

연속 최적화 문제에 대한 수렴성이 개선된 순차적 주밍 유전자 알고리즘

권영두[†], 권순범*, 구남서*, 진승보**
(2001년 9월 12일 접수, 2001년 11월 26일 심사완료)

Convergence Enhanced Successive Zooming Genetic Algorithm for Continuous Optimization Problems

Young-Doo Kwon, Soon-Bum Kwon, Nam Seo Goo and Seung Bo Jin

Key Words : Successive Zooming Genetic Algorithm(순차적 주밍 유전자 알고리즘), Micro Genetic Algorithm(마이크로 유전자 알고리즘), Zooming Factor(주밍 인자)

Abstract

A new approach, referred to as a successive zooming genetic algorithm (SZGA), is proposed for identifying a global solution for continuous optimization problems. In order to improve the local fine-tuning capability of GA, we introduced a new method whereby the search space is zoomed around the design point with the best fitness per 100 generation. Furthermore, the reliability of the optimized solution is determined based on the theory of probability. To demonstrate the superiority of the proposed algorithm, a simple genetic algorithm, micro genetic algorithm, and the proposed algorithm were tested as regards for the minimization of a multimimima function as well as simple functions. The results confirmed that the proposed SZGA significantly improved the ability of the algorithm to identify a precise global minimum. As an example of structural optimization, the SZGA was applied to the optimal location of support points for weight minimization in the radial gate of a dam structure. The proposed algorithm identified a more exact optimum value than the standard genetic algorithms.

1. 서 론

유전자 알고리즘(genetic algorithm, GA)은 생물 진화의 원리와 유전학에 바탕을 둔 확률론적 전역 탐색 방법이다. 유전자 알고리즘의 개념은 Holland⁽¹⁾에 의해 처음으로 제안되었고, Goldberg와 Santani^(2,3)에 의해 구조 최적화 문제에 처음으로 도입되었다.

그러나 유전자 알고리즘은 정밀도, 신뢰도, 연산시간 등이 서로 상충되는 문제의 경우, 해의 정밀도가 요구되는 경우에 있어 수렴이 느리거나, 정밀도의 부족 등과 같은 단점을 드러내었다. 따

라서 이러한 단점을 극복하기 위해 부분 탐색(local search)의 성능을 높이기 위한 방법으로 유전자 알고리즘과 지역적 탐색을 병행한 하이브리드 알고리즘,^(4,5) 민감도를 고려한 유전자 알고리즘,⁽⁶⁾ 조기 수렴을 막기 위한 개선된 유전자 알고리즘⁽⁷⁾ 등과 같은 연구들이 진행되고 있다.

실수 최적화 문제를 이진수로 부호화할 때 생기는 어려움을 극복하기 위한 방법들도 제안되었다. 대표적인 방법으로는 이진수 코드 대신 실수 코드를 사용하여 연속 최적화를 수행하는 것이 있다. 표준형 유전자 알고리즘(standard GA)에서 일반적으로 사용되는 이진수 표현 대신에 부동점 표현이 연속 최적화에 대해 편리하다는 것이 보고되었다.⁽⁸⁻¹²⁾

부동점 표현의 중요한 특징은 변수 값들이 교배 연산 동안 바뀌거나 파괴되지 않으며, 십진수를 이진수로 또는 이진수를 십진수로의 변환이 필

[†] 책임저자, 회원, 경북대학교 기계공학부
E-mail : ydkwon@knu.ac.kr
TEL : (053)950-5575 FAX : (053)950-6550
* 회원, 경북대학교 기계공학부
** 회원, 경북대학교 대학원 기계공학부

요없게 된다. 또한 이진수 GA에서 본질적으로 나타나는 해밍절벽(Hamming-cliff) 문제가 사라진다.⁽²⁾ 실수 코드를 사용함으로써 문제의 차원을 줄이고 기만(deceit)에 대한 가능성이 줄어드는 장점이 있는 반면 수렴에 대한 문제가 존재한다.

본 연구에서는 유전자 알고리즘이 가지는 부분 탐색의 문제점을 해결하기 위한 방법으로 순차적 주밍 방법(successive zooming method)을 제안하였다. 이 방법은 해 영역을 효율적으로 줄이면서 최적화를 수행하는 방법이다. 또한 실수 코드를 사용할 경우에 발생하는 수렴성 문제를 피하기 위하여 이진수 표현을 그대로 사용하면서 연속 최적화에 있어 부분 탐색의 효율을 높이는 동시에 연산시간도 줄이는 알고리즘을 제안하였다.

표준형 유전자 알고리즘의 단점을 극복하기 위해 여러 가지 개선된 방법들이 제안되었지만 그에 대한 수학적 근거가 제시된 예는 흔하지 않다. 본 연구에서는 제안된 유전자 알고리즘의 수학적 근거를 보이는 동시에 탐색한 해에 대해서 얼마나 신뢰할 수 있는가를 평가하였다.

제안된 알고리즘의 우월성을 검증하기 위해, 해석적으로 해가 존재하는 상당히 복잡한 시험 함수를 사용하여 표준형 유전자 알고리즘 및 마이크로 유전자 알고리즘의 결과와 비교하였다. 이 시험 함수는 상당히 많은 국부 최소해 때문에 최적해를 찾기가 어려운 함수들이다. 본 연구에서 제안한 순차적 주밍 유전자 알고리즘이 연속 최적 문제를 풀기 위한 신뢰할 만한 도구인 것을 입증한 후에 구조 최적화 문제로서 회전식 수문의 지지점 위치에 따른 중량 최소화 문제에 적용하였다.

