

파레토 지배순위와 밀도의 가중치를 이용한 다목적 최적화 진화 알고리즘

장 수 현[†]

요 약

진화 알고리즘은 여러 개의 상충하는 목적을 갖는 다목적 최적화 문제를 해결하기에 적합한 방법이다. 특히, 파레토 지배관계에 기초하여 개체의 적합도를 평가하는 파레토 기반 진화알고리즘들은 그 성능에 있어서 우수한 평가를 받고 있다. 최근의 파레토 기반 진화알고리즘들은 전체 파레토 프론트에 균일하게 분포하는 해집합의 생성을 위해 개체들의 밀도를 개체의 적합도를 평가하기 위한 하나의 요소로 사용하고 있다. 그러나 밀도의 역할은 전체 진화과정에서 중요한 요소가 되기보다는 파레토 프론트에 어느 정도 수렴된 후, 개체의 균일 분포를 만들기 위해 사용된다. 본 논문에서 우리는 파레토 지배 순위와 밀도에 대한 임의가중치를 적용한 다목적 최적화 진화알고리즘을 제안한다. 제안한 알고리즘은 진화 개체의 적합도를 평가하기 위해 파레토 순위와 밀도에 대한 임의의 가중치를 적용하므로 전체 진화과정에서 파레토 순위와 밀도가 비슷한 영향을 미치도록 하였다. 또한, 제안한 방법을 6개의 다목적 최적화 문제에 적용한 결과 비교적 우수한 결과를 보였다.

Evolutionary Multi-Objective Optimization Algorithms using Pareto Dominance Rank and Density Weighting

Su-Hyun Jang[†]

ABSTRACT

Evolutionary algorithms are well-suited for multi-objective optimization problems involving several, often conflicting objective. Pareto-based evolutionary algorithms, in particular, have shown better performance than other multi-objective evolutionary algorithms in comparison. Recently, pareto-based evolutionary algorithms uses a density information in fitness assignment scheme for generating uniform distributed global pareto optimal front. However, the usage of density information is not important elements in a whole evolution path but plays an auxiliary role in order to make uniform distribution. In this paper, we propose an evolutionary algorithms for multi-objective optimization which assigns the fitness using pareto dominance rank and density weighting, and thus pareto dominance rank and density have similar influence on the whole evolution path. Furthermore, the experimental results, which applied our method to the six multi-objective optimization problems, show that the proposed algorithms show more promising results.

키워드 : 다목적 최적화(Multi-Objective Optimization), 진화알고리즘(Evolutionary Algorithms), 파레토 최적해(Pareto Optimal Solution), 파레토 지배순위(Pareto Dominance Rank), 밀도(Density)

1. 서 론

많은 실세계 문제들은 서로 상충하는 여러 개의 목적을 갖는 다목적 최적화(Multi objective optimization) 문제이다. 그리고 다목적 최적화 문제는 하나의 최적 해를 갖기 보다는 파레토 최적해(Pareto optimal solution)라 불리는 절충하는 해들의 집합을 갖는다. 이러한 다목적 최적화 문제를 해결하기 위한 방법으로 최근에는 진화알고리즘을 이용한 방법들이 주된 연구 방향이 되어 가고 있다.

진화알고리즘은 선형프로그래밍(linear programming)이나

기울기 탐색(gradient search)과 같은 엄격한 방법으로 해결하기 어려운 다목적 최적화 문제에서 비교적 최적 파레토 프론트에 근사하는 해들을 만들어 낼 수 있는 대안으로 알려지고 있다. 1980년대 중반부터 진화알고리즘을 이용하여 다목적 최적화 문제를 해결하기 위한 관심이 증대되었고[1-3], 1993년과 1994년에는 MOGA(Multi Objective Genetic Algorithms)[4], NPGA(Niched Pareto GA)[5], NSGA(Non-dominated Sorting GA)[6]와 같은 파레토 기반 진화알고리즘들이 제안되어지면서 진화적 다목적 최적화 알고리즘들이 한번의 실행으로 최적의 절충 해들의 집합에 근사할 수 있음을 보여주었다. 그리고 최근에 제안된 NSGA-II, SPEA-II와 같은 다목적 최적화 진화알고리즘들은 전

[†] 정 회 원 : 명지대학교 대학원 컴퓨터공학과
논문접수 : 2003년 8월 7일, 심사완료 : 2004년 3월 25일

체 파레토 프론트에 균일하게 분포된 해집합의 생성을 위해 개체들의 밀도를 개체의 적합도 평가를 위한 하나의 요소로 사용하고 있다[7, 8]. 그러나 NSGA-II, SPEA-II에서 사용된 개체의 밀도는 같은 파레토 순위의 개체들에 대한 순서화를 위해 부가적인 요소로만 사용된다. 따라서 개체 밀도 평가의 역할은 전체 진화과정에서 중요한 요소가 되기보다는 파레토 프론트에 어느 정도 수렴된 후, 개체의 균일 분포를 만들기 위해 사용된다. 그러나 밀도정보의 부가적인 사용은 균일 분포의 파레토 프론트 생성을 보장하지만 많은 지역 파레토 프론트를 가지는 문제에서 개체의 다양성을 보장하지 못하기 때문에 전역 파레토 프론트에 근사하지 못할 수 있는 약점을 가지고 있다.

본 논문에서 우리는 파레토 순위와 밀도의 임의의 가중치를 적용한 다목적 최적화 진화알고리즘을 제안한다. 제안한 알고리즘은 진화 개체의 적합도를 평가하기 위해 파레토 순위와 밀도에 대한 임의의 가중치를 적용하므로 전체 진화과정에서 파레토 순위와 밀도가 비슷한 영향을 미치도록 한다. 따라서, 현 세대에서 비지배 개체는 아니지만 밀도가 작은 개체가 선택될 가능성이 존재하기 때문에 개체의 다양성을 전체 진화과정에서 유지하도록 하였다. 또한, 제안한 방법을 다양한 형태의 문제들에 적용한 결과를 NSGA-II의 결과와 비교한다.

