기서 $c_3=0.10471$
 $c_4=0.04811$

subject to

$$\begin{aligned}
 g_1(x) &= \tau(x) - \tau_d \leq 0 \\
 g_2(x) &= \sigma(x) - \sigma_d \leq 0 \\
 g_3(x) &= x_1 - x_4 \leq 0 \\
 g_4(x) &= F - P_c(x) \leq 0 \\
 g_5(x) &= \delta(x) - \delta_d \leq 0 \\
 0.125 &\leq x_1 \leq 10. \\
 0.100 &\leq x_2 \leq 10. \\
 0.100 &\leq x_3 \leq 10. \\
 0.100 &\leq x_4 \leq 10.
 \end{aligned}
 \tag{20}$$

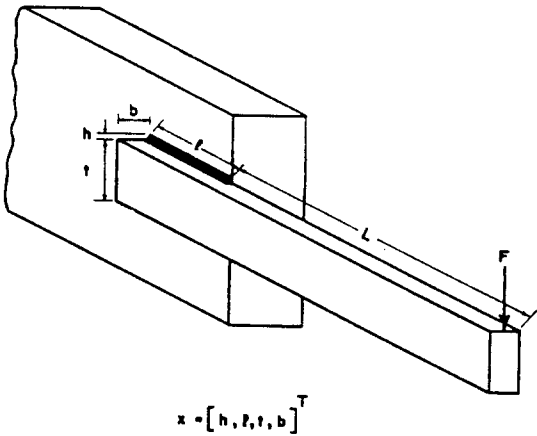


Fig. 8 The welded beam

여기서 g_1 : shear stress constraint ($\tau_d=13600$ psi)
 g_2 : bending stress constraint ($\sigma_d=30000$ psi)
 g_4 : buckling constraint
 g_5 : deflection constraint ($\delta_d=0.25$)

그의 각 설계응력, 좌굴응력, 변위에 대한 식들은 참고문헌 [7]에 의해 계산하였으며, 최소치는 다음과 같다.

$$\begin{aligned}
 x^* &= [0.2444, 6.2187, 8.2915, 0.2444] \\
 F^*(x) &= 2.38116 (\$)
 \end{aligned}$$

4.2. 10-bar truss의 최소중량설계 [5]

그림 10의 트러스 구조물에 대해 각 부재의 면

적을 설계변수 (x_i)로 하고 전체중량을 목적함수로 하며 각 부재에 작용하는 축응력을 제한조건으로 하여 최소중량설계를 행하였다.

$$\min. F(x) = \gamma \sum_{i=1}^{10} x_i l_i \tag{21}$$

$$g_i(x) = \frac{|\sigma_i|}{\sigma_i^a} - 1.0 \leq 0, \quad i=1,2, \dots, 10$$

$$g_i(x) = 1.0 - \frac{x_i}{x_i^a} \leq 0, \quad i=11,12, \dots, 20$$

허용응력 σ_i^a 는 25ksi, 최소단면적 x_i^a 는 0.1 inch²로, 재료의 밀도는 0.1로 하였으며, 최소점의 위치 및 크기는 다음과 같다.

$$\begin{aligned}
 x^* &= (7.938, 0.1, 8.062, 3.938, 0.1, 0.1, 5.745, \\
 &\quad 5.569, 5.569, 0.1) \\
 F^* &= 1593.2
 \end{aligned}$$

4.3. 계산결과 검토

(1) 용접부의 계산결과

각 최적화방법에 의한 결과를 그림 9에 나타내었는데, 초기점은 (1,7,4,2)로 하였다. Genetic Algorithm에 의한 계산은 $P_c=0.09$, $P_m=0.01$ 및 $Lchrom=25$ 로 하였는데, 계산이 최소점으로 진행되다가 약 15%의 오차에서 더 진전 되는데 너무 많은 시간이 소요되어 초기 $P_m=0.1$ 로 하고 점점 감소시켜 마지막 계산시 $P_m=0.01$ 로 줄여주어 계산시간을 줄여줄 수 있었다. 그리고 Gradi-