2. 유전자 알고리즘

이 장에서는 전통적으로 많이 사용하는 표준형 유전자 알고리즘과 마이크로 유전자 알고리즘에 대해 간단히 소개한다.

2.1 표준형 유전자 알고리즘

표준형 유전자 알고리즘의 핵심은 자연의 진화 원리를 알고리즘 형태로 모방한 유전 연산자(genetic operator)라 할 수 있으며, 그 종류로는 선택 및 재생산(selection and reproduction), 교배(crossover), 돌연변이(mutation)로 이루어진다.

본 연구에서는 재생산 과정의 개체 선택도구로 토너먼트 선택(tournament selection)을 채용하였다. 토너먼트 선택이란 모집단으로부터 임의의 수의 개체를 무작위로 선택하여 그 중 적합도가 높은

개체는 살아 남고 그렇지 못한 개체는 도태되는 선택 방법이다.

또한 엘리트 전략(elitist strategy)을 채택하였다.⁽¹³⁾ 엘리트 전략은 집단 내에서 가장 강한 개체가 변경되지 않고 다음 세대로 전달되는 것을 보장하기 위해 고안된 것이다. 하나의 스트링이 다음 세대에서 후손을 가질 확률은 다음과 같다.

$$F(i) = \frac{f(i)}{\sum_{i=1}^N f(i)} \quad (1)$$

여기에서 $f(i)$ 는 i 번째 스트링의 적합도이다.

교배는 교배 급원으로부터 부모 염색체 쌍을 임의로 선택하고, 교배점 이후의 비트들을 서로 교환 결합함으로써 적합도가 우수한 새로운 개체들을 생성한다. 본 연구에서는 가장 간단하면서도 자주 이용되는 방법인 일점 교배를 적용하였으며 교배 확률은 0.95로 고정하였다.

돌연변이는 유전자의 다양성 결핍으로 인하여 발생하는 지역해(local solution)와 같은 원치 않는 해로부터 벗어나기 위한 메커니즘이다. 여기서 사용한 돌연변이 과정은 점프 돌연변이(jump mutation)과 크립 돌연변이(creep mutation)를 동시에 적용하였다. 본 연구에서는 점프 돌연변이 확률은 0.01, 크립 돌연변이 확률은 0.04로 고정하였다.

2.2 마이크로 유전자 알고리즘

Krisnakumar⁽¹⁴⁾는 매우 작은 집단을 취급하는 데 효율적인 마이크로 유전자 알고리즘을 제안하였고 Carroll⁽¹⁵⁾은 다른 연산자를 적용하는 문제를 다루었다. De Jong⁽¹³⁾은 마이크로 유전자 알고리즘이 표준형 유전자 알고리즘보다 초기 성능과 후기 성능을 동시에 개선할 수 있음을 보였다.

본 논문에서 사용된 연산자를 정리하면 다음과 같다. 재생산 연산자로는 토너먼트 선택을 사용하였고 엘리트 전략을 채용하였다. 교배 연산자로는 균등교배를 채용하였으며 어버이 쌍에는 반드시 교배가 일어나도록 하였다. ($P_c = 1$) 마이크로 유전자 알고리즘은 개체들이 어느 한 점 가까이로 수렴하게 되면 수렴된 집단 내의 최적 개체와 임의로 발생된 나머지 개체로 재구성되는 새로운 집단을 가지고 재시동하기 때문에 이 때마다 충분한 정도의 다양성이 도입된다. 따라서 별도의 돌연변이 연산이 필요없다. 따라서 돌연변이 확률은 $P_m = 0$ 으로 고정된다.

3. 순차적 주밍 유전자 알고리즘

유전자 알고리즘에는 두 가지 문제점이 있다. 첫 번째는 다봉함수에 있어 조기수렴으로 인한 낮은 정밀도이고, 둘째는 유전자 알고리즘이 가진 안정성의 결핍이다. 따라서 이러한 문제를 개선하기 위해 부동점 표현, 개선된 선택, 교배 및 돌연변이 연산자의 사용, 구간 축소와 같은 유전자 알고리즘의 성능 향상을 위한 방법들이 보고되고 있다.

본 연구에서는 마이크로 유전자 알고리즘의 골격을 유지하면서 탐색 공간을 확률 이론에 근거하여 적응적으로 줄여나가는 순차적 주밍 기법을 제안하였다. 또한 알고리즘을 사용하여 구한 최적해가 얼마나 정해에 가까운지를 나타내는 해의 신뢰도를 확률론적인 접근 방법을 통해 유도하였다.

3.1 순차적 주밍 기법

이 방법은 후보 최소점 주위로 탐색 공간을 적응적으로 줄여가는 것으로 일반적으로 모든 유전자 알고리즘에 적용될 수 있으나 본 연구에서는 마이크로 유전자 알고리즘에 적용하였다. 먼저, 초기 집단이 생성되고 마이크로 유전자 알고리즘이 실행된다. 정해진 세대(본 연구에서는 100 세대)가 진행된 후에 최적의 적합도를 나타내는 개체를 선정한다. 엘리트 전략을 사용하였으므로 항상 최적의 개체는 다음 세대가 진행하더라도 보존된다.

