

기생적 공진화 알고리즘을 이용한 퍼지 제어기 설계

Design of Fuzzy Controller Using Parasitic Co-evolutionary Algorithm

심귀보*, 변광섭
(Kwee-Bo Sim and Kwang-Sub Byun)

Abstract : It is a fuzzy controller that it is the most used method in the control of non-linear system. The most important part in the fuzzy controller is a design of fuzzy rules. Many algorithms that design fuzzy rules have proposed. And attention to the evolutionary computation is increasing in the recent days. Among them, the co-evolutionary algorithm is used in the design of optimal fuzzy rule. This paper takes advantage of a schema co-evolutionary algorithm. In order to verify the efficiency of the schema co-evolutionary algorithm, a fuzzy controller for the mobile robot control is designed by the schema co-evolutionary algorithm and it is compared with other parasitic co-evolutionary algorithm such as a virus-evolutionary genetic algorithm and a co-evolutionary method of Handa.

Keywords : design of fuzzy controller, schema co-evolutionary algorithm, co-evolutionary method of Handa, virus-evolutionary genetic algorithm

I. 서론

비선형 시스템의 제어에서 가장 많이 사용되는 방식이 퍼지 제어기이다. 이때, 가장 중요한 것은 퍼지 룰의 설계이다. 퍼지 룰을 설계하는 많은 기법들이 제안되어 있는데, 그 중에 진화 연산에 대한 연구가 활발히 진행되어 왔다. 진화 연산의 대표적인 유전 알고리즘(genetic algorithm : GA)은 자연세계의 진화과정을 컴퓨터상에서 시뮬레이션 함으로써 복잡한 실세계의 문제를 해결하고자 하는 계산 모델이다. 또한 GA는 구조가 간단하고 방법이 일반적이어서 응용범위가 매우 넓으며, 특히 적응적 탐색과 학습 및 최적화를 통한 공학적인 문제의 해결에 많이 이용되고 있다.

그런데 GA가 모든 최적화 문제에 좋은 결과를 가져오는 것은 아니다. 예를 들면, 최적해를 포함하는 스키마의 적합도가 경쟁하는 최적해를 포함하지 않는 스키마의 적합도보다 낮아서 결국 최적해를 포함하지 않는 스키마 쪽으로 수렴하기 때문에 최적해를 찾지 못하고 국소해에 머무는 경우가 발생하는데 이것을 GA-hard problem이라고 한다[1]. 즉 GA는 하나의 개체군으로 이루어져 있어서, 미리 정의되어지는 적합도 함수에 의하여 성능의 제약을 받게 된다. 이와 달리 자연계는 두 개체군간의 상호작용과 공진화를 통하여 동적인 적합도 환경을 제공받는다. 이러한 관점에서 공진화에 대한 관심이 높아지고 있다.

공진화는 두 개의 개체군이 서로 영향을 주고 받으면서 진화해가는 알고리즘으로, 크게 경쟁적 공진화와 협조적 공진화로 분류된다. 퍼지 시스템과 같은 규칙 기반 시스템의 설계에 공진화를 적용하는데 있어서는 일반적으로 퍼지 룰 안에서 두 개체군을 설정하기 때문에 협조적 공진화가 적합하다. 즉, 퍼지 룰이 서로 긍정적인 영향을 미치기 때문이다.

협조적 공진화 중에서 기생적 공진화는, 전체 룰의 정보를 모두 주 개체군이 갖고 있고, 부 개체군에서는 주 개체군의 적합도를 높이기 위해 효율적인 스키마를 공급한다. 즉, 기생적 공진화는 진화가 빠르고 강인하게 일어날 수 있도록 유용한 스키마를 공급하는 스키마 정리와 빌딩 블록 가설을 적용한 알고리즘이다[2][3].

