차분진화 알고리듬을 이용한 전역최적화

정 재 준* · 이 태 희†

(2002 년 3 월 30 일 접수, 2003 년 7 월 29 일 심사완료)

Global Optimization Using Differential Evolution Algorithm

Jae Joon Jung and Tae Hee Lee

Key Words : Differential Evolution Algorithm(차분진화 알고리듬), Global Optimization(전역최적화), Genetic Algorithm(유전 알고리듬), Crossover Process(교배과정), Mutation Process(변이과정)

Abstract

Differential evolution (DE) algorithm is presented and applied to global optimization in this research. DE suggested initially for the solution to Chebychev polynomial fitting problem is similar to genetic algorithm (GA) including crossover, mutation and selection process. However, differential evolution algorithm is simpler than GA because it uses a vector concept in populating process. And DE turns out to be converged faster than GA, since it employs the difference information as pseudo-sensitivity. In this paper, a trial vector and its control parameters of DE are examined and unconstrained optimization problems of highly nonlinear multimodal functions are demonstrated. To illustrate the efficiency of DE, convergence rates and robustness of global optimization algorithms are compared with those of simple GA.

기호설명

$\mathbf{x}_{j,G}$	: 파라미터 벡터(parameter vector)
$\mathbf{v}_{i,G+1}$	: 시행벡터(trial vector)
ϕ	: 변이상수(mutation constant)
CR	: 교배확률(crossover probability)

1. 서 론

차분진화 알고리듬(differential evolution algorithm: DE)은 Price 와 Storn 에 의해 벡터 차분(vector difference)을 사용하여 Chebychev 다항곡선의 내삽문제(polynomial fitting problem)를 해결하는 과정에서 개발되었다.[1,2] 차분진화 알고리듬은 현재 널리 사용되고 있는 통계적 임의 탐색법인 유전 알고리듬(genetic algorithm : GA)과 유사한 알고리듬이다. 유전 알고리듬과 차분진화 알고리듬은 초기 개체군을 이루는 개체를 샘플링하고, 이들의 교배(crossover), 변이(mutation), 선택(selection) 과정을 거쳐 적합도(fitness)가 개선되는 새로운 개체들을 추출해 낸다는 공통점이 있다. 그러나 이런 연산과정에서 GA 알고리듬이 개체들의 표현형(phenotype)을 유전형(genotype)으로 바꾸는 코딩(coding) 과정이 필요한 반면, 차분진화 알고리듬은 개체를 벡터로 표현하기 때문에 코딩과정이 필요 없으며 이들의 산술적 연산을 통해 새로운 개체를 생성한다는 차이점이 있다. 따라서 차분진화 알고리듬은 유전 알고리듬보다 쉽게 구현할 수 있을 뿐 아니라 적은 수의 제어인자(control parameter)로 최적화 과정을 조절할 수 있다는 특징이 있다.

현재 유전 알고리듬은 수학적 모델링이 어렵거나 민감도 정보를 사용할 수 없는 비선형적, 비오목적(nonconvexity) 최적설계문제에 많이 적용되고 있다. 그러나 민감도 정보를 사용하지 않고 확률에 의한 샘플링 과정은 필연적으로 급격한 계산비용의 증가로 나타나 유전 알고리듬의 단점

* 한양대학교 대학원 기계설계학과
† 책임저자, 회원, 한양대학교 기계공학부
Email : thlee@hanyang.ac.kr
TEL : (02) 2290-0449

으로 지적되고 있다.

본 논문에서는 이러한 문제점에 대한 개선 방안으로 차분진화 알고리듬을 제안한다. 차분진화 알고리듬을 이용하여 비선형 함수와 최적화 검정 함수들에 대해 전역최적화(global optimization)를 수행하고, 그 결과의 정확성과 수렴성을 비교하여 차분진화 알고리듬이 전역최적화에 적합한 방법임을 보이고자 한다.

