

## PSO의 특징과 차원성에 관한 비교연구

### Comparative Study on Dimensionality and Characteristic of PSO

박 병 준, 오 성 권, 김 용 수, 안 태 천\*

(Byoung-Jun Park, Sung-Kwun Oh, Yong-Soo Kim, and Tae-Chon Ahn)

**Abstract :** A new evolutionary computation technique, called particle swarm optimization(PSO), has been proposed and introduced recently. PSO has been inspired by the social behavior of flocking organisms, such as swarms of birds and fish schools and PSO is an algorithm that follows a collaborative population-based search model. Each particle of swarm flies around in a multidimensional search space looking for the optimal solution. Then, Particles adjust their position according to their own and their neighboring-particles experience. In this paper, characteristics of PSO such as mentioned are reviewed and compared with GA which is based on the evolutionary mechanism in natural selection. Also dimensionalities of PSO and GA are compared throughout numeric experimental studies. The comparative studies demonstrate that PSO is characterized as simple in concept, easy to implement, and computationally efficient and can generate a high-quality solution and stable convergence characteristic than GA.

**Keywords :** particle swarm optimization, genetic algorithm, dimensionality, comparative study

#### I. 서론

많은 지역극소와 전역극소를 가진 비선형 함수의 최적해 탐색은 다양한 영역에서 자주 발생하는 문제이다. 그러나 이 문제는 지난 몇 십 년 간 많은 방법들이 제시되고 있음에도 불구하고 아직까지 해결하기 어려운 문제로 남아 있다. 이 문제를 해결하기 위해 제시된 방법들 중 효과적이고 두드러진 특징을 보이는 것은 확률적 최적화 방법이다. 확률적인 탐색방법 중 하나인 진화연산(evolutionary computation) 기법은 전통적인 확률적 방법과 비교해 볼 때 군집(population)이라는 유용한 해 집단을 이용하고 있으며, 군집에 속한 개체들 사이의 협력과 경쟁을 통해 최적 해에 접근하는 특징을 가지고 있다. 이러한 군집 기반 탐색 알고리즘은 병렬처리 능력을 가지며, 이것은 복잡한 문제에 직면했을 때 탐색 능력의 향상과 전역극소점으로서의 빠른 접근을 보여준다. 가장 많이 사용되고 있는 군집 기반 진화연산 기법으로 GA(Genetic Algorithm[1-3,23]), GP(Genetic Programming[4]), ES(Evolution Strategies[5,6]), EP(Evolution-ary Programming[7])이 있으며, 이들 알고리즘은 자연의 진화 메카니즘에 바탕을 두고 있다.

최근에 PSO(Particle Swarm Optimization[8-14])라는 새로운 진화연산 기법이 소개되었다. PSO는 자연선택의 진화 메카니즘이 아닌 새 떼와 물고기 떼와 같은 생체군집의 사회적 행동양식을 바탕으로 하고 있다. PSO 역시 군집 기반 알고리즘으로 병렬처리 특징을 가지며, 군집과 개체는

Swarm과 Particle로 표현된다. PSO는 Kennedy and Eberhart [8]에 의해 처음 소개되었으며, 어려운 최적화 문제를 수행함에 있어서 그 효과는 응용영역의 확대로 나타나고 있다. 응용분야로는 multiobjective problem[15], minimax problems [16], integer programming problems[17] 등 다양하다. PSO의 각 particle은 최적의 해를 얻기 위해 다차원 탐색공간은 날아다니며, 그들 자신과 그들 이웃의 경험에 대한 정보를 이용하여 최적의 위치로 이동해 간다. 이를 위해 particle은 이전에 경험했던 최적의 위치정보에 대한 기억을 유지한다. 또한 PSO는 파라미터의 조절을 통해 particle의 전역탐색과 지역탐색 능력의 균형을 유지할 수 있다[13]. 일반적으로 PSO는 이론의 간결성, 구현의 용이성, 연산의 효율성과 같은 특징을 보이며, 짧은 계산시간 안에 최적의 해를 생성할 수 있고, 다른 확률적 방법보다 안정적인 수렴특징을 나타내고 있다[14].

