# 개체 구조에 따른 유전자 알고리즘 기반의 문서 클러스터링 성능 비교†

# (Comparison of Document Clustering algorithm using Genetic Algorithms by Individual Structures)

최 임 천*, 쑹 웨 이**, 박 순 철***
(Lim Cheon Choi, Wei Song, and Soon Cheol Park)

**요 약** 유전자 알고리즘을 문서 클러스터링에 적용하기 위해서는 적절한 개체 구조가 필요하다. 기존의 유전자 알고리즘을 이용한 문서 클러스터링(DCGA)은 센트로이드 벡터 형식의 개체 구조를 사용하였다. 새로운 유전자 알고리즘을 이용한 문서 클러스터링(NDAGA)은 문서 할당 형식의 개체 구조를 사용한다. 본 논문에서는 문서 클러스터링에 더 적합한 개체 구조와 연산을 결정하기 위해 두 개체 구조의 차이에 따른 연산, 연산량, 클러스터링 수행 시간, 성능을 구체적으로 비교, 분석한다. 본 논문에서 수행한 다양한 실험에서 NDCGA가 DCGA와 비교하여 15%정도 더 빠른 수행 시간과, 약 5~10%정도 더 높은 성능을 보여, 문서 할당 형식의 개체 구조가 센트로이드 벡터 형식의 개체 구조 보다 문서 클러스터링에 적합한 것을 증명한다. 또한 NDCGA는 전통적인 클러스터링 알고리즘들(K-means, Group Average)에 비해서 15~20% 더 좋은 성능을 보였다.

**핵심주제어** : 유전자 알고리즘, 한글 문서 클러스터링, K-means, Group Average

**Abstract** To apply Genetic algorithm toward document clustering, appropriate individual structure is required. Document clustering with the genetic algorithms (DCGA) uses the centroid vector type individual structure. New document clustering with the genetic algorithm (NDAGA) uses document allocated individual structure. In this paper, to find more suitable object structure and process for the document clustering, calculation, amount of calculation, run-time, and performance difference between the two methods were analyzed. In this paper, we have performed various experiments using both DCGA and NDCGA. Result of the experiment shows that compared to DCGA, NDCGA provided 15% faster execution time, about 5~10% better performance. This proves that the document allocated structure is more fitted than the centroid vector type structure when it comes to document clustering. In addition, NDCGA showed 15-25% better performance than the traditional clustering algorithms (K-means, Group Average).

**Key Words** : Genetic Algorithm, Document Clustering, K-means, Group Average

† 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2011-0004389)
* 전북대학교 컴퓨터 공학과, 제1저자
** School of Information Technology, Jiangnan University
*** 전북대학교 전자정보공학부, 교신 저자

## 1. 서 론

대용량의 디지털 문서 정보를 효율적으로 분석하고 활용하기 위해서 문서 클러스터링에 대한 연구가 활발히 진행되고 있다[1,2,3]. 최근에는 트위터, 페이스북으로 대표되는 소셜 네트워크 서비스에 의하여 수집되는 정보를 분석하여 새로운 정보를 창출하는 데에도 많이 사용된다.

문서 클러스터링은 주어진 문서 집합에서 사전 정보 없이 유사한 문서들을 그룹화 하는 방법이다[1,2]. 문서를 그룹화 하는 방법에 따라서 크게 계층적 클러스터링과 비 계층적 클러스터링으로 나눌 수 있다[3,4]. 계층적 클러스터링 알고리즘은 모든 문서와 각각의 문서 사이에 계층 구조를 형성하도록 클러스터링 하는 것으로 상향식 클러스터링이 많이 사용된다. 계층적 클러스터링 알고리즘 중 가장 좋은 성능을 보이는 것으로 알려진 그룹 평균 클러스터링 알고리즘은 두 클러스터에 포함되는 문서들 사이의 유사도 평균값을 기준으로 문서를 클러스터링 하는 기법이다[3,5]. 대표적인 비 계층적 클러스터링 알고리즘인 K-means 클러스터링 알고리즘은 임의의 센트로이드 벡터를 생성한다. 생성된 센트로이드 벡터를 기준으로 문서를 재배치하고, 재배치된 문서들을 기준으로 센트로이드 벡터를 재정의 하는 과정을 통해서 문서를 클러스터링 한다[3,6]. 이러한 기존의 알고리즘들은 이해가 쉽고 수행 시간이 빨라 많이 사용하지만 높은 성능을 보이지 못한다는 단점이 있다[3,4].

이러한 성능상의 한계성을 해결하기 위한 방법 중 하나는 인공지능 알고리즘을 문서 클러스터링에 적용하는 것이다. 문서 클러스터링에 인공지능 알고리즘을 적용하기 위해서는 각 알고리즘의 수행에 필요한 요소와 문서 클러스터링에 필요한 요소를 대입하는 과정이 필요하다. 논문 [7,8]에서는 유전자 알고리즘의 요소들을 센트로이드 벡터 형식의 구조법에 대입하여 좋은 성능을 보이는 문서 클러스터링 알고리즘을 제안, 구현 하였다. 하지만 이러한 구조는 GA의 한 개체 표현에 있어서 필요한 정보량이 너무 많고, 표현가능한 상태가 광범위해진다. 따라서 최적해를 찾기 위한 연산이 복잡해지고 알고리즘에 긴 수행시간이 필요하게 된다.

