

論文95-32B-12-15

유전자 집단의 크기 조절을 통한 Genetic Algorithm의 조기 포화 방지

(Preventing Premature Convergence in Genetic Algorithms with Adaptive Population Size)

朴來正*, 朴哲薰*

(Lae-Jeong Park and Cheol Hoon Park)

요 약

GA는 자연계의 적자 생존과 유전학의 이론에 바탕을 둔 효과적인 찾기 알고리즘으로 최근 많은 응용 분야에서 사용되고 있다. 그러나, 유전자 집단(population)내의 유전자(gene)가 충분하지 않을 경우에 종종 최적해를 찾기전에 유전자 집단내의 개체(chromosome)가 서로 유사해지는, 조기 포화현상을 보인다. 본 본문에서는 변이에 의한 새로운 개체를 진화 도중 유전자 집단의 크기의 변화를 허용하여 유전자 집단에 보충함으로써, 다양한 유전자를 유지하는 방법을 제안한다. 시뮬레이션을 통하여 유전자 집단의 크기가 작고 개체의 길이가 길지 않을 경우에 제안된 방법이 효과적임을 보인다.

Abstract

GAs, effective stochastic search algorithms based on the model of natural evolution and genetics, have been successfully applied to various optimization problems. When population size is not large, GAs often suffer from the phenomenon of premature convergence in which all chromosomes in the population lose the diversity of genes before they find the optimal solution. In this paper, we propose that a new heuristic that maintains the diversity of genes by adding some chromosomes with random mutation and selective mutation into population during evolution. And population size changes dynamically with supplement of new chromosomes. Experimental results for several test functions show that when population size is rather small and the length of chromosome is not long, this method is effective.

I. 서 론

유전 알고리즘(GA)은 J. Holland에 의해 개발된 효과적인 찾기(search) 알고리즘으로, 최근 최적화 문제뿐만 아니라 여러 분야에서 사용되고 있다. GA는 자연계의 진화 과정의 적자 생존과 유전적 부호화와 처리 과정을 모델링함으로써, 해석적인 방법으로 해결

하기 힘든 문제, 함수 최적화나 조합(combinatorial) 최적화 문제에 널리 사용되고 있다.

GA의 조기 포화현상(premature convergence)은 최적해를 찾기전에 유전자 집단내의 개체가 최적해를 찾기위해 필요한 일부 유전자들을 잃어버림으로써 최적해를 찾을 수 없는 현상인데, 오랫동안 GA의 난제로 인식되어 왔다. 유전자 집단의 크기가 제한되어 있으므로, 즉, 각 진화(evolution)에서 사용되는 유전자의 개수가 제한되어 있으므로 GA가 진화되어 감에 따라 자연히 선택과정(selection)에서 높은 적합도(fi-

* 正會員, 韓國科學 技術院 電氣 및 電子工學科

(Dept. of Electrical Engineering, KAIST)

接受日字: 1994年10月5日, 수정완료일: 1995年12月5日

tness)를 가진 개체내의 유전자들이 유전자 집단에서 많아지게 되며, 낮은 적합도를 가진 개체내의 유전자들은 선택과정에서 점차 도태되어 진다(schemata 정리^[11]).

특히, 주어진 최적화 문제의 복잡도에 비해 유전자 집단의 크기가 작을 경우에는 진화 도중에 최적해를 찾기위한 충분한 유전자들을 유지하는 것이 더욱 어렵다. 상당히 큰 크기의 유전자 집단을 사용할 경우에는 진화도중에 최적해를 찾기 위한 충분한 유전자를 유지할 수도 있다. 그러나, 유전자 집단의 크기가 클 경우에 수렴 속도가 늦은 경향을 나타내며, 상당히 큰 유전자 집단을 사용하기 어려운 경우에는 다른 효과적인 방법이 필요하다. 즉, 적절한 유전자 집단의 크기를 사용할 경우에 GA가 최적해를 찾는 과정에 크게 영향을 주지 않으면서, 효과적으로 유전자 집단내의 다양한 유전자를 유지하는 것이 중요하다.

조기 포화는 GA의 성능을 저하시키는 중요한 현상이므로, 조기 포화를 방지하기 위한 heuristic에 대해 많은 연구가 있었다.