2. 차분진화 알고리듬을 이용한 전역최적화

차분진화 알고리듬에서는 초기 개체군을 이루는 모든 개체들을 벡터로 표현하는데, 이들을 파라미터 벡터라고 한다.[1,2]

$$\mathbf{x}_{j,G} \in R^D \quad j = 1,2,\cdots,NP \tag{1}$$

여기서 NP 는 파라미터 벡터의 개수이고, 이 값은 최적화 과정에서 일정하며, 설계자가 문제에 따라 정의할 수 있는 샘플링 개수이다. G는 파라미터 벡터가 몇 번째 세대인가를 표현하며 알고리듬에서는 최적화 횟수(iteration)를 의미한다. 파라미터 벡터의 차원은 목적함수를 이루는 설계변수의 개수 D 와 같으며, 초기 파라미터 벡터는 가용영역(feasible region)에서 임의로 선택하되, 일반적으로 균등확률분포를 따르도록 한다.

차분진화 알고리듬은 두 개의 파라미터 벡터의 가중차분(weighted difference)에 임의의 세 번째 벡터를 더함으로써 새로운 파라미터 벡터를 생성한다. 이 벡터를 시행벡터(trial vector)라 하며, 이것을 정의하는 방법에 따라 차분진화 알고리듬의 다양한 형태가 있을 수 있다. 본 연구에서는 아래와 같은 시행벡터를 사용하였다.

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{best,G} + \phi\,(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) \tag{2}$$

여기서 벡터 $\mathbf{x}_{r1,G}$, $\mathbf{x}_{r2,G}$ 는 G 번째 세대에서 선택된 임의의 파라미터 벡터이고, $\mathbf{x}_{best,G}$ 는 G 세대에서 가장 작은 목적함수값을 갖는 파라미터 벡터이다. 가중치 ϕ 는 벡터 차분 $(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G})$ 의 크기를 결정하는 변이 제어상수(mutation control constant)로 0 과 1 사이의 값으로 최적화 문제에 따라 다르게 정의한다.

변이 제어상수는 $\mathbf{x}_{best,G}$ 에 $\phi\,(\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G})$ 만큼의 변동량을 주어 시행벡터가 설계 영역 내에서 보다 다양한 값을 갖게 하여 해가 국소최소값(local minimum)으로 수렴하는 것을 방지하는 역할을 한다.

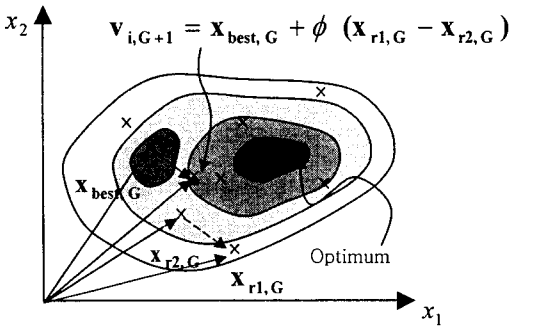

Fig. 1 An example of a two-dimensional objective function showing relation between parameter vector and trial vector

Fig. 1 은 목적함수가 설계변수 (x_1, x_2) 의 함수일 때, 파라미터 벡터 $\mathbf{x}_{r1,G}$, $\mathbf{x}_{r2,G}$, $\mathbf{x}_{best,G}$ 를 이용하여 시행벡터를 생성한 결과를 보여주는 그림이다. Fig. 1 에서 폐곡선은 동일한 목적함수값을 갖는 점들을 연결한 선이고, × 로 표시된 점이 임의로 분포하고 있는 초기 파라미터 벡터의 위치를 나타낸다.