2. 다목적 최적화문제

일반적으로 다목적 최적화 문제는 n 개의 목적들에 대해 m 개의 결정변수를 사상시키는 벡터함수 f 로 설명될 수 있다. 즉,

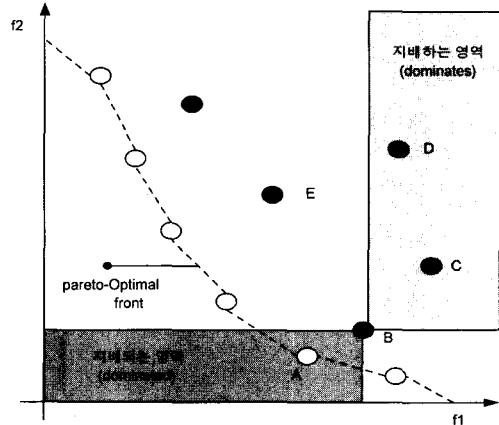
$$\begin{aligned} \min. / \max. \vec{y} = f(\vec{x}) &= (f_1(\vec{x}), f_2(\vec{x}), \dots, f_n(\vec{x})) \\ \vec{x} &= (x_1, x_2, \dots, x_m) \in X \\ \vec{y} &= (y_1, y_2, \dots, y_n) \in Y \end{aligned} \quad (1)$$

식 (1)에서 \vec{x} 는 결정벡터(decision vector), X 는 파라미터 공간(parameter space), \vec{y} 는 목적벡터(objective vector), Y 는 목적 공간(objective space)이라 한다. 다목적 최적화 문제의 해의 집합은 어떤 목적함수의 값을 감소시키지 않고는 다른 어떤 함수의 값도 향상시킬 수 없는 목적 벡터들과 상응하는 모든 결정벡터 들로 구성되고, 이러한 벡터들을 파레토 최적해(pareto optimal solution)라 한다. 파레토 최적화에 대한 정의는 다음과 같다. 최소화 문제를 가정하고, 두개의 결정벡터 $a, b \in X$ 가 존재한다고 할때,

$$\begin{aligned} \forall i \in 1, 2, \dots, n : f_i(a) \leq f_i(b) \wedge \\ \exists j \in 1, 2, \dots, n : f_j(a) < f_j(b) \end{aligned} \quad (2)$$

인 결정벡터 a 는 결정벡터 b 를 지배(dominate)한다고 한다 (또한 $a < b$ 로 표기한다).

또한, 임의의 결정벡터 a 에 대하여, 결정벡터들의 부분 집합 X' 내의 어떤 벡터도 a 를 지배하지 못한다면, 결정벡터 a 는 X' 에 의해 비지배(nondominated)된다고 한다. 그리고 전체 탐색 공간 내에서 비지배 되는 모든 결정벡터들을 파레토 최적해 집합 또는 최적 파레토 프론트라 한다.



(그림 1) 목적 공간에서 최적 파레토 프론트와 결정벡터 사이의 지배관계의 예

(그림 1)에서 점선은 최소화 문제를 가정한 경우의 최적 파레토 프론트가 되는 결정벡터와 상응하는 목적공간에서의 목적벡터들을 의미하고, 이 목적벡터들은 어떤 다른 목적벡터에 의해서도 지배되지 않는다. 또한 (그림 1)에서 목적벡터 B는 목적벡터 C와 D를 지배하고(dominates), 목적벡터 A에 의해 지배되는(dominated) 관계를 가지고 있다. 그리고 목적벡터 A는 어떤 벡터에 의해서도 지배되지 않는(nondominated) 파레토 최적해 집합의 원소가 된다.

3. NSGA-II(Nondominated Sorting GA-II)

이 장에서는 우리가 제안하는 파레토순위와 밀도의 가중치에 기반한 진화알고리즘과 실험적 비교 대상이 되는 NSGA-II[7] 알고리즘을 설명한다. NSGA-II는 NSGA[6]를 개선한 알고리즘으로 비지배 개체 정렬(non-dominated sorting)의 높은 연산 복잡도를 개선하고, 엘리티즘(elitism)을 보다 강화하였다. 그리고 밀도측정(density estimation) 연산자를 추가 함으로서 적합도 평가에서 NSGA와 같이 공유 매개변수(sharing parameter) σ_{share} 를 정의할 필요성을 제거한 알고리즘이다.

NSGA-II의 다목적 최적화 과정은 다음과 같다. 먼저 개체집단내의 모든 개체들이 파레토 지배관계에 따라 서로 다른 프론트로 분리된다. 그리고 각 프론트에 속하는 개체들에 순위(rank)를 할당한다. 이때 첫 번째 비지배 프론트에 속하는 개체는 가장 높은 순위가 할당되고, 두 번째 비지배 프론트에 속하는 개체들은 두 번째 높은 순위가 할당

되는 방식으로 모든 프론트에 순위가 할당된다. 같은 순위
에 속하는 개체들에 대하여 각 목적에 따라 인접한 두 개
체의 거리의 합으로 밀도를 측정한다. 각 개체의 밀도는 같
은 순위의 개체들에 대한 부분 순서로서 사용된다. 즉, 같
은 순위를 가지고 있는 개체일지라도 높은 밀도를 가지는
개체는 낮은 밀도를 가지고 있는 개체와의 토너먼트 선택
연산에서 탈락된다. NSGA-II는 엘리티즘을 강화하는 방법
으로 부모 개체집단과 자식 개체집단을 결합한 후, 결합된
개체집단에 대한 순위 할당과 밀도측정을 한다. 그리고 성
능이 낮은 50%를 전지하여 버린다. 나머지 50%는 다음 세
대의 자식 개체집단을 만들기 위한 이진 토너먼트 선택
(binary tournament selection)에 사용된다.

