

# 입실론-다중 목적함수 진화 알고리즘에 대한 비교 연구

이인희<sup>o</sup> 신수용 장병탁

서울대학교 컴퓨터 공학부 바이오 지능 연구실

{ihlee<sup>o</sup>, syshin, btzhang}@bi.snu.ac.kr

## Comparison Analysis of $\epsilon$ -Multiobjective Evolutionary Algorithm

In-Hee Lee<sup>o</sup> Soo-Yong Shin Byoung-Tak Zhang

Biointelligence Laboratory, School of Computer Science and Engineering

Seoul National University, Seoul 151-742, Korea

### 요 약

실제 응용에서 제기되는 많은 최적화 문제는 실제로 여러 개의 목적함수를 가진 최적화 문제로 분류될 수 있다. 이러한 다중 목적함수 최적화 문제에 적용되는 방법 중에서 다중 목적함수 진화 알고리즘은 해 집합을 이용한다는 특성 및 목적함수 처리의 용이성 때문에 많은 연구가 이루어지고 있다. 본 논문에서는 대표적인 다중 목적함수 진화 알고리즘이라 할 수 있는 입실론-다중 목적함수에 대하여 다양한 최적화 문제에 대하여 실험적으로 비교 분석해 보았다.

### 1. 서 론

다중 목적함수 최적화 문제 (Multi-objective Optimization Problem, MOP)는 복수 개의 서로 상충하는 목적함수를 최적화하는 문제이다. DNA 염기서열의 최적화 문제[1]를 비롯한 실제계의 디자인 문제들은 대부분 MOP에 속하는 문제들이다. 이러한 MOP에 대하여 여러 가지의 해결방법들이 제시되어 왔는데, 최근에는 다양한 해집합을 다룰 수 있고, 다양한 종류의 목적함수를 다룰 수 있는 진화 알고리즘의 장점을 이용한 다중 목적함수 진화 알고리즘 (Multi-Objective Evolutionary Algorithm, MOEA)이 새로운 해결방법으로 각광을 받고 있다[2-3].

1984년 Schaffer가 VEGA를 제안한 이래 현재까지 NSGA[4], PAES[5], SPEA[6] 등 다양한 MOEA가 제안되었다. 본 논문에서는 다양한 MOEA 중에서 대표적 MOEA라 할 수 있는 NSGA-II (Controlled Non-dominated Sorting Genetic Algorithm)[7]와  $\epsilon$ -MOEA ( $\epsilon$ -MultiObjective Evolutionary Algorithm)[8]를 다양한 테스트 문제에 대하여 비교 분석해 보았으며, 이를 통하여 MOP, MOEA에 대한 이해 및 각 알고리즘의 특성을 파악하여 이후의 MOEA 연구에 도움을 주고자 하였다.

### 2. 다중 목적함수 진화 알고리즘 (MOEA)

MOP에서는 주어진 변수 집합  $X$ 에 대하여 복수 개의 목적함수  $F = (f_1, \dots, f_n)$ 가 존재하며, 특히 모든 목적함수에 대하여 모두 최적화시킬 수 없고, 각각의 목적함수 사이에 trade-off 관계가 성립될 수 밖에 없는 경우를 말한다. 복수 개의 목적함수가 존재하므로 각각의 해  $X$ 에 대하여 목적함수 값의 벡터  $F(X) = (f_1(X), \dots, f_n(X))$ 가 대응되게 된다. 그리고 두 개의 해  $X, Y$ 사이의 비교

는 각각의 목적함수 값의 벡터를 비교하여 이루어진다.  $F$ 의 모든 목적함수가 최소화되어야 한다고 가정하고 (최대화 문제일 경우에도 간단한 과정을 통해 최소화 문제로 바꿀 수 있다), 모든  $i$ 에 대하여  $f_i(X) \leq f_i(Y)$ 이고, 적어도 하나 이상의  $i$ 에 대하여  $f_i(X) < f_i(Y)$ 이면,  $X$ 가  $Y$ 보다 좋은 해로 간주되며, ' $X$ 가  $Y$ 를 dominate한다'라고 말한다. 또한 일정한 해 집합 내에서 집합 내의 어떤 다른 해에게도 dominate되지 않는 해들의 집합을 non-dominated set이라 하며, MOP의 전체 해공간 내에서 dominate되지 않는 해집합을 Pareto-optimal이라고 한다. 하나의 non-dominated 또는 Pareto-optimal set 내의 각각의 해들은 위와 같은 관점에서 보면 우열 관계가 존재하지 않으므로, MOEA나 기타 MOP를 해결 방법에서는 이들 Pareto-optimal set에 속하는 모든 해를 찾는 것을 목표로 하고 있다.

