

퍼지 클러스터링 기반의 국소평가 유전자 알고리즘 (Partially Evaluated Genetic Algorithm based on Fuzzy Clustering)

유 시 호 [†] 조 성 배 ^{**}

(Si-Ho Yoo) (Sung-Bae Cho)

요 약 유전자 알고리즘은 원하는 최적해를 찾기 위해서 개체 집단의 크기를 가능한 크게 유지하여야 한다. 하지만 실제 문제에서 개체의 적합도를 평가하는 것이 어려운 경우가 많기 때문에 큰 집단의 모든 개체에 대하여 적합도를 평가하는 것은 많은 시간과 비용을 요구한다. 이에 본 논문에서는 집단의 크기를 크게 유지하되 클러스터링에 의해 대표 개체만을 평가함으로써 효율을 높이는 퍼지 클러스터링 기반의 국소 평가 유전자 알고리즘을 제안한다. 나머지 개체들은 대표 개체로부터 간접적으로 적합도를 분배받는다. 다수의 집단에 소속되는 개체들의 경우, 하드 클러스터링 방법으로는 정확한 적합도 분배를 하기 어렵기 때문에 퍼지 c-means 알고리즘을 사용하였고, 클러스터 결과인 퍼지 소속 행렬에 의해 적합도를 배분하였다. 9개의 벤치마크 적합도 함수에 대하여 6가지 하드 클러스터링 알고리즘을 적용한 유클리디안 거리와 피어슨 상관계수에 의한 적합도 배분 방법과 본 논문에서 제안하는 방법을 비교 실험한 결과, 제안한 방법의 우수한 성능을 확인할 수 있었다.

키워드 : 유전자 알고리즘, 국소 평가 유전자 알고리즘, 적합도 배분, 하드 클러스터링, 퍼지 c-means 알고리즘

Abstract To find an optimal solution with genetic algorithm, it is desirable to maintain the population size as large as possible. In some cases, however, the cost to evaluate each individual is relatively high and it is difficult to maintain large population. To solve this problem we propose a novel genetic algorithm based on fuzzy clustering, which considerably reduces evaluation number without any significant loss of its performance by evaluating only one representative for each cluster. The fitness values of other individuals are estimated from the representative fitness values indirectly. We have used fuzzy c-means algorithm and distributed the fitness using membership matrix, since it is hard to distribute precise fitness values by hard clustering method to individuals which belong to multiple groups. Nine benchmark functions have been investigated and the results are compared to six hard clustering algorithms with Euclidean distance and Pearson correlation coefficients as fitness distribution method.

Key words : genetic algorithm, partially evaluated genetic algorithm, fitness distribution, hard clustering, fuzzy c-means algorithm

1. 서 론

유전자 알고리즘은 적자생존, 유전자 교차, 돌연변이 등 자연의 진화 메커니즘에 기반하여 문제를 해결하고자 하는 알고리즘이다[1]. 즉, 다수의 개체들 중에서 뛰어난 생존 능력을 가진 개체가 많은 자손을 퍼뜨리고 많은 세대를 거치면서 각 세대의 평균적인 생존 능력이

진화되어 나간다는 점에 착안하였다. 이 방법은 문제를 풀기위한 해결방법을 염색체로 코딩한 후 다수의 해 후보를 사용한다는 점에서 다른 탐색이나 최적화 방법과 다르다. 즉, 집단 탐색방법의 사용, 적합도 함수에 의해 얻어지는 비용정보를 이용한 탐색, 결정론적인 규칙 없이 확률 연산자를 이용한 탐색 등이 유전자 알고리즘의 특징이라 할 수 있다. 이런 특징들로 인해 다른 탐색 또는 최적화 방법에 비해 전역적인 해를 구할 가능성이 높다[2]. 그러나 상대적으로 작은 크기의 집단을 사용하면 지역적인 탐색에 의한 지역해에 빠질 가능성이 커지므로 집단의 크기를 가능한 크게 유지하여야 한다.

하지만 응용 분야에서 개체의 적합도를 평가할 때 큰

· 이 연구는 과학기술부가 지원한 뇌과학연구프로그램에 의해 지원되었음

† 학생회원 : 연세대학교 컴퓨터과학과

bonanza@sclab.yonsei.ac.kr

** 종신회원 : 연세대학교 컴퓨터과학과 교수

sbcho@cs.yonsei.ac.kr

논문접수 : 2004년 3월 11일

심사완료 : 2004년 6월 17일

비용이 소요되어 큰 집단을 유지하기가 힘든 경우가 있다. 예를 들어 사람의 감성을 적합도로 평가하고자 하는 대화형 유전자 알고리즘의 경우, 적합도 평가 함수를 정의하기가 어렵고 집단의 크기가 커지면 사람에 의한 평가가 불가능해진다[3]. 공학의 역 문제(inverse problem)를 유전자 알고리즘을 이용하여 풀고자 할 때도 적합도를 평가하는 데 드는 시간과 비용이 크다는 문제가 있다[4,5]. 이러한 문제의 경우 최적해를 찾기 위해 집단의 크기를 크게 유지하는 것은 실질적으로 불가능하다. 이를 해결하기 위하여 다양한 방법들이 연구되고 있다. 일부 집단의 평가는 컴퓨터가 처리하고 나머지 남은 집단에 대해서만 사람이 직접 평가하는 변형된 대화형 유전자 알고리즘이 제안되었고[6], 소수의 개체에 대해서만 사람이 직접 선택하여 평가를 한 후 나머지 개체는 선택된 소수의 개체와의 유사도에 따라 평가를 하는 대화형 유전자 알고리즘도 연구되었다[7]. 그리고 개체들을 클러스터링하여, 각 클러스터의 대표만으로 적합도를 평가하고 나머지 개체는 이로부터 추정하는 효율적인 유전자 알고리즘을 제안한 연구도 있었다[8]. 하지만 이 연구는 일반적인 하드 클러스터링 알고리즘을 사용하였기 때문에 다수의 집단에 속하는 개체가 존재할 경우, 올바른 클러스터를 형성하지 못하여 정확히 적합도를 배분하기 어렵다.

본 논문에서는 이러한 문제를 해결하기 위하여 퍼지 클러스터링 방법에 기반한 국소평가 유전자 알고리즘을 제안한다. 진화시키고자 하는 집단을 10개의 클러스터로 그룹화하고, 각 클러스터의 대표 개체만의 적합도를 계산한 후, 이를 기준으로 클러스터의 나머지 개체에 대하여 그 개체의 소속 행렬값에 따라, 소속된 다수의 클러스터 대표의 적합도를 배분받는다. 제안한 방법의 성능을 평가하기 위해 9개의 벤치마크 적합도 평가 함수에 대해 대표적인 6가지 하드 클러스터링 알고리즘에 유클리디안 거리와 피어슨 상관계수에 의한 적합도 배분 방법을 적용한 결과와 퍼지 c-means 알고리즘을 이용한 소속 행렬에 의한 적합도 배분 방법을 적용한 결과를 체계적으로 비교분석하였다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장은 연구의 배경이 되는 유전자 알고리즘과 클러스터링 알고리즘에 대하여 소개한다. 3장은 제안하는 방법의 구성과 각 세부사항에 대하여 설명하고, 4장은 실험 과정과 결과를 설명한다. 그리고 5장은 논문의 결론에 대해 언급한다.

2. 배경

이 장에서는 유전자 알고리즘과 대표적인 하드 클러스터링 알고리즘에 대하여 설명한다.