최적 개체가 선정되면 이 개체 주위로 탐색 영역을 $(X_k - \frac{\alpha^{N_{zoom}}}{2}, X_k + \frac{\alpha^{N_{zoom}}}{2})$ 로 좁혀 들어가고 이 축소 영역에서 새로운 초기 집단을 생성하여 계속 진화되어 나가도록 한다. 이러한 탐색 영역의 축소를 통하여 해의 resolution과 탐색 효율을 높일 수 있으며, 또한 구한 해가 실제의 최적해일 확률을 제시 할 수 있다. 여기에서, X_k 는 100세대 후의 최적 개체, α 는 주밍 인자(zooming factor), N_{zoom} 은 주밍 횟수이다. 이와 같은 과정을 해가 수렴될 때까지 계속 진행한다. Fig. 1에는 이 알고리즘의 순서도를 나타내었다.

3.2 해의 신뢰도

제안된 순차적 주밍 방법으로 구한 해가 가지는 신뢰도는 다음과 같다.

식 유도에 사용된 변수의 정의는 다음과 같다.

N_{var} ; 염색체를 이루는 변수의 개수

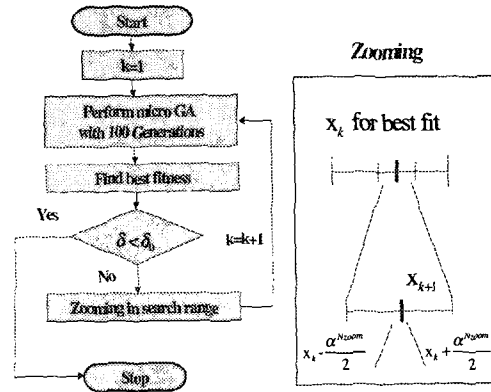


Fig. 1 Flow chart of successive zooming genetic algorithm (δ : convergence ratio)

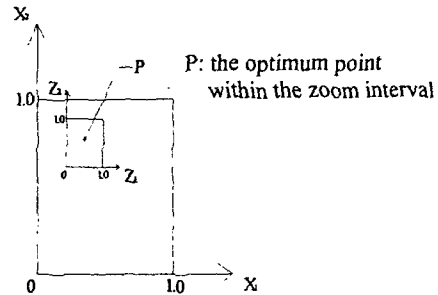


Fig. 2 Solution space (2-D case)

- N_{gen} ; 세대수
- N_{sub} ; 부반복 계산수
- N_{zoom} ; 주밍 횟수
- N_{pop} ; 집단수
- $N_{sp} = N_{sub} \times N_{pop}$
- α ; 주밍 인자

3.2.1 N_{zoom} 와 Z_{range} 사이의 관계

주밍 영역의 국부 좌표계를 X , 전역 좌표계를 Z 라 하자. Fig. 2에는 간단한 예로 2차원 해공간을 나타내었다. X 와 Z 의 범위는 항상 0과 1 사이가 되도록 영역을 조정(scale)하였다. 또한 적합도인 $F(i)$ 역시 0과 1 사이의 값으로 정의하였다.

N_{zoom} 와 Z_{range} 사이에는 $Z_{range} = \alpha^{N_{zoom}-1}$ 의 관계가 있으며 이를 Table 1에 나타내었다. 여기에서 Z_{range} 는 해의 분해능과 관계되며 N 번 주밍 후의 탐색 영역을 나타낸다. 예를 들어 $N_{zoom} = 6$, $\alpha = 0.05$ 일 때는 3.1×10^{-7} 이 된다. 따라서 원하는 분해능에 따라 N_{zoom} 이 결정된다.

Table 1 Relation between N_{zoom} and Z_{range}

N_{zoom}	1	2	3	4	5	6
Z_{range}	α^0	α^1	α^2	α^3	α^4	α^5

3.2.2 무작위 선택 알고리즘의 신뢰도

계질의 변화와 세대에 따라 집단 크기를 달리하는 자연계의 일부 곤충들의 경우와 같이 가변 집단을 채택하고 있는 유전자 알고리즘⁽¹⁵⁾도 있지만 대부분 집단 크기는 세대 변화와는 무관하게 항상 고정된 것으로 한다. 또한 집단 내의 염색체들의 길이는 서로 같고 세대 변화와는 상관없이 일정한 것으로 간주한다. N_{sp} 는 100 세대가 진행되는 동안에 탐색 영역에 떨어지는 총 개체수를 나타낸다. 최적해가 주밍 구간의 내부에 있다고 가정하면 무작위 탐색일 경우 적어도 하나의 탐색점이 주밍 구간에 떨어질 확률은 다음과 같다.

$$P_{sub}^{RA} = 1 - (1 - \alpha^{N_{var}})^{N_{sub} \times N_{pop}} \quad (2)$$

오른쪽 변의 두 번째 항은 모든 개체가 주밍 구간의 밖에 있을 확률을 의미한다. 100 세대 동안 단지 하나의 탐색점만이 주밍 구간 내에 떨어지더라도 최적점을 놓치지 않을 것이다.