본 논문에서는 최근 많은 분야에서 적용되고 있는 PSO와 GA의 특징을 알아보고 비교해본다. 이들은 각각의 독립된 영역에서 뚜렷한 결과를 보여주고 있으며, 상호 공통된 특징과 구별된 특징을 가진다. 이 중 구별된 특징은 PSO를 GA보다 우수한 해를 찾도록 해준다. 실험예제를 통해 그 이유를 설명하고, PSO의 특징들을 검증하고자 한다. 특히 차원에 증가에 따른 두 알고리즘의 특징을 알아본다.

#### II. PSO(Particle Swarm Optimization)와 GA(Genetic Algorithm)

최근 많은 분야에서 적용되고 있는 생물학적 특징을 이용한 방법들 중 뚜렷한 성과를 보이고 있는 것은 PSO와 GA이다. 이 장에서는 이들 알고리즘과 특징을 간략히 기술하였다.

##### 1. PSO

Kennedy and Eberhart에 의해 처음 소개된 PSO[8]는 사회심리학 특징과 진화연산의 결합에 의한 기술이다. PSO는

\* 책임저자(Corresponding Author)

논문접수 : 2005. 12. 15., 채택확정 : 2006. 2. 25.

박병준, 안태천 : 원광대학교 전기전자공학부

(lcap2005@gmail.com/tcahn@wonkwang.ac.kr)

오성권 : 수원대학교 전기공학부(ohsk@suwon.ac.kr)

김용수 : 대전대학교 컴퓨터공학과(kystj@dju.ac.kr)

※ 본 논문은 2004년 교육인적자원부의 재원으로 한국학술진흥재단 (KRF-2004-214-M01-2004-000-20175-0)에서 지원하여 연구하였음.

물고기, 새 떼와 같은 생물의 행동양식을 바탕으로 이루어진다.

### 1.1 PSO의 연산자

PSO의 기본요소와 연산자는 다음과 같다[8-14].

*Particle-x(t)*: 집단(swarm) 안에 속해 있는 각각의 개체는 하나의 particle이다. swarm의 모든 particle들은 같은 체제 아래에서 개별적으로 동작한다. 이것은 하나의 해를 나타내며  $m$ -차원 실수 벡터로 표현된다. 여기서  $m$ 은 최적 파라미터의 수가 된다.  $t$  시간에,  $j$ 번째 particle  $\mathbf{x}_j(t)$ 는  $[x_{j1}(t), x_{j2}(t), \dots, x_{jm}(t)]$ 로 표현되며,  $x_{jk}(t)$ 는  $j$ 번째 particle의  $k$ 번째 차원(파라미터)의 위치(값)를 나타낸다.

*Swarm-S(t)*: swarm은  $n$ 개 particle의 집합이다. 즉,  $\mathbf{S}(t) = [\mathbf{x}_1(t), \mathbf{x}_2(t), \dots, \mathbf{x}_n(t)]^T$ .

*Particle best-pbest(t)*: 탐색 공간을 움직이는 particle들은 현재 위치정보에 대한 적합도를 구하고 최적의 위치정보를 위해 비교된다. 최적의 위치정보를 가진 particle을 pbest라고 한다.

*Global best-gbest(t)*: 최적의 위치정보를 가지는 particle들, 즉 pbest의 모든 particle 중 가장 최적의 위치정보를 나타내는 particle이다. 따라서 gbest는 주어진 문제의 최적해가 된다.