하나의 MOP에 대하여 임의의 알고리즘이 찾아낸 non-dominated set에 대한 평가는 두가지 관점에서 이루어질 수 있다. 하나는 수렴도(convergence), 즉 얼마나 Pareto-optimal set에 가까운 해집합을 찾았는가 하는 것이고, 다른 하나는 다양성(diversity), 즉 얼마나 다양하고 많은 Pareto-optimal 해를 찾아내었는가 하는 것이다. 이후에 소개될 두 알고리즘에 대한 비교 실험에서는 이 두가지 관점에 대하여 평가해 보았다.

#### 2.1 $\epsilon$ -MOEA

$\epsilon$ -MOEA는 steady-state GA에 기반한 MOEA로서, 그림 1과 같은 구조로 이루어져 있다.

먼저, 일반 population에서 domination 관계를 사용한 토너먼트를 통하여 하나의 부모를 선택하고, archive에서 임의로 또 하나의 부모를 선택한 다음, 교차 및 변이 연산을 통하여 새로운 해를 생성해 낸다. 그 후, archive 내의 해 중에서 새로운 해에 의해  $\epsilon$ -dominate되는 것이 있거나, archive의 어떤 해도 새로운 해를  $\epsilon$ -dominate하

지 못하면 새로운 해에 의해  $\epsilon$ -dominate되는 해들을 모두 archive에서 제거하고, 새로운 해를 넣는다. 또한, population의 해 중에서 새로운 해에 의해 dominate되는 해가 있으면, 이 중에서 하나와 새로운 해를 바꾸고 다음 세대로 넘어가게 된다.

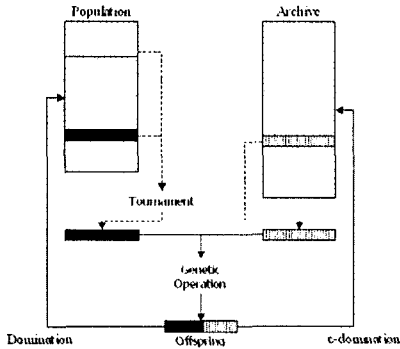


그림 1  $\epsilon$ -MOEA의 구조

여기서  $\epsilon$ -dominate 관계는 비슷한 목적함수 벡터끼리 묶기 위해 도입된 개념으로 하나의 해가 다른 해를  $\epsilon$ -dominate하기 위해서는 모든 목적함수에 대하여  $\epsilon$ 이상 크거나 같아야 한다(그림 2 참고).

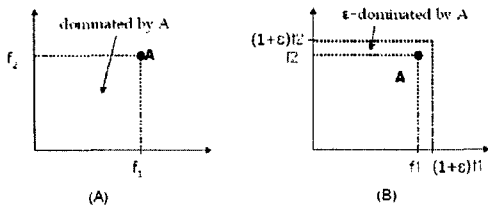


그림 2 dominate(A)와  $\epsilon$ -dominate(B)의 비교

2.2 NSGA-II

NSGA-II는 elitist generational GA에 기반한 MOEA로서 그림 3과 같은 과정으로 이루어져 있다.