3.2.3 유전자 알고리즘에 대한 신뢰도

유전자 알고리즘에서 적어도 하나의 탐색점이 주밍 구간 내에 떨어질 확률은 무작위 알고리즘의 경우보다 크거나 같아야 한다. 따라서 다음과 같은 식으로 표현된다.

$$P_{sub}^{GA} \geq P_{sub}^{RA} = 1 - (1 - \alpha^{N_{var}})^{N_{sp}} \quad (3)$$

따라서 유전자 알고리즘에서 100 세대 진행 후에 적어도 하나의 탐색점이 주밍 구간 내에 떨어질 확률은 다음과 같이 표현된다.

$$P_{sub}^{GA} = 1 - (1 - \alpha^{N_{var}} \times P_{avg})^{N_{sp}} \quad (4)$$

여기에서 P_{avg} 는 개선의 정도를 나타내기 위해 도입한 개선도 인자(improvement factor)이다. 개선도 인자란 무작위 알고리즘과 유전자 알고리즘사이의 주밍 구간 $(X_k - \frac{\alpha^{N_{zoom}}}{2}, X_k + \frac{\alpha^{N_{zoom}}}{2})$ 내에 떨어지는 개체수의 비로서 정의한다.

3.2.4 최종 신뢰도

N번 주밍 후 주밍 구간 내에 최종적으로 최적해가 존재할 확률, 즉 최적점을 놓치지 않을 확률은 다음과 같다. 식 (5)는 무작위 탐색의 경우를 나타낸다.

$$P_{final}^{RA} = [1 - (1 - \alpha^{N_{var}})^{N_{sp}}]^{N_{zoom}-1} \quad (5)$$

식 (6)은 유전자 알고리즘의 경우를 나타낸다.

$$P_{final}^{GA} = (P_{sub}^{GA})^{N_{zoom}-1} = [1 - (1 - \alpha^{N_{var}} \times P_{avg})^{N_{sp}}]^{N_{zoom}-1} \quad (6)$$

따라서, 식 (6)의 신뢰도가 1에 가까우면 수렴하는 것으로 하였다. 예로서 2차원의 경우를 고려해 보자. 사용된 변수 값들은 다음과 같다.

$$N_{var} = 2; \quad N_{gen} = 600; \quad N_{sub} = 100 \\ N_{zoom} = 6; \quad N_{pop} = 10; \quad \alpha = 0.05$$

식 (2)과 식 (5)에 위의 변수 값들을 대입하면 무작위 탐색인 경우의 확률은 다음과 같다.

1회 주밍 시

$$P_{one}^{RA} = 0.9182$$

N회 주밍 시

$$P_{final}^{RA} = 0.6526$$

무작위 탐색의 경우 한 번의 주밍 시의 확률은 0.92이고, N번의 주밍 시의 확률은 0.65가 된다. 식 (4)과 식 (6)에 변수 값들을 대입하면 유전자 알고리즘에 대한 확률은 Table 2와 같이 나타낼 수 있다. 이 표는 가정된 개선도 인자에 대한 확률을 나타낸다. Table 2에서 알 수 있듯이 개선도가 클수록 P_{sub}^{GA} 와 P_{final}^{GA} 의 값이 큰 것을 알 수 있다. 수치적인 계산 결과 개선도 인자는 보통 80이 넘는다.

위의 결과를 바탕으로 더 나은 신뢰도를 가진 해를 얻기 위해서는 다음 두 가지 사항에 유의해야 한다. 1) $N_{sp} = N_{sub} \times N_{pop}$ 가 충분히 커야 한다. 2) 적절한 주밍 인자 α 을 선택해야 한다.

최종적으로 우리가 취급한 확률은 실제 해와는 다른 최종 주밍 구간, $Z_{range} = \alpha^{N_{zoom}-1}$ ($= 3.1E-0.7$; $N_{zoom} = 6$, $\alpha=0.05$ 인 경우)내에서 우리가 찾은 최적해에 대한 확률이다. 간단하게 말하면, 실제해와 근사한 차이가 있는 순차적 주밍 유전자 알고리즘에서 찾은 최적해에 대한 확률이다.

Table 2 Probability in case of genetic algorithm

P_{avg}	1	2	8	$P_{avg} \geq 80$
P_{one}^{GA}	0.9182	0.9933	$1-\epsilon$ ($\epsilon=1.0E-11$)	$1-\epsilon$ ($\epsilon \leq 1.0E-15$)
P_{final}^{GA}	0.6526	0.9672	$1-\epsilon$ ($\epsilon=1.0E-11$)	$1-\epsilon$ ($\epsilon \leq 1.0E-14$)

4. 검증 및 구조 최적화에의 적용

제안된 알고리즘에 대한 수학적 타당성을 10 개 이상의 시험함수에 대하여 검토하였다. 또한 표준형 유전자 알고리즘과 마이크로 유전자 알고리즘을 사용하여 같은 시험 함수를 다루어 그들의 성능을 서로 비교하였다. 또한 각 시험함수에 대한 수치 결과로부터 개선도 인자의 평균을 구한 뒤 P_{final}^{GA} 을 계산하였다. 또한 제안된 알고리즘의 타당성을 검증한 뒤 구조 최적문제⁽¹⁸⁾를 고려하였다.

4.1 시험함수를 통한 검증

다음과 같은 시험함수에 대하여 제안된 알고리즘을 적용하여 해를 구하였다.

