클러스터링 기반의 효율적 유전자 알고리즘의 체계적인 성능 평가

원홍희 조성배
연세대학교 컴퓨터과학과
cool@candy.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Systematic Performance Evaluation of Efficient Genetic Algorithm based on Clustering

Hong-Hee Won and Sung-Bae Cho
Computer Science Department, Yonsei University

요 약

기존의 유전자 알고리즘은 우리가 원하는 최적해를 찾기 위해서 개체 집단의 크기를 가능한 크게 유지하여야 한다. 하지만 일반적인 문제들에 있어 개체의 적합도를 평가하는 것은 어렵기 때문에 큰 집단의 모든 개체에 대하여 적합도를 평가하는 것은 커다란 시간과 비용을 소모한다. 이에 본 논문에서는 집단의 크기를 크게 유지하되 적합도 평가 과정을 줄이는 방안으로 클러스터링에 기반한 효율적인 유전자 알고리즘을 제시하고 체계적인 평가를 한다. 9개의 벤치마크 적합도 함수에 대하여 여러 클러스터링 방법을 적용하여 실험한 결과, 제안한 방법의 유용성을 확인할 수 있었다.

1. 서론

유전자 알고리즘은 적자생존, 유전자 교차, 돌연변이 등 자연의 진화 메커니즘에 기반하여 문제를 해결하고자 하는 알고리즘이다. 즉, 다수의 개체들 중에서 뛰어난 생존 능력을 가진 개체가 많은 자손을 퍼뜨리고 많은 세대를 거치면서 각 세대의 평균적인 생존 능력이 진화되어 나간다는 점에 착안하였다. 또한 유전자 교차와 돌연변이에 의해 더 좋은 개체가 우연적으로 생길 수 있게 된다는 점도 유전자 알고리즘의 중요한 특성이다.

이 알고리즘은 풀고자 하는 문제에 맞는 적합도 평가 함수가 존재하여 다수 개체들의 적합도를 평가하여 적합도가 큰 우수한 개체들을 생존시키는 전략을 이용하여 원하는 해에 접근한다. 상대적으로 작은 크기의 집단을 사용한다면 지역적인 탐색에 의한 지역해에 빠질 가능성이 커지므로 집단의 크기를 가능한 크게 유지하여야 한다.

하지만 대화형 유전자 알고리즘이나 유전자 알고리즘의 많은 응용 분야에서 개체의 적합도를 평가하는 것에는 어려움이 있다. 예를 들어 사람의 감성을 적합도로 평가하고자 할 때 적합도 평가 함수를 정의하기 어렵고, 공학의 역 문제(inverse problem)를 유전자 알고리즘을 이용하여 풀고자 할 때 적합도를 평가하는 데 드는 시간과 비용이 크다는 문제가 있다.

이러한 문제를 해결하기 위하여 본 논문에서는 개체 집단의 크기를 유지하되 적합도 평가 과정을 줄이는 방안으로 클러스터링 방법에 기반한 효율적인 유전자 알고리즘을 제시하고, 9개의 적합도 평가 함수에 대해 대표적인 클러스터링 방법들을 적용하여 실험하였다.

2. 유전자 알고리즘

유전자 알고리즘은 1970년대 초에 John Holland에 의해 제안되었다. 이 알고리즘은 적자생존, 유전자 교차, 돌연변이 등 자연의 진화 메커니즘을 기반으로 하여 기계 학습, 최적화 등의 문제에 적용 되었으며, 그 외에도 효율적인 탐색 방법과 많은 분류 문제에도 응용되었다. 일반적인 유전자 알고리즘의 과정은 다음과 같다.

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

적합도 값에 따른 우수한 개체 선택방법에는 여러 가지가 있는데 그 중 대표적인 것이 룰렛 선택이다. 즉 모든 개체에 적합도 값을 누적시켜 각각의 누적 적합도 값을 기준으로 임의로 발생시킨 값과 그 누적 값을 비교하여 선택하는 방식이다. 교차연산은 선택된 두 개의 부모 개체로부터 유전자 스트링의 임의의 위치를 중심으로 양쪽의 스트링을 바꾸는 과정이다. 돌연변이 연산은 선택된 한 개의 부모 개체의 유전자 스트링의 임의의 부분을 반대 값으로 바꾸는 것이다.

