

# 스키마 공진화 알고리즘과 GA의 성능 비교

## A Performance Comparison between GA and Schema Co-Evolutionary Algorithm

\*전호병, 전호병, 심귀보  
중앙대학교 전자전기공학부

Tel : 02-820-5319; Fax : 02-817-0553; E-mail : kbsim@cau.ac.kr

Ho-Byung Chun, Hyo-Byung Jun, and Kwee-Bo Sim  
School of Electrical and Electronics Engineering, Chung-Ang University  
Tel : 02-820-5319; Fax : 02-817-0553; E-mail : kbsim@cau.ac.kr

**Abstract :** Genetic algorithms(GAs) have been widely used as a method to solve optimization problems. This is because GAs have simple and elegant tools with reproduction, crossover, and mutation to rapidly discover good solutions for difficult high-dimensional problems. They, however, do not guarantee the convergence of global optima in GA-hard problems such as deceptive problems. Therefore we proposed a Schema Co-Evolutionary Algorithm(SCEA) and derived extended schema theorem from it. Using co-evolution between the first population made up of the candidates of solution and the second population consisting of a set of schemata, the SCEA works better and converges on global optima more rapidly than GAs. In this paper, we show advantages and efficiency of the SCEA by applying it to some problems.

### I. 서론

유전자 알고리즘(genetic algorithms : GAs)은 어려운 고차원의 문제들에 있어서도 최적해를 빠르게 구할 수 있다는 이점 때문에, 최근 들어 최적화 문제의 해결 방법으로써 여러 분야에 널리 사용되고 있다. 그런데 GA가 최적해를 찾지 못하고 국소해에 머무는 경우가 있다. 그 이유는 최적해를 포함하는 스키마의 적합도가, 경쟁하는 다른 스키마들의 적합도보다 낮아서 결국 최적해를 포함하지 않는 스키마 쪽으로 수렴하기 때문이다[1][2]. 즉, GA는 하나의 개체군으로 이루어져서, 미리 정의되어지는 적합도 함수에 의하여 성능의 제약을 받게 된다. 그러나 자연계는 두 개체군간의 상호작용과 공진화를 통하여 동적인 적합도 환경을 제공받는다. 이런 관점에서 공진화에 대한 관심이 높아지고 있다[3][4].

이와 같은 문제를 해결하기 위한 방법으로써 본 연구자들은 스키마 공진화 알고리즘(Schema Co-Evolutionary Algorithm: SCEA)을 제안하였다[5].

SCEA는 스키마로 구성된 부가적인 개체군을 도입

하여 기존의 해의 후보들로 구성된 주 개체군과의 공진화를 통해, 최적해를 포함하면서 평균 적합도가 낮은 스키마들도 기존의 GA에서보다 개체군의 기대치를 높여서 선택 될 가능성을 높였다. 또한 본 연구자들은 확장 스키마 정리(extended schema theorem)를 제안하여, SCEA가, 정의장(defining length)이 짧고, 차수가 낮고, 적합도가 높은 스키마들의 기대치를 GA보다 증가시켜서 더욱더 빠른 수렴성을 가지게 함을 보였다[5].

본 논문에서는 몇 가지 문제에 SGA와 SCEA를 적용해봄으로써 일반적인 GA에 대한 SCEA의 효율성과 장점들을 살펴본다. 본 논문의 구성은 다음과 같다. 2장에서는 SCEA, 3장에서는 확장 스키마 정리를 살펴본다. 4장에서는 실험 결과의 고찰을 통하여 SCEA의 효율성과 장점을 보이고, 5장에서 결론을 맺는다.

### II. 스키마 공진화 알고리즘

다른 공진화 알고리즘들과 마찬가지로 SCEA도 공진화를 위한 두 개의 서로 다른 개체군을 갖는다.

첫 번째 개체군은 숙주 개체군(host-pop.)이라 하며, 최적해를 찾기 위한 해의 후보들로 구성되어 주어진 환경으로부터 적합도의 평가를 받으며 선택, 교차, 돌연변이, 재생의 과정을 거친다. 두 번째 개체군은 기생 개체군(parasite-pop.)이라 하며, 유용한 스키마들을 찾기 위한 스키마들로 이루어져 있다. 이 개체군은 숙주 개체군으로부터 적합도를 평가받으며 마찬가지로 선택, 교차, 돌연변이, 재생의 과정을 거친다.

그림 1은 SCEA의 개관을 나타낸다.