F1

$$f(x, y) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$

탐색공간: $-1.28 \leq x_1, x_2 \leq 1.27$

F2

$$f(x, y) = -(\cos 2\pi x + \cos 2.5\pi x - 2.1)(2.1 - \cos 3\pi y - \cos 3.5\pi y)$$

탐색공간: $-1 \leq x, y \leq 1$

F3⁽¹³⁾

$$F(X) = (0.002 + \sum_{j=1}^{25} (j + \sum_{i=1}^2 (x_i - a_{ij})^6)^{-1})^{-1}$$

탐색공간 : $-65.536 \leq x_1, x_2 \leq 65.536$

여기에서

$$[a_j] = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & \dots & -32 & -16 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & \dots & 32 & 32 & 32 & 32 & 32 \end{bmatrix}$$

Branin⁽⁷⁾

$$f(x, y) = a(y - bx^2 + cx - d)^2 + h(1 - f)\cos(x) + h$$

탐색공간: $-5 \leq x \leq 10 \quad 0 \leq y \leq 15$

여기에서

$$a=1, b=5.1/4\pi^2, c=5/\pi, d=6, h=10, f=1/8\pi$$

Camelback⁽⁷⁾

$$f(x, y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2$$

탐색공간: $-3 \leq x \leq 3 \quad -2 \leq y \leq 2$

Goldprice⁽⁷⁾

$$f(x, y) = [1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)] \times [30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)]$$

탐색공간 : $-2 \leq x \leq 2 \quad -2 \leq y \leq 2$

Shubert⁽⁷⁾

$$f(x, y) = \left\{ \sum_{i=1}^5 i \cos[(i+1)x + i] \right\} \left\{ \sum_{i=1}^5 i \cos[(i+1)y + i] \right\}$$

탐색공간 : $-10 \leq x \leq 10 \quad -10 \leq y \leq 10$

F4⁽¹⁶⁾

$$f(a, c_1, c_2) = \sum (ax_1^{c_1} x_2^{c_2} - f_{simulated})^2$$

탐색공간 : $-\infty \leq a, c_1, c_2 \leq \infty$

여기에서

$f_{simulated}$	x_1	x_2
2.122	5.0	10.0
9.429	3.0	1.0
23.57	0.6	0.6
74.25	0.1	2.0
6.286	3.0	1.8

F5⁽¹⁷⁾

$$F(X) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$$

탐색공간: $-10 \leq x_i \leq 10, (1 \leq i \leq 4)$

Brown3⁽⁷⁾

$$f(X) = \sum_{i=1}^{10} [(x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}]$$

탐색공간 : $X = [x_1, \dots, x_{20}]^T$
 $-1 \leq x_i \leq 4 \text{ for } 1 \leq i \leq 20$

F10N⁽⁷⁾

$$f(X) = (\pi/20)(10\sin^2(\pi x_1) + \sum_{i=1}^{10} [(x_i - 1)^2(1 + 10\sin^2(\pi x_{i+1}))]) + (x_{20} - 1)^2$$

탐색공간 : $X = [x_1, \dots, x_{20}]^T, -10 \leq x_i \leq 10$

Table 3 Results of SGA, MGA and SZGA for test functions

Function	N_{var}	Theoretical minimum	SGA		MGA		SZGA	
			Minimum	Function evaluation	Minimum	Speed ratio	Minimum (P_{final}^{GA} : Reliability)	Speed ratio
F1	2	0	0.44108E-5	1(4000)	0.40842E-5	0.22	0.29802E-7 (1.- ϵ)	1
F2	2	16.09172	16.09171	1(4000)	16.09171	0.5	16.09172(1.- ϵ)	1
F3	2	0.99800	0.99800	1(8000)	0.99800	4	0.99800(1.- ϵ)	4
Branin	2	0.39789	0.39789	1(8000)	0.39789	0.8	0.39789(1.- ϵ)	2
Camel Back	2	-1.03163	-1.03163	1(6000)	-1.03163	2	-1.03163 (1.- ϵ)	2
Gold Price	2	3	3.0001	1(6000)	3.0001	1.2	3 (1.- ϵ)	1.5
Shubert	2	-186.73091	-186.72744	1(6000)	-186.72825	0.5	-186.73091 (1.- ϵ)	2
F4	3	8.01276	8.01359	1(200000)	8.01293	0.74	8.01276(1.- ϵ)	10
F5	4	0	1.63594	1(200000)	0.44407E-2	0.63	0.28015E-6 (1.- ϵ)	1.5 \uparrow
Brown3	20	0	0.15029E-1	1(400000)	0.13565E-1	1.25	0.26122E-10 (1.- ϵ)	1.5 \uparrow
F10N	20	0	0.89927	1(400000)	0.16851	1.67	0.21058E-11 (1.- ϵ)	2.0 \uparrow

(\uparrow : This symbol is used because it is difficult to compare the computing time of SZGA with those of SGA and MGA as, in the case of many variables, SGA and MGA converged to a solution with a lower precision than that of SZGA. So this symbol is used. ϵ : very small value, $\leq 1.0E-14$)

위의 시험 함수들에 대해 표준형 유전자 알고리즘, 마이크로 유전자 알고리즘과 개선된 유전자 알고리즘에 적용하여 목적함수 값, 함수 계산 횟수 및 개선도 평균을 Table 3에 나타내었다. 속도 증가비(speed ratio)는 표준형 유전자 알고리즘의 함수 계산 횟수에 대한 순차적 주밍 유전자 알고리즘의 함수 계산 횟수의 비로 정의된다. 예를 들어 F3 함수는 표준형 유전자 알고리즘을 사용했을 때 8000번의 함수 계산을 하기 때문에 속도 증가비 4는 2000번의 함수 계산량을 의미한다.