차분진화 알고리듬은 새로운 파라미터 벡터의 생성을 위해 교배과정을 거친다. 교배과정은 부모의 DNA 를 자식이 물려받는 자연현상을 모방한 것으로, 차분진화 알고리듬에서는 다음과 같은 벡터를 사용하여 교배효과를 얻는다.

$$\mathbf{u}_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \ldots, u_{Di,G+1}) \tag{3}$$

이때 벡터 $\mathbf{u}_{i,G+1}$ 의 성분 $u_{ji,G+1}$ 는 식(4)와 같이 정의한다.

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{for} \quad \text{rand}\,() < CR \\ x_{ji,G} & \text{for} \quad \text{others} \end{cases} \tag{4}$$

여기서 함수 rand ( ) 는 [0,1] 사이의 무작위수(random number)이며, CR 은 교배확률을 결정하는 상수로 0 에서 1 사이의 값으로 한다. 예를 들어, 교배확률 CR=1 로 설정할 경우는 앞에서 생성한 시행벡터의 모든 성분이 $\mathbf{u}_{i,G+1}$ 을 구성하며, CR=0.5 일 경우는 시행벡터와 그 전세대의 파라미터 벡터의 성분이 동등한 확률로 $\mathbf{u}_{i,G+1}$ 을 구성함을 나타낸다.

차분진화 알고리듬은 G 번째 세대의 파라미터 벡터와 이들 벡터로 만든 시행벡터 $\mathbf{u}_{i,G+1}$ 의 상호 경쟁을 통해 전역최적해를 구한다. 시행벡터는 초기 파라미터 벡터의 개수 NP 만큼 생성하

는데, 구한 시행벡터가 다음 세대의 파라미터 벡터가 되기 위해서는 비교대상이 되는 파라미터 벡터의 목적함수값보다 작은 목적함수값을 가져야 한다. 이 조건을 만족하는 시행벡터는 G 번째 세대의 파라미터 벡터를 대체하여, (G+1) 번째 세대의 새로운 파라미터 벡터가 된다. 만약, 이 조건을 만족하지 못하면 이 시행벡터를 버리고 이미 존재하고 있던 G 번째 세대의 파라미터 벡터가 (G+1) 번째 세대의 파라미터 벡터로 유지된다. 이러한 과정은 설계자가 정해준 세대횟수만큼 반복하며, 결국 목적함수값이 작은 파라미터 벡터들로 구성된 개체군이 만들어지게 된다. 이 벡터 중에 가장 작은 목적함수값을 갖는 파라미터 벡터를 추출해 내면 이 값이 바로 전역최적해가 된다.

3. 예 제

본 논문에서는 제시된 차분진화 알고리듬의 수렴성과 정확성을 유전 알고리듬[3]과 비교한다. 예제함수는 비선형 함수와 최적화 검정함수들로 채택하였으며, 각 함수마다 제어인자를 바꾸어가며 최적화를 수행하였다. DE 와 GA 모두 확률적인 방법에 근거한 알고리듬이므로 수렴속도 평가에서는 수렴하기까지의 평균 함수계산회수(number of function evaluation: NFE)를 그 기준으로 삼았다. 또한 해를 찾아주는 알고리듬의 강건성(robustness)을 비교하기 위해서 매 20 번의 최적화 수행 후의 수렴횟수(convergence number: CN)를 강건성의 기준으로 채택하였다. 수렴 기준은 엄밀해(exact solution)와의 오차가 0.01% 이하일 때로 하였다.

3.1 Haupt 함수의 전역최적화

Haupt 함수는 많은 다중국소점을 갖지만 전역최적해는 하나만 존재하는 함수로 다음과 같은 식으로 표현된다.[4]

$$f = x_1 \sin(4x_1) + 1.1x_2 \sin(2x_2) \qquad (5)$$

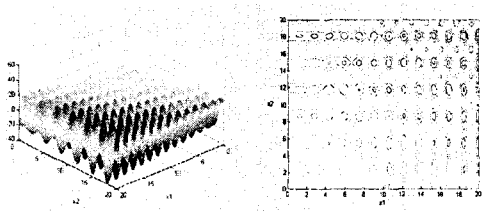

Fig. 2 Haupt function and its contour plot

Fig. 2 는 Haupt 함수와 목적함수값이 동일한 폐곡선을 나타내고, $(x_1, x_2) \in [0,20]$, 전역최적해는 $\mathbf{x}^* = (20.0, 18.077)$, 최소값은 $f(\mathbf{x}^*) = -39.7559$ 를 갖는다.