2.1 유전자 알고리즘

유전자 알고리즘은 1970년대 초에 John Holland에 의해 제안되었다. 이 알고리즘은 적자생존, 유전자 교차, 돌연변이 등 자연의 진화 메커니즘에서 동기를 얻은 학습 방법이다. 유전자 알고리즘은 염색체라고 하는 유전자 개체의 형태로 표현되며 알고리즘의 반복을 통해 자손을 생성하기 위해 조합된다. 적합도가 높은 개체의 집합이 살아남아 다음 세대의 자손을 생성하는 부모가 되며, 자손은 선택(Selection), 교차(Crossover), 돌연변이(Mutation)라고 하는 유전 연산자를 통해 생성된다. 일반적인 유전자 알고리즘의 절차는 다음과 같다[9,10].

- ① 유전자 집단을 초기화한다.
- ② 집단내 각 개체의 적합도를 평가한다.
- ③ 적합도 값에 근거하여 우수한 개체들을 선택하여 새로운 집단을 만든다.
- ④ 집단의 개체에 유전자 교차와 돌연변이 연산을 수행한다.
- ⑤ 일정한 조건이 만족될 때까지 (2)에서 (4)까지의 과정을 반복한다.

적합도 값에 따른 우수한 개체 선택방법에는 토너먼트, 순위-기반, 룰렛 휠 등의 여러 가지 방법이 있다. 토너먼트 선택은 두 개의 개체가 경쟁하여 보다 우수한 개체가 다음 세대까지 살아남는 방법이며, 순위-기반 방법은 개체의 적합도에 순위를 매겨 순위가 높은 개체일수록 높은 선택확률을 부여받게 하는 방법이다. 마지막으로 룰렛 휠 선택방법은 한 세대의 다른 모든 개체의 적합도 값과 각 개체의 적합도에 비례하여 개체를 선택하는 방법이다. 이러한 선택방법을 통해 상대적으로 높은 적합도의 개체들이 선택되며, 낮은 적합도를 가진 개체들은 선택되지 않기도 한다. 선택된 개체들은 두 개씩 짝을 지어 교차연산을 거치게 되고, 마지막으로 돌연변이 연산을 통해 새로운 세대의 집단이 생성된다. 교차연산에서는 짝지어진 개체끼리 유전정보를 교환하는 작업이 일어나며, 돌연변이 연산에서는 유전정보 중 일부가 사라지거나 새로 나타난다. 새롭게 생성된 집단은 그들의 부모세대가 했던 과정을 반복하게 되며 이는 미리 정의한 수준에 이를 때까지 반복된다. 교차와 돌연변이 연산 과정은 그림 1과 같다.

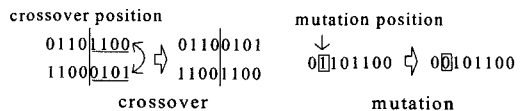


그림 1 교차 및 돌연변이 연산

2.2 클러스터링 알고리즘

하드 클러스터링 알고리즘은 각 데이터의 유사성에

근거하여 데이터를 그룹화하며 같은 그룹 내의 데이터는 높은 유사도를 갖고 다른 그룹 내의 데이터와는 낮은 유사도를 갖도록 군집화한다[11,12]. 이 때 각 그룹을 클러스터라 한다. 클러스터링 방법은 크게 계층적 클러스터링, 분할 클러스터링, 중복 클러스터링으로 나눌 수 있다.

(1) 계층적 클러스터링

계층적 클러스터링(Hierarchical clustering)은 클러스터들이 더 작은 클러스터로 이루어진 하부 구조를 가지도록 하는 방법이며, 클러스터 간의 거리를 구하는 방법에 따라 그림 2와 같이 단일연결(single linkage), 완전연결(complete linkage), 평균연결(average linkage) 등의 방법으로 나뉜다[13,14].

단일연결 알고리즘은 두 클러스터 간의 가장 가까운 개체의 거리를 클러스터 간의 거리로 정의하고, 완전연결 알고리즘은 두 클러스터 간의 가장 먼 개체의 거리를 클러스터 간의 거리로 정의하며, 평균연결 알고리즘은 두 클러스터 내의 모든 개체 사이 거리의 평균을 클러스터 간의 거리로 정의한다.

(2) 분할 클러스터링

분할 클러스터링(Partitional clustering)은 클러스터 간에 중복이 없으며, 각 개체를 가장 가까운 클러스터에 할당하는 과정을 반복하여 가장 적합한 클러스터를 구성한다. 하드 c-means (HCM) 알고리즘과 k-means 알고리즘이 분할 클러스터링의 대표적인 예이다. K-means 알고리즘은 가장 많이 쓰이는 클러스터링 알고리즘의 하나로 다음의 목적함수를 최소화시키는 방향으로 알고리즘이 진행된다[15].

$$E = \sum_{i=1}^c \sum_{x \in C_i} d(x, v_i) \tag{1}$$

v_i 는 i 번째 클러스터의 중심이고, $d(x, v_i)$ 는 x 와 v_i 의 유클리디안 거리이다. 이와 같이 각 클러스터에 속한 개체들과 클러스터 중심간의 거리를 최소화시키는 방향으로 각 개체를 가장 가까운 클러스터에 소속시킨다. 하드 c-means 알고리즘의 목적함수는 다음과 같다[15].

$$J(X, U, V) = \sum_{i=1}^c \sum_{j=1}^n u_{ij} d^2(x_j, v_i) \tag{2}$$

여기에서 u_{ij} 는 개체의 소속여부에 대한 정보를 가진 행렬로 i 번째 개체가 v_i 에 속할 경우 1의 값을 가지며 속하지 않으면 0의 값을 가진다. $d^2(x_j, v_i)$ 는 x_j 와 v_i 의 유클리디안 거리의 제곱이다. k-means와 마찬가지로 하드 c-means 알고리즘은 목적함수를 최소화시키는 방향으로 각 개체를 해당 클러스터에 소속시킨다.

(3) 중복 클러스터링

중복 클러스터링은 클러스터 간에 계층적 구조를 갖지 않으며, 클러스터 간의 중복을 허락하여 각 개체를 가장 가까운 클러스터로 근접시키는 과정을 반복하여 가장 적합한 클러스터를 구성한다. 즉, 분할 클러스터링은 하나의 개체가 가장 가까운 하나의 클러스터로 할당되는 데 반해 중복 클러스터링에서는 하나의 개체가 여러 클러스터에 소속될 수 있으며 근접 정도에 따라 소속 정도에 차등을 두는 방법에 의해 할당된다[16].