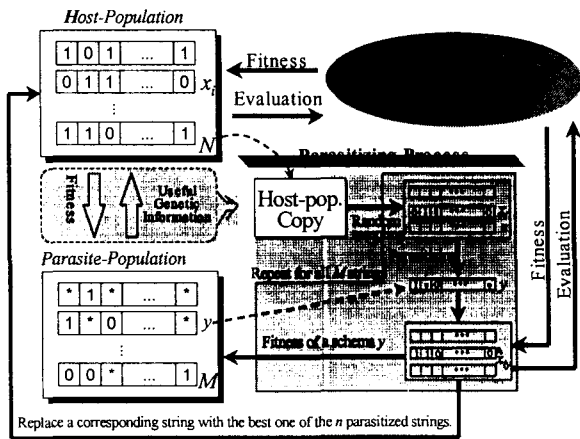


그림 1. 스키마 공진화 알고리즘의 개관

SCEA는 이와 함께 기생 프로세스(parasitizing process)라는 과정을 더 갖고 있다. 여기서 기생(寄生)이란, 기생 개체군은 숙주 개체군에 의해서만 적합도를 평가받고 그에 따라 그 다음 세대에서의 생존이 결정되는 것을 의미한다.

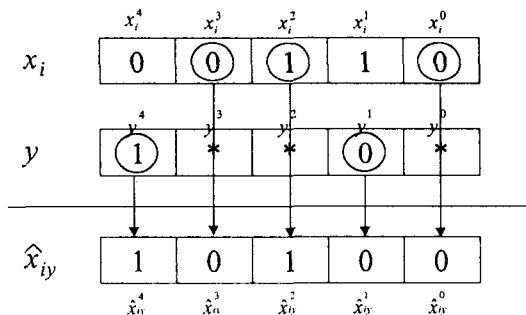


그림 2. 마스크

기생 프로세스는 숙주 개체군의 어떤 스트링의 일부분의 값을 마스크(masking) 과정을 통하여 스키마의 고정된 값으로 대체하고, 기생 개체군의 각 스키마들에게 새로운 적합도를 주며, 이를 통하여 생성된 새로운 스트링은 마스크한 스키마의 인스턴스가 된다. 마스크 과정을 다음과 같이 표현할 수 있다.

$$\hat{x}_{iy}^p = \begin{cases} y^p, & y \text{의 위치 } p \text{의 값이 고정값일 때} \\ x_i^p, & \text{그 밖의 경우} \end{cases} \quad (0 \leq p \leq l-1) \quad (1)$$

여기서  $y$ 는 마스크를 하는 스키마,  $p$ 는 스트링에서의 위치 인덱스,  $x_i^p$ 는 스트링  $x_i$ 의 위치  $p$ 의 값,  $l$ 은 스트링의 길이를 나타낸다. 기생 개체군은 기생 프로세스에 의하여 유용한 스키마들을 찾고 마스크를 통하여 숙주 개체군에 유용한 스키마들의 인스턴스들을 새롭게 증가시킴으로써 유전 정보를 전달한다.

그림 2는 마스크 과정을, 그림 3은 기생 프로세스를 보여준다.

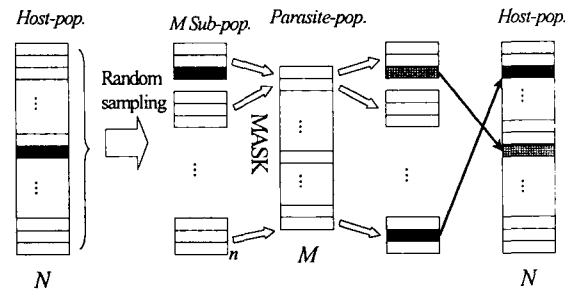


그림 3. 기생 프로세스

여기서  $N$ 은 숙주 개체군의 크기,  $M$ 은 기생 개체군의 크기,  $n$ 은 기생 개체군 내의 각각의 스키마에 의하여 마스크 될 개체군의 크기이다.

다음은 기생 개체군의 스트링  $y$ 에 대한 기생 프로세스의 진행과정과 스트링  $y$ 의 적합도  $F_y$ 를 구하는 순서이다.

단계 1 : 먼저 숙주 개체군의 복사본을 만들고, 그곳에서 기생 개체군의 스트링  $y$ 에 의하여 기생시킬  $n$ 개의 스트링들을 무작위로 뽑는다.

