

## 最適値 計算을 위한 漸減多點探索法과 그 學習알고리즘의 提案

正會員 金 周 弘\* 正會員 孔 徽 植\* 正會員 李 洸 植\*

### A Proposal of Descent Multi-point Search Method and Its Learning Algorithm for Optimum Value

Joo Hong Kim\*, Whue Sik Kong\*, Kwang Jeek Lee\* *Regular Members*

#### 要 約

본 논문에서는 目的函數에 대한 最適値 計算法으로 점감 다점 탐색법과 그 학습 알고리즘을 제안하였다. 漸減 多點 探索法에서는 一變數 變換法으로 결정되어 직접 탐색법을 적용하기 위한 다점 탐색점 수를 점차 감소하였다. 學習 알고리즘에서는 앞서 탐색된 결과를 이용하여 새로운 탐색 구간을 설정하고 동일한 탐색을 반복 수행하였다.

제안한 알고리즘의 성능을 검토하기 위하여 종래의 직접 탐색법으로 최적치 산출이 불가능하거나 불확실한 것으로 알려진 Rosenbrock, Powell, Colville 등의 함수에 적용하였다.

적용결과 모든 함수의 최적치 탐색에 성공하였으며, 특히 절대 최적치 산출 알고리즘으로 우수한 계산 범위를 확인하였다.

#### ABSTRACT

In this paper, the decrease multipoint search method and its learning algorithm for optimum value computation method of object function is proposed. Using this method, the number of evaluation point according to searching time can be reduced multipoint of the direct search method by applying the univariate method. And the learning algorithm can repeat the same search method in a new established boundary by using the searched result.

In order to investigate the efficiency of algorithm, this method is applied to Rosenbrock and Powell, Colville function that are impossible or uncertain in traditional direct search method.

And the result of application, the optimum value searching on every function is successful. Especially, the algorithm is certified as a good calculation method for producing global(absolute) optimum value.

---

\*東國大學校 電子工學科  
Dept. of Electronics Engineering Dongguk Univ.  
論文番號: 92-85(接受1992. 3. 28)

## I. 序 論

컴퓨터 계산 능력의 향상에 힘입어 최적 설계, 제어, 추정 등의 최적화 기법에서 설정하는 목적함수(cost function or performance indices or object function)의 최적치(최소치 또는 최대치) 문제는 공학을 위시한 자연과학은 물론, 사회과학의 여러분야에서 필요성이 증대하고 있다.

Unimodal형 1차원 목적함수의 최소치 탐색법에는 golden section<sup>[2,3,4,5,6]</sup>, dichotomous<sup>[3]</sup>, Fibonacci<sup>[5]</sup> 탐색법등이 있으며, 다차원 목적함수의 탐색법으로는 random walk<sup>[3]</sup>, modified Newton-Raphson<sup>[2]</sup>, Hooke와 Jeeves의 Pattern 탐색<sup>[3]</sup>, Rosenbrock의 rotating coordinate 탐색<sup>[3]</sup>, Burhardt의 adaptive search(ASA)<sup>[3]</sup>, 3점 탐색(3 point search algorithm)<sup>[11]</sup>등의 수많은 알고리즘들이 제안되었으며, multimodal형 목적함수의 최소치 탐색법에는 Synman의 multi-start<sup>[10]</sup>, Anderssen의 random<sup>[11]</sup>, Ge와 Qin의 filled function<sup>[13]</sup>등이 있으나 신뢰성, 속도, 적용 가능한 함수의 종류 등과 같은 모든 요구를 동시에 만족시키는 계산법은 찾기 어려우므로 수치 계산법의 주요 연구과제이다.

본 논문에서는 다차원 목적함수의 절대(global or absolute) 최소치를 탐색하는 알고리즘으로 3점 탐색법<sup>[1]</sup>을 확장하여 탐색구간 내에 임의로 설정한 다점(Po)에서 직접 목적함수를 계산하는 탐색법으로 최소치를 탐색하고, 다점 탐색을 반복 할 때마다 탐색점의 갯수를 점차로 감소시키는 漸減 多點 探索法과 그 결과를 참고로 탐색구간을 변경하고 이같은 과정을 계산된 횟수만큼 반복 수행하여 절대 최소치를 산출하는 漸減 多點 探索法과 그 學習알고리즘을 提案한다. 그리고, 제안한 알고리즘의 성능을 검토하기 위하여 어려운 문제로 알려져 있는 Burhardt<sup>[13]</sup>, Optimum Design<sup>[2]</sup>, Optimization<sup>[3]</sup>, Foundation<sup>[5]</sup>에서 시험(test)함수로 채택한 Rosenbrock, Powell, Colville 함수 등에 대한 최소치를 탐색하여 정확성, 안정성, 신뢰성 및 유용성을 확인하였다.

## II. 最小值 探索 計算法

최소치 문제에서 극치(extremum value) 계산의 대상인 목적함수  $F(\mathbf{X})$ 는 일반적으로 식(2.1)과 같이

변수  $\mathbf{X}$ 의 2차 형식(quadratic form)으로 표시된다.

$$F(\mathbf{X}) = \mathbf{X}^T \cdot \mathbf{R} \cdot \mathbf{X} \quad (2.1)$$

$$\mathbf{X} = [x_1, x_2, \dots, x_N]$$

여기서,  $\mathbf{X}$ 는  $N$ 개의 변수를 성분으로 하는  $N$ 차 벡터(vector)이며,  $\mathbf{R}$ 은 가중함수로  $N \times N$  차인 계수행렬(coefficient matrix)이다.