3. 방법

국소 평가 유전자 알고리즘은 집단내의 개체들을 클러스터링하고, 각 클러스터의 중심만을 평가하여 클러스

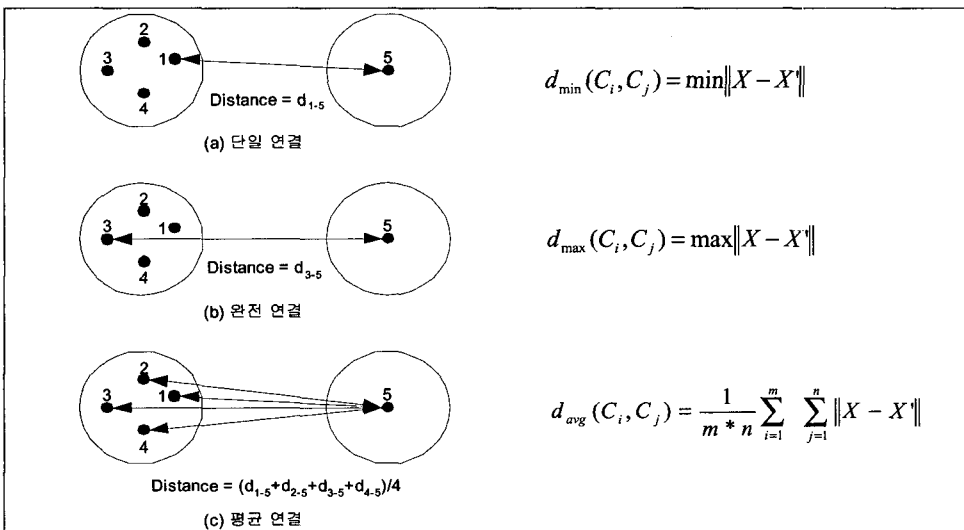


그림 2 계층적 클러스터링 알고리즘의 클러스터 간의 거리

터내의 개체들에게 중심과의 유사도에 따라 적합도를 배분하는 방법이다[8]. 이 방법은 크게 두 가지 과정에 의해 진행된다. 개체들을 유사도에 따라 집단화시키는 클러스터링 과정과 클러스터 중심의 적합도를 평가하고 해당 클러스터에 속한 개체들의 유사도에 따라 각 개체에 적합도를 분배하는 과정으로 나뉜다.

본 논문에서 제안한 퍼지 클러스터링 기반의 국소 평가 유전자 알고리즘의 개요는 그림 3과 같다. 우선 클러스터링 과정에서는 퍼지 c-means 알고리즘에 의해 개체들을 10개의 클러스터로 나누고, 각 클러스터의 중심을 구한다. 그리고 적합도 분배과정에서 각 클러스터 중심의 적합도를 평가하고, 나머지 개체들은 그 소속 행렬 값과 소속된 클러스터 중심과의 유사도에 따라 간접적으로 적합도를 분배받는다. 이렇게 모든 개체들의 적합도를 평가한 후, 유전자 알고리즘의 과정인 선택, 교차, 그리고 돌연변이를 거쳐 다음 세대의 개체들을 생성한다. 정해진 세대까지 위의 과정을 반복하여 최적해를 찾아나간다. 개체들을 10개의 클러스터로 나누고 10번의 적합도 평가만으로 모든 개체들의 적합도를 구할 수 있기 때문에 적합도 평가에 따른 많은 시간과 비용을 효과적으로 줄일 수 있다.

제안하는 방법에서는 각 개체를 하나의 염색체로 설계한다. 염색체 설계 시, 적합도 평가 함수의 x 를 30bit의 2진수로 표현하여 각 유전자 알고리즘의 염색체 길이(chromosome size)를 $30 \times (x$ 의 수)로 정하였다. 각 평가함수마다 x 의 개수가 다르기 때문에 각 평가함수별로 염색체의 길이도 모두 다르게 설계하였다. 이렇게 설계된 각 개체들은 그림 3의 클러스터링과정과 적합도 분배과정을 거친 후, 유전자 연산인 선택, 교차, 돌연변이

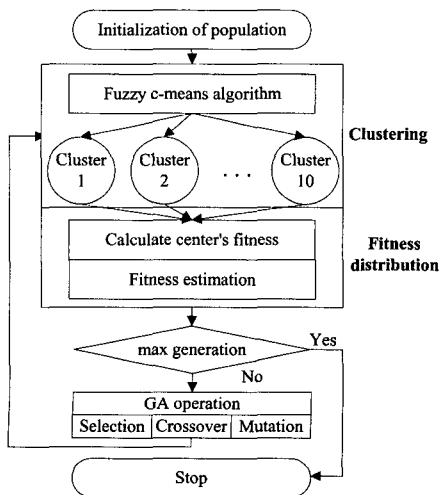


그림 3 제안하는 방법의 개요

를 한다. 선택방법은 룰렛 휠을 사용하였다. 한 세대의 진화가 끝나면 다시 클러스터링을 하고 새로 업데이트된 클러스터 중심으로부터 적합도를 분배받는 과정이 되풀이된다.

3.1 클러스터링

퍼지 클러스터링 알고리즘은 하나의 샘플이 반드시 하나의 클러스터에만 속하는 것이 아니라, 다수의 클러스터에 속하는 정도를 나타내는 클러스터링 알고리즘이다. 퍼지 클러스터링 알고리즘은 하드 클러스터링 알고리즘에 비해 지역해에 빠질 가능성이 적으며, 소속 행렬 값으로 각 개체가 다수의 클러스터에 소속되는 정도를 표현한다[17]. 본 논문에서는 대표적인 퍼지 클러스터링 알고리즘인 퍼지 c-means 알고리즘을 사용하였다.

퍼지 c-means 알고리즘은 Bezdek에 의해 제안된 것으로, 가장 널리 이용되는 퍼지 클러스터링 방법이다[18]. 이 방법은 퍼지이론을 적용한 목적함수의 반복 최적화에 기반을 둔 방식으로 각 데이터가 특정 클러스터에 속하는 소속 정도를 줌으로써 데이터에 대한 보다 정확한 정보를 제공한다[19]. 그림 4에 퍼지 c-means 알고리즘이 나와 있다. 먼저 클러스터 수(c)와 퍼지 계수(m)의 값을 결정한다. 그리고 x_j 의 소속 정도인 u_{ij} 를 합이 1이 되도록 초기화 한다. u_{ij} 는 퍼지 소속행렬이며, 행렬값은 $[0,1]$ 사이의 값을 가지며 1에 가까울 수록 해당 클러스터에 속하는 정도가 크다. 이렇게 계산된 u_{ij} 를 가지고 클러스터의 중심 v_i 를 구한다. 마지막으로 클러스터의 중심과 x_j 의 거리를 계산하여 소속 행렬 u_{ij} 를 구한다. 그리고 나서 전 단계의 목적함수와 현 단계의 목

- 단계 1: 클러스터 수(c)와 퍼지 계수(m)값을 설정한다.
- 단계 2: 다음의 조건을 만족하도록 x_j 의 소속 정도인 u_{ij} 를 초기화 한다.

$$\sum_{i=1}^c u_{ij} = 1, 1 \leq j \leq n$$

- 단계 3: 각 클러스터의 중심 v_i 를 계산한다. ($i=1,2,\dots,c$)

$$v_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m}$$

- 단계 4: 소속 행렬 U 를 계산한다.

$$u_{ij} = \frac{\left(\frac{1}{d^2(x_j, v_i)} \right)^{\frac{1}{m-1}}}{\sum_{k=1}^c \left(\frac{1}{d^2(x_j, v_k)} \right)^{\frac{1}{m-1}}}$$

- 단계 5: 목적함수 J_m 를 계산한다.

$$J_m(X, U, V) = \sum_{j=1}^n \sum_{i=1}^c (u_{ij})^m d^2(x_j, v_i)$$

- 단계 6: 다음의 종료 조건이 만족할 때까지 단계 3부터 단계 5를 반복한다.

$$|J_m^{(t)} - J_m^{(t-1)}| \leq \epsilon$$

그림 4 퍼지 c-means 알고리즘

적합도의 차를 계산하여 그 차이가 특정 임계값(ϵ)보다 작아질 때까지 위의 과정을 반복한다.