기존 논문들의 보고에 따르면[7-9], NSGA-II의 수렴 성
능과 만들어진 해의 균일 분포는 대부분의 진화적 최적화
방법들 보다 우수한 결과를 보이고 있다.

4. 파레토 지배순위와 밀도의 가중치를 이용한 다목적 최적화 진화 알고리즘(Pareto Dominance Rank And Density Weighting Evolution Algorithm)

다목적 최적화를 위한 진화알고리즘들의 가장 큰 어려움
은 주어진 최적화 문제에 대해 보다 완벽하게 전체 파레토
프론트를 포함하면서 최적 파레토 프론트에 근사할 수 있
도록 하는 적합도 할당 전략을 만드는 것이다. 그러나 이
두 가지 목적들은 서로 상충하는 성질을 갖는다. 즉, 진화
알고리즘에서의 "유전적 편류(genetic drift)" 현상은 최적
해에 수렴하기 위해 필요한 특징이 되는 반면에 균일 분포
로 전체 파레토 프론트를 포함하는 해집합을 만들기 위해
서는 피하여야 하는 현상이다.

기존의 다목적 최적화 진화알고리즘 들은 유전적 편류현
상을 이용하여 최적 파레토 프론트에 근사하는 해들을 만
든 후, 근사한 해들의 밀도측정 연산에 의해 해들 사이의
부분 순서를 결정하여 선택 연산에 사용하였다. 즉, 해들의
밀도측정 연산은 같은 순위에 있는 개체들과의 밀도만을
반영하게 된다. 이는 많은 지역 파레토 프론트를 가지고 있
는 문제에서 이미 지역 파레토 프론트에 수렴한 해들 사이
의 밀도만을 반영하게 된다. 또한 선택연산의 우선적인 기
준이 순위에 있기 때문에 많은 수의 지역 파레토 프론트에
수렴한 비지배 개체들이 선택될 가능성을 내재하고 있다.
이러한 극도의 유전적 편류 현상은 개체의 다양성을 유지
하며 전역해로 수렴 해야 하는 다목적 최적화 문제에서 중
요한 약점이 될 수 있다.

DDWEA(Pareto Dominance Rank And Density Weight-
ing Evolution Algorithm)는 이러한 다목적 최적화 진화알
고리즘의 특징을 고려하여 진화 개체의 적합도를 평가하기
위한 방법으로 파레토 순위와 밀도에 대한 임의의 가중치

를 적용하도록 하였다. 적합도 평가에서 사용되는 밀도는
현세대의 전체 개체집단내의 모든 개체를 반영한 밀도 측
정방법이 사용되고, 파레토 순위와 밀도의 임의 가중치 적
용은 전체 진화과정에서 파레토 순위와 밀도가 비슷한 영
향을 미치게 되어 현 세대에서 비지배 개체는 아니지만 밀
도가 작은 개체가 선택될 가능성이 존재하기 때문에 극도
의 유전적 편류 현상을 보완할 수 있다.

DDWEA에서는 진화과정에서 찾아지는 엘리트 개체를
보관 하기위해 SPEA-II와 같이 비지배 개체집합(nondomi-
nated archive set)을 사용하고, 이를 진화를 위한 단서로
사용하기 위해 선택연산에서 비지배 개체집합의 개체를 일
부 선택하도록 하였다. DDWEA의 전체 알고리즘은 (그림
2)와 같다.

```

Input :  $N$  (population size)
         $\bar{N}$  (archive size)
         $T$  (maximum number of generation)
Output :  $A$  (nondominated archive set)
Step 1 : Initialization : Generate an initial population  $P_0$ 
        and create empty archive set  $\bar{P}_0 = \emptyset$ . Set  $t = 0$ .
Step 2 : Estimate individual's Dominance Rank and
        Density in  $P_t$ 
Step 3 : Fitness Assignment (random weighting)
Step 4 : Elitist Selection (maintaining archive)
        Copy all nondominated individuals in  $P_t$  to  $\bar{P}_t$ .
        If size of  $\bar{P}_t$  exceeds  $\bar{N}$  then pruning  $\bar{P}_t$  by
        means of density based truncation.
Step 5 : Termination check : if  $t \geq T$  then stop.
Step 6 : Mating selection : perform binary tournament
        selection from
         $\bar{P}_t$  while number of individual  $\leq \frac{N}{2}$  and  $P_t$  for rest.
Step 7 : Reproduction : apply crossover and mutation
        operators to the mating pool and set  $P_{t+1}$  to the
        resulting population.
        set  $t = t + 1$  and goto 2.
    
```

(그림 2) DDWEA 알고리즘

4.1 파레토 지배순위 할당

DDWEA에서 한 개체의 파레토 지배순위는 다른 개체들
과의 파레토 지배 관계에 기초하여 계산되어진다. 즉, 한
개체의 순위는 그 개체를 지배하는(dominate) 다른 개체들
의 순위 값의 합으로 정의된다. 예를 들어, t 세대에서 개체
 y 가 n 개의 개체 y_1, y_2, \dots, y_n 에 의해 지배되고, y_1, y_2, \dots, y_n
의 순위값이 $rank(y_1, t), rank(y_2, t), \dots, rank(y_n, t)$ 라면, 개
체 y 의 순위는 식 (3)과 같이 계산된다.