최소치 계산 문제는 목적함수  $F(\mathbf{X})$ 가 最小值  $F^*(\mathbf{X}^*)$ 로 되는  $\mathbf{X}$ 의 탐색점  $\mathbf{X}^*$ 를 구하는 것이다. 이때, 변수  $\mathbf{X}$ 는 일반적으로 다른 독립변수들의 함수로 구성되므로, 범함수(functional)에 따른 해석법을 적용한다.

최소치 계산법으로 목적함수의 미분 및 보조 함수에 대한 계산이 필요한 conjugate gradient, steepest descent 등의 간접 탐색법과 보조 함수의 계산이 필요없이 여러가지 방법으로 결정된 탐색점에서 목적함수만을 계산하여 최소치를 산출하는 직접 탐색법(direct search method)으로 나뉘어지고 있다.

직접 탐색법에서 일변수 변환법(univariate method)은 多變數系에서 하나의 변수만을 선택하여 값을 변화시키고, 나머지 다른 변수는 상수로 취급하는 방법으로 목적함수를 계산하여 최소치를 탐색하고, 이를 교대로 모든 변수에 적용하는 방법이다.

또한, 1차원으로 구성되는 목적함수  $F(\mathbf{X})$ 의 값이 탐색구간의 시작점  $X_i$ 와 종료점  $X_r$ 내에서 단지 하나의 최소치만 존재하는 그림 1-(a)의 unimodal 형과 그림 1-(b)와 같이 최소치가 2개 이상 존재하는 multimodal 형이 있다.

전자인 unimodal형 목적함수의 최소치를 탐색하는 기법에는 독립변수가 1차원인 경우의 최소치 탐색법으로 exhaustive, golden section, dichotomous, Fibonacci 및 interpolation method<sup>[2,6]</sup>등이 있으며, 2개 이상의 다차원으로 구성되는 경우의 최소치 탐색법으로 random walk, Hooke and Jeeves, Rosenbrock<sup>[2,6]</sup>, 3점 탐색 알고리즘<sup>[1]</sup>, ASA<sup>[8]</sup>등이 있으나 steep valley, ridge 및 sharp corner에서의 처리와 적용 가능한 종류가 제한되는 문제가 있었다. 특히 Rosenbrock, Powell, Colville 및 Rao에서 예제로 사용한 시험함수들은 여러 다항식에 분포되어 있는 독립변수의 차수(order)와 계수가 커다란 차이를 갖고 있어 계수에 따른 기울기의 변화율이 매우 심하고 상이하며, 그림 1-(b)와 같이 목적함수  $F(\mathbf{X})$ 가

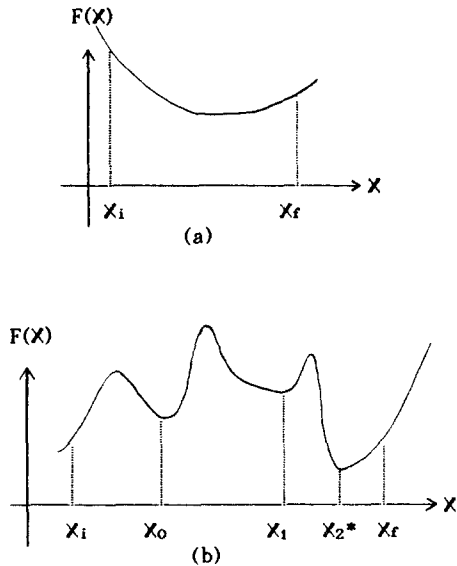


그림 1. 1차원 목적함수의 (a)unimodal와 (b)multimodal 특성

Fig. 1. Properties (a) unimodal and (b) multimodal of one-dimensional object function.

multimodal 형을 갖는 탐색구간에서는 적용이 불가능하였다.

후자인 multimodal 형의 경우는 局部 最小值(local minimum value)가 2개 이상 존재하므로 節對 最小值를 탐색하기 위하여 목적함수를 미분하는 multi-start<sup>10)</sup>방법과 보조함수인 filled function을 이용하는 방법<sup>13)</sup> 등의 간접 탐색법과 탐색구간내에서 임의의 점을 결정하여 목적함수를 계산하고 최소치를 산출하는 직접 탐색법을 적용하는 random search<sup>11)</sup> 등이 있다.

2개 이상의 변수로 구성되어 다차원이고, multimodal 형의  $F(X)$ 에서 임의의 탐색구간  $R$ 내에 존재하는  $\{X_i\}$  점들에서 최소치는 식(2.2)에 의해 결정되므로<sup>2.11)</sup>,

$$F^*(X^*) = \min_{1 \leq i \leq m} F_i(X_i) \quad (2.2)$$

여기서,  $m$ 는  $R$  내에서 결정되는 목적함수 탐색점의 전체 갯수를 나타낸다.

여러개 국부 최소치를 갖는  $F_i(X_i)$ 의 절대 최소치  $F^*(X^*)$ 를 탐색하기 위한 알고리즘으로 본 논문에서

는 3점 탐색법을 확장하여 최소치 탐색을 위한 탐색구간 내에 목적함수의 탐색점을 임의의 다점( $P_0$ )으로 설정하고, 설정한 다점에서 직접 탐색법과 일변수 변환법으로 탐색하며, 다점의 수를 반복하여 감소시켜 탐색을 계속하는 漸減 多點 探索法과 그 결과를 기억하여 학습하고, 학습한 결과를 중심으로 탐색구간을 변경하며, 계산된 학습 횟수만큼 동일한 탐색을 반복 수행하는 알고리즘을 제안한다.