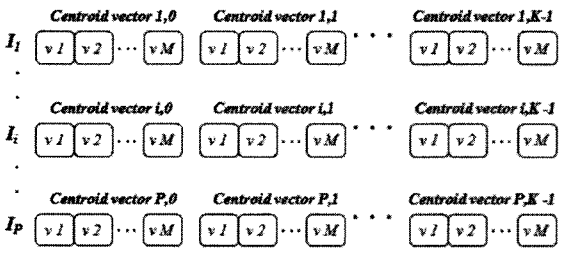
우리는 [9]에서 문서 할당 형식의 개체 구조를 가진 유전자 알고리즘을 이용한 문서 클러스터링에서 다양한 클러스터 측정법과 함께 적용하여 최적의 적합도 함수를 발견하였다. 본 논문에서는 기존의 유전자 알고리즘을 이용한 문서 클러스터링의 개체 구조[7,8]와 우리가 제안한 개체 구조[9]를 구체적으로 분석, 비교한다.

## 2. 기존의 유전자 알고리즘을 이용한 문서 클러스터링 알고리즘

유전자 알고리즘은 자연선택과 적자생존의 원리에 기반한 최적화, 검색 알고리즘이다[7,8]. 유전자 알고리즘의 기본 구성 요소는 개체, 염색체, 유전자, 유전자 연산(선택, 교배, 변이), 적합도 함수가 있다[10,11]. 유전자 알고리즘을 문서 클러스터링에 적용하기 위해서는 이러한 구성 요소들을 적절히 대입한 개체 구조와 연산에 대한 정의가 필요하다[8,10].

### 2.1 개체 구조

논문 [7,8]의 문서 클러스터링에 이용한 유전자 알고리즘(Document Clustering with Genetic Algorithm : DCGA)은 센트로이드 벡터 형식의 개체 구조를 가진다. 따라서 DCGA의 개체는 센트로이드 정보를 가지고, 각 센트로이드 정보는 특징 벡터로 나타난다. <그림 1>은 DCGA의 개체 구조를 보여준다.




<그림 1> DCGA의 개체 구조

<그림 1>에서 DCGA는 $P$개의 개체를 가진다. 각 개체($I_i$)는 $K$개의 센트로이드 정보를 가지고, 각각의 센트로이드는 $M$개의 특징 벡터($v$)로 표현된다. <그

림 1>에서 나타난 DCGA의 전체 개체의 크기는 다음과 같다.

$$Size\ of\ DCGA_I = K \times M \times P \quad (1)$$

식 (1)의 $K$는 클러스터의 수를 의미하고, $M$은 특징 벡터의 수를, $P$는 개체의 수를 의미한다. 각각의 특징 벡터 값이 가지는 값의 범위는 0~1 사이의 실수가 된다.

일반적으로 $M$의 크기는 매우 크고 실수로 표현되다. 따라서 개체를 저장하기 위한 데이터 크기가 크고, 표현 가능 상태가 다양해, DCGA는 최적의 해를 구하기 어렵다.

### 2.2 DCGA의 연산 구조

DCGA는 개체 초기화, 유전자 연산(선택, 교배, 변이)의 연산 구조를 가진다[7,8]. DCGA 연산구조의 세부적인 내용은 다음과 같다.

#### 개체 초기화

센트로이드 형식의 DCGA는 개체 초기화 연산에서 임의 문서의 특징 벡터로 센트로이드 정보를 정의한다[8]. <그림 2>는 DCGA의 개체 초기화 연산을 프로그램으로 기술한 것이다.

```
개체 생성 : I_DCGA = {c_0, ..., c_{K-1}}
// I_DCGA : DCGA의 개체 형식
// c_i : i번째 센트로이드 벡터
for i = 0 to K-1
  j ← IRNAD(0, N-1)  // j : 임의의 문서 번호
  c_i in I_DCGA ← d_j
  // d_j : j번째 문서 벡터, N : 전체 문서의 수
end for
```

<그림 2> DCGA의 개체 초기화

<그림 2>는 임의의 문서를 선택하여, 선택된 문서의 벡터 정보를 개체의 센트로이드에 할당하는 DCGA의 개체 초기화 과정을 보여준다. <그림 2>에서 사용된 $IRAND(i,j)$함수는 $i \sim j$범위의 임의의 정수를 반환한다.