대표적인 기생적 공진화 알고리즘으로 스키마 공진화 알고리즘, 바이러스-진화 유전 알고리즘, Handa의 공진화 알고리즘 등이 있다[4][5][6]. 이 세 알고리즘은 기생적 공진화 알고리즘으로서 구조적으로는 매우 유사하지만, 각각 독특한 특성을 보인다. 특히 스키마 공진화 알고리즘은 가장 빠르고 강인하게 유용한 스키마를 주 개체군에 공급함으로써 공학적으로 응용 가능성이 가장 크다. 따라서 본 논문에서는 퍼지 규칙의 자동생성을 위해 스키마 공진화 알고리즘을 이용한다.

본 논문의 2장에서는 스키마 공진화 알고리즘의 자세한 프로세스를 설명하고 3장에서는 다른 기생적 공진화와 알고리즘 차원에서 비교를 한다. 4장에서는 실제 퍼지 룰 설계에 대한 시뮬레이션을 수행하고 세 알고리즘을 비교한다. 마지막으로 5장에서는 결론에 대해 논한다.

II. 스키마 공진화 알고리즘

1. 스키마 공진화 알고리즘

최적의 퍼지 제어기를 자동으로 설계하기 위해 본 논문에서는 스키마 공진화 알고리즘(Schema Co-Evolutionary Algorithm: SCEA)을 이용한다. SCEA의 기본적인 프로세스는 1998년에 제안되었다[7]. 다른 공진화 알고리즘과 같이 SCEA는 두 개의 협조적으로 작용하는 개체군을 갖는다. 호스트-개체군은 최적해의 후보군으로 구성되고 일반적인 유전 알고리즘과 같은 방식으로 수행되고, 기생-개체군은 스키마타의 집합으로서, '빌딩 블록' 이라 불리는 유용한 스키마를 찾는 데 사용된다[2][8]. 그림 1은 SCEA 프로세스를 보여준다.

SGA(Simple Genetic Algorithm)는 한 세대동안 네 가지 중요한 단계를 갖는데, 그것은 적합도 평가, 선택, 교차, 돌연변이이다. 그러나 SCEA는 선택과정 전에 기생 프로세스를 추가

* 책임저자(Corresponding Author)

논문접수 : 2004. 9. 25., 채택확정 : 2004. 10. 13.

심귀보, 변광섭 : 중앙대학교 전자전기공학부

(kbsim@cau.ac.kr/ks4070@wm.cau.ac.kr)

※ 본 연구는 과학기술부의 뇌신경정보학연구사업인 뇌정보처리에 기반한 감각정보 융합 및 인간행위 모델 개발의 연구비 지원으로 수행되었습니다. 연구비 지원에 감사드립니다.

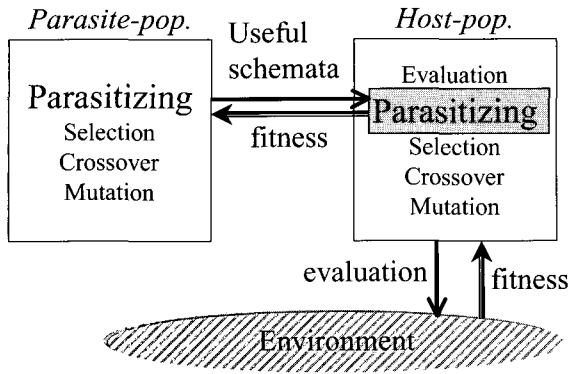


그림 1. SCEA 프로세스.
Fig. 1. Process of the SCEA.

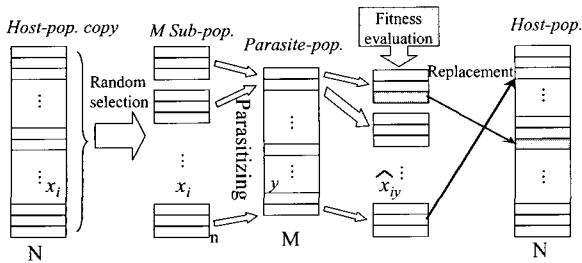


그림 2. 기생 프로세스. N 은 호스트-개체군 크기, M 은 기생-개체군 크기, n 은 M 개의 sub-개체군 크기.
Fig. 2. Parasitizing process. N is the size of host-pop., M is the size of parasite-pop., n is the size of M sub-pop.