번식(reproduction)과 대체(replacement) 선택 정도를 변화시켜서 다양성을 유지하거나(예를 들면, 선택될 확률을 개체의 적합도에 비례하지 않게 하는 방법)^[21], 균일 상호교체(uniform crossover)와 같이 더욱 disruptive한 연산자를 사용하여 다양성을 유지하거나^[2,3], 번식을 위한 선택과정에서 짝짓기(mating)에 제약을 가하는 방법(incest prevention)^[3], 대체 과정에서 동일한 개체의 배제(duplicate exclusion)^[3,4] 등의 여러 방법들이 제안되었다. 또한, 여러개의 소유전자 집단(subpopulation)으로 나누어 국소적인 짝짓기와 선택을 통해서 다양성을 유지할 수 있다^[5].

GA의 여러 변수(parameter)를 조절함으로써, 조기 포화를 방지하는 여러 방법이 가능하나, 본 논문에서는 유전자 집단이 작을 때에 변이(mutation)가 조기 포화를 방지하는 한 방법임을 보이고, 이를 효과적으로 이용하기 위한 방법으로 진화도중에 변이에 의한 새로운 개체를 유전자 집단에 삽입하는 방법을 제안한다^[6].

다음 장에서 일반적인 GA에 대하여 설명하고, 3장에서 제안하는 유전자 보충 방법에 대하여 설명한다. 4장에서 몇 가지 표준 예제를 통한 시뮬레이션 결과를 보이고, 마지막으로 결론을 맺는다.

II. 유전 알고리즘(Genetic Algorithms)

그림 1은 일반적인 GA의 개념적인 구조를 나타낸다. 유전자 집단의 개체들은 부호화된(coded) 여러 해를 나타내며, 새로운 해를 생산하기 위하여 유전자 집단에서 두 개체를 선택한다. 선택된 두 개체에 유전 연산자를 이용하여 새로운 개체를 생산하고 부호화(decoding)를 통하여 평가(evaluatoin)한 후, 만들어진 개체와 현재의 유전자 집단내의 개체중에서 선택적으로 일정한 갯수의 개체를 선택하고, 다음 진화과정을 위한 유전자 집단을 형성한다. 이와 같은 한 단위의 동작을 최적 해를 찾을 때까지 반복적으로 수행한다.

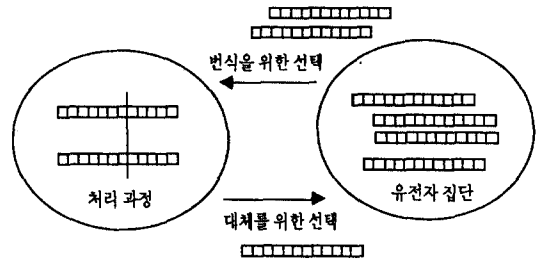


그림 1. 일반적인 GA의 개념적인 구조

Fig. 1. A conceptual diagram for Genetic Algorithms.

GA의 중요한 요소는 사용하는 부호화(coding), 번식과 대체를 위한 선택과정과 유전 연산자, 적합도 평가 방법이라 할 수 있다. 사용하는 부호화는 문제에 따라 다르나, 대부분 단순한 0과 1로 구성한다. 번식과 대체 과정에서 각각 offspring을 만들 개체와 다음 유전자 집단에서 살아남을 개체의 선택이 일어난다. 일반적으로, 각 개체의 적합도에 비례하게 확률적으로 선택한다. 선택하는 경향(selection pressure)에 따라 GA의 성능이 달라진다. 만약 높은 적합도를 가진 개체를 선택하는 정도를 높일 수록, 수렴 속도는 빠르지만 찾은 해가 국소해(local minimum)일 가능성이 많고, 선택정도를 낮추면 랜덤하게 해를 찾는 경우와 다름없어진다. 따라서, 주어진 문제에 따라 적절한 선택의 정도를 유지하는 것이 중요하다.