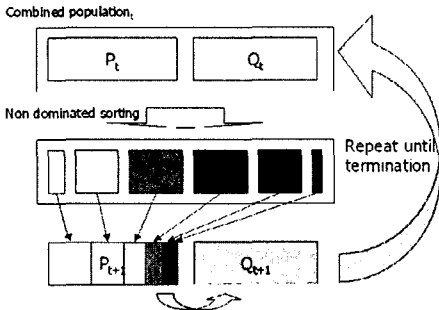


그림 3 NSGA-II의 구조

각 세대에서 부모와 자손 집합을 합친 다음, 이들 사이의 dominate 관계에 따라 집합을 나눈다. 즉, 전체에서 non-dominated set을 분리해낸 다음, 이들을 제외한 집합에서의 non-dominated set을 분리하고, 이러한 과

정을 반복하여 여러 개의 non-dominated set으로 분리한다. 그 다음, 이들 집합에서 일정 분포에 따른 개수만큼 다음 세대로 부모 population으로 복사한다. 단, 먼저 분리된 non-dominated set일수록 더 많은 해를 다음 세대로 복사할 수 있으며, 현재 집합의 크기가 복사할 수 있는 크기보다 크면 토너먼트를 통해 주변 해와의 거리가 큰 해를 선호하여 복사한다. 그리고 이 부모 집합으로부터 교차, 변이 연산을 통해 새로운 자손 집합을 만들어서 위의 과정을 반복하게 된다.

3. 비교 실험 결과

3.1 Test Function

두 MOEA를 비교하기 위하여 널리 사용되는 multi-objective test function 중의 하나인 DTLZ2 함수를 사용하였다[9]. DTLZ2 함수는 목적함수의 개수 및 난이도를 여러 가지 목적에 맞도록 변형하기 쉽게 만들어져 있어 다양한 MOEA의 비교 실험에 사용되었으며, 본 논문에서는 원래 형태의 DTLZ2 및 두가지 다른 변형을 사용하였다.

원래 형태의 DTLZ2는 최소화 함수들로 이루어져 있으며, 목적함수 값들의 공간에서 원점을 중심으로 하는 반지름 1인 구 (또는 hyper ball)의 표면이 Pareto-optimal set이며, 전체 탐색공간은 이 구의 바깥쪽 제1사분면이다. 따라서 구의 바깥쪽에서 표면을 향하여 좁혀들어오는 형태의 탐색을 하게 되어 수렴도에 대한 비교가 가능하다 (그림 4 참고). 두가지 변형된 형태의 DTLZ2 (DTLZ2-1, DTLZ2-2)는 모두 최대화 함수들로 이루어져 있으며, 그림 4에서와 같이 Pareto-optimal set에 가까워짐에 따라 점점 넓은 공간에서 해를 찾아야 하도록 되어 있다. 따라서 수렴도 및 다양성에 대한 비교가 가능하다.

그림 4에서 (B)와 (C)를 비교해보면 같은 최대화 문제이지만, 임의로 선택된 점에 대하여 함수값의 분포가 다름을 알 수 있다. (B)의 경우, Pareto-optimal set에 가까운 점들이 비교적 많은 것에 비해, (C)의 경우는 Pareto-optimal set에 가까운 해가 매우 드물다. 임의의 점 5만개에 대한 결과임을 고려해 볼때, (C)가 (B)보다 수렴하기 어려운 문제임을 알 수 있다.

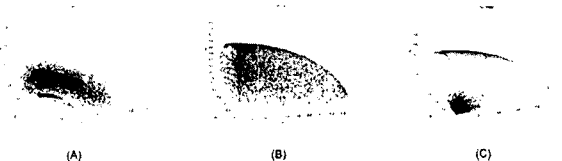


그림 4 (A)원래의 DTLZ2 함수. (B,C) 변형된 형태의 DTLZ2 함수. (점:임의의 50000개의 점에 대한 함수값, 구면: Pareto-optimal set)