단계 2 : 단계 1에서 뽑은  $n$ 개의 스트링들을  $x_1, \dots, x_n$ 이라 하고, 이 스트링들을 스트링  $y$ 에 의하여 마스크를 거쳐  $\hat{x}_{1y}, \dots, \hat{x}_{ny}$ 로 만든다.

단계 3 : 기생 개체군의 스트링  $y$ 의 적합도를 결정하기 위하여  $n$ 개의 스트링들 각각에 대하여 식 (2)와 같이 적합도를 계산한다.

$$\hat{f}_{iy} \triangleq f(\hat{x}_{iy}) - f(x_i) \quad (i=1, \dots, n) \quad (2)$$

여기서  $f(x_i)$ 는 스트링  $x_i$ 의 적합도이며,  $f(\hat{x}_{iy})$ 는 스트링  $y$ 에 의하여 마스크를 거친 스트링  $\hat{x}_{iy}$ 의 적합도이다.

단계 4 : 기생 개체군의 스트링  $y$ 의 적합도  $F_y$ 를 다음과 같이 구한다.

$$F_y \triangleq \sum_{i=1}^n \max[0, \hat{f}_{iy}] \quad (3)$$

그리고  $\hat{x}_{1y}, \dots, \hat{x}_{ny}$ 중에서 적합도가 가장 높은 스트링 하나만을 숙주 개체군 내의 뽑아왔던 위치로 되돌린다.

### III. 확장 스키마 정리

일반적인 GA에 적용되는 스키마 정리에서는  $k$  세대에 개체군에 있는 스키마  $H$ 의 인스턴스들의 수  $m(H, k)$ 는,  $k+1$ 세대에, 그 스키마의 평균 적합도와 교차 확률, 돌연변이 확률 등에 의하여 새로이 증가 또는 감소한 양으로 대체되어  $m(H, k+1)$ 이 된다. 그와 마찬가지로 SCEA에서의  $k$ 세대에 기생 프로세스를 거친 후의 숙주 개체군에 있는 스키마  $H$ 의 인스턴스들의 수  $m'(H, k)$ 를 다음 식 (4)와 같이 나타낸다.

$$m'(H, k) = m(H, k) + \widehat{m}(H, k) \quad (4)$$

여기서  $m(H, k)$ 는  $k$ 세대에 기생 프로세스를 거치기 전 숙주 개체군에 있는 스키마  $H$ 의 인스턴스들의 수이고,  $\widehat{m}(H, k)$ 는  $k$ 세대에 기생 프로세스를 거쳐서 숙주 개체군 내에 새로 증가하거나 감소한 스키마  $H$ 의 인스턴스들의 수이다.  $\widehat{m}(H, k)$ 는 다음 식 (5)와 같이 계산되어진다.

$$\widehat{m}(H, k) = \sum_{y \in S_H, x_q \in I_H} \lambda([f(\widehat{x}_{qy}) - f(x_q)] > 0) - \sum_{y \notin S_H, x_q \in I_H} \lambda([f(\widehat{x}_{qy}) - f(x_q)] > 0) \quad (5)$$

여기서  $\lambda(A)$ 는  $A$ 가 참이면 1의 값을,  $A$ 가 거짓이면 0의 값을 갖는다. 그리고  $I_H$ 는 스키마  $H$ 의 인스턴스들의 집합이고,  $q = \arg \max_i \hat{f}_{iy}$ ,  $S_H$ 는 스키마  $H$ 에 포함되는 자기자신을 포함한 교차 스키마들의 집합이다. 예를 들어 스키마  $H$ 가  $1**$ 라면  $S_H = \{1**, 1*0, 1*1, 10*, 11*, 100, 101, 110, 111\}$ 이 된다.

식(4)와 식(5)에 의하여 스키마  $H$ 의 평균 적합도  $f'(H, k)$ 를 구하면 다음 식 (6)과 같다.

$$f'(H, k) = \frac{\sum_{x \in I_H} f(x) + \sum_{\widehat{x}_{iy} \in \widehat{I}_H^+} f(\widehat{x}_{iy}) - \sum_{x \in I_H} f(x)}{m(H, k) + \widehat{m}(H, k)} \quad (6)$$

여기서  $\widehat{I}_H^+$ 는 기생 프로세스에 의해서 증가된 스키마  $H$ 의 인스턴스들의 집합이고,  $\widehat{I}_H^-$ 는 기생 프로세스에 의해서 감소한 스키마  $H$ 의 인스턴스들의 집합이다.