GA의 유전 연산자로서 상호교체(crossover), 변이(mutation), 반전(inversion)등이 있는데, 그림 2는 0과 1로 구성된 개체인 경우에 대한 각각의 수행과정을 나타낸다. 상호교체로 인해 GA는 한꺼번에 여러 유전자를 처리함으로써, 은닉 병렬성(implicit para-

lletism)^{11,21}이라는 특징을 가지고 있으며, 변이는 진화도중에 새로운 유전자를 생성하는 역할을 한다. 전통적인 GA에서는 주로 상호교체가 주된 연산자이며, 변이는 백그라운드 연산자로 사용된다.

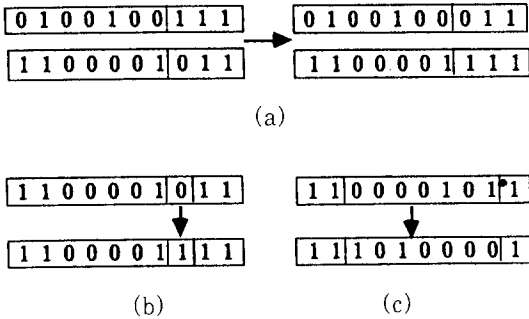


그림 2. GA의 연산자: (a) 상호교체 (b) 변이 (c) 반전

Fig. 2. Genetic operators: (a) Crossover (b) Mutation (c) Inversion.

개체의 적합도 평가는 우선 부호화된 개체의 복호화를 통해 구해진 해(변수)를 문제에 따른 비용 함수(cost function)에 의해 평가하는 것으로, 이를 이용하여 유전자 집단내에서 각 개체의 순위를 고려하여 개체 선택이 이루어진다.

III. 변이를 이용한 유전자 보충

조기 포화를 방지하기 위해서는 해를 찾는 공간을 빨리 줄이지 않아야 하며, 효과적으로 점차 찾는 공간을 넓혀나가야 한다¹⁷⁾. 따라서, 유전자 집단이 작을 때는 조기포화 현상이 유전자 집단이 클 경우에 비해 상대적으로 빨리 그리고 빈번히 발생하며, 따라서 상호교체 연산자만으로 좋은 해를 찾기란 매우 어렵다. 본문에서 이러한 상황에서 진화도중 유전자집단내의 유전자의 손실을 감시(monitor)하여 유전자의 손실이 발생하였을 경우에 새로운 개체를 유전자 집단에 보충함으로써 유전자의 다양성을 유지하게 하는 방법을 제시한다. 보충되는 개체는 유전자 집단내의 상위 적합도를 가지는 개체와 하위 적합도를 가지는 개체중에서 선택된 후 변이과정을 통해 만들어진다. 하위 적합도를 가지는 개체는 비록 적합도는 떨어지나, 새로운 찾기 공간을 제공할 가능성이 있다.

위의 방법에는 여러 가지 조건의 선택이 가능한데, 하나는 보충되는 개체를 현재의 유전자 집단내의 어떤

개체를 이용하느냐에 관한 것이고, 다른 하나는 변이를 어떻게 선택된 개체에 적용하느냐이다. 완전히 랜덤하게 만들어진 개체의 보충도 가능하다, 보충된 개체가 유전자 집단내의 개체들에 비해 상대적으로 낮은 적합도로 인해, 대체를 위한 선택과정에서 즉시 도태되기 쉽다. 따라서 진화과정에서 유전자 집단내의 개체에 변이를 통하여 얻어진 개체를 사용하는 것이 합리적이다. 유전자 집단내의 개체를 이용하는 경우 상위 적합도를 가지는 개체만을 이용하는 경우와 하위 적합도를 가지는 개체를 이용하는 경우, 그리고 두 가지를 이용하는 경우 모두 가능하다. 함수 최적화의 시뮬레이션 결과 두 개체를 동시에 이용한 경우가 가장 좋은 결과를 나타내었다. 사용되는 개체는 유전자 집단내의 전체 개체의 적합도를 이용하여 전체의 상위와 하위 개체로 결정하며, 유전자 집단내의 개체들중 같은 유전자를 가지는 개체가 정해진 수준이상일 경우를 유전자의 손실로 간주한다. 이 방법은 유전자 집단의 크기가 작을수록 개체 보충시 해의 평가 횟수가 적으므로 효과적이다.