### III. 漸減 多點 探索 計算法

Multimodal형 목적함수의 최소치 탐색을 위한 계산법으로 3점 탐색법<sup>14)</sup>을 확장한 점감 다점 탐색 계산법은 다음과 같다.

#### 1. 1次元 函數인 경우

$F(X)$ 에서  $X=[x]$ 인 목적함수가 그림 2와 같이 탐색구간  $x_i$ 에서부터  $x_f$  사이에 최소치가 여러개 존재하는 multimodal 형인 경우에는 설정한 임의의 탐색점의 수  $P$ 로 탐색구간을 등분하고, 분할구간 내에서 unimodal 형으로 가정하면 절대최소치의 탐색이 가능하다.

즉, 설정한 초기 탐색점의 수를  $P=P_0$ , 탐색구간  $R_0=x_f-x_i$ , 탐색점 간의 間隔  $d_0=R_0/P_0$ 로 정하고, 그림 2의 중앙점  $x_0=(x_i+x_f)/2$ 를 중심으로  $d_0$ 의 간격으로 등분한  $r=P_0/2$ 개의 점들에서  $F(x)$ 를 계산하여 그들 중에서 최소치인  $F^*(x^*)$ 를 구한다. 그리고, 이때  $F^*(x^*)$ 가 되는 변수  $x^*$ 를 탐색하여 최소치를 구한 것으로 제 1회 탐색을 종료한다.

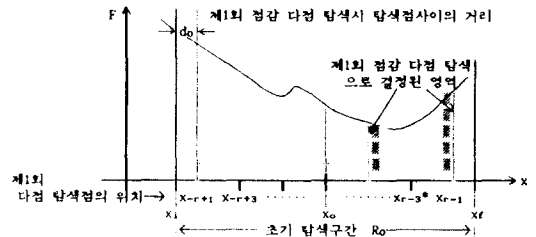


그림 2. 1차원 목적함수의 점감 다점( $P^N$ 점) 탐색 알고리즘

Fig. 2. Decrease multi( $P^N$ )-point search algorithm of one dimensional object function.

그러므로, 제  $i$  회 다점 탐색은

$$\begin{cases} P_i = P_{i-1} - 2 \\ R_{i0} = (2 \cdot d_i - 1) / P_{i-1} \\ d_i = R_{i0} / P_{i-1} \end{cases} \quad (3.1)$$

이 된다.

식 (3.1)은 제  $i$  회 다점 탐색을 표현한 것으로, 목적함수 탐색점의 수  $P_i$ 는 전회의 탐색 절차  $i-1$  회에서 2씩을 줄여 漸次 減少시켰으며, 다점 탐색의 반복 횟수가  $P_i = P / 2$ 로 될 때 종료하였다.

### 2. 多次元 函數의 경우

$X = [x_1, x_2]$ 로 2차원 함수인 경우에 제 1 회의 탐색점은 각 변수의 탐색구간 내에서 동일한 간격을 갖는  $P$ 개의 점을 그림 3과 같이 점선의 교점으로 결정하고, 일변수 변환법으로 탐색하되  $F^*(X^*)$ 가 되는 탐색점  $X^* = [x_1^*, x_2^*]$ 를 구하고, 1차원의 경우를 類推 適用하면 된다.

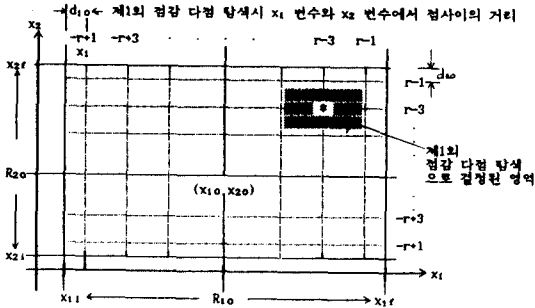


그림 3. 2차원 목적함수의 점감 다점( $P^N$ ) 탐색법  
Fig. 3. Decrease multi( $P^N$ )-point search method of 2 dimensional object function.

$N$ 차원의 목적함수인 경우에도 2차원 목적함수의 경우를 유추 적용하여 탐색을 실행한다. 이때 초기에 설정하는 1개 변수에 대한 탐색점의 수를  $P_0$ 라 하고, 탐색구간을  $R_{00} = x_{1r} - x_{1f}$ 로 하여, 다점 탐색의 1회 탐색시에 필요한  $F(X)$ 의 계산 횟수  $C_1$ 은 식(3.2)로 계산된다.

$$\begin{aligned} C_1 &= P_0^N + (P_0 - 2)^N + (P_0 - 4)^N + \dots + 3^N + 1^N \\ &= \sum_{i=0}^{(P_0-1)/2} (P_0 - 2i)^N \end{aligned} \quad (3.2)$$

그리고, 최소 탐색 구간  $d_m$ 은 식 (3.3)과 같이 되어 탐색의 정밀도가 계산된다.

$$d_m = \frac{R_0}{P_0 \cdot (P_0 - 2)(P_0 - 4) \dots 3 \cdot 1}$$

이므로,  $P_0 = 2 \cdot P + 1$ 로 분모를 정리하면,

$$d_m = R_0 \cdot 2^{P-1} \cdot P! / P_0! \quad (3.3)$$

로 된다. 여기서,  $P_0$ 는 초기 계산점의 수로 기수이다.