확률 값에 의존한 우수한 개체 선택과 유전자 교차, 돌연변이 연산 과정에서 적합도 값이 가장 큰 개체가 소멸될 우려가 있다. 이러한 우수한 개체의 생존을 보존하는 방법으로 엘리티즘(Elitism)이 있다. 즉 이전 세대에서 가장 우수한

개체를 저장하였다가 다음 세대를 생성할 때 그 개체를 꼭 포함하도록 하는 것이다. 그 외에도 기본적인 유전자 알고리즘의 한계를 해결하기 위한 개선들이 많이 연구되어지고 있다.

3. 클러스터링 유전자 알고리즘

3.1 클러스터링 알고리즘

클러스터링 알고리즘은 각 데이터의 유사성에 근거하여 데이터를 그룹화하며 같은 그룹 내의 데이터는 높은 유사도를 갖고 다른 그룹 내의 데이터와는 낮은 유사도를 갖도록 군집화 한다. 이 때 각 그룹을 클러스터라 한다. 클러스터링 방법은 크게 계층적 클러스터링, 분할 클러스터링, 중복 클러스터링으로 나눌 수 있다.

유사도를 측정하는 방법에는 Pearson Correlation Method, Cosine Correlation Method 등의 상관 계수 관계법과 Euclidean Distance Method 등의 거리 측정법이 있다. 다음은 Euclidean Distance Method를 나타낸다.

$$d_{ij} = d(X_i, X_j) = \sqrt{\sum_{k=1}^{N} |x_{ik} - x_{jk}|^2} \quad (1)$$

A. 계층적 클러스터링(Hierarchical Clustering)

계층적 클러스터링은 클러스터들이 더 작은 클러스터로 이루어진 하부 구조를 가지도록 하는 방법이며, 다시 bottom-up 방식의 agglomerative 알고리즘과 top-down 방식의 divisive 알고리즘으로 나뉜다.

계층적 방법으로는 단일연결(single linkage), 완전 연결(complete linkage), 평균 연결(average linkage), 워드 기법(Ward's Method) 등이 있다. Agglomerative 알고리즘이 그림 1에 기술되어 있다.

❑ Algorithm 1.(Agglomerative Hierarchical Clustering)
(1) 모든 개체를 하나의 클러스터로 정의한다.
(2) 가장 가까운 클러스터를 merge 한다.
각 클러스터의 거리를 계산하는 식은 방법별로 수식 2~4 에 기술되어 있다.
(3) 우리가 원하는 클러스터 수가 될 때 까지 (2)를 반복한다.

그림 1. Agglomerative Algorithm

$$d_{\min}(C_i, C_j) = \min \|X - X'\| \quad (2)$$

$$d_{\max}(C_i, C_j) = \max \|X - X'\| \quad (3)$$

$$d_{avg}(C_i, C_j) = \frac{1}{m * n} \sum_{i=1}^{m} \sum_{j=1}^{n} \|X - X'\| \quad (4)$$

Single lingkage 방법은 수식 (2)와 같이 두 클러스터 간의 가장 가까운 개체의 거리를 클러스터 간의 거리로 정의한다. Complete lingkage 방법은 수식 (3)과 같이 두 클러스터 간의 가장 먼 개체의 거리를 클러스터 간의 거리로 정의한다. Average lingkage 방법은 수식 (4)와 같이 두 클러스터 내의 모든 개체 사이의 거리의 평균을 클러스터 간의 거리로 정의한다.

B. 분할 클러스터링(Partitional Clustering)

분할 클러스터링은 클러스터 간에 중복이 없으며, 각 개체를 가장 가까운 클러스터에 할당하는 과정을 반복하여 가장 적합한 클러스터를 구성한다. Hard c-means (HCM) 알고리즘과 k-means 알고리즘, ISODATA 알고리즘이 분할 클러스터링의 대표적인 예이다. K-means 알고리즘이 그림 2에 기술되어 있다.