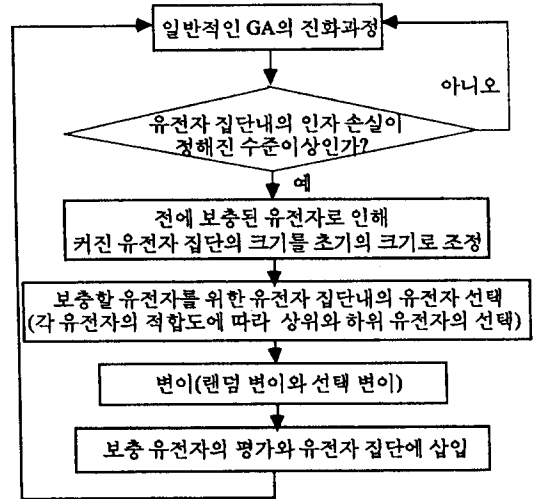


그림 3. 유전자 보충 알고리즘의 흐름도

Fig. 3. Flow chart of GA with supplement of new strings allowing for adaptive populatin size.

이 방법에서 사용하는 변이는 다음과 같다. 우선 유전자 집단의 손실된 유전자의 갯수까지의 랜덤 숫자를 이용하여 보충되는 개체의 변이될 유전자의 갯수(스트링에서 비꾸어야 하는 비트 수)를 결정하고, 정해진 갯수만큼 유전자를 랜덤하게 선택하여 유전자를 바꾼다 (0에서 1 혹은 1에서 0). 유전자를 바꿀 경우에 잃어

버린 유전자만을 회복하는 선택 변이(selective mutation)와 랜덤하게 바꾸는 랜덤 변이(random mutation)의 두 가지의 변이가 가능하다. 그림 3은 변이를 이용한 유전자 보충 알고리즘의 흐름도를 나타낸다.

IV. 시뮬레이션

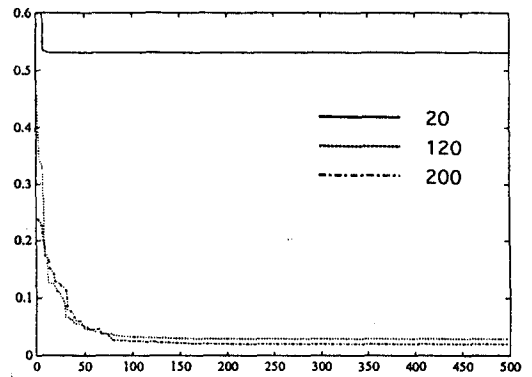
시뮬레이션에 사용한 그림 3의 GA의 진화과정은 다음과 같다: 1) 번식을 위한 선택은 linear-rank 방법을 사용하며, 2) 대체를 위한 선택과정은 한번에 한 개체를 생산해서 중복된 개체가 유전자 집단내에 존재하는지를 검사하고, 만약 없다면(중복된 개체가 존재한다면, 새로운 개체가 생산될 때까지 1000번의 시도를 하고, 1000번이상에도 중복된 개체가 있다면 그냥 삽입). 유전자 집단에 삽입하고 유전자 집단내에 최하의 적합도를 가지는 개체를 도태시킨다. 3) 유전 연산자는 상호 교체와 변이를 사용하며, 각각의 확률은 1과 0.05이다. 4) 이진 부호화(binary coding)를 사용하였으며 각 변수당 16비트를 사용하였다. 제안된 알고리즘은 위의 GA 알고리즘의 조건을 똑같이 사용하고, 추가적인 조건은 다음과 같다: 1) 10번의 진화마다 유전자의 손실을 감시한다. 2) 전체 개체의 80%가 같은 유전자를 가질때에를 유전자 손실로 간주한다. 3) 변이에 의해서 보충에 사용하는 개체는 유전자 집단내의 상위 10%와 하위 10%이다. 유전자 보충 알고리즘의 조기 포화를 방지하는 성능을 다음 4개의 함수 최적화(최소화)에 적용함으로써 살펴보았다(각 함수의 최소값은 각각 0, 0, 0, -1.002이다)¹²¹.