시험 함수는 알고리즘의 강건성을 보이기 위하여 다양한 형태의 함수를 사용하였다. 본 논문에서 집단 수는 표준형 유전자 알고리즘의 경우 200개로 고정하였고 마이크로 유전자 알고리즘은 10개로 하였다. 순차적 주밍 유전자 알고리즘은 변수가 작은 경우는 10개로 하였지만 변수가 증가할수록 초기 단계에서 떨어지는 개체의 수에 따라 성능이 차이가 있기 때문에 변수의 수에 따라 가장 적당한 집단을 여러 번의 반복 수행에 의해 구하였다.

결과를 보면 마이크로 유전자 알고리즘은 표준형 알고리즘에 비해 작은 집단으로도 좋은 성능을 나타낼 수 있다는 것을 볼 수 있다. 그러나 F1함수의 경우 표준형 유전자 알고리즘은 개체 수 200으로 20세대 만에 수렴하는데 반해, 마이크로 유전자 알고리즘은 개체 수 20개의 작은 집단이지만 세대가 900 세대까지 가야 수렴한다. 앞에서 언급했듯이 마이크로 유전자 알고리즘은 작은 집단을 다루면서 좋은 성능을 보이지만 함수 계산 횟수는 표준형 유전자 알고리즘보다 커지게 된다. 개선된 순차적 주밍 유전자 알고리즘은 개체 수 10으로 초기 집단을 형성하여 400 세대에서 수렴하였다. 순차적 주밍 유전자 알고리즘의 함수 계산 횟수는 표준형 유전자 알고리즘과 같고, 마이크로 유전자 알고리즘보다 4배 적다. 따라서 순차적 주밍 유전자 알고리즘이 해의 정밀도가 더 높고 연산시간도 상당히 단축시킬 수 있다는 것을 알 수 있다.

F3함수는 일명 fox hole 함수라고도 불리며 연속, 비블록, 다봉 함수로서 25개의 지역 해를 갖는다. 전역 해는 $X = [-32, -32]^T$ 에서 존재하고 이 때의 함수 값은 0.998이다. 순차적 주밍 유전자 알고리즘의 연산시간은 표준형 유전자 알고리즘과 마이크로 유전자 알고리즘보다 4배 더 빠르다.

설계변수가 4개인 F5함수의 경우 $X = [1111]^T$ 에서 전역 해를 가지며 이 때의 함수 값은 0이다. 참고문헌⁽¹¹⁾에서, 실수코딩 유전자 알고리즘으로 이 함수의 해를 구하였을 때 약 2700 세대에서 근

사 해 $X = [0.993 \ 0.986 \ 1.007 \ 1.014]^T$ 를 찾았고 이 점에서의 함수 값은 약 0.181×10^{-3} 이다. 그러나, 본 연구에서 제안한 순차적 주밍 유전자 알고리즘은 근사 해 $X = [1.000 \ 0.999 \ 1.000 \ 1.000]^T$ 를 찾았고 이 점에서의 함수 값은 0.28015×10^{-6} 로 해의 정밀도가 더 높다는 것을 알 수 있다.

변수가 20개인 Brown3와 F10N함수의 경우도 이전의 연구⁽¹⁷⁾에 비해 순차적 주밍 유전자 알고리즘이 상당히 좋은 탐색 성능을 보였다.

Table 3의 F5, Brown3, F10N함수의 결과에서 알 수 있듯이 변수가 증가할수록 표준형 유전자 알고리즘과 마이크로 유전자 알고리즘의 수렴성이 상당히 떨어진다는 것을 볼 수 있다. 이것은 유전자 알고리즘의 부분 탐색 능력이 떨어지기 때문에 탐색 공간이 증가할수록 해의 정밀도가 떨어진다는 것을 나타낸다.

제안된 순차적 주밍 유전자 알고리즘은 초기 주밍 구간 내에 후보 최적점을 발견하기 위해 적어도 하나의 개체가 주밍 구간 안에 존재하여야 한다. 따라서, 초기 단계에서 집단수를 증가시키는 것이 좋은 방법일 것이다. 순차적 주밍 유전자 알고리즘의 탐색성능에 영향을 미치는 인자인 주밍 인자 α 는 경험적으로 0.05가 적당하였다.

4.2 구조 최적화예의 적용

제안된 순차적 주밍 유전자 알고리즘을 회전식 수문의 중량 최소화 설계에 적용하였다. 이 문제는 저자들이 다른 최적화 알고리즘을 사용하여 연구한 바 있다.⁽¹⁸⁾ Fig. 3은 동압과 정수압(hydro-static pressure)이 동시에 작용하는 주판(skin plate)을 3개의 지지점(arm)으로 지지하는 회전식 수문(radial gate)의 구조를 나타낸 그림이다.

ELH, ELS 및 P_{ice} 는 각각 수면과 주판(skin plate) 최하부의 높이 및 동결기 수면의 결빙에 관련된 동압이다. B는 주판(skin plate)의 폭을 나타낸다. 회전식 수문(radial gate)에 대해 수압(hydrostatic pressure), 수면의 동결과 관련된 동압(ice pressure), 주판(skin plate)의 자중 등을 고려하여 유도한 지배 방정식을 근거로 중량에 대한 정식화를 수행하였다. 설계변수로는 주판에서 각 지지점 사이의 길이가 차지하는 비(ratio)를 사용한다. 따라서 최적화 문제는 다음과 같이 정식화 될 수 있다.