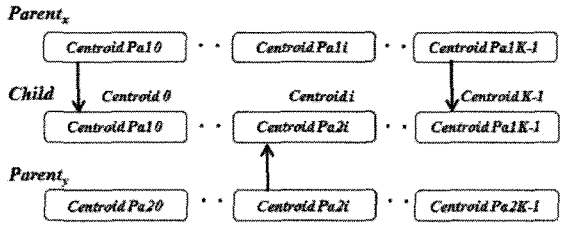
#### 선택, 교배, 변이 연산

DCGA는 fittest concept 선택 연산으로 개체를 선택하고, 일점 교차 연산으로 개체를 생성한다[7,8]. 일점 교차 연산은 두 개체에서 한 점의 정보를 교환하는 것이지만, 센트로이드 벡터의 한 점을 교환하는 것은 큰 의미가 없다. 이에 따라 DCGA에서는 센트로이드 벡터를 하나의 단위로 삼아 교배 연산에 사용하였다[7,11]. 이를 프로그램으로 기술하면 <그림 3>와 같다.

```
개체 생성 : Child              // DCGA의 개체 형식
개체 선택 : Parent_x  Parent_y // DCGA의 개체 형식
c_num ← IRAND(0, K-1); // 센트로이드 정보 선택
for i = 0 to K-1
  if(i == c_num) c_i in Child ← c_i in Parent_y
  else c_i in Child ← c_i in Parent_x
end for
```

<그림 3> DCGA의 교배 알고리즘

DCGA의 교배 연산에서 새로 생성되는 개체(Child)는 선택된 하나의 정보를 $Parent_y$로부터, 그 이외의 정보를 $Parent_x$로부터 센트로이드 벡터 단위로 받게 된다. <그림 4>는 DCGA의 교배 연산의 예를 그림으로 표현한 것이다.




<그림 4> DCGA의 교배 연산

<그림 4>에서 새로 생성되는 개체(Child)는 선택된 $i$번째 센트로이드 정보를 $Parent_y$로부터 그 외의 다른 센트로이드 정보를 $Parent_x$로부터 물려받는다.

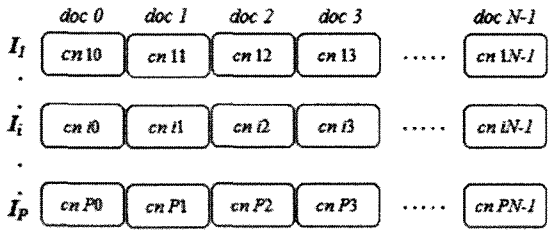
<그림 3>에서 나타나듯이 DCGA의 교배 연산은 $K$개의 센트로이드와 $M$개의 특징 벡터 정보를 연산하게 된다. 따라서 DCGA의 교배 연산은 $T(K \times M)$의 시간복잡도를 가진다.

## 3. 새로운 개체 구조의 유전자 알고리즘을 이용한 문서 클러스터링

DCGA는 개체 구조에 클러스터 센트로이드 벡터를 적용하여 문서 클러스터링 알고리즘을 구현하였다[7,8]. 이러한 개체 구조에 의하여 유전자 알고리즘의 개체를 표현하는 데이터의 크기가 커지고 표현 가능한 상태가 많아 최적화 알고리즘의 수행 시간이 매우 오래 걸리는 단점을 가지게 되었다[12]. 본 논문에서는 [9]에서 이용한 문서 할당 형식의 개체 구조를 문서 클러스터링에 적용하여 알고리즘의 수행시간을 줄이면서 성능을 높이는 것을 증명한다. 문서 할당 형식의 유전자 알고리즘을 이용한 문서 클러스터링을 NDCGA라고 명명한다.

### 3.1 개체 구조

NDCGA는 각각의 문서와 문서를 포함하는 클러스터의 관계를 직접적으로 표현하여 유전자 알고리즘을 문서 클러스터링에 적용하였다. <그림 5>는 본 논문에서 사용한 유전자 알고리즘의 개체 구조를 그림으로 표현한 것이다.




<그림 5> NDCGA 개체 구조

<그림 5>의 NDCGA는 $P$개의 개체를 가진다. 개체 $I_i$는 $N$개의 유전자를 가지게 된다. 유전자의 인덱스는 문서의 번호를 나타내고, 그 값(CN)은 문서가 포함되는 클러스터의 번호를 나타낸다. 따라서 NDCGA의 개체 크기는 다음과 같다.

$$Size\ of\ NDCGA_I = N \times P \qquad (2)$$

식 (2)의 $N$은 전체 문서의 수를 의미하고, $P$는 개체의 수를 의미한다. 또한 값은 $0 \sim K-1$의 정수가 된다. 특징 벡터의 수($M$)가 문서의 수보다 훨씬 크고, 실수가 한정된 범위의 정수보다 광대한 표현 범위를 가진다는 것을 감안한다면, 식 (2)의 NDCGA 개체 크기는, 식 (1)에서 보여진 DCGA 개체 크기와 비교하여 개체를 표현하는데 필요한 정보와 개체의 표현 가능 범위가 획기적으로 줄어든 것을 확인할 수 있다.

### 3.2 NDCGA의 연산 구조

**개체 초기화**

문서 할당 형식의 NDCGA는 개체 초기화 연산에서 임의의 클러스터 번호로 문서의 할당 정보를 정의한다. <그림 6>는 NDCGA의 개체 초기화 연산 프로그램으로 기술한 것이다.

```
개체 생성 : I_NDCGA = {cn_0,...,cn_N-1}
// I_NDCGA : NDCGA의 개체 형식
// cn_i : i번째 문서의 클러스터 번호
for i = 0 to N-1
  cn_i in I_NDCGA ← IRNAD(0,K-1)
end for
```

<그림 6> 개체 초기화(NGA)

NGACD는 개체 초기화 과정에서 각각의 문서에 문서가 속하는 클러스터의 번호($0 \sim K-1$)를 각 개체에 할당한다.