*Particle velocity-v(t)*:  $\mathbf{v}(t)$ 는 PSO의 유일한 연산자이며, particle들의 이동속도를 나타내는  $m$ -차원 실수벡터이다.  $t$  시간의  $j$ 번째 particle 이동 속도  $\mathbf{v}_j(t)$ 는  $[v_{j1}(t), v_{j2}(t), \dots, v_{jm}(t)]$ 로 구성된다. 초기 PSO[8]의 particle velocity는 다음 식에 의해 계산되었다.

$$v_{jk}(t+1) = v_{jk}(t) + c \cdot r_1 \cdot (pbest_{jk}(t) - x_{jk}(t)) + c \cdot r_2 \cdot (gbest_k(t) - x_{jk}(t)) \quad (1)$$

여기서,  $c$ 는 가속상수(acceleration constant),  $r_1$ 과  $r_2$ 는  $[0, 1]$ 의 랜덤상수이다.

초기 PSO의 particle velocity 연산자는 swarm 파열과 발산의 위험을 촉진하는 속도 크기의 제어 부족 현상을 가지고 있다[11]. 이러한 문제를 피하기 위해 particle velocity의 최대 값  $v_{max}$ 을 가정한다. 즉  $k$ 번째 차원에 대한 particle velocity는 최대값  $v_{max}$ 로 제한된다. 이 제한조건은 해 공간의 지역적 탐색을 강화하고 인간의 학습 증가를 실제적으로 시뮬레이션 하는 특징을 가진다[9,14]. 만약  $v_{max}$ 가 매우 큰 값을 가진다면, particle들은 최적 해를 지나쳐버릴 수 있다. 반대로 만약  $v_{max}$ 가 매우 작은 값을 가진다면, particle들은 해 공간을 충분히 탐색하지 못하게 된다. 기존의 많은 실험연구를 통해,  $v_{max}$ 는 각 변수 공간의 10-20%로 설정되었다[10]. 종종  $v_{max}$ 는  $k$ 번째 변수에 대해 다음 식과 같이 주어진다[9].

$$v_k^{max} = \frac{x_k^{max} - x_k^{min}}{N} \quad (2)$$

여기서  $N$ 은  $k$  변수에 대한 탐색 간격의 수이다.

그러나 PSO 성능의 효과적인 개선을 위해서는  $v_{max}$ 만으

로 불충분하였으며, 이전 속도 값과 하중계수를 결합한 방법이 제안되었다. 따라서 particle velocity는 다음과 같다.

$$v_{jk}(t+1) = w(t) \cdot v_{jk}(t) + c_1 \cdot r_1 \cdot (pbest_{jk}(t) - x_{jk}(t)) + c_2 \cdot r_2 \cdot (gbest_k(t) - x_{jk}(t)) \quad (3)$$

여기서,  $w$ 는 관성하중(inertia weight),  $c_1$ 과  $c_2$ 는 가속상수(acceleration constant)이며, 각각 인지성(cognitive)과 사회성(social)을 나타낸다. PSO의 인지성은 그 자신의 생각과 메모리를 바탕으로 velocity를 조절하며, 사회성은 지식의 사회적 심리적응을 바탕으로 velocity를 조절한다.  $c_1$ 과  $c_2$ 는 각 particle이 pbest와 gbest로 향하는 확률적인 가속의 가중치를 표현한다. 이들의 낮은 값은 목적지가 아닌 곳을 떠돌게 만든다. 반면에, 높은 값은 목적지를 지나치거나 갑작스런 방향전환을 유도한다. 기존에 많은 실험들은 이들 값을 2.0으로 설정하였다.

*Inertia weight-w(t)*: 이것은 현재 velocity에 대한 기존 velocity의 영향을 조절하기 위한 제어 파라미터이다. 따라서 이 값은 particle의 전역탐색 능력과 지역탐색 능력 사이의 관계를 조절한다. 탐색과정의 초기에는 전역탐색의 강화를 위해 큰 관성하중(inertia weight) 값을 설정하고, 반대로, 탐색 후반부에는 좀 더 나은 지역탐색을 위해 관성하중을 감소시킨다. 전역탐색과 지역탐색 사이의 균형을 위해 적절한  $w$ 의 선택은 필요하다. 최적 해를 탐색하는 동안  $w$  값은 0.9에서 0.4까지 선형적으로 감소시키는 방법이 자주 이용된다.

$$1) w(t) = \alpha w(t-1) \quad (4)$$

여기서  $\alpha$ 는 감소상수이며, 1에 가까운 값을 가진다.

$$2) w(t) = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times t \quad (5)$$

여기서  $iter_{max}$ 는 세대(generation) 수이다.