Minimize $f(R)$ = volume of gate(vertical girders and supporting arms)

$$R = \{R_1 \ R_2 \ R_3 \ R_4\}^T$$

subject to

$$R_1 + R_2 + R_3 + R_4 = 1$$

R_i : dimensionless lengths

($\sigma_w \leq \sigma_a = S_y / SF$: vertical girders and arms, σ_w = 사용 응력, σ_a = 허용 응력, S_y = 항복 응력, SF = 안전 계수)

본 문제에서는 밀도가 일정하므로 부피를 최소화시키면 결국 중량이 최소가 된다.

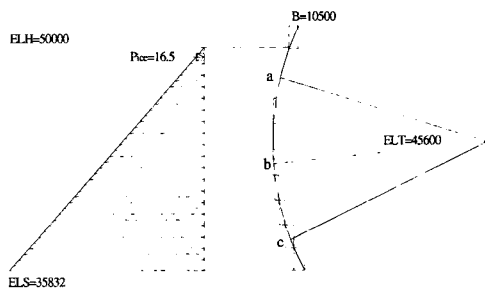


Fig. 3 Model of the radial gate with 3 arms (unit: mm)

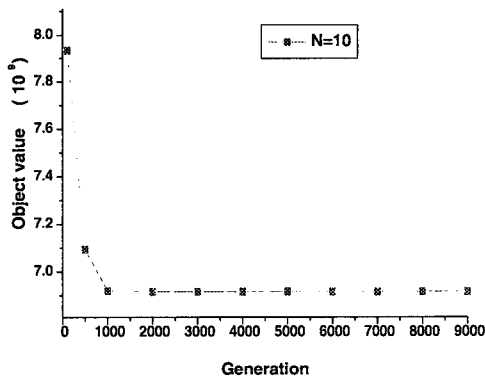


Fig. 4 Convergence according to generation for micro genetic algorithm

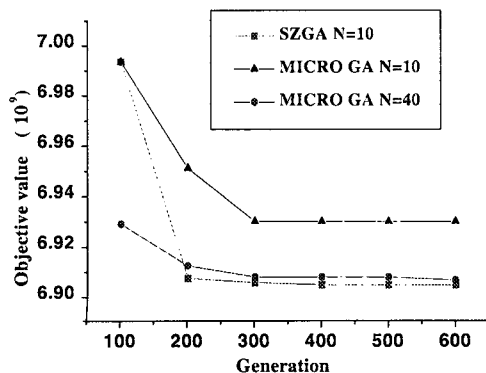


Fig. 5 Convergence according to generation for SZGA and micro GA

해석 결과를 Fig. 4와 Fig. 5에 나타내었다. Fig. 4에는 마이크로 유전자 알고리즘을 사용한 해의 추이를 나타내었다. 2000 세대 이후에 목적함수의 변화가 아주 작으나 세대가 계속 진행하면 함수 값이 작아지기 때문에 함수 값의 변화가 없을 때까지 진행시켜본 결과 9000 세대에서 수렴하는 것을 볼 수 있었다

Fig. 5은 순차적 주밍 유전자 알고리즘의 집단 수를 10개로 한 것과 마이크로 유전자 알고리즘의 집단 수를 10개와 40개로 한 것에 대해 세대가 진행함에 따라 수렴해 가는 정도를 나타낸다. 마이크로 유전자 알고리즘의 경우 Fig. 4와 같이 계속 최적값이 변화하나 600 세대까지만 나타내었다. 집단수가 10개로 같을 때는 순차적 주밍 유전자 알고리즘이 초기 성능 및 후기 성능도 좋다. 또한 마이크로 유전자 알고리즘의 집단수가 40개로 늘어나면 처음 100세대까지의 성능은 마이크로 유전자 알고리즘이 좋지만 그 이후의 성능은 순차적 주밍 유전자 알고리즘이 더 좋다는 것을 볼 수 있다.

그러나 Fig. 4에서 알 수 있듯이 마이크로 GA는 9000 세대가 지나서야 최종적으로 수렴된 결과를 내게 된다. 그러나 순차적 주밍 유전자 알고리즘은 400 세대만 진행하더라도 수렴한다는 것을 알 수 있다. 순차적 주밍 유전자 알고리즘의 경우 집단수가 10개만으로도 좋은 결과를 보이는데 반해 집단수 40개의 마이크로 유전자 알고리즘은 여전히 최적점에 수렴하지 못하고 있다는 것을 알 수 있다.

이상의 결과를 Table 4에 정리하였다. Table 4에서 DFP(Davidon-Fletcher-Powell)법은 지역 탐색 알고리즘으로서 여기서는 하이브리드 형태로 유전자 알고리즘과 결합하여 해를 구하여 순차적 주밍 유전자 알고리즘이 국부적인 탐색 성능도 우수함을 밝히기 위해서 적용한 것이다. Table 4를 보면 순차적 주밍 유전자 알고리즘이 마이크로 유전자 알고리즘 보다 수렴성이 좋으며 연산시간도 상당히 단축되는 것을 볼 수 있다.