### IV. 多點 探索 學習 알고리즘

전술한 다점 탐색법은 다차원의 목적함수에서 절대 최소치를 탐색하는 방법으로는 적합하나,  $\partial F(x_i) / \partial x_i$ 가 각 변수에 따라 큰 차이를 갖거나  $F(X)$ 에 있는 각 항에서 가중함수의 크기가 큰 차이를 갖는 경우에는 탐색이 불가능한 경우가 있다.

예로써, Rosenbrock의 시험함수인 식(4.1)과 같은

$$F_R = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (4.1)$$

경우에  $x_2$ 과  $x_1$ 의 차수에서 2배의 차이가 나고, 계수의 값이 100배의 차이가 있으므로 3장의 점감 다점 탐색법만을 적용할 때에는 표 1의 결과와 같이 그 탐색구간의 조건에 따라서 절대 최소치에 도달하지 못하는 경우가 발생한다.

표 1은 초기 조건에서 점감 다점 탐색의 반복 횟수는 10으로 고정하고, 각 변수에 대한 탐색점의 수  $P_0$ 는 5가지 경우로 하였으며, 탐색구간  $R_{00} = x_{1r} - x_{0f}$ 를 4가지로 변경한 경우의 결과이다.

이같은 경우에는 3장의 점감 다점 탐색법에서 탐색한  $F^*(X^*)$ 에서  $F^*$ 가 최소치는 아니지만 중앙점에서 목적함수 계산값  $F_0$  보다 최소치에 근접하여 있으므로  $F^*$ 와  $X^*$ 를 기억시키는 학습 활동을 한다. 그리고, 학습한 내용을 참고로 탐색구간을 새로이 설정하고, 설정한 탐색구간 내에서 점감 다점 탐색을 실시한다.

점감 다점 탐색시에 탐색점 사이의 거리가 0으로 수렴할 때에 정밀도가 향상되므로 식(4.2)와 같은 계산식을 사용하여 학습의 반복 횟수  $rr$ 를 산출하였다.

$$rr = \sqrt{P_0 + 1} \quad (4.2)$$

표 1. 점감 다점 탐색법의 계산 결과

Table. 1. The result of decrease multipoint search.

탐색구간	결과	15 점	20 점	25 점	30 점	35 점
$-10 \leq X \leq 10$	F	0.0156	0.7892	0.0334	0.0317	0.0399
	$x_1$	1.1248	0.1116	1.1827	0.8222	1.1996
	$x_2$	1.2645	0.0125	1.3996	0.6749	1.4399
$-5 \leq X \leq 10$	F	0.2978	0.2417	0.1022	0.0121	0.1350
	$x_1$	1.5456	1.4914	0.6812	1.1101	0.6326
	$x_2$	2.3898	2.2258	0.4617	1.2327	0.4001
$-1 \leq X \leq 5$	F	2.8557	0.0506	0.2171	0.0389	1.7416
	$x_1$	-0.6897	0.7756	0.5350	0.8027	2.3194
	$x_2$	0.4779	0.6001	0.2832	0.6440	5.3824
$-0 \leq X \leq 50$	F	0.4825	0.0761	0.0030	0.0000	0.0000
	$x_1$	0.3055	0.7247	1.0549	1.0012	1.0005
	$x_2$	0.0924	0.5233	1.1131	1.0025	1.0010

반복 학습 계산의 횟수를 점차 k로 하고, k회 반복 시의 탐색구간  $d_k$ 은

$$d_k = \frac{R_0 \cdot 3^{k-1} \times (k-1)!}{P_0^{k-1}} \times d_{k-1}$$

$$= \frac{3^{k-1} \times (k-1)!}{P_0^{k-1}} \times \frac{2^{P_1} \times P_1!}{P_0!} \times \dots \quad (4.3)$$

으로 확장된다. 여기서  $R_0 = x_f - x_i$ ,  $P_0 = 2 \cdot P_1 + 1$  이다.

학습을 반복할 때마다 탐색구간은 k-1 회의 탐색구간에서  $d_k$ 을 감산하여 k회 탐색구간의 시작점으로, 가산하여 종료점으로 하는 새로운 탐색구간을 설정

하였다. 새로이 설정된 탐색구간 내에서 점감 탐색을 계속 실행한다. 이것을 1차원 목적함수의 경우에 적용하면 그림 4의 사선 부분이다. 그리고, 새로운 탐색구간 내에서 점감 다점 탐색을 위하여 준비되는 초기 탐색점 사이의 거리는  $d_1$ 이다. 목적함수가 2차원 이상인 다차원의 경우에는 이를 확장하여 유추 적용하면 된다.

또한, 학습을 k회 반복한 후 목적함수의 계산 횟수는

$$C_k = k \times \sum_{i=0}^{(P_0-1)/2} (P_0 - 2 \cdot i) \quad (4.5)$$

로 계산 된다.

### V. 본 探索法의 性能

제 3 장과 4장에서 기술한 탐색법은 그림 5에서 그림 8까지의 흐름도로 구성하고, 그 성능을 검토하였다.