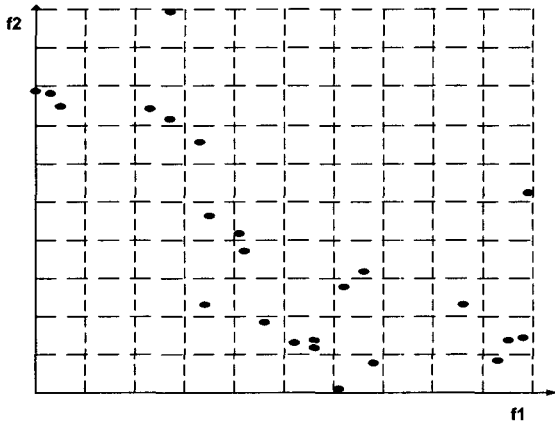
$$rank(y, t) = 1 + \sum_{j=1}^n rank(y_j, t) \tag{3}$$

순위를 계산하는데 있어 어떤 개체에 의해서도 지배되지

않는(nondominated) 개체들의 순위는 모두 1이 되고, 다른 개체에 의해 지배되는 개체의 순위는 개체집단의 밀도와 중복을 감소시키기 위해 순위가 높아진다.

4.2 밀도 평가방법

DDWEA에서 균일 분포의 파레토 프론트를 유지하기 위해 우리는 (그림 3)과 같이 "Hyper-grid based method"에 기반한 적응적 셀 밀도 평가방법을 사용한다[10, 11].



(그림 3) 밀도 평가를 위한 목적 공간의 셀 구성

각 개체의 밀도는 같은 셀에 있는 개체의 수가 되고, 각 셀의 넓이는 식 (4)와 같이 각 목적 함수 차원의 최대, 최소 값의 차를 사용자 정의 셀의 개 수 K 로 나눔으로써 계산되어진다.

$$d_i = \frac{|Max(f_i) - Min(f_i)|}{K} \tag{4}$$

식 (4)에서 d_i 는 i 번째 함수 차원에서의 셀의 넓이이고, 진화의 각 세대에서 목적 공간에서 각 차원의 최대, 최소 값은 매 세대마다 다르므로 셀의 넓이는 세대마다 다르다.

4.3 적합도 할당

DDWEA의 적합도 할당은 식 (5)와 같이 각 개체에 대하여 계산되어진 지배순위와 밀도에 대하여 매 세대에 임의의 가중치 w_r, w_d 를 적용하여 스칼라 값으로 계산된다.

$$Fit(i) = w_r Rank(i) + w_d Density(i) \tag{5}$$

식 (5)에서 w_r 과 w_d 는 0보다 큰 가중치이고 $w_r + w_d = 1$ 이다. 지배순위와 밀도의 가중치를 이용한 적합도 할당방법은 NSGA-II, SPEA-II등 대부분의 방법에서 파레토 지배 순위가 적합도 평가의 주된 기준이 되고, 밀도는 부가적인 요소로만 사용되는 것과는 구별된다. 즉, NSGA-II방법에서 밀도는 같은 순위에 속하는 개체들 사이의 거리를 계산하여 같은 순위 내에서의 순서를 구하는 것이기 때문에 다음 순위의 개체들 중 밀도가 아주 작은 개체일 지라도 상

위 순위의 어떤 개체와 토너먼트 선택(tournament selection)연산을 적용하여도 선택되지 않는다. 그러나 DDWEA에서는 다음 순위의 개체일 지라도 밀도가 작으면 가중치 값에 따라 토너먼트 선택연산에 의해 선택될 가능성을 가지기 때문에 개체의 다양성 유지의 측면에서 NSGA-II방법 보다는 유리하다. 물론 같은 순위 내에서는 밀도가 작은 개체가 선택된다.

4.4 복제 연산(Reproduction)

DDWEA의 복제 연산은 SEAMO[12]에서 제안한 3가지 규칙을 따르는 복제 연산을 수행한다. SEAMO는 모든 개체가 부모개체가 되는 기회를 부여하여 교배연산과 돌연변이 연산을 수행한 후 다음과 같은 복제연산 규칙을 적용하여 개체집단을 구성한다.

- ① 부모 개체는 그들 자신의 자식개체에 의해서만 대체된다.
- ② 자식 개체는 부모개체를 지배할 때에만 부모개체를 대체할 수 있다.
- ③ 개체집단 내의 중복된 개체는 제거된다.

DDWEA에서는 이진 토너먼트 선택을 하기 때문에 개체집단 내의 모든 개체가 부모 개체가 되는 기회를 부여받지는 못하지만 두 번째 복제 연산 규칙을 적용함으로써 자식 개체의 성능을 보장 할 수 있도록 하였다.

4.5 비지배(nondominated) 개체집합 구성

비지배 개체집합 즉, archive set의 구성은 현재 세대까지 찾은 파레토 프론트가 되는 개체들을 다음 세대들에서 잃어버리지 않도록 사용하는 부가적 메모리이다. 또한 다목적 최적화 진화 알고리즘의 실행 결과가 된다. 따라서 비지배 개체집합을 구성할 때 고른 분포의 개체집합을 구성하여야 한다. 이를 위해 DDWEA에서는 밀도를 기반으로 하는 초과 개체 전지(density based truncation)를 한다. 즉, 비지배 개체집합의 최대 크기보다 많은 비지배 개체들이 만들어지면 비지배 개체집합 내에서 4.2에서 설명한 밀도평가방법을 사용하여 밀도가 높은 개체부터 비지배 개체집합으로부터 제거된다.

5. 실험 및 토의

5.1 테스트 함수

제안한 DDWEA의 성능을 평가하기 위해 Deb[13, 14]이 제시한 6개의 테스트 함수들을 실험대상으로 하였다. 테스트 함수들은 다목적 최적화 진화알고리즘들이 최적 파레토 프론트에 수렴하고 균일분포의 파레토 해들을 만드는데 있어 어려움의 원인이 될 수 있는 몇 가지 특징을 가지고 있

다. 각 실험 함수들과 그 특징은 다음과 같다.