DE 에서 $\phi = 0.95$, 교배확률 CR=0.8, 파라미터 벡터의 개수(*NP*)는 각각 20, 40 으로 선택하였으며 최대 함수계산회수는 3000 으로 하였다. GA 에서는 변이확률 P_m=0.05, 균등교배 확률 P_c=0.5 로 선택하였으며, 토너먼트 경쟁방식을 채택하였다. Fig. 3 은 샘플링 개수가 20 개인 경우, DE 와 GA 의 세대에 따른 최소 목적함수값의 수렴 과정을 보여주는 그림이다.

Table 1 은 개체군 크기가 20 일 때 DE 와 GA 의 수렴 결과를 비교한 것이다. DE 가 GA 보다 1.66 배정도 함수계산횟수가 적었고, 알고리듬의 강건성을 나타내는 수렴횟수(CN) 면에서도 DE 가 100 퍼센트(%) 수렴성을 보인 반면, GA 는 45%의 수렴성만을 보였다.

Fig. 4 는 개체군의 크기가 40 일 때, 각 세대의 최소 목적함수값의 변화과정을 나타낸다. Fig. 4 에서는 GA 와 DE 의 수렴과정이 비슷한 양상을 나타내었고, Table 2 에서 GA 의 수렴성이 *NP* = 10 일 때보다 70%로 증가하였음을 볼 수 있다. 이것은 개체군의 크기가 커짐에 따라 초기 최적화 과정에서 최적해에 근접한 해를 얻을 확률이 증가되었기 때문이다. 하지만 Table 1 과 비

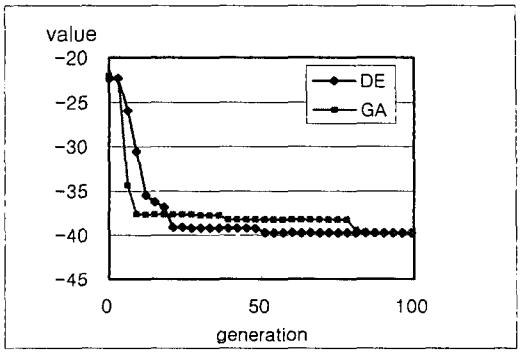

Fig. 3 Convergence history of minimum cost (NP=20)

Table 1 Number of function evaluation (NFE) and control parameters for Haupt function at NP=20

DE				GA			
NP	CR	ϕ	NFE	NP	P_c	P_m	NFE
20	0.8	0.95	1060	20	0.5	0.05	1760
CN: 20/20 (100%)				CN: 9/20 (45%)			

교해 두 알고리듬 모두 함수계산회수가 증가하였는데 이것는 매 세대마다 효과적이지 못한 샘플링 횟수가 보다 많이 포함됨으로써 전체 함수 계산 횟수가 증가 되었기 때문이다. 따라서 단순히 개체군의 개수를 늘리는 것이 수렴속도를 증가시키는 것은 아니며 적절한 개체군 선정이 수렴속도에 중요한 인자임을 알 수 있다.

3.2 Rosenbrock 함수의 전역최적화

Rosenbrock 함수는 전체 설계영역에서 함수값의 변화가 급격하고 최적해 주위라 하더라도 $y=x^2$ 의 관계를 만족하지 않을 경우 계수의 영향으로 수렴이 잘 되지 않는 비선형 함수이다.[5]

$$f=100(x_2-x_1^2)^2+(1-x_1)^2 \tag{6}$$

여기서 두 알고리듬의 최대 함수계산 회수는 3000 으로 하였고 설계영역은 $(x_1,x_2)\in[0,5]$ 로 하였다. 최적해는 $\mathbf{x}^*=(1,1)$ 이고, 최소값은 $f(\mathbf{x}^*)=0.0$ 이다.