흐름도는 그림 5의 내용이 주 프로그램이며, 제 3 장에서 기술한 점감 다점 탐색과정의 흐름도는 그림 6이고, 제 4 장에서 설명한 학습 알고리즘의 흐름도는 그림 7이며, 제 2 장에서 설명한 일변수 변환법 및 목적함수를 계산하여 최소치를 탐색하는 흐름도는 그림 8이다.

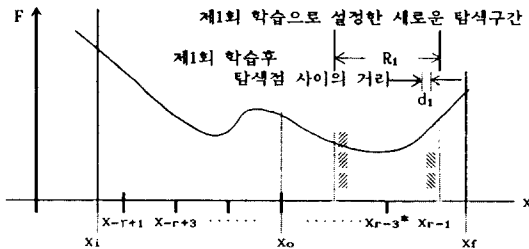


그림 4. 1차원 목적함수의 점감 다점(PN) 탐색과 그 학습 알고리즘

Fig. 4. Decrease multi(PN)-point search and its learning algorithm of 1 dimensional object function.

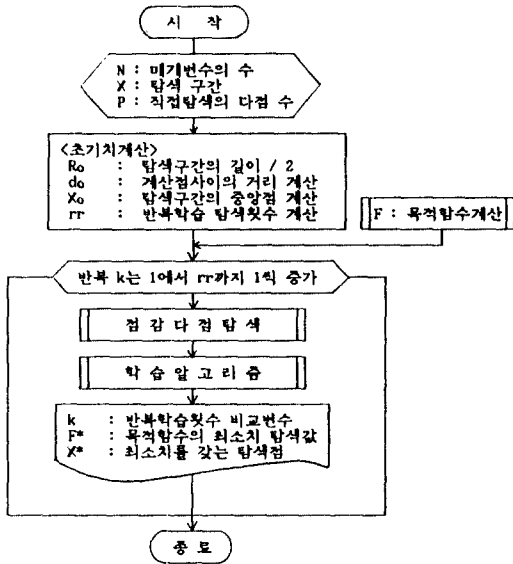


그림 5. 제안 알고리즘의 주 프로그램 흐름도  
Fig. 5. Main flowchart of the proposed algorithm.

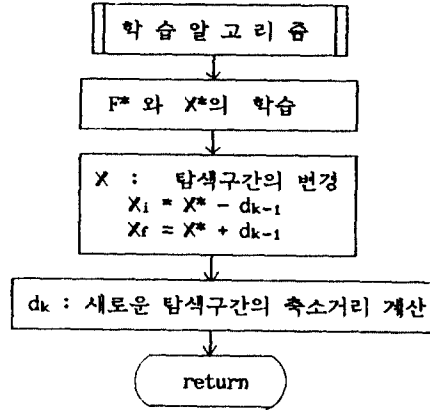


그림 7. 학습 알고리즘  
Fig. 7. Algorithm of learning method.

또한, 제한 탐색 알고리즘의 성능을 검토하기 위하여 어려운 함수로 알려져 있는 Rosenbrock 함수<sup>[27]</sup>를 시험함수로 채택하였다.

$$F_R = 100 \cdot (x_2 - x_1)^2 + (1 - x_1)^2 \quad (5.1)$$

표 2는 본 논문에서 제안한 알고리즘의 결과로 탐색 구간 및 점감 다점 탐색 횟수는 표 1의 경우와 동일하게 하였고, 학습 반복 횟수는 식 (4.2)에 의해 계산하였다. 점감 다점 탐색과 그 학습 알고리즘을 실행한 경우에 (1.0000, 1.0000)에서  $F^*(X^*)$ 의 값이 0인 최소치로 정확하게 산출한 학습 횟수를 표시하였다.

표 3은 Rosenbrock 함수에서 각 변수의 탐색구간을 다르게 설정하여 최소치가 출력된 반복학습의 탐색 횟수를 표시하였다.

표 4는 식(5-1)의 Rosenbrock 함수에 대한 최소치

표 2. 제안 알고리즘에 의한 최적치 탐색횟수  
Table 2. The search number for optimum value of proposed algorithm.

탐색구간	15 점	20 점	25 점	30 점	35 점
$-10 \leq X \leq 10$	4	4	3	2	1
$-5 \leq X \leq 10$	3	3	7	4	2
$-1 \leq X \leq 5$	4	3	1	1	31
$0 \leq X \leq 50$	5	1	1	1	1

그림 6. 점감 다점 탐색 알고리즘  
Fig. 6. Algorithm of decrease multipoint search.

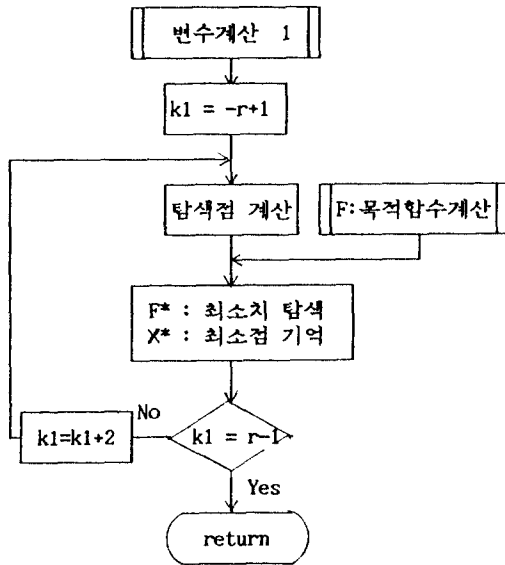
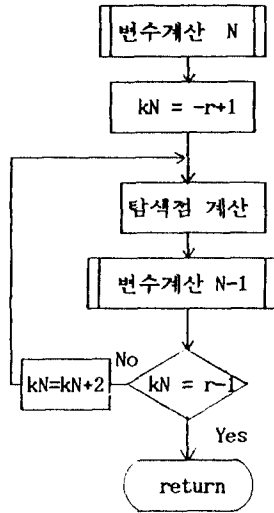


그림 8. 일변수 변환법과 직접탐색법의 흐름도  
Fig. 8. Flowchart of univariate and direct search method.