목적 함수는 주어진 데이터 집합이 $X = \{x_1, x_2, \dots, x_n\}$ 이고 퍼지 클러스터링의 중심 벡터가 $V = \{v_1, v_2, \dots, v_c\}$ 일 때, 각 데이터 x_j 와 각 클러스터 중심 v_i 와의 거리와 클러스터 소속 정도(membership) 값으로 정의된다. u_{ij} 는 x_j 가 i 번째 클러스터에 대한 소속 정도를 나타내며 $(c \times n)$ 의 소속 행렬 $U = [u_{ij}]$ 의 원소이다. $d^2(\cdot, \cdot)$ 는 유클리디안 거리(Euclidean distance)의 제곱이고, 매개 변수 m 은 각 데이터의 소속 정도에 대한 퍼지 값을 나타내며 1에서 2사이의 값을 사용한다. 본 논문에서는 $m=1.2$ 의 값을 사용하여 실험하였다.

3.2 적합도 배분

클러스터의 중심으로부터 적합도를 각 개체에 분배하는 적합도 간접 평가는 개체들의 진화과정에서 매우 중요한 역할을 한다. 개체의 적합도를 대체할 수 있는 값을 클러스터 중심으로부터 정확하게 배분받지 못하면, 올바른 평가가 이루어지지 못하여 국소 평가 유전자 알고리즘의 효율성이 감소한다. 적합도 간접 평가 방법의 선택에 따라 방법의 성능이 크게 좌우되기 때문에 정확하고 효율적인 적합도 간접 평가 방법이 필요하다.

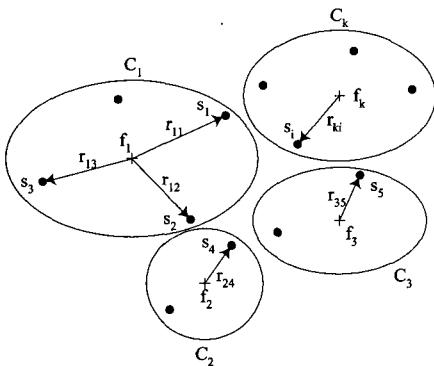


그림 5 유사도 기반의 적합도 배분 방법

하드 클러스터링을 사용하면 클러스터 중심과 해당 클러스터에 속한 개체들의 유사도를 측정하여 적합도를 배분할 수 있다. 그림 5와 같이 주어진 개체들의 집합이 $S = \{s_1, s_2, \dots, s_i\}$ 이고 클러스터 집합은 $C = \{c_1, c_2, \dots, c_k\}$ 일 때, 각 클러스터 중심의 적합도를 $F = \{f_1, f_2, \dots, f_k\}$ 라고 하자. 이 때 각 개체가 받는 적합도 s_i 는 다음과 같다.

$$s_i = r_{ki} f_k, \quad s_i \in c_k \quad (3)$$

i 번째 개체인 s_i 는 자기가 속한 하나의 클러스터 중심과의 유사도 계수 r_{ki} 에 비례하여 적합도를 분배받는다. 유사도 계수 r_{ki} 는 i 번째 개체와 k 번째 클러스터 중심과

의 유사도를 의미하며, 유클리디안 거리나 피어슨 상관계수를 사용할 수도 있다.

피어슨 상관계수(PC)는 상관계수분석에서 자주 이용되는 계수로 하나의 변수가 다른 변수와 관련성이 있는지, 또 관련 정도가 어느 정도인지 알아보기 위한 방법으로 i 번째 개체와 c_k 의 피어슨 상관계수 PC_i 는 다음과 같다[20].

$$PC_i = \frac{\sum s_i c_k - \frac{\sum s_i \times \sum c_k}{D}}{\sqrt{\left(\sum s_i^2 - \frac{(\sum s_i)^2}{D}\right) \left(\sum c_k^2 - \frac{(\sum c_k)^2}{D}\right)}, \quad s_i \in c_k \quad (4)$$

s_i 은 i 번째 개체를 의미하고 k 번째 클러스터(c_k)에 속한다. c_k 는 k 번째 클러스터의 중심이며 D 는 개체 속성의 차원일 때, 피어슨 상관계수 PC_i 는 $[-1, 1]$ 의 값을 갖는다. 1에 가까운 값을 가질수록 해당 개체는 자기가 속한 클러스터 중심과의 유사도가 크다. 피어슨 상관계수에 의한 유사도 계수 r_{ki} 는 다음과 같다.

$$r_{ki} = \alpha \cdot PC_i \quad (5)$$

PC값이 클수록 클러스터 중심과 개체의 유사도가 크기 때문에 PC값에 비례하여 r_{ki} 를 구한다. 유클리디안 거리는 두 변수의 유사성을 측정하는 방법으로 기하학적 공간에서의 거리를 나타내며 i 번째 개체 s_i 와 c_k 의 유클리디안 거리는 다음과 같다[21].

$$ED_i = \sqrt{\sum_{k=1}^P (c_k - s_i)^2}, \quad s_i \in c_k \quad (6)$$

s_i 가 속한 클러스터 중심(c_k)과의 거리를 구한 후, 개체의 적합도를 간접적으로 평가한다. 여기서 P 는 집단의 크기를 의미한다. ED값이 크게 나올수록 유사한 정도가 낮은 것이므로, 이 값에 반비례하게 다음과 같이 적합도 분배에 필요한 유사도 계수 r_{ki} 를 구한다.

$$r_{ki} = \alpha \cdot \frac{1}{ED_i} \quad (7)$$

PC와 ED는 모두 하드 클러스터링 방법을 사용한 적합도 간접 평가 방법으로 하나의 개체가 하나의 집단에 소속되는 경우만을 가정하여 평가하는 방법들이다. 각 개체가 소속된 클러스터의 대표치에 의해서 적합도를 선형으로 분배받기 때문에 정확한 적합도를 계산하기 어렵다.

본 논문에서는 이러한 단점을 극복하기 위하여 퍼지 클러스터링 결과인 소속 행렬을 이용하여 개체들의 적합도를 간접적으로 평가하는 방법을 제안한다. 퍼지 소속행렬에 의한 배분 방법을 사용할 경우, 따로 특정 유사도 측정 도구가 필요하지 않으며 퍼지 클러스터링 알고리즘의 결과인 퍼지 소속 행렬을 그대로 사용할 수 있다. 소속 행렬에 의한 적합도 배분 방법은 그림 6과 같다.

소속 행렬에 의한 적합도 배분 방법은 그림 6과 같이,

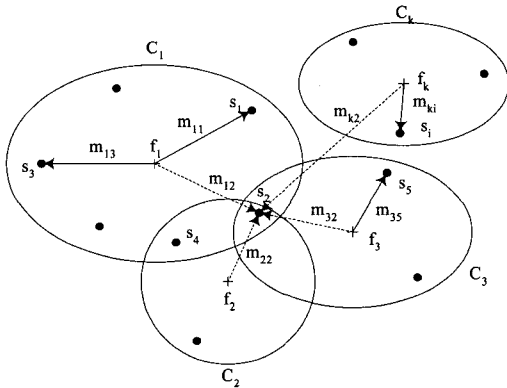


그림 6 소속 행렬에 의한 적합도 분배 방법

각 개체가 다수 클러스터의 중심으로부터 소속 멤버십 값에 의해 적합도를 배분받는다. m_{ki} 는 i 번째 개체가 k 번째 클러스터에 속하는 소속 멤버십값을 의미한다. 이때 각 개체가 배분 받는 적합도는 다음과 같다.