Fig. 6 은 $NP=10$ 일 때, DE 와 GA 의 수렴과정의 차이를 나타낸다. 즉, DE 가 초기부터 빠른 수렴속도를 보인 반면 GA 는 초기 수렴경향은 비슷하나 이후 계단형 수렴 특성을 보여주고 있다.

Table 3 에서 DE 가 함수계산 횟수가 훨씬 적은 것으로 나타났으며, 수렴의 강건성 면에서도 GA 가 25%로 수렴성을 보인 반면, DE 는 100%의 수렴성을 보였다.

Fig. 7 은 $NP=20$ 인 경우 각 세대의 최소 목적함수값의 수렴 과정을 나타낸다.

Fig. 8 은 Fig. 7 의 수렴과정 그래프 중에서 일부분만을 확대하여 표시한 것이다.

Fig. 8 에서 DE 가 시행벡터의 방향성을 이용하여 최적해 주위에서 정밀도를 높여 좋은 수렴성을 보인 반면 GA 는 확률에 근거한 교배와 변이 과정으로 인해 계단식 수렴과정을 반복하고 있음을 알 수 있다.

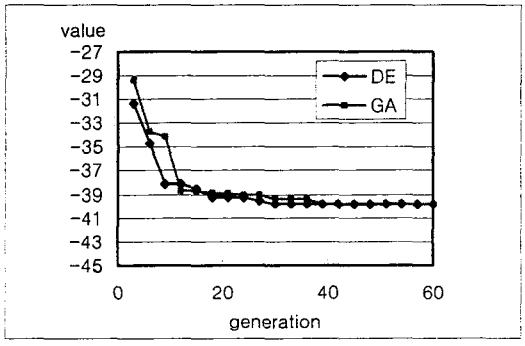

Fig. 4 Convergence history of minimum cost (NP=40)

Table 2 Number of function evaluation (NFE) and control parameters for Haupt function at NP=40

DE				GA			
NP	CR	ϕ	NFE	NP	P_c	P_m	NFE
40	0.8	0.95	1980	40	0.5	0.05	2440
CN: 20/20 (100%)				CN: 14/20 (70%)			

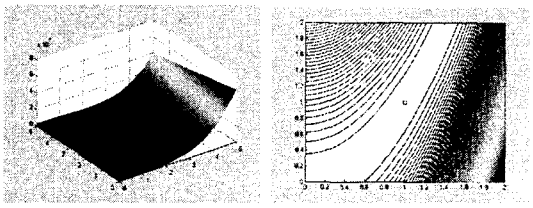

Fig. 5 Rosenbrock function and its contour plot at near minimum

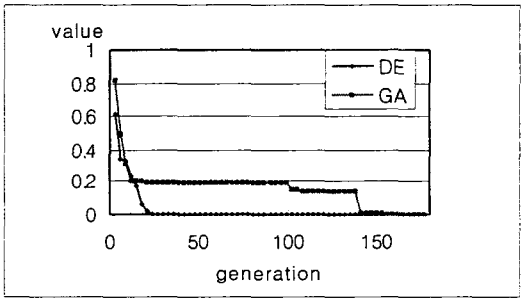

Fig. 6 Convergence history of minimum cost (NP=10)

Table 3 NFE and control parameters for Rosenbrock function

DE				GA			
NP	CR	ϕ	NFE	NP	P_c	P_m	NFE
10	0.8	0.98	480	10	0.5	0.03	1520
CN: 20/20 (100%)				CN: 5/20 (25%)			