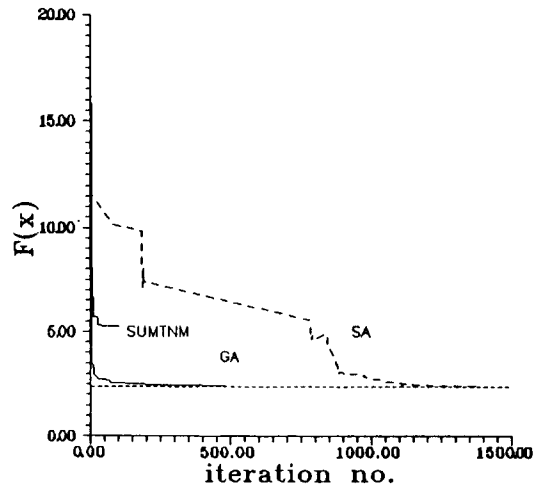


Fig. 9 Optimization of welded beam

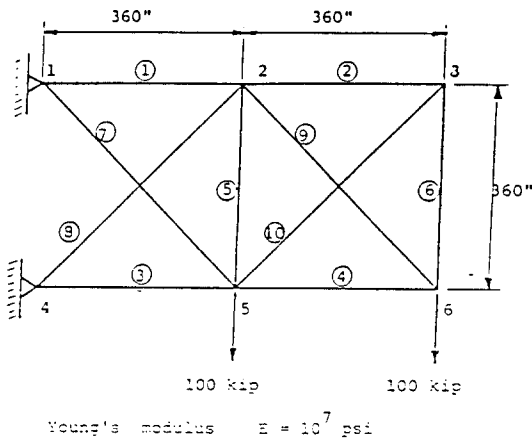


Fig. 10 The 10-bar truss

ent Projection 방법에 의한 최적설계를 위해 dg/dx 의 미분량이 필요한데 그 값을 엄밀하게 구하기 어려워 유한차분법에 의해 근사적으로 구했으며 δx 의 계산중 이들 미분값으로 구성되는 행렬에 singular가 발생하여 계산을 더이상 수행할 수 없었다. 이러한 현상은 미분값은 근사적으로 구하지 않더라도 발생할 가능성이 있으므로 추후 연구가 요망된다. Simplex방법에 의해서는 국부최소점에 수렴하였고 Simulated Annealing과 Genetic Algorithm은 전체최소점에 잘 도달했음을 보여준다. 계산시간은 3.절의 예제에서와 마찬가지로 Genetic Algorithm이 많이 걸렸다.

(2) 10-bar truss의 최소중량설계 결과

본 모델에서는 국부최소점을 발견할 수 없었다. 초기점을 모든변수를 10으로 하여 최소중량설계를 한 결과를 그림 11에 나타내었다. 모든 방법에서 전체최적치에 잘 도달하였고 계산시간은 Genetic Algorithm이 가장 많이 소요되었다. Gradient Projection 방법으로 계산시 응력에 대한 설계변수의 미분량이 필요한데 이는 설계감도해석을 통해 구하였으며 ϵ -active한 제한조건에 대한 감도만이 필요하므로 보조변수법(adjoint variable method)을 사용하였다[8,9]. 그리고 Simplex방법의 계산중 형성된 다면체로는 목적함수의 감소없이 수많은 시행착오를 되풀이 하는 경향을 보여 목적함수의 감소가 계속 없을 때는 그때까지의 계산결과를 시작점으로 하고 다면체를 다

시 구성하여 주어 계산을 진전시켜 주었다. 그림 11의 결과는 계산횟수 1000번과 1500번에서 다면체의 재구성을 통해 계산된 결과이다.

각 방법에 의한 상기의 두 구조물에 대해 계산된 정도를 표4와 표5에 수록하였다. 3절에서의 결과와 마찬가지로 전체최적점에의 도달할 신뢰도는 Simulated Annealing과 Genetic Algorithm이 우수하고 계산시간은 Simulated Annealing이 뛰어났다.

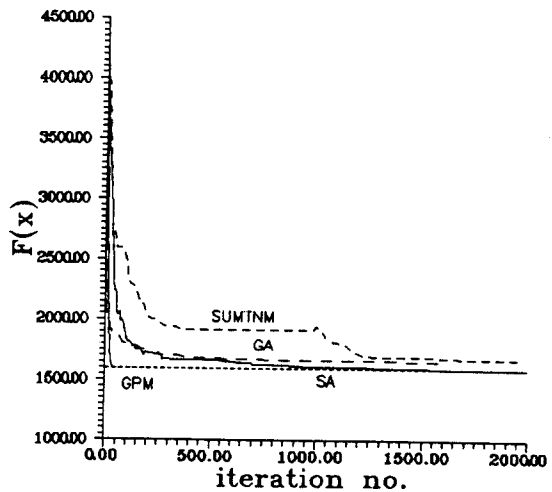


Fig. 11 Optimization of 10-bar truss

5. 결론

이상의 연구과정에서 각 방법의 유용성을 검토한 결과 다음과 같은 결론을 얻었다.

(1) 전체최적점에 도달할 신뢰도를 기준으로 Simulated Annealing과 Genetic Algorithm은 매우 우수한 결과를 나타내었으며, 계산시간까지 고려하면 Simulated Annealing이 더 좋은 결과를 주었다.

(2) Simulated Annealing은 문제의 특성에 따라 적절한 초기 목적함수 증가의 허용확률의 사용이 효과적이며, Genetic Algorithm은 전체최적점 부근까지의 도달에는 적은 계산이 소요되는 특성이 있다.

(3) Simulated Annealing과 Genetic Algo-

ithm은 목적함수와 제한조건에 대한 미분량을 요구하지 않은 직접 탐색기법을 사용하므로 실제 구조물의 최적설계에의 적용이 용이하다.

(4) 앞으로의 연구과제는 Simulated Annealing과 재해석기법 또는 감도해석을 결합하고, Genetic Algorithm은 각 변수마다 서로 다른 string 길이를 사용 등의 과정을 통해 계산의 효율을 증대시켜야 한다.

참 고 문 헌

- [1] Kirkpatrick S., Gelate C.D., Jr., Vecchi M.P. "Optimization by simulated annealing", Science, 220, 671-680, 1983.
- [2] Aarts E., Korst J. "Simulated annealing and Boltzmann machines", John Wiley & Sons, 1989.
- [3] Goldberg D.E. "Genetic algorithms in search, optimization & machine learning", Addison-Wesley, 1989.
- [4] 이경호 "중양단면 최적설계를 위한 전문가시스템 개발에 관한 연구", 서울대학교 조선공학과 공학석사논문, 1990.
- [5] Haug E.J., Arora J.S. "Applied optimal design", John Wiley & Sons, 1979.
- [6] Nelder J.A., Mead R. "A simplex methods for function minimization", Computer J., Vol.7, 1965.
- [7] Reklaitis G.V., Ravindran A. and Ragsdell K. M. "Engineering optimization", John Wiley & Sons, 1983.
- [8] Haug E.J., Choi K.K. and Komkov V. "Design sensitivity analysis of structural systems", Academic Press, 1986.
- [9] 양영순, 김기화 "정적구조응답에 대한 설계감도 해석", 선박구조연구회, 1990.

(접수일자 : 1992. 1. 31)