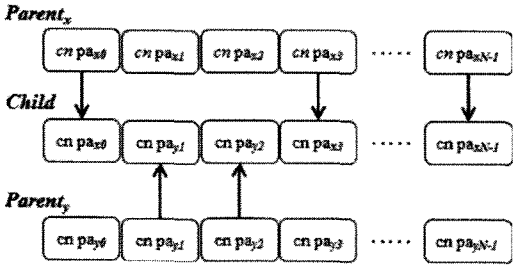
**선택, 교배 연산**

NDCGA에서는 룰렛 휠 선택 연산을 이용하여 부모 개체를 선택하고, 임계값 0.5의 균등 교차 연산을 이용하여 새로운 개체의 정보를 생성하였다[13]. NDCGA에서의 교배 연산 단위는 각각의 문서 할당 정보로 진행된다. 이를 프로그램으로 기술하면 <그림 7>과 같다.

```
개체 생성 : Child                    // NDCGA의 개체 형식
개체 선택 : Parent_x  Parent_y  // NDCGA의 개체 형식
for i = 0 to K-1
  p_num ← IRAND(0,K-1);  // 부모 개체 선택
  if(p_num == 0) cn_i in Child ← c_i in Parent_x
  else c_i in Child ← c_i in Parent_y
end for
```

<그림 7> NDCGA 교배 알고리즘

NDCGA의 새로 생성되는 개체는 $N$개의 문서 할당 정보를 50%의 확률로 $Parent_x$와 $Parent_y$에게서 물려받는다[13]. <그림 8>는 NDCGA의 교배 연산의 예를 그림으로 표현한 것이다.



<그림 8> NDCGA의 교배 연산

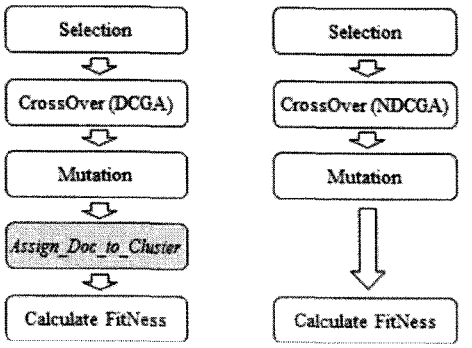
<그림 8>의 새로 생성되는 개체 : *Child*는 0, 3,..., $N-1$번째의 정보는 $Parent_x$로부터 1, 2,...,번째의 정보는 $Parent_y$로부터 정보를 받는다.

<그림 7>에 나타나듯이 NDCGA의 교배 연산은 N개의 정보를 연산한다. 따라서 NDCGA의 교배 연산은 $T(N)$의 시간 복잡도를 가진다. 일반적으로 $K \times M > N$이므로 DCGA의 교배 연산이 NDCGA의 교배 연산보다 복잡함을 알 수 있다.

### 3.3 기존 시스템과의 비교

**DCGA, NDCGA연산 비교**

NDCGA와 DCGA은 개체 구조 차이에 의하여 연산 과정의 차이가 생긴다. <그림 9>는 DCGA, NDCGA의 연산 과정을 그림으로 표현한 것이다.



(a) DCGA 연산 과정 (b) NDCGA 연산 과정

<그림 9> DCGA, NDCGA 연산 과정

<그림 9>에서는 NDCGA와 DCGA의 한 세대에 필요한 연산들을 보여준다. <그림 9>를 통해서 우리는 앞서 언급한 선택연산의 차이 이외에도 DCGA에 *Assign_Doc_to_Cluster* 연산이 더 필요한 것을 알 수 있다. *Assign_Doc_to_Cluster* 연산은 문서가 어느 클러스터에 속하는 지에 대한 정보를 계산하는 것으로 K-means 클러스터링 알고리즘의 문서 재배치 연산과 같다[3]. 따라서 $K$개의 센트로이드와 $N$개의 문서 사이의 유사도를 연산하는 것으로 $T(K \times N)$의 시간복잡도를 가진다[3]. 반면에 NDCGA에서는 문서와 클러스터의 할당 정보를 개체 구조로 가지기 때문에 그러한 과정이 필요하지 않게 된다.

앞서 설명한 선택 연산의 복잡도와 DCGA에 필요한 추가적인 연산에 의해서 한 개체, 한 세대 당 DCGA가 NDCGA에 비하여 더 복잡한 연산과 계산비용을 가짐을 알 수 있다.

## 4. 실험 및 결과 분석

본 논문에서는 유전자 알고리즘을 이용하는 문서 클러스터링 알고리즘에 적용된 기존의 개체 구조와 새로운 개체 구조의 차이에 따른 알고리즘 수행 시간과 성능을 비교, 분석한다. "한국일보-20000/한국일보-40075 문서범주화 실험문서 집합"의 데이터 셋을 사용하여 알고리즘을 테스트하였다[14]. 4개의 주제를 선택하고, 각각의 주제에는 50개의 문서를 할당하였다. 4개의 주제를 포함하는 Topic Set을 2개 만들어 실험을 진행하였다. <표 1>은 Topic Set에 포함되는 주제를 보여준다.