스키마  $H$ 의 적합도는 스키마  $H$ 에 포함되는, 개체군 내의 모든 스트링들, 즉 개체군 내에서의 스키마  $H$ 의 인스턴스들의 적합도를 평균한 값으로 정의되어지므로, SCEA에서 기생 프로세스를 거친 후의 스키마  $H$ 의 평균 적합도  $f'(H, k)$ 는  $f'(H, k) \approx f(H, k)$ 로 근사할 수

있다. 즉 식 (6)에서 분자와 분모에 첨삭되는 양이 비례적으로 볼 때는 식 전체에 큰 값의 변화를 가져오지 않는다는 것이다. 특히 그림 3에서  $N \gg n$ 인 경우 위의 근사는 긴 세대의 변화를 살펴보았을 때 타당하다[6].

결과적으로 우리는 SCEA로부터 다음 식 (7)과 같은 확장 스키마 정리를 얻을 수 있다.

$$m(H, k+1) \geq [m(H, k) + \widehat{m}(H, k)] \cdot \frac{f(H, k)}{f(k)} \cdot \left[ 1 - p_c \cdot \frac{\delta(H)}{l-1} - p_m \cdot \alpha(H) \right] \quad (7)$$

본래의 스키마 정리와 비교해 보면, 정의장이 짧고, 차수가 낮으며, 적합도가 개체군의 평균 적합도보다 높은 스키마들이 일반적인 GA에 비하여 더욱 빨리 증가함을 알 수 있다.

### IV. 실험 결과

첫 번째로 SGA와 SCEA 모두 최적해를 찾을 수 있는 간단한 함수를 골랐다[7]. 적용한 함수는 다음 식(8)과 같다.

$$f(x) = x \cdot \sin(10\pi \cdot x) + 2.0 \quad (8)$$

[7]에서와 같이 스트링의 길이를 22비트로, SGA는 개체군 크기를 50으로, SCEA는  $N$ 을 30,  $M$ 을 10,  $n$ 을 2로 하여 계산량을 SGA와 맞추고 각각 150세대씩 50회동안 적용하였다. 표 1은 교차확률과 돌연변이 확률을 변화시켜가며 적용해본 결과이다. SCEA의 경우는 기생 개체군의 교차확률을 0.6, 돌연변이 확률을 0.02로 고정시키고 숙주 개체군만 확률을 변화시켜가며 적용하였다.

표 1 식(8)의 함수에 SGA와 SCEA를 적용시킨 결과

p <sub>c</sub>	p <sub>m</sub>	최적값 근처로 수렴한 회수		기존 최적값을 넘은 회수	
		SGA	SCEA	SGA	SCEA
0.25	0.001	50	50	8	36
	0.01	50	50	37	43
	0.1	50	50	48	34
	0.5	50	50	38	31
0.60	0.001	50	50	24	28
	0.01	50	50	43	35
	0.1	50	50	49	37
	0.5	50	50	36	34
0.95	0.001	50	50	16	25
	0.01	50	50	42	43
	0.1	50	50	49	38
	0.5	50	50	32	37

표1에서는 [7]에서 나타낸 최적값을 넘어선 값을 찾아낸 경우를 표시한 것으로, 식(8)의 함수가 최적해를 갖는 근처를 자세히 살펴보면 분해능이

$2^{22}$ 이기 때문에 정확한 최적해를 찾는 것이 상당히 힘든 문제임을 알 수 있다. 그러나 일반 GA로도 최적해가 속해있는 근처로의 수렴은 빠르게 할 수 있는 문제이다. 실제 계산으로 나온 최적해는  $x = 1111001100111111001001_{(2)}$ 에서 3.850273766... 이지만 [7]에서는 3.850227 정도를 찾은 것으로 만족하였다. 실험에서는 SGA와 SCEA가 50회 모두 최적해 근처의 값을 찾아내었다. 그러나 SCEA가 더 최적해에 가까운 값을 찾은 회수가 많은 것을 볼 수 있었다. 즉 최적해의 바로 앞의 함수값인 3.850273765... 이상의 값도 자주 찾아내는 것을 볼 수 있었다.