*Stopping criteria*: 만약 종료조건을 만족한다면, 탐색과정을 종료한다.

### 1.2 PSO 알고리즘

PSO 알고리즘에서 swarm은  $n$ 개의 particle을 가지며 이들은 주어진 문제의 해가 된다. 각 particle은  $m$  차원 실수 벡터로 구성된다. 여기서  $m$ 은 파라미터의 수가 된다. 따라서 각 파라미터는 주어진 문제의 차원을 나타낸다. PSO 알고리즘은 다음과 같다.

Step 1: *Initialization* -  $n$ 개의 particle을 탐색공간 내에서 랜덤하게 생성한다. 이들은 초기 swarm이 된다. 마찬가지로 초기 particle velocity를  $[-v_{max}, v_{max}]$  내에서 랜덤하게 생성한다. 초기 swarm의 각 particle은 목적함수를 이용하여 평가되고, 이들 모두는 pbest로 설정한다. 초기 particle 중 최적의 값은 다시 gbest로 선정된다.

Step 2: *Inertia weight & velocity updating* - 관성하중 값을 계산하고, (3)을 이용하여  $j$ 번째 particle velocity를 계산한다. 제한조건  $[-v_{max}, v_{max}]$ 을 확인한다.

Step 3: *Position updating* - (6)에 의해 *particle*의 위치정보는 조절된다.

$$x_{jk}(t) = v_{jk}(t) + x_{jk}(t-1) \quad (6)$$

여기서  $x_k^{\min} \leq x_{jk}(t) \leq x_k^{\max}$ 이다.

Step 4: *Individual & global best updating* - 새로운 위치정보를 가진 *particle*들은 목적함수에 의해 평가된다. 각 *particle*의 적합도는 이전 *pbest*의 적합도와 비교되며 *pbest*를 재설정한다. *pbest*의 *particle* 중 최적해를 가지는 *particle*의 적합도와 이전 *gbest*의 적합도를 비교하여 최적 위치정보를 가진 *particle*을 *gbest*로 재설정한다.

Step 5: *Stopping criteria* - 종료조건을 만족하면 탐색과정을 종료하고, 그렇지 않으면 Step 2부터 반복한다.

Step 6: *Optimal parameter* - 최종적으로 생성된 *gbest*는 최적의 위치정보를 가진다.

### 1.3 PSO의 특징

PSO 알고리즘은 자연선택의 진화 메커니즘이 아닌 새, 물고기 떼와 같은 생명체의 사회적 행동양식을 기반으로 이루어진다. 사회과학측면에서 PSO는 사회(social) 모델과 인지(cognition) 모델의 결합이다. 사회적 요소의 특징으로 각 개체들은 그들 자신의 경험을 무시하고 이웃 개체의 성공적인 확신에 따라 그들의 행동양식을 조절한다. 반대로 인지적 요소에서 각 개체들은 독립적인 존재로서 다루어진다. 이러한 특징들을 이용하여 PSO는 각 개체의 위치정보를 조절한다. PSO의 특징은 다음과 같다[8-14,18].