로 포함한다. 호스트-개체군의 모든 개체들이 평가되어진 후에, 그 중의 몇 개가 기생-개체군의 스키마를 위해 임의로 선택된 다음, 기생 프로세스가 수행된다. 이것을 통해 새롭게 생성된 개체를 평가하고, 원래의 개체와 기생 프로세스에 의해 변한 개체 사이의 적합도 개선 정도를 측정한다. 가장 큰 개선 값을 갖는 기생 개체를 각각의 스키마에 대응하는 개체와 대치한다. 개선된 양을 이용하여, 기생-개체군에서 각각의 스키마의 적합도를 부여할 수 있다. 그러므로 기생-개체군의 스키마의 적합도는 스키마의 유용성을 가리킨다. 적합도 평가 후의 기생-개체군에는 SGA와 동일한 과정을 적용한다. 기생-개체군의 개체의 적합도는 기생 프로세스에 의해 계산된다. 따라서 기생-개체군은 기생, 선택, 교차, 돌연변이의 네 단계를 거친다. 기생 프로세스는 다음절에 자세히 나타나 있다[7].

2. 기생 프로세스

기생-개체군은 유용한 스키마를 탐색하고 기생 프로세스에 의해 유전 정보를 호스트-개체군으로 전달한다. 그림 2는 기생 프로세스를 보여준다.

기생-개체군의 스키마의 적합도는 호스트-개체군에서 추출된 n 개의 스트링에 의해 결정된다. 기생 프로세스는 스트링의 유전자가 스키마의 특정한 유전자에 의해 대체되어지는 것을 의미한다. 스트링의 다른 유전자좌(*: don't care)는 자신의 값을 갖는다. 따라서

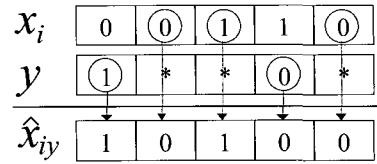


그림 3. Parasitizing의 예 : x_i 는 스키마 y 에 의해 \hat{x}_{iy} 로 parasitizing 됨.

Fig. 3. Example of the parasitizing : x_i is parasitized to \hat{x}_{iy} by the schema y .

$$\hat{x}_i^p = \begin{cases} y^p, & \text{if } p\text{th character of } y \text{ is fixed} \\ x_i^p, & \text{otherwise} \end{cases}, \quad (0 \leq p \leq l-1) \quad (1)$$

여기에서 p 는 유전자좌, x_i^p 는 스트링 x_i 의 p 번째 자리 값, l 은 스트링의 비트 개수를 나타낸다. 그림 3은 기생화의 한 예를 보여준다. SCEA의 주요 프로세스는 기생-개체군에서 찾은 유용한 스키마를 적합도에 따라 호스트-개체군에 전달하는 것이고, 기생-개체군의 진화 방향은 호스트-개체군에 의해 결정된다.

기생-개체군에서 스트링 y 의 적합도 F_y 는 다음 단계에 의해 결정된다.

Step 1: 호스트-개체군에서 n 개의 스트링이 임의로 선택되어 복사된다.

Step 2: 그림 3과 같이 기생화가 수행된다.

Step 3: 기생-개체군에서 스트링 y 의 적합도를 결정하기 위해, 적합도 함수를 다음 식과 같이 적합도의 개선 값으로 설정한다.

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

이 식에서 x_1, \dots, x_n 는 추출된 스트링, $\hat{x}_{1y}, \dots, \hat{x}_{ny}$ 는 기생화된 스트링, $f(x_i)$ 는 스트링 x_i 의 적합도, $f(\hat{x}_{iy})$ 는 \hat{x}_{iy} 의 적합도를 나타낸다.

