

# 순위-기반 클러스터링 이용한 PSO

강신호, 임동현, 안창욱  
성균관대학교 컴퓨터공학과

e-mail:compro83@gmail.com, logicboom@naver.com, cwan@skku.edu

## Rank-based PSO Using Clustering

Shinho Kang, Dong-Hyun Lim, Chan-Wook Ahn  
School of Information and Communication Engineering  
SungKyunKwan University

### 요 약

본 논문은 기존 PSO의 문제점으로 지적되는 Local minimum에의 고착을 해결하여 평균적인 성능을 향상시킬 수 있는 방법을 제안한다. 이를 위해 기존에는 하나의 Best Value만을 향해서 이동하는 것에서 벗어나, 각 Particle들을 Cluster로 나누고 각각의 Best Value를 중심으로 이동하며, 각 Cluster Leader들은 다른 Cluster Leader와 통신하며 순위에 따라 이동한다. 제안 방법으로 수렴속도와 Local minimum 회피에 대해 실험을 통해 비교 분석한다. 실험 결과로부터 제안 방법이 Local minimum 회피에서 성능 향상이 있음을 고찰하였다.

### 1. 서론

PSO(Particle Swarm Optimization)는 1995년 Kennedy와 Eberhart에 의해 제안되었으며, 새 떼나 물고기 무리 등의 사회적 행동양식에 대한 규칙성을 증명하는 것에 착안하여 개발되었다[1]. PSO는 Holland가 개발한 GA(Genetic Algorithm)[2]와 유사하게 집단을 토대로한 최적화 기법으로 초기 해집단을 무작위로 생성하여 최적점을 탐색하지만, 그 방법에서 유전 알고리즘의 교배나 돌연변이 연산의 개념을 사용하지 않고, 입자(Particle)의 위치가 입자의 최적점과 집단 전체의 최적점의 두가지 요소들에 의해 결정되는 방향으로 탐색 공간을 누비는 개념을 사용했다. 때문에 PSO는 GA를 사용하였을 때와 거의 동일한 결과 값을 얻을 수 있다는 장점이 있다. 하지만 Local minimum에 빠지기 쉽다는 단점 또한 있다[3].

본 논문에서는 PSO의 Local minimum을 피하고 성능 향상을 위하여, Particle들의 Clustering 기법과 Cluster Leader 간 Rank에 따른 위치 이동 기법을 제안한다.

### 2. 배경지식

#### 2.1 Particle Swarm Optimization[3]

PSO에서 하나의 해는 입자(Particle) 하나로 표현된다. 초기에 각 입자는 랜덤하게 생성되며, 이렇게 생성된 입자는 적합도(Fitness)함수에 의해 평가된 적합도 값을 가지며, 탐색을 위한 속도를 가지고 있다. 이때 각 입자의 속도는 두 가지의 현재 최적점( $pbest, gbest$ )에 의해 결정된다.  $pbest$ (Particle-best)는 각 입자가 현재와 과거를 통틀어서 발견한 해당 입자의 최적점이며,  $gbest$ (Global-best)는 모든 입자들을 통틀어서 현재의 최적점을 말한다. 각

입자의 속도는 다음의 수식에 의해 결정된다.

$$v_t = wv_{t-1} + c_1r_1(pbest_t - x_t) + c_2r_2(gbest_t - x_t) \quad (1)$$

여기서  $w$ 는 이전 속도에 대한 관성 가중치,  $c_1, c_2$ 는 관계 상수,  $r_1, r_2$ 은 0에서 1 사이의 난수이다. 이렇게 결정된 속도에 의해 각 입자는 다음 수식과 같이 이동한다.

$$x_{t+1} = x_t + v_t \quad (2)$$

이렇게  $pbest$ 와  $gbest$ 를 찾아 속도를 결정하고, 각 입자의 새로운 위치를 찾는 과정을 정해진 횟수만큼 반복하여 최적점을 찾을 수 있다.

#### 2.2 Rank-based Clustering PSO

PSO의 Local minimum에 빠지기 쉽다는 단점을 극복하여 평균적인 성능향상을 피하기 위한 방법을 제안한다. 기존의 PSO는 Best Value를 곧바로 찾아가는 형식으로 탐색이 이루어지지만 제안하는 알고리즘은 Clustering을 통해서 search space를 일정하게 나누어서 각 Cluster 안에서 best값을 향해 Local search를 한다. 그리고 각 Cluster Leader(best값을 가지는 Particle)는 모든 Cluster Leader들과 순위를 정하여 자신의 순위보다 한단계 높은 Cluster Leader를 따라 이동하게 된다. 기존의 PSO에 비해 더 넓은 영역을 탐색하면서도 최적점에 쉽게 수렴할 수 있는 방식이다.