- PSO는 생명체의 사회적 행동양식을 기반으로 한다.
- PSO는 개체군을 기반으로 한 알고리즘(병렬처리)으로, 지역극소에 빠질 위험성이 적고 해 공간 전체를 탐색할 수 있는 능력을 가진다.
- PSO의 각 개체들(particles)은 그들의 속도와, 경험치 그리고 이웃 동료들의 경험을 바탕으로 위치정보를 조절한다.
- PSO는 주어진 문제의 해결을 위해 목적함수(성능지수)를 이용한다. 따라서 PSO는 미분가능하지 않은 함수들을 쉽게 다룰 수 있다.
- PSO는 확률 최적화 알고리즘의 한 종류라 할 수 있으며, 복잡하고, 불확실한 영역에서의 탐색이 가능하다. 이것은 PSO를 기존 방법들 보다 좀 더 유연하고 강인하게 만든다.
- PSO는 전역탐색과 지역탐색 사이에서의 관계를 유연하게 조절할 수 있다. 이것은 빠른 수렴과 탐색능력을 강화할 수 있는 PSO의 특징이다.
- PSO는 기존 방법론들과는 달리 최적해의 탐색이 초기 값에 의존하지 않는다. 즉 탐색공간의 어느 곳에서 출발하든지, 최적 해의 수렴을 보여준다.
- PSO는 짧은 시간 안에 최적의 해를 생성할 수 있으며, 다른 확률적 방법들보다 안정적인 수렴 특징을 나타낸다.
- PSO는 이론의 간결성, 구현의 용이성, 연산의 효율성의 특징을 가진다. PSO는 단순한 수학 연산자만이 요구되며, 연산처리를 위한 많은 메모리나 빠른 속도를 요구하지 않는다.

## 2. GA

유전자 알고리즘(Genetic Algorithm; GA[1-3,23])은 생물학적 진화, 자연선택, 유전자 재조합의 원리에 바탕을 둔 탐색방법으로, 유전적 계승과 생존경쟁이라는 자연현상을 알고리즘 형태로 모델링한 확률적 탐색방법이다. 유전적 계승은 각 세대에 새로운 자손을 생산하는 유전학 과정을 흉내낸 것이며, 생존경쟁은 세대에서 세대로 집단을 만들어 가는 다윈의 진화과정을 흉내낸 것이다.

### 2.1 GA의 연산자

자연계의 진화 현상을 모방하여 주어진 문제에 대한 최적의 해를 이끌어내는 GA의 구성요소는 목적함수(objective function)와 적합도(fitness), 염색체(chromosome), 염색체 표현방법(encoding schemes), 군집(population), 유전 연산자(genetic operators) 등이며, 자연계 진화의 기본 요소와 매우 유사하다.

*Fitness, objective function:* 먼저 풀고자하는 문제, 즉 환경이 존재해야 한다. 환경이란 그 자체로서의 의미보다는 그 환경 속에 살고 있는 개체(individual)가 얼마나 잘 적응해서 살고 있는지, 얼마나 적합한지를 나타내는 값(fitness)을 정하는 역할 한다. 풀고자하는 문제를 정의하는 GA의 환경을 적합도함수 또는 목적함수라고 한다.

*Individual, chromosome:* 주어진 환경 속에 살고 있는 각 개체들은 염색체(chromosome)로 구성되어 있기 때문에 개체의 형태는 염색체의 표현이라 생각할 수 있다. 염색체의 표현은 적합도 함수(목적 함수)의 관점에서 보면 함수의 가능한 해의 표현이라 할 수 있다.

*Population:* 염색체의 형태로 나타내어진 개체들은 GA에 적용될 수 있도록 개체들의 군을 이루는데 이를 군집(population)이라 부른다. 군집은 GA가 다른 탐색 알고리즘과 가장 크게 구별되는 부분이다. 다른 탐색 알고리즘이 점에 의한 탐색이라고 말한다면 GA는 군집에 의한 즉 개체군에 의한 탐색이라고 말할 수 있다. 즉, 군집에 속하는 잠재적인 해를 나타내는 많은 개체들이 GA의 연산자들을 사용하여 서로의 유전 정보를 교환하고 때로는 새로운 유전 정보를 획득함으로써 효율적으로 전역 최적점(global optimum)을 향해 탐색해 나갈 수 있다.