$$s_i = \sum_k m_{ki} \times f_k, \quad s_i \in C_k \quad (8)$$

i 번째 개체는 식 (8)과 같이 모든 클러스터의 중심으로부터 해당 소속 멤버십값에 비례하여 적합도를 분배 받는다. 그림 6에서 개체와 클러스터 중심간의 실선은 개체가 하나의 클러스터에 속하는 경우를 의미하고 점선은 하나의 개체가 다수의 클러스터에 속하는 것을 의미한다. s_2 와 같은 개체는 그 소속 정도가 확실하게 하나의 클러스터에 속하지 않고, c_1 , c_2 , 그리고 c_3 에 동시에 속한다. 이런 경우 단지 하나의 클러스터 중심으로부터만 적합도를 분배받는 하드 클러스터링 알고리즘과는 달리 제안하는 방법은 s_2 가 속하는 3개의 모든 클러스터 중심들을 이용하여 적합도를 분배받는다. 식 (8)과 같이 각 클러스터마다 소속된 정도(m_{ki})에 따라 해당 클러스터 중심의 적합도(f_k)를 나누어가지게 하여 보다 정확히 적합도를 배분할 수 있다. c_k 와 같이 소속 행렬값이 0에 가까운 클러스터의 경우, s_2 가 속하는 정도가 매우 낮기 때문에 적합도 배분에 큰 영향을 주지 않는다. 따라서 적합도 계산과정에서는 $m_{ki} > 0$ 인 조건을 만족하는 소속 행렬값의 클러스터 중심만을 포함시킨다.

4. 실험 및 결과

4.1 실험 환경

제안한 국소 평가 유전자 알고리즘의 성능을 평가하기 위해 9개의 적합도 평가 함수에 대하여 100개의 개체와 10개의 개체를 가지고 simple GA(기존의 유전자 알고리즘)에 적용한 결과와 6가지 하드 클러스터링 알고리즘과 퍼지 클러스터링 알고리즘을 이용하여 100개

의 개체에 대하여 국소평가 유전자 알고리즘을 적용한 결과를 비교 분석하였다. 각 실험은 10번 반복하여 그 평균값을 실험 결과로 사용하였고, 6가지 하드 클러스터링 방법에는 유클리디안 거리와 피어슨 상관계수를 유사도 측정 도구로 사용하였다.

실험에 사용된 방법은 표 1에, 실험에 사용된 환경 변수들은 표 2에 정리하였다. Pop 100과 Pop 10은 국소평가 유전자 알고리즘의 성능을 비교하기 위하여 기존의 유전자 알고리즘을 적용하였다. Cluster 10은 100개의 개체를 10개의 클러스터로 클러스터링하여 10번의 적합도 평가만을 하며 진화시켰다. 유전자 교차 확률과 돌연변이 확률은 각각 0.9와 0.001로 정하였는데, 이는 각 값을 변경하면서 실험 한 결과 우수한 개체의 보존 확률이 가장 높게 나왔기 때문이다. Pop100을 각 적합도 평가 함수에 적용하여 실험한 결과, 대부분의 경우 총 200세대 근처에서 수렴하였기 때문에 진화의 종료 범위를 총 세대수 200으로 정하였다.

표 3은 실험에 사용된 벤치마크 적합도 평가 함수를 보여준다[21]. 표 3의 벤치마크 함수들은 다양한 종류의 해 공간을 갖기 때문에 유전자 알고리즘의 성능을 평가하는 데 널리 사용된다. 함수의 성격에 따라 최적해와 국부해의 분포가 매우 다양하며 국부해가 많이 존재하는 Schwefel 함수나 Griewangk 함수의 경우 유전자 알고리즘의 개체 수에 따라 그 성능이 크게 좌우된다.

표 1 실험에 사용된 방법

방법	설명
Pop 100	100개의 개체에 대하여 simple GA에 적용하여 진화 시킨 경우
Pop 10	10개의 개체에 대하여 simple GA에 적용하여 진화 시킨 경우
Cluster 10	100개의 개체에 대하여 clustering GA에 적용하였으며 10개의 클러스터로 클러스터링하여 10번의 적합도 평가만을 하며 진화시킨 경우

표 2 실험에 사용된 환경 변수

	Pop 100	Pop 10	Cluster10
집단의 크기	100	10	100
세대 당 적합도 평가 횟수	100	10	10
클러스터 수	-	10	-
유전자 길이	30		
유전자 교차 확률	0.9		
돌연변이 확률	0.001		
총 세대 수	200		
퍼지 파라미터	1.2		
종료 조건	0.0000001		

표 3 벤치마크 적합도 평가 함수

인덱스	적합도 평가 함수	함수	정의역
EF1	De Jong Function 1	$f(x) = \sum_{i=1}^n x_i^2$	$n=3, -5.12 \leq x_i \leq 5.12$
EF2	De Jong Function 2	$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	$-2.048 \leq x_i \leq 2.048$
EF3	De Jong Function 3	$f(x) = \sum_{i=1}^n \text{int}(x_i)$	$n=5, -5.12 \leq x_i \leq 5.12$
EF4	De Jong Function 4	$f(x) = \sum_{i=1}^n (ix_i^4 + \text{Gauss}(0,1))$	$n=30, -1.28 \leq x_i \leq 1.28$
EF5	De Jong Function 5	$f(x) = 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}$ $(a_{ij}) = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & -16 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 16 & 32 & 32 & 32 \end{pmatrix}$	$-65.536 \leq x_i \leq 65.536$
EF6	Rastrigin Function	$f(x) = 3.0n + \sum_{i=1}^n x_i^2 - 3.0 \cos(2\pi x_i)$	$n=20, -5.12 \leq x_i \leq 5.12$
EF7	Schwefel Function	$f(x) = 418.9829n + \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	$n=10, -500.0 \leq x_i \leq 500.0$
EF8	Griewangk Function	$f(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$n=10, -600.0 \leq x_i \leq 600.0$
EF9	Ackley Function	$f(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)$	$n=30, -30.0 \leq x_i \leq 30.0$

4.2 실험 결과

실제로 본 논문에서 제안하는 방법이 유사도 측정 도구로 널리 사용되는 유클리디안 거리나 피어슨 상관계수보다 더 실제 개체의 적합도에 근접한 평가를 내리는 것을 확인하기 위한 실험 결과는 표 4와 같다. 각 평가 함수별로 실제 개체의 적합도와 국소 평가를 하였을 때의 적합도 차를 구하여 총 100개의 개체에 대한 평균을 계산하였다.

DeJong Function 4를 제외하고는 나머지 모든 함수들에 대해서 퍼지 소속 행렬에 의한 국소 평가 방법이 실제 개체의 적합도와 가장 유사한 결과를 보인다. 개체가 다수 클러스터의 중심으로부터 소속된 정도에 따라 적합도를 배분받는 것이 유클리디안 거리나 피어슨 상관계수에 의한 적합도 배분 방법에 비해 실제 개체의 적합도와 차이가 적기 때문에 단 10개의 대표 개체만으로도 충분히 정확한 평가가 이루어졌다.