각 테스트 함수들은 식 (6)과 같이 3개의 함수 f_1, g, h 로 정의된다.

$$\begin{aligned} &\text{Minimize } T(X) = (f_1(x_1), f_2(X)) \\ &\text{subject to } f_2(X) = g(x_2, \dots, x_m)h(f_1(x_1), g(x_2, \dots, x_m)) \quad (6) \\ &\text{where } X = (x_1, \dots, x_m) \end{aligned}$$

함수 f_1 은 첫 번째 결정변수 x_1 만을 파라메타로 가지는 함수이고, g 는 나머지 $m-1$ 개의 변수로 이루어진 함수이다. 그리고 h 는 f_1 과 g 의 함수 값을 파라메타로 가진다.

- 테스트 함수 T_1 은 볼록한 최적 파레토 프론트(Convex)를 가지는 문제이다.

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned} \quad (7)$$

식 (7)에서 $m = 30$ 이고 $x_i \in [0, 1]$ 이다. 볼록한 파레토 프론트를 가지는 문제는 중간해가 지배하는 영역이 양쪽 끝에 있는 해가 지배하는 영역보다 커지는 경향이 있기 때문에 순수 파레토 지배관계만을 가지고 적합도를 평가하는 FFGA[4] 같은 진화알고리즘들은 보다 많은 중간해를 찾으려는 경향을 갖게 된다.

- 테스트 함수 T_2 은 T_1 과 반대로 오목한 최적 파레토 프론트를 가지는 문제이다.

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1) \\ h(f_1, g) &= 1 - (f_1/g)^2 \end{aligned} \quad (8)$$

식 (8)에서 $m = 30$ 이고 $x_i \in [0, 1]$ 이다. 오목한 파레토 프론트를 가지는 문제는 볼록한 파레토 프론트를 가지는 문제와는 반대로 양쪽 끝에 존재하는 해를 찾으려는 경향을 가지게 된다.

- 테스트 함수 T_3 은 파라메타 공간에서는 연속적이지만 목적 공간에서는 몇 개의 비연속적인 볼록한 부분으로 이루어진 최적 파레토 프론트를 가지는 이산적인 특징을 갖는다.

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1) \end{aligned} \quad (9)$$

식 (9)에서 $m = 30$ 이고 $x_i \in [0, 1]$ 이다. 이산적인 문제는

각 부분에서 해들을 찾을 수는 있지만, 서로 다른 부분에 속한 해들 사이의 경쟁으로 인하여 어떤 부분의 해들이 멸종될 수도 있다.

- 테스트 함수 T_4 는 21^9 개의 지역적인(local) 최적 파레토 프론트를 가지는 multi-modality 문제이다. 그리고 모든 지역적인 최적 파레토 해들은 목적 공간에서 구분 가능하지 않다.

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2, \dots, x_m) &= 1 + 10(m-1) + \sum_{i=2}^m (x_i^2 - 10 \cos(4\pi x_i)) \quad (10) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \end{aligned}$$

식 (10)에서 $m = 10$ 이고 $x_1 \in [0, 1], x_2, \dots, x_m \in [-5, 5]$ 이다. 이 문제는 많은 지역적인 파레토 프론트를 가지게 되기 때문에 다목적 최적화 진화 알고리즘이 전역해에 수렴하기 어려운 문제이다.

- 테스트 함수 T_5 는 속이는(deceptive) 문제이다. 또한 다른 테스트 함수와는 다르게 x_i 가 이진 스트링으로 표현되어야 한다.

$$\begin{aligned} f_1(x_1) &= 1 + u(x_1) \\ g(x_2, \dots, x_m) &= \sum_{i=2}^m v(u(x_i)) \\ h(f_1, g) &= 1/f_1 \end{aligned} \quad (11)$$

여기서 $u(x_i)$ 는 이진 스트링 x_i 내에서 1의 비트수 이고,

$$v(u(x_i)) = \begin{cases} 2 + u(x_i) & \text{if } u(x_i) < 5 \\ 1 & \text{if } u(x_i) = 5 \end{cases}$$

식 (11)에서 $m = 11$ 이고 $x_1 \in \{0, 1\}^{30}, x_2, \dots, x_m \in \{0, 1\}^5$ 이다. 이 문제에서 완전한 최적 파레토 프론트는 $g(X) = 10$ 에서 만들어 지는 반면에 가장 잘 속이는 최적 파레토 프론트는 $g(X) = 11$ 에서 만들어 진다.

- 테스트 함수 T_6 은 탐색공간의 비균일 분포에 의해 만들어 지는 두 가지 어려움을 가지고 있다. 첫 번째는 최적 파레토 해들이 전역적인 파레토 프론트를 따라 균일하지 못하게 분포 한다는 것이고, 두 번째는 해들의 밀도가 전역 파레토 프론트 근처에서 가장 작고 프론트로부터 멀어질수록 커진다.

$$\begin{aligned} f_1(x_1) &= 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\ g(x_2, \dots, x_m) &= 1 + 9 \cdot \left(\sum_{i=2}^m x_i / (m-1) \right)^{0.25} \\ h(f_1, g) &= 1 - (f_1/g)^2 \end{aligned} \quad (12)$$

식 (12)에서 $m = 30$ 이고 $x_i \in [0, 1]$ 이다.

5.2 실험 결과

본 논문에서 우리는 최적화 알고리즘의 성능을 평가하기 위해 C metric(Coverage metric)[15]을 사용한다. C metric은 한 알고리즘의 결과가 다른 알고리즘의 결과를 얼마나 지배하는가를 보여줄 수 있는 평가방법이다. C metric은 다음과 같이 정의된다.