Step 4: 각각 M 개의 스키마에 대해, 가장 큰 개선 값을 갖는 최적 개체가 원래의 호스트-개체군의 대응되는 개체에 대체된다. 대체 결과, 호스트-개체군의 유용한 스키마는 증가하고 유용하지 않은 스키마는 감소한다. 이때, 원래의 호스트-개체군에 존재하지 않는 새로운 스키마가 추가될 수 있다.

Step 5: 기생-개체군이 유용한 스키마를 찾는 역할을 하기 때문에 스키마 y 의 적합도 F_y 는 (3)과 같이 적합도 개선 값의 합으로 정의된다.

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

표 1. 기생적 알고리즘의 비교.

Table 1. Comparison in the parasitic algorithms.

	VEGA	Handa의 공진화	SCEA
유사성	호스트와 두 번째(바이러스 또는 기생)개체군으로 구성. 두 번째 개체군이 호스트 개체군에 효율적인 스키마를 제공.		
차이점	호스트 개체에서 다른 호스트 개체로 스키마가 수평적으로 전파	기생 GA 과정이 SGA와 유사하게 수행. 유용한 스키마가 확률적으로 공급됨.	기생 프로세스에서 큰 다양성 확보. 각 기생 프로세스에서 최대 스키마가 결정적으로 공급됨.
자연 진화에 근접도	가장 가까움	많은 부분에서 일치	기본적인 구조에서 유사
특징	GA 연산자에 민감. 많은 파라미터 조정이 필요	기계 학습에 응용됨	많은 지역 최적해가 존재하는 GA로 풀기 어려운 문제에 적용됨.

(3)은 기생-개체군에서 스키마의 적합도가 호스트-개체군에서 대체된 스트링에 의존하고 있는 것을 의미한다.

III. 기생적 공진화 알고리즘 비교

기생적 공진화에서, 하나의 개체군은 최적해의 후보군을 갖고 있고 다른 개체군은 유용한 스키마를 공급해주는 역할을 한다. 대표적인 기생적 공진화 알고리즘은 바이러스 진화 유전 알고리즘(Virus-Evolutionary Genetic Algorithm: VEGA), Handa의 공진화(coevolution mechanism of Handa), 스키마 공진화 알고리즘(schema co-evolutionary algorithm: SCEA)이 있다 [4][5][6]. 비례 선택 기법은 유용한 스키마와 비효율적인 스키마를 동시에 증가시키기 때문에, 이러한 기생적 공진화 알고리즘이 제안되었다[4]. 이 세 알고리즘은 유용한 스키마를 공급한다는 면에서는 유사하지만, 몇 가지 다른 특징을 갖는다.

VEGA는 진화의 바이러스 이론에 기반하고 있다. 두 가지 중요한 연산자로서, reverse transcription과 transduction이 있다.

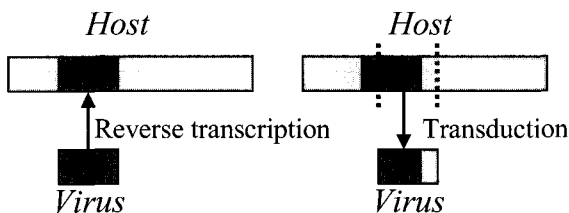


그림 4. VEGA의 주요 연산자.
Fig. 4. Main operators of the VEGA.

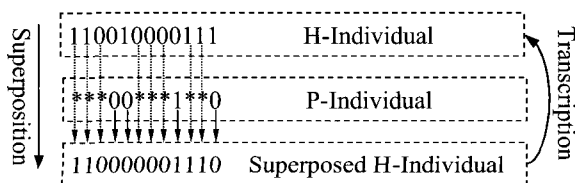


그림 5. Handa의 공진화의 주요 연산자.
Fig. 5. Main operators of the co-evolution of Handa.

바이러스 transduction은 중간에 DNA의 일부를 전달하는 주요 메커니즘이다. VEGA는 호스트-개체군과 이 개체군의 부분 스트링 집합으로 만들어진 바이러스-개체군으로 구성된다. 호스트 개체의 부분 스트링으로 생성된 바이러스는 다른 호스트 개체의 스트링에 복제된다. 그리고 바이러스는 어떤 호스트 개체로부터 유용한 스키마를 다른 호스트 개체로 수평적으로 전달하기 위해 부분 스트링을 복사해온다. 이러한 바이러스 감염 연산이 그림 4에 나타나 있다.