를 여러가지 탐색구간에서 본 제안 알고리즘과 golden section 알고리즘을 2차원으로 확장하여 최소치를 구한 결과이다. 이 표에서와 같이 golden section 알고리즘에서는 탐색구간의 변화에 따라 다른 값이 출력되었으나 본 제안 알고리즘에서는 ASA<sup>[8]</sup>와 같이 정확한 값이 출력되었음을 알 수 있다.

표 3. 각 변수의 탐색구간을 변경한 경우의 최적치 탐색 횟수

Table 3. The search number for optimum value with difference of search region on each variable.

$x_1$ 탐색구간	$x_2$ 탐색구간	15 점	20 점	25 점	30 점	35 점
$-5 \leq x_1 \leq 10$	$0 \leq x_2 \leq 50$	4	2	6	3	2
$0 \leq x_1 \leq 50$	$-10 \leq x_2 \leq 10$	6	3	4	3	2
$-2.5 \leq x_1 \leq 18.5$	$0 \leq x_2 \leq 5.5$	7	8	4	7	3
$0 \leq x_1 \leq 30$	$0 \leq x_2 \leq 70$	5	2	4	2	2
$0 \leq x_1 \leq 5$	$0 \leq x_2 \leq 3$	3	5	2	4	2

표 4. 식(5.1)을 사용한 제안 알고리즘과 golden section 알고리즘과의 비교

Table 4. The comparison for proposed and golden section algorithm using eq. (5.1).

탐색구간	본 알고리즘			golden section 알고리즘		
	$x_1$	$x_2$	$F^*$	$x_1$	$x_2$	$F^*$
$-1 \leq X \leq 5$	1.0000	1.0000	0.0000	1.9351	3.7496	0.8767
$0 \leq x_1 \leq 5$ $0 \leq x_2 \leq 3$	1.0000	1.0000	0.0000	0.9711	0.9426	0.0009
$-1.5 \leq x_1 \leq 3$ $-0.5 \leq x_2 \leq 4$	1.0000	1.0000	0.0000	0.7249	0.5367	0.0883
$-1 \leq x_1 \leq 4$ $0 \leq x_2 \leq 5$	1.0000	1.0000	0.0000	0.6502	0.4015	0.1678

그리고, 식(5.2)와 같은 Rao의 저서<sup>[3]</sup>의 시험 함수와 식(5.3)의 Powell FP, 식(5.3)의 Colville Fc와 같은 함수들에 대한 최소치를 구하고, 그들의 결과치와 비교하여 정확함을 확인하였다.

$$F_H = x_1 - x_2 + 2 \cdot x_1^2 + 2 \cdot x_1 \cdot x_2 + x_2^2 \quad (5.2)$$

$$F_P = (x_1 + 10 \cdot x_2)^2 + 5 \cdot (x_3 - x_4)^2 + (x_2 - 2 \cdot x_3)^4 + (10 \cdot x_1 - x_4)^4 \quad (5.3)$$

$$F_C = 100 \cdot (x_1^2 - x_2)^2 + (1 - x_1)^2 + 90 \cdot (x_4 - x_3^2)^2 + (1 - x_3)^2 + 10 \cdot 1 \cdot ((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8 \cdot (x_2 - 1) \cdot (x_4 - 1) \quad (5.4)$$

표 5는 식 (5.2)에서 탐색구간을  $-10 \leq X \leq 10$ 으로

고정하고 본 논문에서 제안한 알고리즘을 적용하였으며, 또한 직접 탐색법인 random walk, univariate, Powell 및 Hooke와 Jeeves 알고리즘들의 결과<sup>[3]</sup>와 비교하였다. 이들 알고리즘 중에서 random walk는 100회의 탐색에서도 정확한 값이 산출되지 않았다. 그리고, univariate, Hooke와 Jeeves 및 Powell에서는 탐색의 시작점을 (0.0000, 0.0000)으로 하고, 탐색점을 결정하기 위한 이동거리(step size)를 계산식을 이용하든지, 또는 일정한 값으로 변화시키는 방법을 채택하였으며, 탐색의 종료율 위하여 오차  $\epsilon$ 의 범위를 0.1로 설정하고 탐색을 수행한 결과이다. 이들 탐색 알고리즘은 허용오차의 범위에 따라 최적치의 정밀도가 결정되는 특징을 갖고 있으나 본 논문에서 제안한 알고리즘은 일정한 탐색횟수만큼 수행하여야 탐색을 종료하는 특징이 있다.

식 (5.3)의 Powell함수는 (0,0,0,0)에서 F\*의 값이 0으로, 식 (5.4)의 Colville함수는 (1,1,1,1)에서 F\*의 값이 0으로 되어 ASA의 결과<sup>[8]</sup>와 동일하였다. 그리고, 여러 참고문헌에서 시험함수로 사용하였거나, 실제 문제에서 설정되는 11가지 종류의 목적함수들에 대한 최소치와 최소점을 구하여 결과를 문헌의 내용과 비교하여 검토하였고, 적용한 시험함수의 종류는 부록 1에서 제시하였다.