두 결정벡터 집합 X', X'' 에 대해, 함수 C는 순서쌍 (X', X'') 을 $[0, 1]$ 사이의 값으로 사상시키는 함수이다.

$$C(X', X'') = \frac{|a'' \in X''; \exists a' \in X' : a' \leq a''|}{|X''|} \quad (13)$$

$C(X', X'') = 1$ 은 X'' 의 모든 결정벡터가 X' 의 결정벡터에 의해 지배되거나 같음을 의미한다. 반대로, $C(X', X'') = 0$ 은 X'' 의 어떤 결정벡터도 X' 의 결정벡터에 의해 지배되지 않음을 의미한다. C metric 평가는 $C(X', X'')$ 과 $C(X'', X')$ 모두를 고려해야 한다. 즉 $C(X'', X')$ 의 평가 값이 $1 - C(X', X'')$ 가 아니기 때문이다.

실험은 각 문제에 대해 각각 30회의 실험을 수행 하였고, 각 알고리즘에서 사용한 공통 파라메타는 다음과 같다.

- 세대수 : 250세대
- 개체집합 크기(Population Size) : 100
- 교배연산 확률(Crossover rate) : 0.8
- 돌연변이연산 확률(Mutation rate) : 1/개체 비트 스트링 수

각 테스트 문제의 결정변수들의 값 표현은 30비트로 표현하였다. 다만 T_5 문제의 x_2, \dots, x_m 변수들의 표현만 5비트로 표현하였다. DDWEA 알고리즘에서 밀도계산을 위한 셀의 개수 K는 목적공간의 각 차원마다 10으로 설정하였다.

(그림 4)부터 (그림 9)까지는 각 테스트 문제에 대해 30번 수행한 결과 중 가장 우수한 수렴결과 이다. T_1 과 같은 볼록한 최적 파레토 프론트를 가지는 문제는 다목적 최적화 문제에서 최소한의 복잡도를 가지는 문제이다. (그림 4)의 결과와 같이 DDWEA와 NSGA-II는 유사한 수렴 결과를 보였다.

T_2 의 오목한 최적 파레토 프론트를 가지는 문제에서 (그림 5)는 30번 수행한 결과 중 가장 잘 수렴된 결과이기 때문에 DDWEA와 NSGA-II가 유사한 수렴 결과를 보이고 있지만 30번 수행한 결과의 평균적인 결과에 있어서는 DDWEA가 NSGA-II보다 약간 우수한 수렴 결과를 보였다.

T_3 과 같이 목적 공간에서 이산적인 파레토 프론트를 가지는 문제에서 두 알고리즘의 수렴은 NSGA-II가 조금 더 파레토 프론트에 근사하였다. 그러나 NSGA-II의 경우는 (그림 6)의 오른쪽과 같이 전체 파레토 프론트 중 일부에서 균일하게 분포된 결과를 만들지 못하는 경우가 있는 반면, DDWEA는 모든 실험에서 비교적 균일한 분포를 갖는 해들을 만들었다. DDWEA가 NSGA-II보다 비교적 고른 분포의 해들을 만드는 원인은 NSGA-II의 경우 일정한 세대

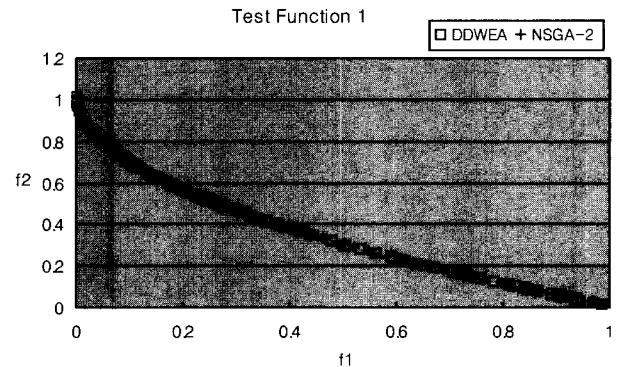
가 지나면 전체 개체들이 비지배 개체들만으로 구성되어 복제되기 때문에 특히 이산적인 형태의 파레토 프론트를 갖는 문제의 경우 개체집단을 구성하는 개체의 다양성이 충분하지 못하기 때문이다. 반면 DDWEA의 경우 현 세대에서는 지배되는 개체이지만 복제를 위해 선택될 수 있는 충분한 가능성을 가지기 때문에 NSGA-II에 비해 개체의 다양성이 보장되기 때문에 서로 다른 부분의 해들과의 경쟁으로 인한 특정부분의 해들이 멸종되는 현상이 줄게 된다.

T_1, T_2, T_3 문제는 전역 파레토 프론트에 대한 수렴의 어려움 보다는 균일 분포의 파레토 프론트를 만들기 어려운 문제들이다. 이런 문제들에서 DDWEA는 NSGA-II와 유사한 수렴결과와 균일 분포의 파레토 프론트를 만들었지만, T_3 문제와 같은 특징을 갖는 문제에서 DDWEA는 NSGA-II보다 비교적 안정적으로 균일 분포의 파레토 프론트를 만들었다.

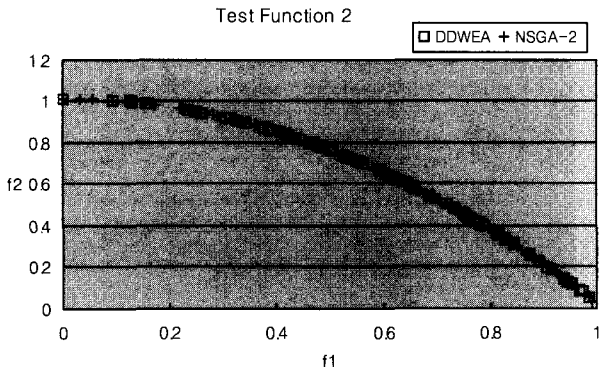
T_4 와 같이 많은 지역적인(local) 해들을 가지는 multimodality 문제에서 DDWEA는 NSGA-II에 비해 월등히 우수한 수렴 결과를 보였다. 이는 DDWEA의 복제 연산과 충분한 개체의 다양성을 보장하는 알고리즘의 특징에 기인하는 것으로 보인다. NSGA-II의 경우에는 선택연산에서 이미 지역해에 수렴한 개체들이 선택될 가능성이 매우 높기 때문에 수렴한 지역해에서 쉽게 벗어나지 못하는 문제점을 가지고 있기 때문이다. T_5 와 같은 속이는 문제에서 DDWEA는 NSGA-II보다 비교적 우수한 수렴성능을 보였다.