Handa가 제안한 공진화 알고리즘은 호스트 GA와 기생 GA로 구성된다. 기생 GA는 호스트 GA에 있는 유용한 스키마를 찾는 역할을 한다. 이 알고리즘의 두 가지 중요한 연산자로서, superposition과 transcription이 있다. 기생 개체의 유전자를 복사된 호스트 개체의 유전자로 복사하고 원래 호스트 개체를 확률적으로 대체함으로써, 기생 개체의 유전 정보가 호스트 GA 개체군으로 복사된다. 이러한 연산과정이 그림 5에 나타나 있다.

VEGA는 바이러스 감염에 의해 한 호스트 개체로부터 다른 호스트 개체로 유용한 스키마를 전달하고, Handa의 공진화는 복사된 호스트 개체에 기생 개체의 유전 정보를 확률적으로 공급한다. 하지만, SCEA는 최대 개선값을 갖는 스키마가 존재하면, 그것을 호스트 개체로 공급한다. 게다가, SCEA는 기생-개체군에의 주요 연산자가 mutation과 replacement이기 때문에, 유용한 스키마를 찾기 위한 다양성을 보장한다. 그러므로, VEGA는 세 알고리즘 중에서 가장 자연계에 가까운 알고리즘이고, SCEA는 기생-개체군에서의 결정적 대체와 큰 돌연변이 확률 때문에 실제 생물의 진화와는 가장 거리가 멀지만, 가장 강력하고 빠른 알고리즘이다. 세 알고리즘에 대한 비교가 표 1에 요약되어 있다.

IV. 실험 및 결과

1. 퍼지 제어기 설계

본 논문에서는 퍼지 제어기에서 그림 6과 같이 세 개의 멤버십 함수를 갖도록 설계한다. 또한 퍼지 추론 엔진에서는 TSK model의 simplified method를 사용한다. 이 방식의 장점은 후건부가 상수로 주어지며, 퍼지 디코더를 포함하고 있어서 전체 퍼지 제어기의 구조가 간단해진다는 것이다[9].

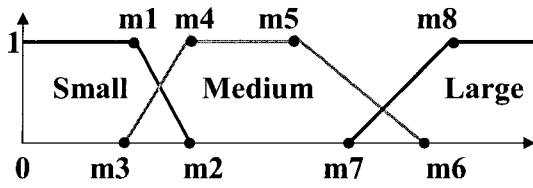


그림 6. 퍼지 제어기의 멤버쉽 함수.
Fig. 6. Membership function in the fuzzy controller.

그림 6에서 알 수 있듯이, 세 개의 멤버쉽 함수는 8개의 파라미터가 필요하다; $m1\sim m8$. 세 개의 입력에 대해 세 개의 멤버쉽 함수를 가지므로, 퍼지 제어기는 27개의 퍼지 룰을 가지며, 마찬가지로 27개의 후건부 상수가 필요하다: $c1\sim c27$. 따라서 총 35개의 파라미터가 결정되어야 한다. 하나의 파라미터를 6비트로 잡으면 거리는 64cm까지를 범위로 하고 로봇의 방향 회전 값은 $\pm 10^\circ$ 를 오차로 했을 경우에는 충분히 모든 방향에 대해 제어가 가능하다. 따라서 진화 연산을 적용하기 위해 필요한 비트 스트링은 210비트가 된다. 즉, 호스트 개체군에서는 모두 210비트 길이를 갖는 개체를 진화시킨다. 본 적용에서는 구현을 간단히 하기 위해, 후건부 상수 값을 로봇의 방향 회전 값으로 할당하였다. 즉 로봇은 일정한 속도로 움직이며, 방향 회전 만으로 장애물을 회피하게 된다.