이 때, 각 Cluster 안에서의 Local search는 다음의 수식에 의해 각 particle의 속도가 정해진다.

$$v_t = wv_{t-1} + c_1R(pbest_t - x_t) + c_2R(clbest_t - x_t) \quad (3)$$

각 Cluster Leader들의 경우는 서로 정해진 순위에 따라서 이동하기 때문에 다음의 수식에 의해 속도가 정해진다.

$$v_r = wv_r + c_1R(pbest - cbest_r) + c_2R(cbest_{r-1} - cbest_r) \quad (4)$$

여기서 r은 랭킹을 의미하며, 낮을수록 높은 순위의 Cluster Leader를 의미한다.

이를 알고리즘화 하여 설명하면 다음과 같다.

```

begin
  initialize
  evaluate
do
  begin
    find Pbest and Gbest
    clustering
    find cluster-best(cluster leader)
  do Cluster Local Search
    begin
      calculate new velocity vector
      move
      evaluate
    while(predetermined local search condition)
      calculate velocity vector for cluster leaders
      move cluster leaders
      evaluate
    while(not termination condition)
  end
end
    
```

### 3. 실험 결과 및 분석

#### 3.1 실험 문제

본 실험에서 사용한 문제는 다음과 같다.

$$f(x,y) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (5)$$

Griewank 함수를 사용하였으며, 기존의 PSO 알고리즘과 본 논문에서 제안된 알고리즘을 적용하여 0과 얼마나 가까운 Fitness Value를 찾을 수 있는지와 Local minimum에 빠지지 않는지를 살핀다.

#### 3.2 결과

본 실험에서 사용된 언어는 C++이며 실험에서 사용된 변수는 다음과 같다.

- 개체수 : 200
- 반복수 : 1000
- 속도 관성 가중치(w) : 0.9
- $c_1 = c_2 = 1$
- sub local search : 2 (제안된 PSO 알고리즘)

기존 PSO와 본 논문에서 제안된 Rank-based Clustering PSO를 적용하여 20회씩 실험한 결과, 약  $x = \pm 3.14E-11$ ,  $y = \pm 4.44E-10$ 에서 더 나은 Fitness Value를 계산하지 못하고 Global minimum을 찾아내었다. 하지만 기존 PSO를 적용하였을 때, 실험횟수 20회중 5회는 1000번의 세대반복에도 Local minimum에 빠져 Global minimum을 찾지 못하였다. <표 1>은 Local minimum의 value값과 x, y 좌표를 나타낸 것이다. 반면 Rank-based

Clustering PSO의 경우 단 한차례도 Local minimum에 빠지지 않고 Global minimum의 좌표를 찾아내었다.

<표 1> 기존 PSO의 Local minimum 고착

	Fitness Value	x	y
1	4.22E-07	-3.14227	-4.442
2	0.000805738	-6.2505	-8.91873
3	0.00537614	-0.0744652	-8.98801
4	0.000672148	-6.30299	-8.92941
5	0.00217586	-0.06592	-8.8818

<표 2> PSO와 Rank-based Clustering PSO를 이용해 400번째 반복점에서 찾은 Best value의 비교

	PSO	Rank-based Clustering PSO
평균값	1.22E-03	3.97E-11
최대값	5.98E-03	2.45E-10
최소값	1.11E-16	6.77E-13

<표 2>는 20회씩 실행하여 Global minimum을 찾기 전인 400번째 반복에서의 Best value의 평균과 최대, 최소값을 비교하여 정리한 것이다. 기존 PSO의 경우 최대값과 최소값의 편차가 커서 평균적으로 best 값을 잘 찾지 못함을 알 수 있었다. 이는 기존 PSO가 Local minimum에 빠지기 때문에 발생한 결과이다. 하지만 Rank-based Clustering PSO의 경우 최대값과 최소값의 편차가 크지 않고, 평균값도 기존 PSO에 비해 훨씬 좋은 값을 찾아내어 그만큼 안정적인 성능을 보임을 알 수 있었다.

### 3. 결론

본 논문에서는 기존의 PSO 알고리즘의 문제점을 고찰하고 이를 보완하기 위한 방안을 제시했다. 기존 PSO에서는 하나의 Best Value만을 목표로 각 particle들이 이동하기 때문에 Local minimum에 빠질 위험이 크다. 그러나 제안된 알고리즘에서는 클러스터링을 통해 지역을 나누고 각 리더들끼리 통신하게 함으로써 더욱 넓은 지역을 search할 수 있게 하였다. 본 논문의 실험의 결과 값에서 볼 수 있듯이, Rank-based Clustering PSO는 Local minimum 문제를 상당 부분 해결하였다. 차원성이 더욱 높은 함수에 적용한다면 더 좋은 효과를 볼 수 있을 것이다.

### 참고문헌

[1] Kennedy J and Eberhart R., "Particle Swarm Optimization," IEEE International Conference on neural Networks, pp 1942-1948, Perth, Australia, 1995.  
 [2] Holland J. H. "Adaptation in Natural and Artificial Systems," University of Michigan Press, 1975.  
 [3] Matthew Settles, "An Introduction to Particle Swarm Optimization", University of Idaho, 2005