$$\begin{aligned}
 f_1(x) &= \sum_{i=1}^3 x_i^2 & -5.12 \leq x_i \leq 5.12 \\
 f_2(x) &= 100(x_1^2 - x_2^2)^2 & -2.048 \leq x_i \leq 2.048 \\
 f_3(x) &= \sum_{i=1}^{20} ix_i^4 + Gauss(0, 1) & -1.28 \leq x_i \leq 1.28 \\
 f_4(x) &= -0.002 - \frac{25}{j + \sum_{i=1}^j (x_i - a_{ij})^6} & -65.536 \leq x_i \leq 65.536
 \end{aligned}$$

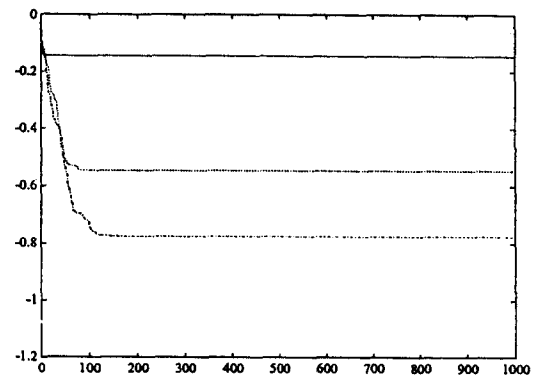
함수 1은 하나의 극소점을 가지는 단순한 함수이고, 함수3도 통계적인 항을 가지고 있으나 역시 한 차원으로는 단순 증가하는 함수이다. 함수 2는 nonconvex하고 함수 4는 여러 극소점을 갖는 함수로서 최적해(여기서는 최소점)를 찾기가 어렵다.

각 함수에 대해 유전자 집단의 크기를 변화시키면서 20번씩 시뮬레이션하여 일반적인 GA의 성능과 제안된

유전자 보충 알고리즘의 성능을 분석하고 비교하였다. 함수 1과 2는 각 5000번의 평가횟수까지, 함수 3과 4는 10000번의 평가횟수까지 시뮬레이션하였다. 변이 확률을 0으로 하였을 경우에는, 즉 상호교체만을 사용하였을 때에는 유전자 집단이 작은 경우에는 진화도중 급속도로 유전자를 잃어버림으로 인해 빨리 조기 포화되고, 성능도 상당히 좋지 않다. 그림 4는 함수 2와 4에서 유전자 집단의 크기를 20, 120, 200으로 변화시키면서 상호교체만을 사용하였을 때의 성능을 나타낸다.



(a)



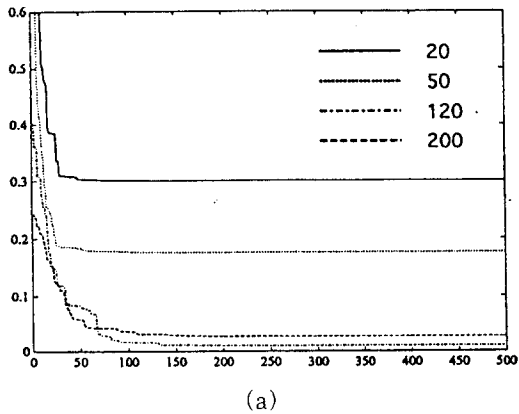
(b)

그림 4. 함수 2와 4에 변이확률이 0일 경우 GA의 성능 (a)함수 2 (b) 함수 4

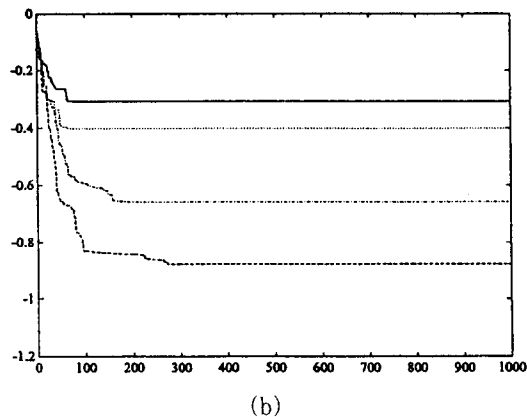
Fig. 4. Performance of general GA with zero mutation probability on F2 and F4. (a) F2; (b) F4.