<표 1> Topic Set

| Topic Set | 주 제 |
|---|---|
| Topic 01 | 여가생활 실내 TV, 사회 사회질서 사건사고(교통)<br>문화와 종교 생활 주거 결혼, 정치 외교(대북) |
| Topic 02 | 여가생활 실내 공연, 사회 사회질서 사건사고(절도)<br>문화와 종교 스포츠 육상, 정치 외교(대일) |

VSM(Vector Space Model)을 이용하여 Topic Set의 문서들을 표현하였다. 사용된 용어 가중치$w_{ij}$는 다

음과 같이 정의된다[3].

$$w_{ij} = tf_{ij} \times \log \frac{N}{df_j} \quad (4)$$

$tf_{ij}$는 $i$번째 문서의 $j$번째 용어의 빈도수, $df_j$는 $j$번째 용어의 문서 빈도수, $N$은 전체 문서의 수를 의미한다.

DCGA의 센트로이드 벡터 형식은 큰 차원의 특징 벡터를 처리하기 어렵다. 이를 해결하기 위해서 DCGA에서는 Latent Semantic Indexing를 이용하여 용어 벡터의 차원 수를 감소시키며 용어 사이의 의미 관계성을 부여한 특징 벡터 행렬을 정의한다[8,15,16]. 특징 벡터 행렬 $C$은 다음과 같이 정의된다[8,15].

$$C = DU_k \quad (5)$$

$D$는 문서 벡터 행렬을, $U$는 SVD를 이용한 특징 벡터 행렬의 $U$ 행렬을 의미한다. 본 논문에서는 실험에서 가장 효율적인 성능을 보여준 $k$ = 200의 값을 이용하였다[8,16].

유전자 알고리즘의 성능에 가장 큰 영향을 미치는 인자는 적합도 함수이다[11]. DCGA에서는 적합도 함수 $CosSum$을 이용하였다. $CosSum$은 클러스터 센트로이드와 클러스터에 배치된 문서 사이의 유사도를 이용한 것으로 계산식 (6)을 이용한다.

$$CosSum = \sum_{i=1}^{K} Clusum_i \quad (6)$$
$$where\ Clusum_i = \sum_{j=1}^{NC_i} CosSim(c_i, d_{ij})$$

$NC_i$는 $i$번째 클러스터에 포함되는 문서의 수를 의미하고, $c_i$는 $i$번째 클러스터의 센트로이드 벡터를, $d_{ij}$는 $i$번째 클러스터의 $j$번째 문서 벡터를 의미한다. 유사도 평가는 코사인 유사도($CosSim$)를 이용하였고, 더 높은 결과 값을 가지는 개체가 더 좋은 클러스터의 결과를 가진다고 평가된다.

NDCGA는 문서가 할당된 클러스터 정보를 이용한다. 따라서 센트로이드 벡터 형식의 적합도 함수를 이용하는 것은 적절하지 못하다. 따라서 NDCGA에서는 클러스터와 문서 사이의 정보를 효율적으로 이용할 수 있는 적합도 함수($AveSim$)를 사용하였다[6,9]. $AveSim$은 클러스터에 포함되는 문서들 간의 유사도 평균을 이용한 것으로 계산식 (7)을 이용한다[6,9].

$$AveSim = \frac{1}{K} \sum_{i=0}^{K} CluSim_i \quad (7)$$
$$where\ CluSim_i = \sum_{j=0}^{NC_i - 1} \sum_{k=j+1}^{NC_i} CosSim(d_{ij}, d_{ik})$$

$K$는 클러스터의 개수를 의미하고, $NC_i$는 $i$번째 클러스터에 포함되는 문서의 수를 의미하며, $d_{ij}$는 $i$번째 클러스터의 $j$번째 문서를 의미한다.

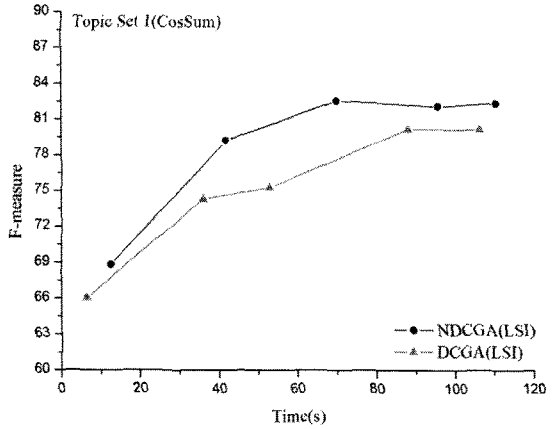
클러스터링의 성능 평가는 F-measure를 사용하였다. F-measure는 정확률과 재현율의 분기점으로 이루어지고, 계산식 (8)을 따른다[2,3].

$$F-measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (8)$$

본 논문에서는 먼저 개체 구조에 따른 유전자 알고리즘의 성능 비교를 위하여, 각각의 적합도 함수와 Topic Set에 대한 알고리즘 수행 시간에 따른 성능을 비교하였다.