다음으로 False-peaks 문제에 적용하여 보았다. False-peaks는 국소해를 많이 가짐으로써 일반 GA로는 최적해를 찾기가 쉽지 않은 문제이다. 적용한 함수와 적합도를 계산하는 식은 각각 식(9), (10)과 같다.

$$f = (x_1 \wedge x_2 \wedge \dots \wedge x_{20}) \vee (\overline{x_1} \wedge \overline{x_2} \wedge \dots \wedge \overline{x_{20}}) \quad (9)$$

$$Fit(f) = \max \left[ \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_{20}^2}{20}}, \sqrt{\frac{x_1^2 + (1-x_1)^2 + \dots + (1-x_{20})^2}{21}} \right] \quad (10)$$

적용한 함수는 스트링의 길이가 20비트로 모든 비트가 1인  $x = 11111111111111111111_{(2)}$ 에서 최대 값 1을 가지며, 0.9759의 국소해를 여러 개 갖는다. SGA는 개체군 크기를 500으로, SCEA는  $N$ 을 200,  $M$ 을 100,  $n$ 을 3으로 하여 계산량을 SGA와 맞추고 각각 50세대씩 50회동안 적용하였다.

표 2 False-peaks에 SGA와 SCEA를 적용시킨 결과

Pc	Pm	최적해를 찾은 회수		최적해를 찾은 평균 세대수	
		SGA	SCEA	SGA	SCEA
0.25	0.001	21	38	37.38	6.42
	0.01	16	35	36.63	6.54
	0.1	4	38	25.75	6.92
	0.5	0	34	-	10.85
0.60	0.001	30	35	35.60	6.03
	0.01	24	39	32.00	6.13
	0.1	3	34	19.67	6.94
	0.5	0	38	-	10.42
0.95	0.001	36	39	35.64	6.10
	0.01	22	33	34.41	6.61
	0.1	4	33	36.75	8.70
	0.5	0	33	-	9.88

표 2는 첫 번째 문제와 같은 방법으로 교차확률과 돌연변이 확률을 변화시켜가며 적용해본 결과이다. SCEA의 경우 교차확률과 돌연변이 확률에 관계없이 SGA보다 최적해를 찾는 확률이 훨씬 높고, 또한 빠르게 수렴하고 있음을 볼 수 있다.

## V. 결론

본 논문에서는 제안된 SCEA와 그에 따른 확장 스키마 정리의 내용을 살펴보고, 일반 GA로도 해결할 수 있는 문제와, 해결하기 힘든 문제 각각에 대하여 SGA와 SCEA를 적용하여 보았다.

실험 결과에서 보는 바와 같이 SCEA는 두 개체군간의 공진화를 통하여 일반 GA보다 최적해를 찾을 확률이 높고, 빠른 수렴속도를 가짐을 볼 수 있었다. SCEA는 GA보다 개체군의 크기가 작더라도 스키마에 의하여 보다 다양한 탐색공간을 속주 개체군에 제공함으로써 최적해를 쉽게 찾을 수 있도록 하며, 확장 스키마 정리에서 보듯이 세대에 따른 스키마의 인스턴스 수의 기대치의 변화를 크게 하여 빠른 수렴성을 제공하며, 마스킹을 통하여 우수한 스트링들만 개체군으로 보냄으로써 우수한 개체가 쉽게 파괴되지 않는 강건성을 가진다.

앞으로 SCEA에서 기생 개체군의 교차 확률과 돌연변이 확률을 변화시켜가며 여러 문제에 적용시켜 보는 것과, 좀 더 다양한 문제들에 적용시켜서 성능의 우수함을 보이는 것이 필요하겠고, 나아가서 속주 개체군, 기생 개체군의 크기, 각 스키마에 의하여 마스킹 될 대체군의 크기를 문제에 따라서 어떻게 정하는 것이 최적인지를 밝혀내는 노력이 요구된다.

## 감사의 글

본 연구는 과학기술부의 뇌과학 프로젝트(Braintech 21)의 지원으로 이루어진 결과임 (과제번호 : 98-J04-01-01-A-07).

## VI. 참고 문헌

- [1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
- [2] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [3] D. Cliff, G. F. Miller, "Tracking Red Queen: Measurements of adaptive progress in co-evolution," *COGS Technical Report CSR 363*, Univ. of Sussex, 1995.
- [4] Jan Paredis, "Co-evolutionary Computation," *Artificial Life*, Vol. 2, No. 4, pp. 353-375, 1995.
- [5] H. B. Jun and K. B. Sim, "Schema Analysis on Co-Evolutionary Algorithm," *Journal of Control, Automation and Systems Engineering*, Vol. 4, No. 5, October, 1998.
- [6] Hans-Paul Schwefel, *Evolution and Optimum Seeking*, A Wiley-Interscience Publication, John Wiley & Sons, Inc., 1995.
- [7] Z. Michalewicz, *Genetic Algorithms+Data Structures=Evolution Programs*, Springer-Verlag, 1995