표 6에서는 3<sup>N</sup>점 탐색 알고리즘<sup>[1]</sup>과 본 논문에서 제안한 알고리즘의 수행 시간과 함수의 계산 횟수를 비교하였다. 적용한 함수는 변수가 1개로 구성되어 있는 경우에서부터 4개로 구성되어 있는 경우로 하였으며, 각 변수의 차수가 2차로 한정되어 있는 경우와 4차 및 다른 여러 차수들이 혼합되어 있는 것을 4차 함수로 선택하였다. 그리고, 각 변수의 탐색구간을 동일하게 하였으며, 구간 내에서 초기에 탐색을 위한 계산점의 갯수를 30으로 설정하였다.

표 4와 표 5에서와 같이 본 논문에서 제안한 점감 다점 탐색법과 그 학습 알고리즘의 결과가 고정밀도를 유지하고 있었으며, 탐색이 진행될 수록 결과치가 최소치에 근접하고 있어 신뢰성을 갖고 있으며, 탐색구간을 이탈하는 경우가 없으므로 안정성이 확인되었다. 표 6에서와 같이 여러 종류의 함수에도 적용이 가능하므로 유용성이 있었으나, 최소치 산출을 위한 함수의 계산 횟수가 크며, 많은 수행 시간을 필요로 하고 있다.

표 5. 식(5.2)의 최소치 탐색 결과 비교표

Table. 5. The comparison for optimum value of search using eq.(5.2).

탐색 횟수	본 알고리즘	random walk	univariate	Hooke & Jeeves	Powell
1	0.3537	-1.1288	0.0000	0.0000	0.0000
2	-1.2487	-1.2136	-0.1250	-0.5700	-0.2500
3	-1.2499	-1.2207	-0.6875	-1.2100	-1.0000
4	-1.2500	-1.2415	-0.9687	-1.2350	-1.0050
5		-1.2422	-1.0746	-1.2500	-1.2500

표 6. 3<sup>N</sup>점과 제안 알고리즘과의 비교

Table. 6. The comparison of 3<sup>N</sup> point and proposed algorithm.

변수의 갯 수	구 분	3 <sup>N</sup> 점		본 알고리즘	
		2 차 함수	4 차 함수	2 차 함수	4 차 함수
1	시 간	0.10"	0.16"	0.87"	0.93"
	계산횟수	271	271	810	810
2	시 간	2S41"	2S74"	13S84"	16S36"
	계산횟수	7371	7371	22110	22920
3	시 간	1M13S32"	1M28S81"	6M54S96"	8M39S64"
	계산횟수	203311	203311	609980	632850
4	시 간	40M16S17"	1H2M17S46"	3H48M1S87"	6H9M3S70"
	계산횟수	5665419	5665419	16996254	17629104



### VI. 結 論

목적함수의 최소치 계산법으로 3점 탐색을 확장한 漸減 多點 探索과 그 學習 알고리즘을 제안하고, 어려운 문제로 알려져 있는 Rosenbrock, Powell, Colville 함수 및 Rao의 예제함수 등을 비롯한 15가지 함수의 최소치를 탐색하여 성능 및 유용성과 적용 가능성, 안정성 등을 검토한 결과는 다음과 같다.

- (1) Rosenbrock을 위시한 15가지의 시험함수에서 실패없이 절대 최소치를 탐색하였다.
- (2) 계산의 정밀도는 탐색구간 내에서 수공간의  $1 \times 10^{-6}$ 까지도 가능한 것으로 확인되었다.
- (3) 탐색구간을 이탈하는 경우가 없으므로 안정성이 확인되었다.
- (4) Powell의 시험함수와 같이 탐색변수가 4개이고, 변수의 차수가 1부터 4까지 서로 다른 경우에는 목적함수의 계산횟수가 방대하여 대단히 많은 탐색시간이 소요되었다.

본 논문에서 제안한 漸減 多點 探索과 그 學習 알고리즘은 이상과 같은 특징을 갖고 있어 최적 설계에서와 같이 계산 시간에 크게 제약 받지 않으며, 정밀한 최소치가 요구되는 경우에는 적용이 가능한 알고리즘으로 사료된다.

그러나, 변수의 갯수가 4개 이상으로 되고, 차수가 높은 경우에는 많은 처리 시간이 요구되므로 최적 제어 등과 같이 실시간 처리의 문제에는 적용이 곤란하게 된다. 그러므로 실시간 처리가 필요한 경우에도 적용이 가능한 알고리즘의 계속적인 연구가 필요하다.