*Encoding scheme:* GA는 최적의 해를 찾기 위해 염색체의 형태로 인코딩된 개체의 염색체를 조작한다. 즉, 염색체 변형을 위한 연산자를 적용시킴으로써 유전자의 군집이 유익한 정보를 바탕으로 해 공간을 탐색한다.

*Genetic operators:* 많이 사용하는 GA의 연산자는 군집의 개체들 중 유익한 정보를 가지고 있는 개체들을 선별하는 선택(selection) 연산자, 두 개체의 유전 정보를 교환하기 위한 교배(crossover) 연산자와 군집 내의 모든 개체들에 존재하지 않는 새로운 유전 정보를 부여하기 위한 돌연변이(mutation) 연산자가 있다. 이들 연산자는 확률이라는 정보와 각 개체들의 환경에 대한 적합도를 바탕으로 적용된다.

*Selection:* 선택 연산자(selection operator)는 다음 세대를 위해 환경에 잘 적응한 해들은 살아남고 그렇지 못한 해들은 도태되도록 교배 유전자가 선택되어지는 과정으로 자연선택(natural selection) 현상을 모델링한다. 각 개체들은 그

자신의 적합도에 따라 선택 연산자에 의해 다음세대를 생산할 수 있는 기회를 차등적으로 부여받는다. 즉 각각의 개체를 적합도에 근거하여 다음 세대에 영향을 미칠 기회를 결정하는 과정이다.

**Crossover:** 교배 연산자(crossover operator)는 한 쌍의 유전자들이 염색체를 교환하는 과정이다. A와 B라는 유전자가 있다면, 특정 부분은 A의 염색체를 취하고 또 다른 부분은 B의 유전자를 취하는 것이다.

**Mutation:** 돌연변이 연산자(mutation operator)는 문자열 위치의 값을 임의로 변경한다. 2진 코딩에서 돌연변이는 0을 1로 또는 1을 0으로 바꾸는 것을 의미하며, 실수코딩에서는 임의의 값을 더하거나 빼주는 방법을 이용하기도 한다. 앞에서 설명한 교배 연산자에 의해서 개체들 사이에 유전 정보가 서로 교환되지만 모든 해 공간을 탐색하기 위한 유전 정보가 현재 군집내의 개체들에 들어 있지 않다면 교배 연산자를 아무리 적용시키더라도 더 이상의 탐색이 이루어질 수 없다. 또한 선택과 교배연산은 종종 유용한 잠재능력을 가지고 있는 유전자를 잃는 경우가 있다. 따라서 새로운 유전 형질을 부여할 수 있는 돌연변이 연산자는 필수적이다.

그밖에 고려할 수 있는 구성요소로, 세대, 군집의 크기, 교배 확률, 돌연변이 확률, 종료 조건 등이 있다. 이들 중 세대와 군집의 크기를 크게 할 경우 최적 해에 도달할 확률은 높으나 많은 기억용량과 계산시간을 필요로 한다. 반면 세대와 군집의 크기를 작게 설정할 경우는 적합도 계산에 필요한 시간을 절약할 수 있으나 개체간 다양성의 빠른 손실로 인해 최적 해를 구하기 전에 수렴할 위험성을 내포한다[14].

## 2.2 GA 알고리즘

GA에서 하나의 개체는 하나의 생명체와 같이 행동을 하며, 군집은 조그마한 사회와 같은 역할을 한다. 개체들은 군집 안에서 짝을 만나 자신들의 우수한 성질을 다음세대에 유전시키면서 군집을 발전시켜 나간다. GA 알고리즘을 단계별로 살펴보면

**Step 1: Initialization** - 적용할 목적함수와 유전자로 이용할 변수를 결정하고, 개체와 군집의 크기를 결정한다. 군집의 크기, 즉 개체의 수는 문제의 난이도나 성질에 의존하지만 일반적으로 수십 개 이상 발생시킨다. 너무 적으면 병렬적 처리를 특징으로 하는 GA의 장점이 발휘되지 않고 반대로 너무 많으면 한 세대 당 연산량이 많아지기 때문에 적절한 개체의 수를 결정해야 한다. 랜덤 발생을 이용해 초기 세대 유전자 개체의 염색체를 형성한다.