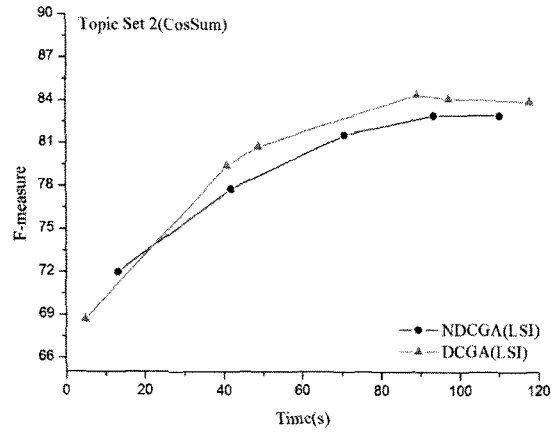
적합도 함수 $CosSum$은 개체, 세대마다 모든 클러스터 센트로이드와 클러스터에 포함되는 문서들 사이의 유사도를 평가하게 된다. 따라서 특징 벡터의 차원이 큰 VSM에서는 많은 연산량에 의해서 적정한 시간에 제대로 된 성능을 보이지 못한다. 이에 따라 적합도 함수 $CosSum$에서는 NDCGA와 DCGA에서 모두 LSI를 이용한 특징 벡터 행렬을 이용하여 알고리즘의 성능을 평가하였다. 또한 개체 구조상 센트로이드 벡터 정보가 없는 NDCGA에서는 추가적인 연산을 통해서 센트로이드 벡터 정보를 계산하였다.

<그림 10>, <그림 11>은 Topic Set 1과 2의 데이터에서 $CosSum$의 적합도 함수를 가질 때의 NDCGA와 DCGA의 알고리즘 수행시간에 따른 성능을 그래프로 표현한 것이다.




<그림 10> 알고리즘 성능(Topic Set1, CosSum)

<그림 10>을 통해 우리는 시간에 대한 성능에서 NDCGA가 DCGA보다 항상 더 좋은 성능을 가지는 것을 확인할 수 있다. 또한 NDCGA는 70초, DCGA는 90초 이후의 시간에서 성능의 변화가 거의 없는 것을 확인할 수 있다.



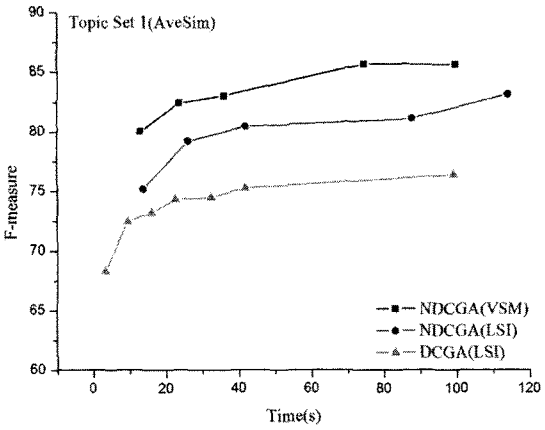

<그림 11> 알고리즘 성능(Topic Set2, CosSum)

<그림 11>을 통해 우리는 시간에 대한 성능에서 DCGA가 20초 이후부터 NDCGA보다 더 좋은 성능을 가짐을 확인할 수 있다. 또한 NDCGA는 95초, DCGA는 90초 이후의 시간에서 성능의 변화가 거의 없는 것을 확인할 수 있다.

<그림 10>, <그림 11>을 통해서 적합도 함수 *CosSum*에서는 NDCGA, DCGA 두 알고리즘의 우열을 가리기 힘들다는 것을 확인할 수 있다.

적합도 함수 *AveSim*은 알고리즘 시행 초기 문서와 문서 사이의 유사도 계산이 완료되면 세대와 개체에 따른 적합도 평가에 많은 연산이 필요하지 않게 된다. 따라서 LSI를 반드시 이용해야 했던 *CosSum*과는 다르게 VSM를 이용하여 그 성능의 평가가 가능하다. 다만 DCGA의 경우에는 개체, 세대마다 문서 재배치 과정이 필요하기 때문에 여전히 VSM에서는 적정시간내의 성능평가가 불가능하였다. 이에 따라 NDCGA는 LSI와 VSM을 이용하여, DCGA은 LSI를 이용하여 알고리즘의 성능을 평가하였다.

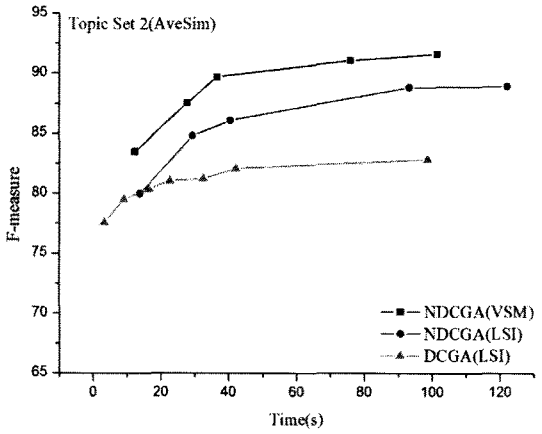
<그림 12>는 Topic Set 1에서 *AveSim*의 적합도 함수를 가질 때의 NDCGA와 DCGA의 알고리즘 수행 시간에 따른 성능을 그래프로 표현한 것이다.