❑ Algorithm 2. (K-means Clustering Algorithm)
(1) 처음 k 개의 개체로 k 개의 클러스터를 만든다.
(2) 남아 있는 $(n-k)$ 개의 개체에 대하여
· 가장 가까운 중심을 갖는 클러스터에 삽입한다.
· 변경된 클러스터의 중심을 다시 계산한다.
(3) n 개의 모든 개체에 대하여
· 가장 가까운 중심을 갖는 클러스터에 삽입한다.
· 변경된 클러스터의 중심을 다시 계산한다.
(4) 모든 클러스터의 중심이 변경되지 않을 때까지 (3)을 반복한다.

그림 2. K-means Algorithm

C. 중복 클러스터링(Overlapping Clustering)

중복 클러스터링은 클러스터 간의 계층적 구조를 갖지 않으며, 클러스터 간의 중복을 허락하여 각 개체를 가장 가까운 클러스터로 근접시키는 과정을 반복하며 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. Fuzzy c-means (FCM) 알고리즘과 b-clump 알고리즘 등이 있다.

3.2 클러스터링 유전자 알고리즘

기존의 유전자 알고리즘이 갖는 문제점은 개체의 수가 충분하지 않을 경우에 지역해에 빠질 우려가 있다는 점이다. 이를 해결하기 위하여 집단의 개체 수를 충분히 크게 유지하여야 하지만, 그럴 경우에 많은 개체들에 대해서 적합도를 평가하는 것이 큰 비용이 드는 경우가 있다. 예를 들어 interactive genetic algorithm (IGA)의 응용 분야는 주로 적합도를 사람이 직접 평가하므로 개체 수가 많아질수록 평가하기가 어렵게 된다.

위의 두 문제를 효과적으로 해결하기 위하여 본 논문에서 많은 개체를 유지하되 모든 개체들의 적합도를 직접 평가하는 것이 아니라 개체의 유사도를 기준으로 클러스터링을 하고 클러스터링의 중심의 적합도만을 평가하여, 클러스터 내의 개체들에 유사도에 비례하여 적합도를 배분하는 방법을 제시하였다. 본 논문에서 제시한 클러스터링 유전자 알고리즘의 전반적인 과정은 그림 3과 같다.

유사도에 의해 적합도를 배분하는 방법은 다음 수식 (5)와 같으며, 개체와 그 클러스터 중심 간의 Euclidean distance와 각 적합도 함수의 정의역의 특성을 고려하여 적합도를 배분하게 된다.

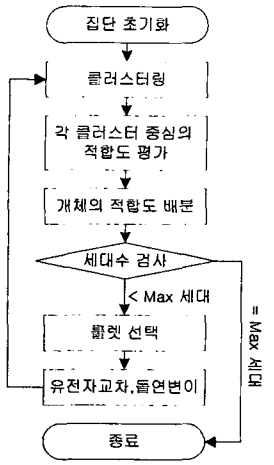

그림 3. 클러스터링 유전자 알고리즘

$$Fitness(X) = Fitness(C) * (1 - \frac{\sqrt{\sum_{i=1}^{N} |X_i - C_i|^2}}{N * |\max\_x - \min\_x|^2}) \qquad (5)$$

4. 실험 및 결과

제안된 클러스터링 유전자 알고리즘의 성능을 평가하기 위해 9개의 적합도 평가 함수에 대하여 100개의 개체, 10개의 개체를 가지고 simple GA(기존의 유전자 알고리즘)에 적용한 결과와 6개의 클러스터링 방법을 이용하여 100개의 개체에 대하여 클러스터링 유전자 알고리즘을 적용한 결과를 비교 분석하였다. 각 실험은 30번 반복하였으며 그 평균값을 실험 결과로 사용하였다. 실험에 사용된 알고리즘은 표 1에, 실험에 사용된 환경 변수들은 표 2에 정리하였다.