**Step 2: Evaluation** - 생성된 군집에 대한 적합도를 계산한다. 모델의 생존능력을 평가하는 GA의 적합도(fitness)는 목적함수로부터 얻어진다. 환경에 대한 생물의 적합도는 장점을 의미하고 크면 클수록 더 많은 보상을 받을 수 있듯이, GA에서도 적합도가 큰 개체가 더 많은 보상을 받을 수 있도록 일반적으로 최대화 문제 형태로 기술된다.

**Step 3: Reproduction** - 주어진 유전자 집단에 대한 적합도를 바탕으로 다음 세대를 생성한다. 재생산 과정은 선택(selection), 교배(crossover), 돌연변이(mutation) 연산자를 이용하여, 개체의 생존 분포를 결정하고 2개의 개체 사이에서

염색체를 바꾸어 넣어 새로운 개체를 발생시킨다. 여기에 더하여, 유전자의 어떤 부분의 값을 강제적으로 바꾸고, 유전자 집단으로서의 다양성, 결국 흠어짐을 크게 한다. 즉 돌연변이 연산을 수행한다.

**Step 4: Elitist strategy** - 진화과정 중 유전 연산자의 확률적 속성 때문에 한 세대의 최적 개체가 다음 세대에서 살아남지 못하고 소멸되는 경우가 발생할 수 있다. 최적 개체의 소멸은 좋은 특성을 가지는 유전자를 잃게 되어 탐색이 침체되는 원인을 제공한다. 엘리트 전략(elitist strategy)은 집단 내에서 가장 강한 개체가 다음 세대로 변경되지 않고 전달되는 것을 보장하기 위해 고안된 것으로 한 세대의 최적 개체를 다음 세대에서도 생존하도록 보장해주는 방법이다.

**Step 5: Stop condition** - 계산된 적합도를 기준으로 만족하는 개체의 유무를 체크한다. 만족한 결과를 보이는 개체가 존재하면 알고리즘을 종료하고, 그렇지 않으면 평가와 재생산 단계로 돌아가 위의 과정을 반복한다.

## 2.3 GA의 특징

GA는 “적자생존”의 생물학 원리에 바탕을 둔 최적화 기법중의 하나로 자연계의 생명체 중 환경에 잘 적응한 개체가 좀더 많은 자손을 남길 수 있다는 자연선택 과정과 생명체의 설계도와 같은 유전자의 변화를 통해서 좋은 방향으로 발전해 나간다는 자연 진화의 과정인 유전자 진화 메커니즘에 바탕을 둔 탐색 알고리즘이다. GA의 특징은 다음과 같다[1-3].

- 자연계에 있어서 생물의 유전과 진화 메커니즘을 공학적으로 모델화함으로써 생물의 환경 적응능력과 자연도태의 원리를 기반으로 만들어진 계산모델이다.
- 실세계의 문제를 풀기 위해 잠재적인 해들을 코딩된 개체로 나타내고, 여러 개의 개체들을 모아 군집을 형성한 뒤, 세대를 거듭하면서 이들의 유전 정보를 서로 교환하거나 새로운 유전 정보를 부여하여 적자생존의 법칙에 따른 진화과정을 통해, 주어진 문제에 대한 최적의 해를 찾아가간다.
- 잠재적으로 좋은 결과를 가져다주는 과정을 추적하고, 반면에 지역 극소에 빠질 가능성을 제거하기 위해 임의의 지점을 검사함으로써, 해 공간 전체를 탐색할 수 있는 능력을 가진다.
- 각 개체들은 선택, 교배, 돌연변이와 같은 진화연산자에 의해 다루어진다.
- 하나의 지점부터 최적지점으로가 아닌 동시에 여러 지점에서 최적지점을 찾아 나간다.
- 미분과 같은 수학적 연산이 아닌 결과의 적합도를 기준(목적함수)으로 수행된다.
- 결정적인 방법이 아닌 확률적인 방법이다.