진화 연산에서 또 하나 중요한 것은 적합도 함수를 설정하는 것이고, 이것은 적용하는 문제에 따라 달라진다. 본 연구에서는 (4)와 같은 적합도 함수를 사용하였다.

$$f = \frac{pos}{11} \times \frac{1 - close}{50} \times \frac{15}{time} \quad (4)$$

로봇이 움직일 거리를 11 부분으로 나누고 시작점은 0점을 도착점은 11점을 주도록 하였다. 'pos'는 이러한 로봇의 움직인 거리를 나타낸다. 또한 로봇이 벽에 너무 가까이 다가가지 않도록 하기 위해 'close'항을 추가하였다. 즉 로봇이 장애물에 5cm 이내로 접근한 횟수를 측정하여 'close'로 값을 준다. 즉 장애물이 가까이 가지 않은 로봇은 'close'값이 0이 되어 높은 적합도를 갖게 된다. 또한 총 주행시간을 적합도 함수에 반영하여 로봇이 장애물에 가까이 가지 않고 도착점까지 갔더라도 더 빨리 주행한 로봇에게 높은 적합도를 부여한다. 물론 로봇이 끝까지 주행하지 않은 경우에는 'time'은 가장 큰 값(20)을 주어 도착점에 가지 않고 이미 주행이 끝난 로봇에 대해서는 낮은 적합도를 부여하도록 하였다. 여기서 15는 전체 적합도의 최대치를 1로 맞추기 위한 스케일링 값이다. 결과적으로, 장애물에 너무 가까이 가지 않으면서 빠른 시간에 도착점까지 충돌 없이 주행한 로봇이 최고의 적합도를 부여 받는다. 물론 장애물의 충돌이 있을 경우에는 로봇의 주행이 바로 정지되며, 적합도는 0을 받는다.

2. 비교 실험

SCEA의 효율성을 입증하기 위해, 세 개의 기생적 공진화 알고리즘(VEGA, Handa의 공진화, SCEA)을 비교하였다. 각각의 실험을 위한 파라미터는 표 2에 나타나있다. 여기에서 p_c

표 2. 각 알고리즘의 파라미터.

Table 2. Parameters of each algorithm for the experiments.

	VEGA	Handa	SCEA
호스트 개체군	100	40	40
기생 개체군	(10) 바이러스	30	30
샘플링 크기 n	-	2	2
호스트 p_c	0.3, 0.5, 0.7	0.3, 0.5, 0.7	0.3, 0.5, 0.7
호스트 p_m	0.001, 0.01, 0.1	0.001, 0.01, 0.1	0.001, 0.01, 0.1
기생 p_c	-	호스트와 동일	0.05
기생 p_m	-	호스트와 동일	0.1
기타	[4]참고	-	-

표 3. 최적해를 찾은 회수와 평균 세대수 (a)VEGA (b)Handa의 공진화 (c)SCEA.

Table 3. The number and the average generation of finding the best solution (a)VEGA (b)method of Handa (c)SCEA.

p_c	p_m	횟수			평균 세대수		
		(a)	(b)	(c)	(a)	(b)	(c)
0.3	0.001	21	22	97	28.92	53.77	30.42
	0.01	0	66	99	-	54.35	31.60
	0.1	100	44	97	21.26	68.14	40.98
0.5	0.001	1	25	87	43.51	56.68	32.82
	0.01	86	71	99	52.60	46.18	24.93
	0.1	100	71	96	17.10	62.20	47.49
0.7	0.001	0	34	93	-	45	31.17
	0.01	0	76	100	-	48.57	23.31
	0.1	99	74	100	33.56	55.76	49.27

는 교차 확률, p_m 은 돌연변이 확률을 나타낸다. 기생-개체군에 대한 호스트-개체군에 대한 비율과 샘플링 크기는 경험적으로 결정된다. 초기 기생-개체군에 대해서는 '0', '1', '*'를 같은 확률로 임의로 생성하였다.