표 1. 실험에 사용된 알고리즘

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

표 2. 실험에 사용된 환경 변수

	Pop_100	Cluster_100	Pop_10
집단의 크기	100	100	10
적합도 평가 횟수	100	10	10
클러스터 수	-	10	-
유전자 교차 확률	0.9		
돌연변이 확률	0.01		
총 세대 수	200		

Clustering GA에 사용된 클러스터링 알고리즘은 single linkage, complete linkage, average linkage, Ward의 방법, hard c-means 알고리즘, k-means 알고리즘으로 총 6개의 방법을 사용하였다.

표 3. 200세대 진화 후 각 알고리즘의 최대 적합도 값

	Pop100	Pop 10	S-L	C-L	A-L	Ward	H C-M	K-M
EF 1	75.15	71.59	74.41	75.13	75.44	74.76	73.99	74.57
EF 2	3835.3	3732.4	3818.2	3826.3	3825.4	3825.5	3795.8	3820.5
EF 3	18.40	13.90	18.10	17.27	16.9	18.17	17.27	17.43
EF 4	569.89	542.83	572.81	557.72	554.21	565.12	553.75	508.14
EF 5	0.8358	0.4942	0.6872	0.6695	0.7098	0.6777	0.6193	0.6654
EF 6	380.53	352.28	372.33	370.38	382.01	372.57	372.99	374.17
EF 7	6445.0	6135.8	6332.2	6446.2	6665.5	6416.6	6422.9	6408.9
EF 8	648.39	617.66	647.94	649.21	640.75	626.37	646.39	651.90
EF 9	21.68	21.61	21.68	21.70	21.67	21.68	21.69	21.69

표 3의 행은 9개의 적합도 평가 함수를 나타낸다. EF는 Evaluation Function을 나타내며, EF 1은 De Jong 1, EF 2는 De Jong 2, EF 3은 De Jong 3, EF 4는 De Jong 4, EF 5는 De Jong 5, EF 6은 Rastrigin, EF 7은 Schwefel, EF 8은 Griewangk, EF 9은 Ackley 함수를 각각 의미한다. 표 3의 열은 총 8개의 유전자 알고리즘을 의미한다. S-L 부터 K-M 까지는 Cluster 100의 각 클러스터링 알고리즘을 나타낸다.

표 3에서 Cluster 100의 경우 적합도 함수의 특성에 따라 각 클러스터링 방법별로 성능의 차이가 보이지만 그 차이는 크지 않았다.

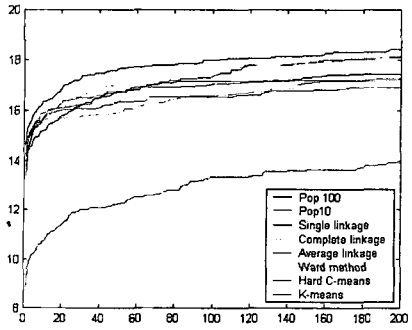

그림 4. De Jong Function 3에 대한 각 알고리즘의 진화 결과

그림 4에서 맨 아래의 가장 좋지 않은 성능을 보이는 곡선은 Pop 10의 결과이고 맨 위의 가장 좋은 성능을 나타내는 곡선은 Pop 100의 결과이다. Pop 100의 밑 부분에 몰려 있는 곡선들이 여러 클러스터링 유전자 알고리즘의 결과이다. 클러스터링 유전자 알고리즘의 성능이 Pop 100에는 다소 못 미치지만 Pop 10과 비교하면 월등히 나은 성능을 나타내는 것을 볼 수 있다.

참고문헌

[1] 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.

[2] Z. Michalewicz , "유전자 알고리즘," 그린, 1996.

[3] R. O. Duda, P. E. Hart and D. G. Stock, *Pattern Classification*, Wiley-Interscience Publication, 2001.

[4] D. Wishart, "Efficient hierarchical cluster analysis for data mining and knowledge discovery," *Computing Science and Statistics*, pp. 257-263, 1998.