Table 4 Results of several algorithms

	N_{var}	N_{pop}	Gene size	Convergence generation	Weight ($\times 10^0$)
3-arm					
Micro GA	3	10	36	9000	0.690722
Hybrid MGA+DFP	3	10	36	4000	0.690488
SZGA	3	10	36	400	0.690488

5. 결론

본 연구에서는 탐색 공간을 유연하게 줄여가는 순차적 주밍 기법을 이용한 수렴성이 개선된 유전자 알고리즘을 소개하였다. 새로운 알고리즘에 대하여 확률론적 접근을 통하여 해의 신뢰도를 예측하였다. 무작위 탐색에 대한 유전자 알고리즘의 개선 정도를 나타내기 위해 개선도 인자 P_{avg} 를 도입하였다. 제안된 순차적 주밍 유전자 알고리즘이 얼마나 강건하며 신뢰할 수 있는지를 확인하기 위하여 10개 이상의 시험함수를 가지고 검증하였다. 검증된 새로운 알고리즘을 실제 댐 구조 해석 문제에 적용하였다.

해석 결과로부터 만족할 만한 수렴성을 보인 설계변수가 작은 함수의 경우 연산 시간이 단축되었다. 또한 설계변수가 많은 불만족한 수렴성을 보인 함수에 대해서도 해의 정밀도가 상당히 좋아졌으며, 연산시간도 단축되었다는 것을 확인하였다. 보다 신뢰도 높은 해를 얻기 위해서는 다음 두 가지 사항에 유의해야 한다. 1) $N_{sp} = N_{sub} \times N_{pop}$ 가 충분히 커야 한다. 2) 적당한 주밍 인자 α 를 선택해야 한다.

제안된 알고리즘은 초기 주밍 구간 내에 후보 최적점을 발견하기 위해 적어도 하나의 개체가 주밍 구간 안에 존재하여야 한다. 따라서, 초기 단계에서 집단수를 증가시키는 것이 좋은 방법일 것이다. 순차적 주밍 유전자 알고리즘의 탐색 성능에 영향을 미치는 인자인 주밍 인자 α 는 경험적으로 0.05가 적당하다는 것을 알 수 있었다. 그러나 설계변수에 따른 최적의 α 값에 대해서는 연구가 더 필요할 것이다.

참고문헌

- (1) Holland, J.H., 1975, *Adaptation in Natural and Artificial systems*, University of Michigan, Ann Arbor, MI, Internal reports.
- (2) Goldberg, D.E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley.
- (3) Goldberg, D.E. and Kuo, C. H., 1987, "Genetic Algorithm in Pipeline Optimization," *Journal of Computers in Civil Engineering*, Vol. 1, No. 2, pp. 128~141.
- (4) Chu, K.C. and Gang, F., 1995, "Accelerated Genetic Algorithms: Combined with Local Search Techniques for Fast and Accurate Global Search," *IEEE International Conference On Evolutionary Computation*, Vol. 1, p. 378.
- (5) Heistermann, J., "Different Learning Algorithms for Neural Networks-A Comparative Study," In Schwefel and Manner.
- (6) 이재관, 신효철, 2000, "민감도가 고려된 유전 알고리즘을 이용한 최적화 방법에 관한 연구," 대한기계학회논문집 A 권, 제 24 권 제 6 호, pp. 1529~1539.
- (7) Andre, J., Siarry, P. and Dognon, T., 2001, "An Improvement of the Standard Genetic Algorithm Fighting Premature Convergence in Continuous Optimization," *Advances in Engineering Software*, Vol 32, pp. 49~60.
- (8) Wong, K.P. and Wong, Y.W., 1993, *Proc. ANZIS-93 Perth*, Western Australia, pp. 512~516.
- (9) Wong, K.P. and Wong, Y.W., 1994, *IEE Proc. Gen. Transm. Distrib.*, Vol. 141, p. 507.
- (10) Goldberg, D.E., 1991, *Complex Systems* 5, p. 139.
- (11) Janikow, C.Z. and Michalewicz, Z., 1991, An Experimental Comparison of Binary and Floating Point Representation in Genetic Algorithm, *Proceedings of the Fourth International Conference on Genetic Algorithm*. San Francisco: Morgan Kaufman, pp. 31~36.
- (12) Davis, L., 1991, *Handbook of Genetic Algorithm*, New York.
- (13) De Jong, K.A., 1975, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Doctoral Dissertation, The University of Michigan, Ann Arbor, Michigan.
- (14) Krishnakumar, K., 1989, "Micro-genetic Algorithms for Stationary and Non-stationary Function Optimization," *SPIE, Intelligent Control and Adaptive Systems*, Vol. 1196, pp. 289~296.
- (15) Carroll, D.L., 1996, "Genetic Algorithms and Optimizing Chemical Oxygen-Iodine Lasers," *Developments in Theoretical and Applied Mechanics*, Vol 18, pp. 411~424.
- (16) Belegundu, A.D. and Chandrupatla, T. R., 1999, *Optimization concepts and applications in engineering*. Prentice Hall.
- (17) Michalewicz, Z., 1996, *Evolution Programs*, Springer-Verlag, Berlin.
- (18) 권영두, 권순범, 박창규, 윤영중, 2001, "회전식 수문의 최적설계," 한국전산구조공학회 논문집, 제 14 권 제 3호, pp. 267~276.