진화 알고리즘의 작동은 확률적이기 때문에, 그 성능은 실행할 때마다 변한다. 그러므로, GA 파라미터(p_c 와 p_m)의 조합에 따라 각각의 진화 알고리즘에 대해 100번씩 반복하였고, 진화는 150세대까지 수행하였다. 세 알고리즘을 이용하여 퍼지 제어기 설계한 결과가 표 3에 나타나있다. 최적해는 적합도가 0.95이상인 경우에 해당한다.

표 3의 결과는 SCEA가 같은 조건에서 다른 알고리즘보다 최적해를 더 잘 찾는 것을 볼 수 있다. 그 이유는 기생 개체

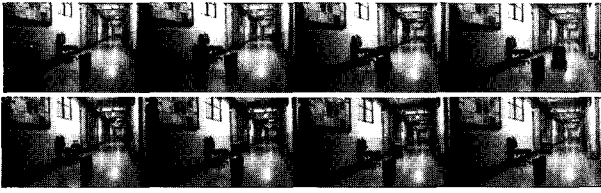


그림 7. SCEA에 의해 설계된 퍼지 제어기를 장착한 이동 로봇의 장애물 회피 실험.

Fig. 7. Experiment on the avoidance of obstacles by a mobile robot with fuzzy controller that is designed by SCEA.

군의 강한 돌연변이가 국소해를 쉽게 탈출할 수 있도록 다양성을 확보해주기 때문이다. VEGA는 호스트의 돌연변이 확률에 매우 큰 영향을 받아서, 0.1 p_m 값에서 가장 많이 최적해를 찾았다. 그러나 그 외의 p_m 값에 대해서는 최적해를 거의 찾지 못했다. VEGA는 국소해에 빠지기 쉬운 경향을 띠고, 높은 돌연변이 없이는 국소해를 탈출하기 힘들다는 것을 알 수 있다. 즉, 그것은 GA 파라미터에 매우 민감하다. 또한 VEGA는 호스트 개체군이 SSGA(steady state GA)를 사용하기 때문에, 한 세대의 진화시간이 다른 알고리즘에 비해 두 배 이상 오래 걸리는 단점이 있었다. 이와는 달리, SCEA와 Handa의 알고리즘은 GA 파라미터의 영향을 잘 받지 않았고, 특히 SCEA는 호스트 개체군의 GA 파라미터 영향을 거의 받지 않는 것을 알 수 있다.

퍼지 규칙은 매우 불규칙적인 적합도 환경을 갖기 때문에, 다른 알고리즘은 최적해를 찾지 못하고 국소해에 자주 빠진다. 그러나 SCEA는 호스트 개체군의 적합도를 높이는 우수한 스키마를 결정적인 방식에 의해 공급하고 기생-개체군의 다양성을 보장하고 있기 때문에, 국소해를 탈출하기 쉽다. 따라서 시뮬레이션의 결과로부터 SCEA가 다른 알고리즘보다 더 최적해를 잘 찾는 것을 알 수 있다.

3. 설계된 퍼지 제어기의 로봇 실험

위의 SCEA 시뮬레이션에 의해 설계된 퍼지 제어기를 이동 로봇에 장착한 후에, 장애물 회피 실험을 수행하였다. 그림 7은 이러한 실험 모습을 보여준다. 이 그림에서 볼 수 있듯이, 로봇은 장애물을 피하면서 벽을 잘 따라가는 것을 볼 수 있다. 이것은 SCEA에서 벽을 따라 가는 것이 높은 적합도를 갖는 개체로 선택되었기 때문이다.