3.2 실험 결과

| 목적<br>함수 | DTLZ2    |          |          |          | DTLZ2-1  |          |          |          | DTLZ2-2  |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|          | ε-MOEA   |          | NSGA-II  |          | ε-MOEA   |          | NSGA-II  |          | ε-MOEA   |          | NSGA-II  |          |
|          | GD       | Spread   | GD       | Spread   | GD       | Spread   | GD       | Spread   | GD       | Spread   | GD       | Spread   |
| 3개       | 0.001884 | 1.012066 | 0.000323 | 0.97623  | 2.67e-18 | 0.880442 | 0.000838 | 0.999702 | 0.000996 | 0.836153 | 0.006165 | 0.929349 |
|          | 0.001497 | 1.008123 | 0.000362 | 0.991853 | 4.08e-18 | 0.872881 | 0.000851 | 0.968693 | 0.000694 | 0.854686 | 0.001842 | 0.958084 |
|          | 0.001490 | 1.007788 | 0.000351 | 0.999624 | 2.67e-18 | 0.874388 | 0.001189 | 0.932447 | 0.000581 | 0.802938 | 0.003261 | 0.928131 |
|          | 0.001708 | 1.009312 | 0.000347 | 0.989116 | 3.11e-18 | 0.881926 | 0.001447 | 0.974922 | 0.000823 | 0.855656 | 0.004689 | 0.948892 |
|          | 0.001895 | 1.00989  | 0.000384 | 0.996079 | 2.18e-18 | 0.875101 | 0.001041 | 0.997676 | 0.001085 | 0.833705 | 0.002288 | 0.975774 |
| 평균       | 0.001695 | 1.00944  | 0.00035  | 0.99058  | 2.94e-18 | 0.87695  | 0.00107  | 0.97469  | 0.000836 | 0.83663  | 0.00365  | 0.94805  |
| 6개       | 0.000655 | 1.078772 | 0.102309 | 1.990517 | 2.01e-5  | 0.967842 | 0.004833 | 0.932627 | 0.003433 | 0.813084 | 0.013503 | 0.858183 |
|          | 0.000474 | 1.044132 | 0.049665 | 1.64836  | 1.69e-5  | 0.960531 | 0.003774 | 0.856672 | 0.004245 | 0.799779 | 0.022071 | 0.784653 |
|          | 0.000506 | 1.061903 | 0.045857 | 1.570792 | 1.69e-5  | 0.971406 | 0.003378 | 0.863198 | 0.002909 | 0.825123 | 0.016888 | 0.84616  |
|          | 0.000487 | 1.064156 | 0.066023 | 1.756121 | 1.76e-5  | 0.972186 | 0.009866 | 0.866238 | 0.002461 | 0.828377 | 0.018631 | 0.807576 |
|          | 0.000508 | 1.067994 | 0.081302 | 1.923748 | 1.73e-5  | 0.965319 | 0.004815 | 0.855462 | 0.004188 | 0.765796 | 0.01502  | 0.784572 |
| 평균       | 0.000526 | 1.06339  | 0.06903  | 1.77791  | 1.78e-5  | 0.96746  | 0.00533  | 0.87484  | 0.003447 | 0.80643  | 0.01722  | 0.81621  |
| 12개      | 0.000719 | 1.173814 | 0.085419 | 1.485254 | 0.000161 | 0.876436 | 0.004142 | 0.859672 | 0.002325 | 0.583672 | 0.016543 | 0.722404 |
|          | 0.000758 | 1.196307 | 0.094013 | 1.587058 | 0.000162 | 0.898879 | 0.003752 | 0.785999 | 0.001648 | 0.67341  | 0.015745 | 0.779446 |
|          | 0.000759 | 1.216018 | 0.097761 | 1.572275 | 0.000154 | 0.906026 | 0.008508 | 0.740732 | 0.001232 | 0.718478 | 0.022799 | 0.72495  |
|          | 0.000634 | 1.132195 | 0.084881 | 1.423913 | 0.000144 | 0.874857 | 0.016064 | 0.741439 | 0.001622 | 0.656164 | 0.015237 | 0.729664 |
|          | 0.00051  | 1.115677 | 0.089085 | 1.426679 | 0.000159 | 0.885111 | 0.008412 | 0.817391 | 0.001902 | 0.645007 | 0.018628 | 0.702087 |
| 평균       | 0.000676 | 1.1668   | 0.09023  | 1.49904  | 0.000156 | 0.88824  | 0.00818  | 0.78905  | 0.001746 | 0.65535  | 0.01779  | 0.73171  |

표 1 DTLZ2 함수들에 대한 실험 결과

실험 결과는 표 1에 정리된 것과 같다.

각 MOEA에 대하여 population 크기는 100, 함수 계산 횟수는 100000번으로 동일하게 하여 실험하였으며, 각 알고리즘마다 해당 문제에 대한 최고의 성능을 보이는 parameter 값을 사용하였다.