표 4 적합도 배분 방법의 성능 비교(실제 개체의 적합도와와의 차)

함수	유클리디안 거리	피어슨 상관계수	퍼지 소속 행렬
DeJong Function1	9.22613	11.2633	9.07400
DeJong Function2	247.037	299.696	240.107
DeJong Function3	2.74428	3.09601	2.61852
DeJong Function4	249.889	250.104	250.043
DeJong Function5	0.00114	0.0015	0.00114
Rastrigin Function	134.572	178.888	134.040
Schwefel Function	648.38	1463.26	622.80
Griewangk Function	173.414	214.815	172.322
Ackley Function	13.2671	20.1320	13.3305

표 5 유클리디안 거리에 의한 적합도 배분 방법과 비교

	Pop 100	Pop 10	Cluster 10						
			S-L	C-L	A-L	Ward	HCM	KM	FCM
EF 1	75.15	71.59	74.41	75.13	75.44	74.76	73.99	74.57	75.43
EF 2	3835.3	3732.4	3818.2	3826.3	3825.4	3825.5	3795.8	3820.5	3849.0
EF 3	18.40	13.90	18.10	17.27	16.9	18.17	17.27	17.43	20.7
EF 4	569.89	542.83	572.81	557.72	554.21	565.12	553.75	508.14	594.98
EF 5	0.8358	0.4942	0.6872	0.6695	0.7098	0.6777	0.6193	0.6654	0.9468
EF 6	380.53	352.28	372.33	370.38	375.24	372.57	372.99	374.17	379.92
EF 7	6445.0	6135.8	6332.2	6446.2	6493.0	6416.6	6422.9	6408.9	6555.4
EF 8	648.39	617.66	647.94	649.21	644.81	626.37	646.39	651.90	642.99
EF 9	21.68	21.61	21.68	21.69	21.68	21.68	21.69	21.69	21.69

표 6 피어슨 상관계수에 의한 적합도 배분 방법과 비교

	Pop 100	Pop 10	Cluster 10						
			S-L	C-L	A-L	Ward	HCM	KM	FCM
EF 1	75.15	71.59	74.11	73.15	74.70	73.92	74.11	75.05	75.431
EF 2	3835.3	3732.4	3825.8	3811.7	3827.3	3813.6	3811.5	3829.1	3849.0
EF 3	18.40	13.90	18.40	18.10	18.10	16.83	17.37	17.00	20.7
EF 4	569.89	542.83	592.04	571.79	577.08	586.6	560.9	498.3	594.984
EF 5	0.8358	0.4942	0.7668	0.7126	0.7206	0.5525	0.6021	0.7392	0.9468
EF 6	380.53	352.28	377.80	378.69	369.15	375.60	376.23	360.46	379.92
EF 7	6445.0	6135.8	6368.5	6379.7	6381.2	6365.4	6422.4	6259.6	6555.4
EF 8	648.39	617.66	660.39	654.84	652.46	649.04	640.57	603.47	642.99
EF 9	21.68	21.61	21.70	21.68	21.69	21.68	21.68	21.61	21.69

표 7 Pop100과 FCM의 t-검정 결과(자유도: 198, 유의수준: 0.05)

평가 함수	세대수	Pop 100		FCM(Cluster 10)		t-통계량
		평균	표준편차	평균	표준편차	
EF1	200	45.2452	11.9514	31.6593	14.0484	7.36583
EF2	200	1649.42	811.311	1600.34	1127.07	0.35228
EF3	200	10.3	4.93161	12.1	4.23906	-2.7371
EF4	200	248.230	54.0054	293.949	111.136	-3.6999
EF5	200	0.10988	0.812	0.66031	0.43471	-12.03
EF6	200	288.509	31.6015	221.370	35.7431	14.0724
EF7	200	4426.47	516.488	4249.8	588.768	2.25480
EF8	200	338.72	63.8574	396.391	77.7296	-5.7322
EF9	200	21.1511	0.20537	20.8383	0.27708	9.07048

표 5는 유클리디안 거리에 의한 적합도 배분 방법과 본 논문에서 제안하는 방법의 성능을 비교한 결과이다. Single-linkage(S-L), Complete-linkage(C-L), Average-linkage(A-L), Ward, Hard c-means(HCM), K-means (KM), 그리고 제안하는 방법인 Fuzzy c-means (FCM) 알고리즘에 대한 실험 결과를 보면, EF 1, EF 6, EF 8 을 제외한 나머지 모든 평가 함수에 대해서 제안하는 방법(FCM)이 다른 방법들에 비해 우수한 성능을 보였다. EF 1의 경우는 A-L이 가장 우수한 성능을 보였고, EF 6은 Pop 100이 가장 우수한 성능을 보였다. 전체적으로는 Pop 10의 성능이 모든 평가 함수에 비하여 낮았고, Cluster 10, Pop 100 순으로 점차 성능이 좋아졌다.

표 6은 피어슨 상관계수에 의한 적합도 배분 방법과

본 논문에서 제안한 방법의 각 방법별로 최대 적합도 비교 결과이다. 여기서도 FCM방법은 EF 6, EF 8, EF 9를 제외한 나머지 모든 평가 함수에 대해 다른 방법보다 우수한 성능을 보인다. EF 9의 경우는 함수 특성상 각 방법별로 성능이 큰 차이를 보이지 않는다. 이와 같이 퍼지 소속 행렬값에 의한 적합도 배분 방법은 기존의 유클리디안이나 피어슨 상관계수에 의한 적합도 배분 방법보다 성능이 우수하며, 적합도 평가면에서 100개 개체 모두의 적합도를 평가하는 Pop 100에 뒤지지 않는 성능을 보인다.

실제로 100개의 모든 개체에 대해 적합도 평가를 하지 않고도 이와 유사한 성능을 보이는 것은 비용과 시간면에서 매우 효율적이다. 표 7은 100개의 모든 개체에

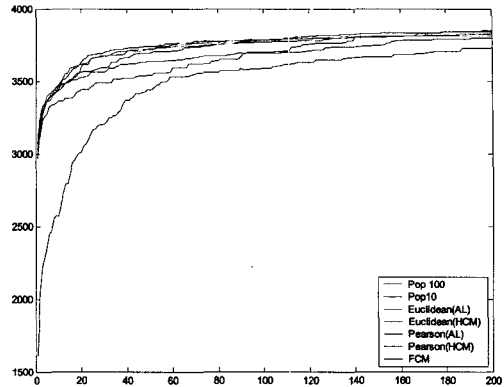
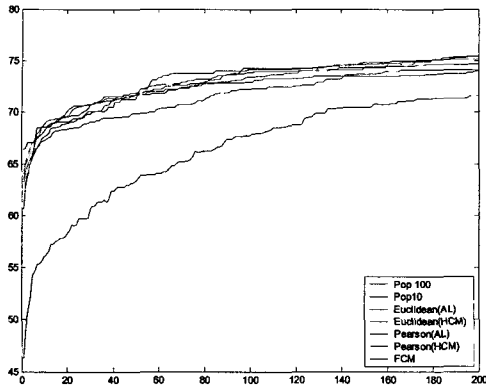


그림 7 DeJong Function 1과 DeJong Function 2의 진화과정

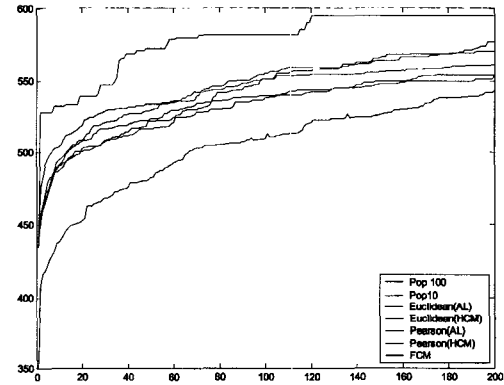
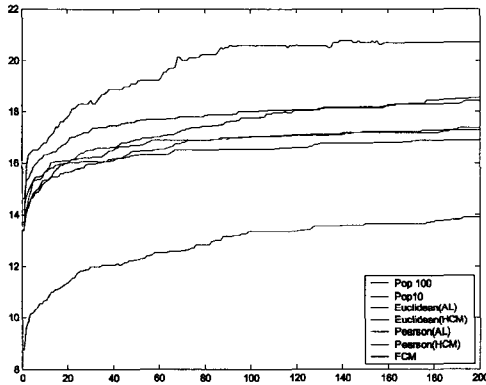


그림 8 DeJong Function 3과 DeJong Function 4의 진화과정

대해 적합도 평가를 한 경우와 단지 10개의 클러스터 중심들의 적합도 평가를 한 두 가지 경우에 대한 t-test 실험 결과이다. 단측 검정에 의하여 t-통계량을 분석해보면 EF 1, EF 6, 그리고 EF 9는 Pop10과 FCM이 유의수준 0.05에서 다르다는 결론에 도달하지만, 나머지 6개의 함수에서는 별 다른 차이를 보이지 않는다. 즉 비슷한 적합도 분포를 보인다고 할 수 있다.