T_4, T_5 문제는 균일 분포의 파레토 프론트를 만드는 어려움 보다 전역 파레토 프론트로의 수렴이 어려운 특징을 가지는 문제이다. 이러한 문제들에서 DDWEA는 NSGA-II와 유사한 분포의 해들을 만들었지만, 보다 최적 파레토 프론트에 근사하는 결과를 보였다.

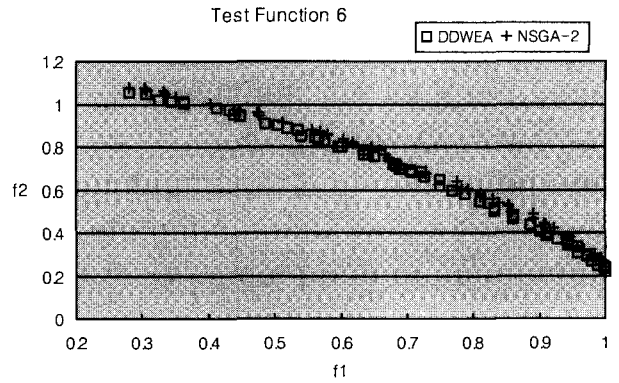
T_6 과 같이 비균일 분포의 최적 파레토 프론트를 가지고 있고, 최적 파레토 프론트 근처에서 해가 드물게 존재하는 문제는 수렴과 균일 분포의 해집합 생성 모두에 어려움이 존재하는 혼합된 문제이다. 이 문제에서 DDWEA는 NSGA-II보다 매우 우수한 수렴 성능을 보였다.



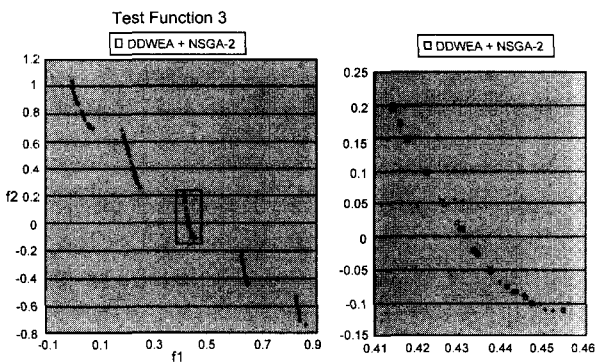
(그림 4) 테스트 함수 T_1 의 수렴결과



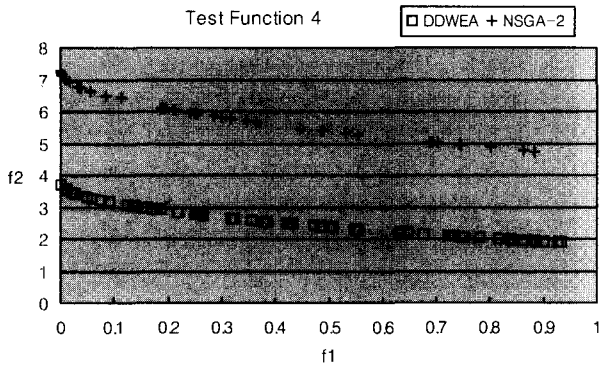
(그림 5) 테스트 함수 T_2 의 수렴결과



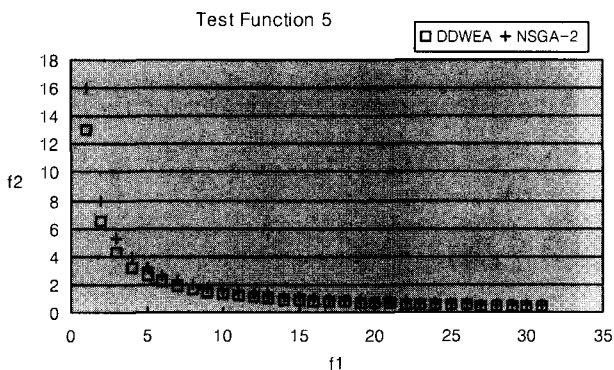
(그림 9) 테스트 함수 T_6 의 수렴결과



(그림 6) 테스트함수 T_3 의 수렴결과(왼쪽)와 확대된 영역(오른쪽)

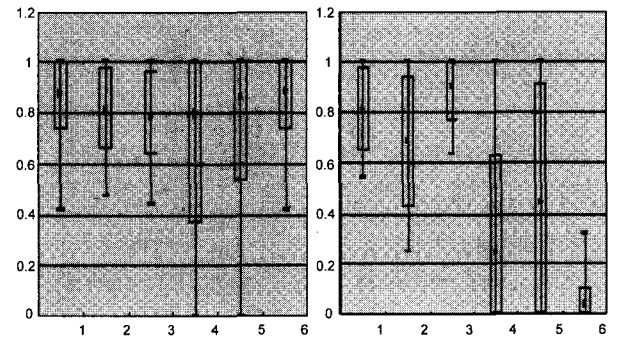


(그림 7) 테스트 함수 T_4 의 수렴결과



(그림 8) 테스트 함수 T_5 의 수렴결과

(그림 10)은 각 문제들에 대한 C metric의 평균과 분산을 표현한 것이다.



(그림 10) C 평가방법에 따른 결과. 왼쪽은 $C(DDWEA, NSGA-II)$ 이고, 오른쪽은 $C(NSGA-II, DDWEA)$ 이다.