<그림 12> 알고리즘 성능(Topic Set1, *AveSim*)

<그림 12>를 통해 우리는 시간에 대한 성능에서 NDCGA가 DCGA 보다 항상 좋은 성능을 가짐을 확인할 수 있다. NDCGA에는 LSI 보다 VSM을 이용할 때 더 좋은 성능을 보이는 것을 확인할 수 있다. 또한 NDCGA(VSM)은 약 70초, NDCGA(LSI)는 약 90초, DCGA(LSI)는 40초 이후의 시간에서 성능의 변화가 거의 없는 것을 확인할 수 있다.

<그림 13>은 Topic Set 2에서 *AveSim*의 적합도 함수를 가질 때의 NDCGA와 DCGA의 알고리즘 수행 시간에 따른 성능을 그래프로 표현한 것이다.




<그림 13> 알고리즘 성능(Topic Set2, *AveSim*)

<그림 13>에서도 <그림 12>와 마찬가지로 시간에 대한 성능에서 NDCGA가 DCGA보다 항상 좋은 성능을 가짐을 확인할 수 있다. NDCGA에는 LSI 보다 VSM을 이용할 때 더 좋은 성능을 보이는 것을 확인할 수 있다. NDCGA(VSM)은 70초, NDCGA(LSI)는 90초, DCGA(LSI)는 40초 이후에서 성능의 변화가 거의 없는 것을 확인할 수 있다.

<그림 12>, <그림 13>에서 DCGA는 NDCGA에 비해서 시간의 증가에도 성능이 저조한 것을 알 수 있다. 이는 센트로이드 벡터의 개체 구조가 *AveSim*을 이용한 최적화에 적합하지 않은 것을 보여준다. 반면 NDCGA는 *CosSum*에도 적합하지만, *AveSim*에 더 적합하다고 할 수 있다.

<표 2> 클러스터링 알고리즘 성능 비교

| Cluster Alorithm | Topic Set 1 | | Topic Set 2 | |
|---|---|---|---|---|
| | Precision | Recall | Precision | Recall |
| K-means | 66.53 | 67.41 | 76.18 | 75.93 |
| Group Average | 56.38 | 55.25 | 83.68 | 79.51 |
| DCGA (LSI, CosSum) | 81.62 | 78.76 | 86.3 | 82.53 |
| NDCGA (LSI, CosSum) | 82.65 | 81.5 | 83.43 | 82.93 |
| DCGA (LSI, AveSim) | 76.58 | 76.16 | 82.32 | 82.58 |
| NDCGA (LSI, AveSim) | 80.9 | 80.33 | 88.85 | 88.83 |
| NDCGA (VSM, AveSim) | 85.56 | 85.71 | 91.12 | 91.04 |

<표 2>는 알고리즘의 객관적인 성능 비교를 위하여 Topic Set 1, 2에 대한 전통적인 클러스터링 알고리즘들(K-means, Group Average)의 성능과 유전자 알고리즘을 이용한 문서 클러스터링 알고리즘들의 성능을 비교한 것이다. <표 2>에서 전통적인 클러스터링 알고리즘에 비하여 제안한 알고리즘이 15~20% 더 좋은 성능을 나타냈다. DCGA의 경우 *CosSum*이라는 적합도 함수에서 최적화된 성능을 보였고, NDCGA는 DCGA에 비해서 적합도 함수의 영향을 크게 받지 않는 것을 확인하였다. 또한 *AveSim*과 VSM을 이용한 NDCGA가 *CosSum*과 LSI를 이용한 DCGA에 비하여 15% 정도 더 빠른 수행시간을 가지며, 5~10% 더 좋은 성능을 나타내는 것을 확인하였다.

## 5. 결론 및 향후 방향

본 논문은 기존의 DCGA가 가지는 센트로이드 백터 형식의 개체 구조에 따른 데이터 표현, 유전자 연산과 NDCGA가 가지는 문서 할당 형식의 새로운 개체 구조에 따른 데이터 표현, 유전자 연산을 비교 분석하여 새로운 개체구조와 연산구조를 가지는 유전자 알고리즘이 문서 클러스터링에 더 적합하다는 것을 증명하였다.

본 논문에서 실행한 다양한 환경의 실험을 통하여 NDCGA는 전통적 클러스터링 알고리즘(Group Average, K-means) 보다 15~20%정도 더 좋은 성능을 가지는 것을 확인하였다. 또한 Topic-Sets, 적합도 함수(*AveSim*, *CosSum*), 문서 표현 방식(VSM, LSI)의 변화를 통한 다양한 실험에서 NDCGA가 기존의 DCGA에 비해서 약 15% 정도의 빠른 수행 시간과 5~10%정도의 성능 향상을 보여, NDCGA가 문서 클러스터링에 더 적합한 형태의 개체 구조와 연산을 가짐을 증명하였다.