각 알고리즘의 수렴도는 Generational Distance (GD)라는 척도를 사용하여 측정하였는데, 이 값이 작을수록 Pareto-optimal에 가깝게 수렴한 것을 나타낸다. 또한 다양성은 Maximum Spread (Spread)라는 척도를 사용하였는데 이 값이 1에 가까울수록 다양한 해를 찾는 것을 의미한다.

표 1을 살펴보면, DTLZ2에서는 ε-MOEA가 수렴도나 다양성의 측면에서 모두 NSGA-II보다 나은 결과를 보임을 알 수 있다. 그러나, 최대화 문제인 DTLZ2-2에서는 다양성의 면에서는 NSGA-II가 우세하고, 수렴도의 면에서는 ε-MOEA가 우세함을 알 수 있다. 같은 최대화 문제인 DTLZ2-1에서는 목적함수의 수가 적을 때만 NSGA-II가 다양한 해를 찾을 수 있었고, 목적함수의 수가 증가하자 ε-MOEA가 우세한 성능을 보였다.

ε-MOEA에서는 이전 세대에서 찾아낸 해를 archive에 모두 보존하여 다음 자손의 생성시 부모중 하나로써 참여하도록 하여 빠른 수렴을 보일 수 있었던 것으로 생각된다. 또한, NSGA-II의 경우에는 그 특성상 이전 세대의 해를 잃어버릴 수 있기 때문에 수렴이 느려지는 하였으나, 다음 세대로의 전환 과정에서 다양성이 고려되기 때문에 비교적 다양한 해를 찾기 어려운 문제에서도 다양성을 유지할 수 있었던 것으로 생각된다.

#### 4. 결론

본 논문에서는 현재 가장 좋은 성능을 보이고, 널리 쓰이는 두 종류의 MOEA (NSGA-II, ε-MOEA)를 다양한 test function에 대하여 실험적으로 비교 분석해 보았다. 비교 결과, NSGA-II는 더 다양한 non-dominated set을 찾는 경향이 있었고, ε-MOEA는

같은 함수 계산 횟수 내에 Pareto-optimal set에 더 가까운 non-dominated set을 찾을 수 있었음을 보였다.

**감사의 글:** 본 연구는 산업자원부 차세대 신기술 과제 및 과학기술부 국가지정 연구실 과제에 의해 지원되었음. 이 연구를 위해 연구 장비를 지원하고 공간을 제공한 서울대학교 컴퓨터 연구소에 감사드립니다.

#### 참고 문헌

- [1] I.-H. Lee, S.-Y. Shin, and B.-T. Zhang, "DNA Sequence Optimization Using Constrained Multi-Objective Evolutionary Algorithm," Proceedings of the 2003 Congress on Evolutionary Computation, v. 4, pp. 2270-2276, 2003.
- [2] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithm*, John Wiley & Sons, Ltd., England, 2001.
- [3] E. Zitzler, M. Laumanns, and S. Bleuler, "A Tutorial on Evolutionary Multiobjective Optimization," *Metaheuristics for Multiobjective Optimization*, Lecture Notes in Economics and Mathematical Systems, Springer, v. 535, pp. 3-37, 2004.
- [4] N. Srinivas and K. Deb, "Multiobjective Optimization using Nondominated Sorting in Genetic Algorithm," *Evolutionary Computation*, v. 2, no. 3, pp. 221-248, 1994.
- [5] J. D. Knowles and D. W. Corne, "Approximating the Nondominated Front using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, v. 8, no. 2, pp. 149-172, 2000.
- [6] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transaction on Evolutionary Computation*, v. 3, no. 4, pp. 257-271, 1999.
- [7] K. Deb and T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithm for Better Convergence," *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 42-50, 2001.
- [8] M. Laumanns, L. Thiele, K. Deb and E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multi-objective Optimization," *Evolutionary Computation*, v. 10, no. 3, pp. 263-282, 2002.
- [9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Functions for Evolutionary Multi-Objective Optimization," *KanGal Report No. 200101*, 2001.