그림 7은 DeJong Function 1과 DeJong Function 2에 대한 진화과정이다. Euclidean(AL)은 Average-Linkage 알고리즘과 유클리디안 거리를 이용한 적합도 배분 방법을 의미하고 Euclidean(HCM)은 Hard c-means 알고리즘과 유클리디안 거리를 이용한 적합도 배분 방법을 의미한다. Pearson(AL)과 Pearson(HCM)은 각각 같은 알고리즘에 피어슨 상관계수를 적합도 배분 방법으로 사용한 것이다. DeJong Function 1은 전형적인 이차 함수로서, 다른 적합도 평가 함수에 비하여 단조로운 해 공간을 갖는다. 따라서 지역해에 빠질 가능성이 적기 때문에 Pop10이 가장 낮은 적합도를 가지고 진화되며, 진화되는 수렴속도 또한 가장 느리다. 또한

유클리디안이 피어슨 상관계수보다 진화 속도가 느리며 적합도도 상대적으로 낮다. DeJong Function 2는 전체적으로 빠른 수렴 속도를 보이며 진화가 된다. 여기서는 FCM의 진화되는 속도가 다른 방법들에 비해 느리지만, 200세대 가까이 진화됨에 따라 Pop10에 가까운 적합도를 보인다. 역시 DeJong Function 1과 마찬가지로 유클리디안 거리의 성능이 피어슨 상관계수에 비해 떨어진다.

그림 8은 DeJong Function 3과 DeJong Function 4의 진화과정이다. DeJong Function 3은 5차원 함수로 계단 모양의 복잡한 형태의 함수이다. 이 함수에서 FCM의 결과가 다른 방법에 비해 큰 차이를 보이며 높은 적합도를 보이는 것은 이 함수 자체가 매우 복잡한 모양을 하고 있기 때문에, 클러스터를 형성하는데 있어서 일반적인 하드 클러스터링 보다는 퍼지 클러스터링이 더 적합하기 때문이다. 특히 개체들의 분산이 매우 적기 때문에 적합도를 배분하는 과정이 매우 까다롭다. DeJong Function 4에서도 FCM이 우수한 성능을 보인다. 이 함수에서 FCM은 매우 빠른 속도로 수렴하는 것

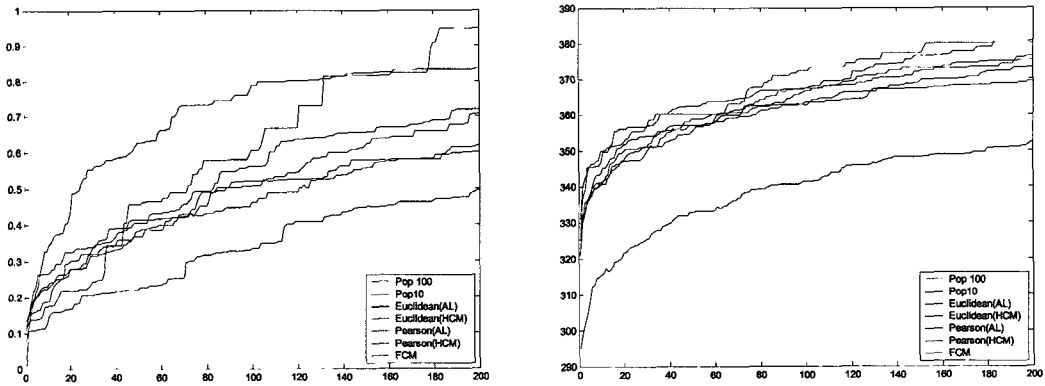


그림 9 DeJong Function 5와 Rastrigin Function의 진화과정

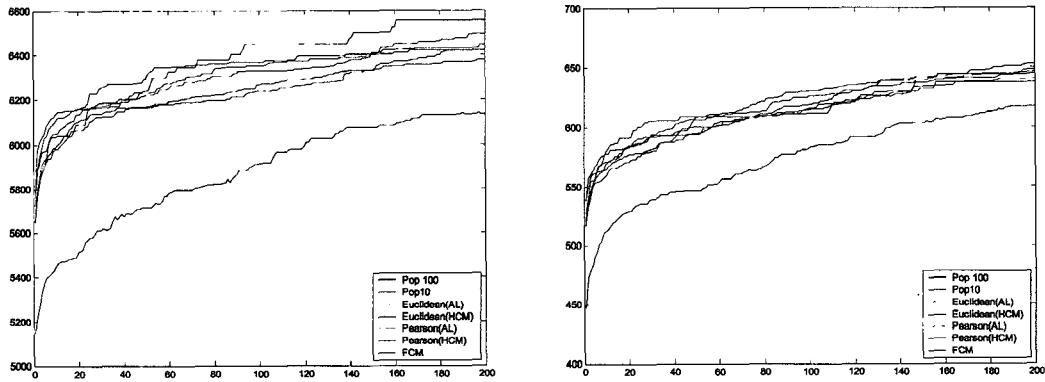


그림 10 Schwefel Function과 Griewangk Function의 진화과정

을 볼 수 있다. 이것은 이 함수가 고원 모양의 평탄한 지붕형태를 띄고 있기 때문에 최적해에 빨리 도달할 수 있기 때문이다.

그림 9는 DeJong Function 5와 Rastrigin Function의 진화과정이다. DeJong Function 5는 복잡한 landscape를 보이고, 많은 지역해를 가지고 있다. 따라서 진화 결과에서도 볼 수 있듯이 다른 적합도 평가 함수의 경우에 비해 많은 부분 수렴점을 볼 수 있으며, 많은 굴곡이 존재한다. 수렴 속도 또한 다른 평가 함수에 비해 느리다. 이 함수에서는 Pop100이 가장 빠른 속도로 최적해를 구하지만, 200세대동안 진화되면 FCM이 보다 최적의 해를 구한다. 역시 복잡한 형태이기 때문에 일반적인 하드 클러스터링의 경우 최적해를 찾는 데 어려움이 따른다. Rastrigin Function은 20차원의 함수로 Pop10을 제외한 나머지 방법들이 비슷한 수렴 속도를 보이며 진화해간다. FCM이 가장 빠르게 수렴하며, Pop100, 유클리디안 순으로 수렴해 간다.

그림 10은 Schwefel Function과 Griewangk Func-

tion의 진화과정이다. Schwefel Function 역시 울퉁불퉁한 모양을 가지고 있기 때문에 FCM방법이 가장 빠른 속도로 수렴한다. 초기 진화 세대에서는 피어슨 상관 계수(HCM)방법이 빠른 속도로 최적해를 찾아가지만, 최적해에 도달하지 못하고 지역해에 빠져 유클리디안 방법이나 FCM에 비해 낮은 적합도를 가진다. Griewangk Function은 Rastrigin Function과 비슷한 모양을 가지지만, 지역해에 빠질 가능성이 훨씬 적게 평탄한 표면을 가지고 있어서 하드 클러스터링 방법과 퍼지 클러스터링 방법의 성능에 거의 차이가 없다. Pop10을 제외한 나머지 방법에서 비슷한 수렴속도를 가지고 진화해가며, 적합도 또한 거의 같다.