(그림 10)의 결과에서 볼 수 있듯이 DDWEA에 의해 구하여진 비지배 개체집합(최적 파레토 해)들은 NSGA-II에 의해 구하여진 해들을 T_3 문제를 제외하고 전반적으로 지배하는 결과를 보였다. 특히 T_4, T_5, T_6 문제의 경우에는 DDWEA의 결과가 NSGA-II의 결과를 월등히 지배한다. 이는 DDWEA가 복잡하고 탐색공간이 큰 문제에 대하여 NSGA-II보다 파레토 최적해에 보다 잘 근사될 수 있음을 의미한다.

6. 결 론

본 논문에서 우리는 파레토 순위와 밀도의 가중치를 이용한 다목적 최적화 진화알고리즘인 DDWEA를 제안하였다. DDWEA는 진화 개체의 적합도를 평가하기 위해 파레토 순위와 밀도에 대한 임의의 가중치를 적용하여 전체 진화 과정에서 파레토 순위와 밀도가 비슷한 영향을 미치게 하여 개체의 다양성을 유지하도록 함으로써 지역 파레토 프론트에 수렴하는 극도의 유전적 편류 현상을 보완할 수 있었다. 즉, 실험을 통하여 ① 다목적 최적화 문제는 전역 파레토 프론트에 수렴을 하기 위해 유전적 편류 현상을 이용

해야 하지만 극도의 유전적 편류현상은 T_4 와 같이 많은 지역 해들을 가지는 문제에서는 오히려 전역 파레토 프론트에 수렴하는데 장애가 될 수 있다는 점과 ② 밀도 정보가 적합도 평가를 위한 부가적인 요소로서 사용되기 보다는 개체의 다양성 유지를 위해 파레토 지배순위와 같은 중도를 가지고 사용되는 것이 극도의 유전적 편류현상을 보완할 수 있음을 확인 할 수 있었다.

본 논문에서의 실험 문제는 2차원의 실험 문제만을 대상으로 하였다. 그러나 직관적으로 DDWEA는 다차원 목적공간을 2차원의 순위와 밀도공간으로 사상하는 특징을 가지고 있기 때문에 3차원 이상의 고차원 문제에서 보다 좋은 성능을 보일 것으로 판단된다. 따라서 3차원 이상의 고차원 문제에서 DDWEA를 적용한 결과를 분석해 보아야 할 것이다. 그리고, 다목적 최적화를 위한 실세계 문제를 모델링하고, 제안한 DDWEA 알고리즘을 적용 하기 위한 연구가 이루어져야 할 것이다.

참 고 문 헌

- [1] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," In Genetic Algorithms and their Applications : Proceedings of the First International Conference on Genetic Algorithms, pp.93-100, 1985.
- [2] M. P. Fourman, "Compaction of Symbolic Layout using Genetic Algorithms," In Genetic Algorithms and their Applications : Proceedings of the First International Conference on Genetic Algorithms, pp.141-153, 1985.
- [3] Frank Kursawe, "A Variant of evolution strategies for vector optimization," In Parallel Problem Solving from Nature. 1st Workshop, PPSN I, volume 496 of Lecture Notes in Computer Science, pp.193-197, 1991.
- [4] Carlos M. Fonseca and Peter J. Fleming, "Genetic Algorithms for Multiobjective Optimization : Formulation, Discussion and Generalization," In Proceedings of the Fifth International Conference on Genetic Algorithms, pp.416-423, 1993.
- [5] Jeffrey Horn and Nicholas Nafpliotis, "Multiobjective Optimization using the Niche Pareto Genetic Algorithm," Technical Report IlliGAI Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.
- [6] N. Srinivas and Kalyanmoy Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Evolutionary Computation, Vol.2, No.3, pp.221-248, 1994.
- [7] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization : SGA-II," Proceedings of the Parallel Problem Solving from Nature VI Conference, pp.849-858, Springer, 2000.
- [8] Eckart Zitzler, Marco Laumanns and Lothar Thiele, "SPEA2 : Improving the Strength Pareto Evolutionary Algorithm," EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, pp.12-21, 2001.
- [9] Carlos A. Coello Coello and Nareli Cruz Cortes, "Solving Multiobjective Optimization Problems using an Artificial Immune System," Technical Report EVOCINV-05-2002, Evolutionary Computation Group at CINVESTAV, Kluwer Academic, 2002.
- [10] Joshua D. Knowles and David W. Corne, "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," Evolutionary Computation, Vol.8, No.2, pp.149-172, 2000.
- [11] Carlos M. Fonseca and Peter J. Fleming, "Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part I : A Unified Formulation," IEEE Transactions on Systems, Man, and Cybernetics, Part : Systems and Humans, Vol.28, No.1, pp.26-37, 1998.
- [12] Christine L. Valenzuela, "A Simple Evolutionary Algorithm for Multi-Objective Optimization(SEAMO)," in Congress on Evolutionary Computation(CEC'2002), Vol.1, pp.717-722, 2002.
- [13] Kalyanmoy Deb, "Multi-Objective Genetic Algorithms : Problem Difficulties and Construction of Test Problems," Evolutionary Computation, Vol.7, No.3, pp.205-230, 1999.
- [14] Eckart Zitzler, Kalyanmoy Deb and Lothar Thiele, "Comparison of Multiobjective Evolutionary Algorithms : Empirical Results," Evolutionary Computation, Vol.8, No.2, pp.173-195, 2000.
- [15] Eckart Zitzler and Lothar Thiele, "Multiobjective optimization using evolutionary algorithms - a Comparative study," In Parallel Problem Solving from Nature V, pp.292-301, 1998.



장 수 현

e-mail : tngus68@naver.com

1993년 명지대학교 전자계산학과(공학사)

1996년 명지대학교 대학원 컴퓨터공학과 (공학석사)

1999년 명지대학교 대학원 컴퓨터공학과 박사과정 수료

2001년~2003년 세경대학 인터넷정보과 전임강사

관심분야 : 인공지능, 지능형에이전트, 진화알고리즘 등