하지만 여전히 유전자 알고리즘이 가지는 연산의 복잡성에 의하여 전통적인 클러스터링 알고리즘들에 비해 긴 수행 시간을 가지는 단점이 존재하였다. 이러한 문제를 해결하기 위해서 유전자 알고리즘을 문서 클러스터링에 적용했을 때 나타나는 특징을 분석하여, 문서 클러스터링에 더 적합한 성능을 보여주는 유전

자 문서 클러스터링 알고리즘을 제안, 구현할 것이다.

## 참 고 문 헌


[1] B. Y. Ricardo and R. N. Berthier, Modern information retrieval, Addison Wesley, 1999.

[2] 정영미, "정보 검색 연구", 구미무역, 2005

[3] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze, "Introduction to Information Retrieval", 2008

[4] Beil, F., Ester, M., & Xu, X. (2002), "Frequent term-based text clustering". International knowledge Discovery and Data Mining, KDD'02, Edmonton, Alberta, Canada, 436-442

[5] S. Selim and M. Ismail, "K-means-type algorithm generalized convergence theorem and characterization of local optimality", IEEE Trans. Pattern Anal. Mach Intell. vol. 6, pp. 81-87, 1984.

[6] YING ZHAS, GEORGE KARYPIS, "Hierarchical Clustering Algorithms for Document Datasets", Data Mining and Knowledge Discovery, 10, 141-168, 2005

[7] W. Song, S.C. Park, Genetic algorithm-based text clustering technique, LNCS 4221 (2006) 779_782.

[8] W. Song, S.C Park, "Genetic algorithm for text clustering based on latent semantic indexing", Computers and Mathematics with Applications, vol. 57, pp. 1901-1907, 2009

[9] 최임천, 박순철, "클러스터 측정과 유전자 알고리즘을 이용한 문서 클러스터링", 한국정보처리학회 추계학술대회 논문집, 제17권, 2호, pp. 490- 493, 2010.11

[10] L. Davis(Ed.), "Handbook of Genetic Algorithms", Van Nostrand Reinhold, New York, 1991

[11] 김대희, 박상호, "분류시스템의 분류 규칙 발견을 위한 유전자 알고리즘", 한국산업정보학회 논문지, 제9권, 4호, pp.16 - 25, 2004

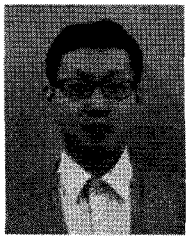
[12] David E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison Wesley, 1989.

[13] U. Maulik, S. Bandyopadhyay, "Genetic algorithm-based clustering technique", Patten Recognition. vol. 33, pp. 1455-1465, 2000

[14] http://www.kristalinfo.com/TestCollections/

[15] 리청화, 변동률, 박순철, "한글문서분류에 SVD를 이용한 BPNN 알고리즘", 한국산업정보학회 논문지, 제15권 2호, pp. 49-57, 2010. 6

[16] S.C. Deerwester, S.T. Dumais, T.K Landauer, G.W. Furnas, R.A. Harshman, "Indexing by latent semantic analysis", J. Amer. Soc. Inform. Sci. 41(1990)

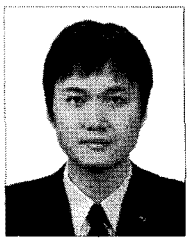


**최 임 천** (Lim Cheon Choi)

- 학생 회원
- 2007년 : 전북대학교 컴퓨터공학 (공학사)
- 2009년 : 전북대학교 컴퓨터 공학 (공학 석사)
- 2009년 3월 ~ 현재 : 전북대학교 컴퓨터공학 (박사 과정)
- 관심분야 : 정보 검색, 문서 클러스터링, 용어 네트워크 구축, 시멘틱 웹 서비스

**박 순 철** (Soon Cheol Park)

- 평생회원
- 1979년 2월 : 인하대학교 공과대학 (공학사)
- 1991년 12월 : 미국) 루이지아나 주립대학 (전산학박사)
- 1991년 ~ 1993년 : 한국전자통신연구원 근무
- 1993년 ~ 현재 : 전북대학교 전자정보공학부 교수
- 관심분야 : 정보검색, 시멘틱 웹, 온톨로지



**쑹 웨 이** (Wei Song)

- 학생 회원
- 2004년 6월 : South-Central University for Nationalities, Computer Science and Technology 학사 졸업
- 2006년 8월 : 전북대학교 정보통신공학과 (정보통신 석사)
- 2009년 8월 : 전북대학교 정보통신공학과 (정보통신 박사)
- 관심분야 : 정보검색, 시멘틱 웹, 온톨로지, 진화 연산 알고리즘

논 문 접 수 일 : 2011년 05월 25일
1차수정완료일 : 2011년 06월 15일
2차수정완료일 : 2011년 07월 05일
게 재 확 정 일 : 2011년 08월 08일