그림 11은 Ackley Function의 진화과정이다. 30차원의 고차원 함수로서 많은 지역해가 존재하지만, 가운데 우뚝 솟은 최적해가 존재하는 함수이다. 그렇기 때문에 많은 지역해의 존재에도 불구하고 하드 클러스터링과 퍼지 클러스터링의 성능 차이가 크지 않으며 Pop100과 기타 다른 방법과도 차이가 적다. 또한 초기 세대에서는

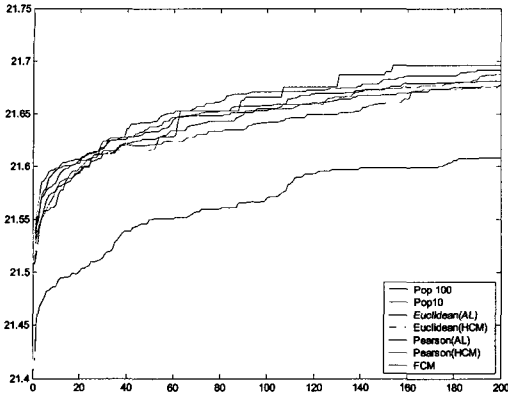


그림 11 Ackley Function의 진화과정

표 8 Simple 유전자 알고리즘(SGA)와 PEGA의 실행 시간 비교(단위: 초)

평가함수	SGA	PEGA
EF1	0.094	0.030
EF2	0.120	0.267
EF3	0.082	0.020
EF4	0.372	0.567
EF5	0.272	0.052
EF6	0.242	0.050
EF7	0.172	0.043
EF8	0.176	0.043
EF9	0.250	0.040

급속하게 적합도가 증가하다가 바로 수렴하는 것을 보아서 바로 가운데 우뚝 솟은 모양의 최적해에 근접해가는 것을 알 수 있다. 200세대에 가까워짐에 따라 FCM이 다른 방법에 비해 높은 해를 찾아간다.

표 8은 모든 100개의 개체에 대해 적합도 평가를 한 simple 유전자 알고리즘과 10개의 대표개체만을 평가한 제안하는 방법인 PEGA의 프로그램 수행시간 비교결과이다. 모든 평가함수에 대해 PEGA는 simple 유전자 알고리즘에 비해 시간면에서 효율이 뛰어난 것을 확인할 수 있다. 즉 적은 수의 적합도 평가만으로도 충분히 효율적이면서도 정확한 적합도 평가를 하는 것을 알 수 있다.

5. 결론

본 논문에서는 기존의 유전자 알고리즘에서 적합도 평가에 소요되는 계산량을 줄이는 국소 평가 알고리즘의 효율을 높이기 위하여 퍼지 클러스터링 기반의 적합도 배분 방법을 제안하였다. 실제 개체의 특성을 고려한 퍼지 클러스터링 기반의 소속 행렬에 의한 적합도 배분 방법은 하드 클러스터링 알고리즘 기반의 유클리디안

거리나 피어슨 상관관계수에 의한 적합도 배분 방법에 비해 우수한 성능을 보였다. 실험 결과에서도 볼 수 있듯이 100개의 개체에 대하여 각 세대 당 10번의 적합도 평가 과정만으로 우리가 원하는 최적해에 근접할 수 있었으며, 모든 100개 개체에 대해 적합도 평가를 한 경우보다 좋은 성능을 보인 평가함수들도 있었다. 실험에 사용된 벤치마크 함수의 성질에 따라 성능차이를 보였지만 빠른 수렴 속도를 보이며 최적해에 접근하였고, 유클리디안 거리와 피어슨 상관관계수에 의한 적합도 배분 방법보다 뛰어난 성능을 보였다.

참고 문헌

- [1] L. Chamber, *Practical Handbook of Genetic Algorithms*, CRC Press, 1995.
- [2] S.-B. Cho, and J.-Y. Lee, "A human-oriented image retrieval system using interactive genetic algorithm," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 32, no. 3, pp. 452-458, May 2002.
- [3] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proc. of the IEEE*, vol. 89, no. 9, pp. 1275-1296, 2001.
- [4] R. Shonkwiler, F. Mendivil, and A. Deliu, "Genetic algorithms for the 1-D fractal inverse problem," *Proc. 4th Int'l Conf. on Genetic Algorithms*, San Diego, pp. 495-501, July 1991.
- [5] M. V. Mannino and M. V. Koushik, "The cost minimizing inverse classification problem: A genetic algorithm approach," *Decision Support Systems*, vol. 29, pp. 283-300, October 2000.
- [6] M. Shibuya, H. Kita, and S. Kobayashi, "Integration of multi-objective and interactive genetic algorithms and its application to animation design," *Proc. 99' Int'l Conf. on Systems, Man, and Cybernetics*, vol. 3, pp. 646-651, 1999.
- [7] F. Sugimoto, and M. Yoneyama, "Hybrid fitness assignment strategy in IGA," *Proc. IEEE Workshop on Multimedia Signal Processing*, pp. 284-287, Dec. 2002.
- [8] H.-S. Kim and S.-B. Cho, "An efficient genetic algorithm with less fitness evaluation by clustering," *Proc. 2001 IEEE Congress on Evolutionary Computation*, pp. 887-894, May 2001.
- [9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publication, 1989.
- [10] R. Eberhart, P. Simpson, and R. Dobbins, *Computational Intelligence PC Tools*, Waite Group Press, 1996.
- [11] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *The Computer Journal*, vol. 41, pp. 578-588, 1998.

- [12] B. K. Lavine, "Clustering and classification of analytical data," *Encyclopedia of Analytical Chemistry*, October 2000.
- [13] R. O. Duda, P. E. Hart and D. G. Stock, *Pattern Classification*, Wiley-Interscience Publication, 2001.
- [14] D. Wishart, "Efficient hierarchical cluster analysis for data mining and knowledge discovery," *Computing Science and Statistics*, pp. 257-263, 1998.
- [15] D. Wishart, "K-means clustering with outlier detection, mixed variables and missing values," *Proc. Conference of the German Classification Society*, pp. 14-16, 2001.
- [16] I. Gath and A.B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 773-781, 1989.
- [17] D. Qzdemir and L. Akrun, "A fuzzy algorithm for color quantization of images," *Pattern Recognition*, vol. 35, no. 8, pp. 1785-1791, 2002.
- [18] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.
- [19] F. Höppner, F. Klawonn, R. Kruse and T. Runkler, *Fuzzy Cluster Analysis*, Wiley, 2000.
- [20] C. Park and S.-B. Cho, "Evolutionary computation for optimal ensemble classifier in Lymphoma cancer," *Lecture Notes in Comp. Sci.*, vol. 2871, pp. 521-530, Oct, 2003.
- [21] X. Yao, Y. Liu and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evolutionary Computation*, vol. 3, no. 2, pp. 82-102, July 1999.



유 시 호

1998년 연세대학교 컴퓨터과학과 학사 졸업. 2002년~현재 연세대학교 컴퓨터과학과 석사 과정. 관심분야는 바이오정보 기술, 패턴인식

조 성 배

정보과학회논문지 : 소프트웨어 및 응용
제 31 권 제 1 호 참조