부록 1: 본 제안 알고리즘에서 목적함수로 이용한 시험함수의 종류

1. Rosenbrock : 참고문헌 [2],[3],[5],[6],[8]

$$F_1 = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2$$

2. Powell : 참고문헌 [2],[8]

$$F_2 = (x_1 + 10 \cdot x_2)^2 + 5 \cdot (x_3 - x_4)^2 + (x_2 - 2 \cdot x_3)^4 + (10 \cdot x_1 - x_4)^4$$

3. Colville : 참고문헌 [2],[8]

$$F_3 = 100 \cdot (x_1^2 + x_2)^2 + (1 - x_1)^2 + 90 \cdot (x_4 - x_3^2)^2 + (1 - x_3)^2 + 10 \cdot ((x_2 - 1)^2 + (x_4 - 1)^2)$$

$$+ 19.8 \cdot (x_2 - 1) \cdot (x_4 - 1)$$

4. Dichotomous, golden : 참고문헌 [2],[3],[6]

$$F_4 = x_1 \cdot (x_1 - 1.5)$$

5. Rao의 예제함수 : 참고문헌 [3]

$$F_5 = x_1 - x_2 + 2 \cdot x_1^2 + 2 \cdot x_1 \cdot x_2 + x_2^2$$

6. Numerical Recipes : 참고문헌 [4]

$$F_6 = |(-10 \cdot x_1 \cdot x_1)^2 + 5 \cdot x_2 \cdot x_2 - 3 \cdot x_1 \cdot x_2 + 4 \cdot x_1 - 2 \cdot x_2 + 500|$$

$$F_7 = 3 \cdot x_1^2 + 2 \cdot x_1 \cdot x_2 + 2 \cdot x_2^2 + 7$$

$$F_8 = (x_1 - 0.5)^2 + (x_2 + 3)^2 + (x_3 + 0.2)^2$$

7. Numerical Recipes and Optimum Design : 참고문헌 [2],[4]

$$F_9 = (x_1 - 1.5)^2 + (x_2 - 1.5)^2$$

8. Optimum Design : 참고문헌 [2]

$$F_{10} = x_1^2 + 2 \cdot x_2^2 + 2 \cdot x_3^2 + 2 \cdot x_1 \cdot x_2 + 2 \cdot x_2 \cdot x_3$$

9. Introduction to Optimization Theory : 참고문헌 [7]

$$F_{11} = (x_1 - 0.1136 \cdot (x_1 + 3 \cdot x_2) \cdot (1 - x_1)^2)^2 + (x_2 + 7.5 \cdot (2 \cdot x_1 - x_2) \cdot (1 - x_2))^2$$

10. Foundations of Optimization : 참고문헌 [5]

$$F_{12} = x_1^2 + 2 \cdot x_2^2 - 4 \cdot x_1 - 2 \cdot x_1 \cdot x_2$$

$$F_{13} = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2$$

$$F_{14} = (x_3^2 - x_2)^2 + (x_2 - x_1^2)^2 + (x_1 - 1)^2$$

$$F_{15} = (x_1^2 - 1.5)^2$$

### 참 고 문 헌

1. 金周弘, 孔徽植, "最適 設計을 위한 3점 探索 알고리즘의 제안," 한국통신학회 논문지 제 16권 7호, pp.640-650.
2. Jasbir S. Arora, "Introduction to Optimum Design," McGraw-Hill Book Company, 1989.
3. S. S. Rao, "Optimization theory and applications," Halsted Press, 1984.
4. William H. Press외 3인, "Numerical Recipes," Cambridge(Pang-Han), pp.166-176, pp.274~334, 1986.
5. Charles S. Beightler 외 2인, "Foundations of Optimization," Prentice-Hall(탑출판사), 2'nd ed., 1982.
6. Lawrence Hasdorff, "Gradient Optimization and Nonlinear Control," A Wiley-interscience Pub. pp.39-85, 1976.

7. Byron S.Gottfried, "Introduction to Optimization Theory," 탑출판사, pp.26-293, 1973.
8. K.K. Burhardt, "An Adaptive Search Optimization Algorithm," IEEE, Trans. on Computer, Vol. c-23, No.9, pp.890-897, Sep. 1974.
9. I.Zang and M. Avriel, "On Functions Whose Local Minima Are Global," JOTA : Vol.16, NOS.3/4, pp.183-190, Aug. 1975.
10. J.A. Snyman and L.P. Fatti, "A Multi-Start Global Minimization Algorithm with Dynamic Search Trajectories," JOTA, Vol. 54, No.1, pp.121-141, July 1987.
11. R.S. Anderssen and P. Bloomfield, "Properties of the Random Search in Global Optimization," JOTA : Vol. 16, Nos. 5/6, pp.383-398, Sep. 1975.
12. L. Grippo, F.Lampariello and S. Lucidi, "Global Convergence and Stabilization of Unconstrained Minimization Methods without Derivatives," JOTA : Vol. 56, No. 3, pp. 385-406, Mar. 1988.
13. R.P. Ge and Y.F. Qin, "A Class of Filled Functions for Finding Global Minimization of a function of Several Variables," JOTA : Vol. 54, No. 2, pp.241-252, Aug. 1987.



金 周 弘(Joo Hong Kim) 正會員  
1929年 1月 5日生  
1952年 : 서울 大學校 工科大學 卒業  
現在 : 東國大學校 電子工學科 教授  
(工博)



孔 徽 植(Whue Sik Kong) 正會員  
1954年 1月 26日生  
1980年 2月 : 東國大學校 電子工學  
科 卒業  
1985年 2月 : 漢陽大學校 產業大學  
院 電子工學 卒業  
1991年 2月 : 東國大學校 大學院 博  
士課程 修了

1985年 3月 ~ 現在 : 東宇專門大學 電子計算科 助教授



李 洸 穉(Kwang Jeek Lee) 正會員  
1949年 2月 11日生  
1971年 2月 : 東國大學校 電子工學  
科 卒業  
1980年 2月 : 延世大學校 產業大學  
院 卒業  
1991年 2月 : 東國大學校 大學院 博  
士課程 修了

1986年 6月 ~ 現在 : 서울 產業大學 媒體工學科 助教授