함수 1과 3은 간단한 함수이기 때문에 유전자 집단의 크기의 증가에 영향을 받지 않는 성질을 보이며, 함수 1은 오히려 유전자 집단의 크기가 작을수록 빠른

수렴 속도를 보였다. 즉, 함수 1과 3과 같은 간단한 함수의 최적화에는 일반적인 GA만으로 최적해를 찾기에 충분하며, 항상 큰 유전자 집단을 필요로 하지 않는다. 그러나, 함수 2와 4는 작은 유전자 집단의 크기로는 좋은 해를 찾기가 어려우며, 좋은 해를 찾기 위해서는 큰 유전자 집단이 필수적임을 알 수 있다 [그림 4].



(a)

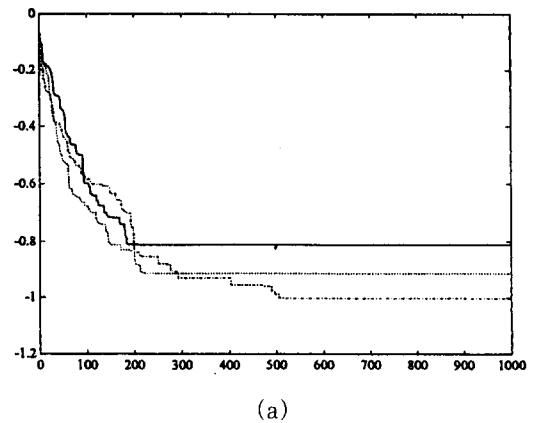


(b)

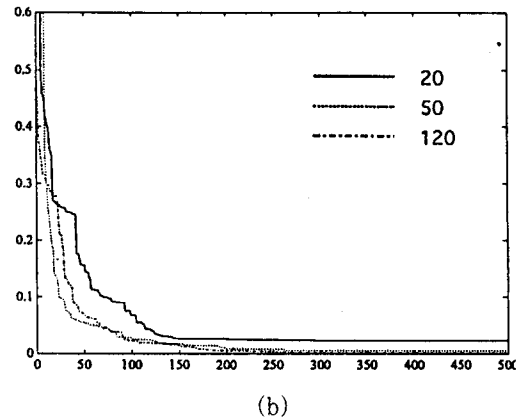
그림 5. 함수 2와 4에 변이확률이 0.05일 경우 GA의 성능 (a) 함수 2 (b) 함수 4
 Fig. 5. Performance of general GA with 0.05 mutation probability on F2 and F4. (a) F2; (b) F4.

그림 5에서 보듯이 함수 2와 4처럼 복잡한 경우에는 유전자 집단의 크기가 작을 경우에, 변이를 사용한 경우가 상호 교체만을 사용한 경우보다 확실히 좋은 성능을 보인다. 그러나, 유전자 집단의 크기가 점차 증가할수록 유전자 집단내의 충분한 유전자가 존재하게 되므로, 상호교체만으로 최적해를 찾을 수 있고, 오히려 높은 변이 확률(이 실험에서의 0.05는 높은 변이 확률

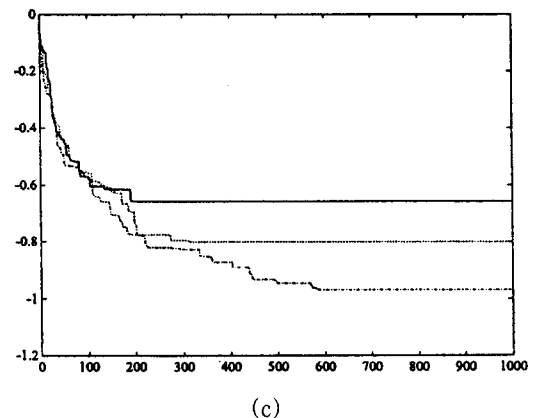
이다)이 성능을 저해할 가능성도 있다 [그림 5].



(a)



(b)



(c)

그림 6. 함수 2와 4에 유전자 보충 알고리즘을 사용한 경우 GA의 성능 (a) 선택 변이; 함수 2와 4. (b) 랜덤 변이; 함수 2와 4
 Fig. 6. Performance of general GA with supplement of new stings on F2 and F4 (a) Selective mutation on F2 and F4 (b) Random mutation on F2 and F4.