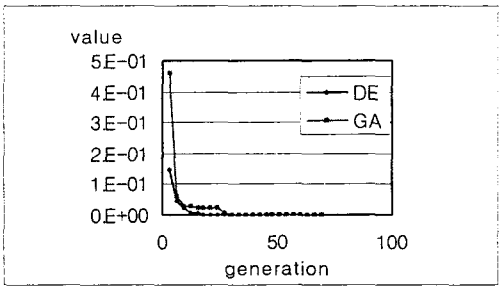

Fig. 7 Convergence history of minimum cost (NP=20)

Table 4 NFE and control parameters for Rosenbrock function (NP=20)

DE				GA			
NP	CR	ϕ	NFE	NP	P_c	P_m	NFE
20	0.8	0.98	620	20	0.5	0.03	1360
CN: 20/20 (100%)				CN: 7/20 (35%)			

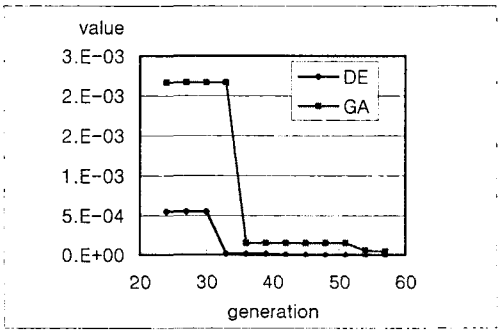

Fig. 8 Convergence history of minimum cost from 20th to 60th generation (NP=20)

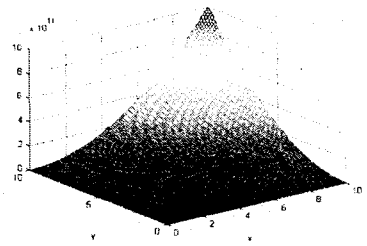

Fig. 9 Powell function

3.3 Powell 함수의 전역최적화

Powell 함수는 $\mathbf{x}^* = (1.098 \times 10^{-5}, 9.106)$ 를 최적해로 갖고 $f(\mathbf{x}^*) = 0.0$ 인 함수이다. Rosenbrock 함수와 같이 함수값의 변화가 급격할 뿐 아니라 최적해의 차수가 큰 차이를 보이는 경우이다.[6]

$$f = (10000x_1x_2 - 1)^2 + \{(e^{-x_1} + e^{-x_2}) - 1.0001\}^2 \quad (7)$$

아래에 Powell 함수와 3 차원 그래프를 나타내었다.

Fig. 10 에서 볼 수 있듯이 DE 의 수렴속도는 매우 빠르게 나타났으며, Fig. 11 은 Fig. 10 의 일부분을 확대하여 수렴과정을 나타낸 것이다

Table 5 에서 Powell 함수는 DE 만이 2226 번의 함수 계산만에 정확하게 전역 최적해로 수렴하였고, GA 는 여러 파라미터를 조정하였으나 모든 경우 수렴하지 않았다.

4. 결 론

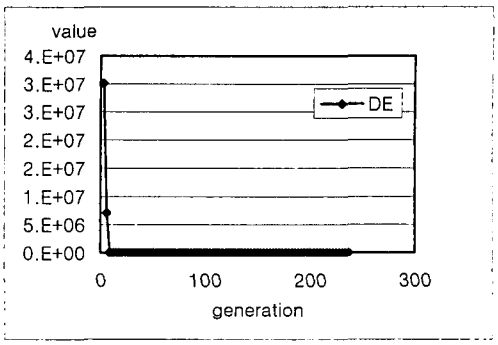

Fig. 10 Convergence history of minimum cost (NP=10)