V. 결론

전통적인 하나의 개체군 기반의 진화 알고리즘과는 대조적으로, 두 개의 개체군이 지속적으로 서로 영향을 미치면서

서로 협조하는 기생적 공진화가 임의의 문제를 더 신뢰적으로 잘 해결한다. 또한 기생적 공진화는 기생-개체군이 스키마 공간을 탐색하고 유용한 스키마를 공급하기 때문에 최적해를 찾을 기회를 더 많이 제공한다. 특히 스키마 공진화 알고리즘(SCEA)은 바이러스-진화 유전 알고리즘(VEGA)이나 Handa의 공진화 알고리즘보다 더 다양한 스키마 탐색 공간을 제공하였다. 그것은 SCEA가 다른 알고리즘보다 더 빠르고 강인하게 최적해를 찾을 수 있는 기회를 갖게 하였다. 요약하면, VEGA는 속도는 느리지만 특정한 조건에서 최적해를 매우 잘 찾을 수 있으나, 일반적인 조건에서는 국소해에 잘 빠진다. 반면에 SCEA는 파라미터 조건에 상관없이 최적해를 매우 잘 찾고, 최적해를 찾는 속도가 가장 빠르다. Handa의 공진화는 그 중간 정도의 성능을 보인다.

본 논문에서는 이동 로봇의 간단한 행동을 위한 퍼지 제어기에서 퍼지 규칙을 설계하는데 SCEA를 적용하였다. 그러나 SCEA의 강인성 때문에, SCEA는 다양한 퍼지 규칙 설계 뿐만 아니라 기존의 GA로는 풀기 어려운 최적화 문제에 적용될 수 있을 것으로 기대된다.

참고문헌

- [1] 심귀보, 전호병, "스키마 공진화 알고리즘의 해석," 한국 퍼지 및 지능시스템학회 논문지, vol. 10, no. 1, pp. 23-27, Feb. 2000.
- [2] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Third Edition, Springer-Verlag, NY, 1995.
- [3] 문병로, 유전 알고리즘, 두양사, 서울, 2003.
- [4] N. Kubota, K. Shimojima, T. Fukuda, "The role of virus infection in virus-evolutionary genetic algorithm," *Proc. of IEEE Int. Conf. on Evolutionary Computation*, pp. 182-187, May 1996.
- [5] H. Handa, N. Baba, O. Katai, T. Sawaragi, T. Horiuchi, "Genetic algorithm involving coevolution mechanism to search for effective genetic information," *Proc. of IEEE Int. Conf. on Evolutionary Computation*, pp. 709-714, Apr. 1997.
- [6] K. B. Sim, D. W. Lee, "Schema co-evolutionary algorithm (SCEA)," *IEICE Trans. on Information & Systems*, vol. E87-D, no. 2, pp. 416-425, Feb. 2004.
- [7] K. B. Sim, H. B. Jun, "Co-evolutionary algorithm and extended schema theorem," *J. of the Korea Society for Industrial and Applied Mathematics*, vol. 2, no. 1, pp. 95-110, June 1998.
- [8] M. Mitchell, *An Introduction to Genetic Algorithm*, Bradford Book, MA, 1996.
- [9] 강훈, 심귀보, 지능 정보 시스템, 브레인코리아, 서울, 2003.



심귀보

1984년 중앙대학교 전자공학과(공학사).
 1986년 동대학원 전자공학과(공학석사).
 1990년 The University of Tokyo 전자공학과(공학박사). 1991년~현재 중앙대학교 전자전기공학부 교수. 2000년~현재 제어·자동화시스템공학회 이사 및 지능시스템연구회 회장. 2002년~현재 중앙대학교 산학연컨소시엄센터 센터장. 2003년~현재 일본계측자동제어학회(SICE) 이사. 2003년~현재 한국퍼지 및 지능시스템학회 부회장. 관심분야는 인공지능, 지능로봇, 지능시스템, 다개체시스템, 학습 및 적응알고리즘, 소프트 컴퓨팅(신경망, 퍼지, 진화연산), 인공면역시스템, 침입탐지시스템, 진화하드웨어, 인공두뇌, 지능형 홈 및 홈네트워킹, 유비쿼터스 컴퓨팅 등.



변광섭

2003년 중앙대학교 전자전기공학부(공학사). 2003년~현재 중앙대학교 전자전기공학부 대학원 석사과정. 관심분야는 지능 로봇, 지능 시스템, 퍼지 시스템, 공진화 알고리즘, 원격 제어.