따라서, 주어진 문제의 복잡도(complexity)에 비해 유전자 집단이 작은 경우에는 변이가 효과적이며, 문제에 필요한 충분한 유전자를 유지할 만큼 유전자 집단의 크기가 큰 경우에는, 상호교체와 낮은 변이 확률로도 해를 찾을 수 있다.

제한된 유전자 보충 방법은 함수 1의 경우에는 일반적인 GA와 비슷한 수렴속도를 나타내며, 함수 3의 경우에는 일반적인 방법보다 유전자 보충시 부가적인 개체의 평가 횟수로 인해 수렴속도가 느다. 이러한 현상은 함수 3의 긴 개체의 길이(30개의 변수)로 인해 보충되는 개체의 영향이 성능을 촉진하지 못하고 오히려 평가 회수만을 늘리기 때문이다. 즉, 개체의 길이가 길수록, 짧은 경우보다 손실되는 유전자들의 갯수가 훨씬 많기 때문이다. 함수 2와 4에서 제안된 방법은 유전자 집단의 크기가 작을 수록 일반적인 방법보다 훨씬 좋은 성능을 보인다. 또한, 선택적 변이를 사용하였을 경우가 랜덤 변이를 사용한 경우보다 약간 좋은 성능을 보였다 [그림 6]. 유전자 보충 방법은 유전자 집단이 작고, 주어진 문제가 복잡하며 개체의 길이가 작을수록 최적해를 찾는 데 효과적이다.

V. 결론 및 고찰

GA의 조기 포화 현상을 막기 위한 하나의 방법으로 진화도중에 유전자 집단의 유전자 손실을 감시하여, 손실이 발생하였을 경우에 유전자 집단내의 적절한 개체에 변이를 가한 후 유전자 집단에 보충하는 새로운 heuristic을 제안하였다. 제안된 heuristic은 특히, 유전자 집단의 크기가 작고 함수가 복잡한 경우에 좋은 성능을 보였다. 랜덤 변이보다 진화과정에 손실된 유전자를 이용하는 선택적 변이를 사용함으로써, 약간 더 좋은 해를 찾을 수 있었다. 또 다른 하나의 방법으로서 조기 포화 현상이 일어났을 경우에 유전자 집단

을 부분적 초기화를 하는 방법과의 비교와, 이 방법을 유전자 보충 알고리즘과의 병행하는 경우에 대한 연구가 진행중이다. 또한, 함수 최적화뿐만 아니라 다양한 문제에서의 적용을 통한 제안된 알고리즘의 분석과 고찰이 필요하다.

참 고 문 헌

- [1] J. H. Holland, *Adaptation in natural and artificial systems*, MIT Press, 1992.
- [2] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, 1989.
- [3] L. J. Eshelman and J. D. Schaffer, "Preventing premature convergence in genetic algorithms by incest prevention," *Procs. of the Fourth Int. Conf. on Genetic Algorithms*, pp. 115-122, 1991.
- [4] D. Lawrence, *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York, 1991.
- [5] H. Muhlenbein, M. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," *Parallel Computing*, vol. 17 pp. 619-632, 1991.
- [6] 박 래정, 박 철훈, "Genetic Algorithm에서 조기 포화를 막는 방법," 제 3회 인공지능, 신경망 및 퍼지 시스템 종합학술대회 논문집, pp. 55-59, 1993
- [7] 박 철훈, 박 래정, "Genetic Algorithm을 이용한 Traveling Salesman Problem 해법," 1992년도 신호처리 종합학술대회 논문집, pp. 590-593, 1992

— 저 자 소 개 —



朴來正(正會員)

1968년 12월 29일생. 1991년 2월 서울대학교 전기공학과 졸업. 1993년 2월 한국과학기술원 전기 및 전자공학과(석사). 현재 한국과학기술원 전기 및 전자공학과 박사과정 재학중.

관심분야는 최적화 기법과 유전 알고리즘등임

朴哲薰(正會員)

1961년 4월 14일생. 1984년 2월 서울대학교 전자공학과 졸업. 1990년 미국 California Institute of Technology 공학 석사 및 박사 학위 취득. 현재 한국과학기술원 전기 및 전자공학과 조교수. 관심분야는 패턴 인식, 음성 인식, 신경망 제어, 유전 알고리즘등임