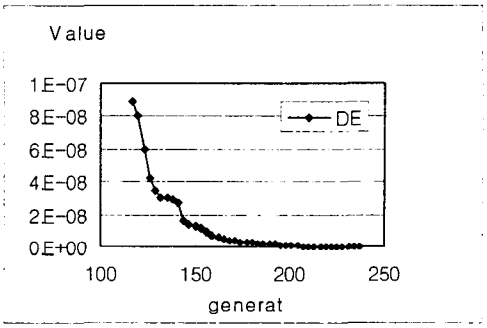

Fig. 11 Convergence history of minimum cost from 110th to 240th generations (NP=10)

Table 5 NFE and control parameters for Powell function

DE				GA			
NP	CR	ϕ	NFE	NP	P_c	P_m	NFE
10	0.9	0.98	2260	10	0.5	0.03	-
CN: 20/20 (100%)				CN: 0/20 (0%)			

본 논문에서는 전역최적화 알고리듬인 차분진화 알고리듬을 소개하고, 전역최적화 검정함수를 사용하여 제안한 알고리듬의 정확성과 수렴성을 검증하였으며, 그 결과를 유전알고리듬과 비교하여 다음과 같은 결론을 얻었다.

(1) 다중국소점을 갖는 Haupt 함수에 대해서는 DE 와 GA 모두 전역최적해로 정확하게 수렴하였으나, 수렴속도와 수렴의 강건성 면에서 DE 가 탁월하였다. 이것은 DE 가 매 세대 마다 임의의 두 벡터로 만들어지는 벡터 차분을 이용하여 그 세대에서 최소 목적함수값을 갖는 위치에서 좀 더 작은 목적함수값을 갖는 개체를 생성해 나가기 때문이다. 이때의 벡터차분은 샘플링에 있어서 방향성을 제공해 주기 때문에 확률적으로 최

적해를 탐색하는 GA 보다 뛰어난 수렴성을 제공하는 것으로 판단된다.

(2) 해의 정확성을 알아볼 수 있는 Rosenbrock 함수의 전역최적화에서도 DE 알고리듬이 GA 보다 정확한 수렴결과를 보였다. 이는 DE 가 충분한 최적화 이후 시행벡터의 변동량이 작아져서 가장 작은 목적함수 주위에서 집중적으로 샘플링을 하기 때문이다.

(3) 최적해가 미소값을 갖는 Powell 함수는 알고리듬의 정밀도에 따라 수렴여부가 결정되는 경우이다. 최적화 결과 DE 만이 최적해로 수렴하였다. 이것은 GA 의 경우 설계영역을 이진수로 맵핑하는 과정에서 정밀도가 결정되는데, 이때의 해가 너무 커서 해상도를 넘거나 차수가 너무 작으면 해의 구별이 어려워 최적화가 진행될 수 없다. 그러나 DE 는 실수 자체를 이용하여 교배와 변이 과정을 거치기 때문에 유효자리에만 영향을 받으므로 매우 작은 값을 갖는 해도 정확히 찾아낼 수 있었다.

후 기

이 연구는 한국과학재단 지정 최적설계신기술 연구센터의 지원에 의해 수행되었습니다.

참고문헌

(1) Price, K., 1997, "Differential evolution vs. the contest functions of the 2nd ICEO," *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation*, pp. 153~157.

(2) Storn, R., 1996, "On the usage of differential evolution for function optimization," *Proceedings of the 1996 North American Fuzzy Information Processing Society (NAFIPS)*, IEEE Press, pp. 519~523.

(3) Genetic Algorithms developed by David. L.Carrol, http://www.aic.nrl.navy.mil/galist/

(4) Haupt, R. L., S.E Haupt, 1998, *Practical Genetic Algorithms*, p. 6, John Wiley & Sons, Inc.

(5) Schittkowski, K., 1987, *More Test Examples for Nonlinear Programming Codes*, p. 32, Spinger-Verlag.

(6) Powell, M. J. D., 1970, "A hybrid method for nonlinear equations," *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz, Ed., Gordon & Breach, pp. 87~114.

후 기

이 연구는 한국과학재단 지정 최적설계신기술 연구센터의 지원에 의해 수행되었습니다.