## III. PSO와 GA의 특징비교와 차원성

PSO와 GA는 각각의 독립된 영역에서 좋은 결과를 보여주고 있다. 그러나 이들은 서로 공통된 특징과 구별된 특징을 가진다. 이 장에서는 이들 특징을 비교하고 실험예제를 통해 PSO의 특징을 검증하고자 한다.

1. PSO와 GA의 비교

GA와 PSO는 다양한 영역에서 성공적으로 적용되었으며 다음과 같은 공통적인 특징으로 유용한 정보를 제공하고 있다.

- 주어진 문제의 해결을 위해 미분과 같은 수학적 연산이 아닌 목적함수를 이용한다.
- 결정적인 방법이 아닌 확률적인 최적화 알고리즘의 한 종류이다
- 개체군을 기반으로 한 알고리즘으로 하나의 지점부터 최적지점으로가 아닌 동시에 여러 지점에서 최적지점을 찾아 나간다.
- 지역 극소에 빠질 가능성을 제거하고 임의의 지점을 검사함으로써, 해 공간 전체를 탐색할 수 있는 능력을 가진다.

두 최적화 기법은 기본적인 접근방법에 있어서 상호 다른 카테고리 안에서 독립적으로 존재한다. 즉, PSO는 swarm의 사회적 행동양식을 바탕으로, GA는 염색체의 자연선택을 기반으로 구현된다. 그러나 PSO는 GA에 비교해 몇 가지 유용한 특징을 가진다.

PSO는 하나의 연산자(velocity)를 가지며, GA는 세 개의 연산자(selection, crossover, mutation)를 가진다. 또한 GA의 연산자들은 다양한 형태로 선택될 수 있다. 예를 들어 선택연산자로서 tournament 또는 roulettewheel을, 교배연산자로서 일점 또는 이점연산자를 선택할 수 있다. 최적화 문제에 대해 많은 수의 연산자는 최적의 연산자 선택이라는 또 다른 문제를 야기할 수 있다. 따라서 적은 수의 연산자는 연산자 선택의 필요성의 제거와 함께 시간의 이득을 가진다[12].

PSO는 GA와 달리 수렴을 조절하는 능력이 있다. 물론 교배연산과 돌연변이 연산은 GA의 수렴에 영향을 미친다. 그러나 관성하중(w)과 같은 제어파라미터를 가지지 못한다. 수렴성의 침체는 모든 개체들이 본질적으로 같은 코드를 처리할 때 발생한다. 그러나 PSO는 이러한 문제를 극복하고 조절할 수 있는 기능을 가지고 있다[12].

또 한 가지 특징은, 정보유지에 있다. PSO에서 좋은 해의 지식정보는 모든 particle에 의해 유지된다. 그러나 GA는 개체군의 변화와 함께 이전 지식정보는 파괴된다. 물론 엘리트 전략에 의해 유지되기는 하지만 모든 유용한 개체정보를 가지고 있지 못하다[19]. PSO의 particle들은 swarm안에서 정보유지 뿐만 아니라 그 정보를 공유함으로써 상호 발전적인 협력관계를 가진다. 그러나 GA의 개체들은 그들의 군집 안에서 독립적으로 존재하며 상호배타성을 가진다 [12,19].

PSO는 간단한 개념, 쉬운 구현, 빠른 수렴 때문에 오늘날 많은 주목을 받게 되었고 다양한 영역에서 광범위하게 응용되고 있다.

2. 실험에 의한 PSO와 GA의 비교

이 절에서는 수치예제를 통해 PSO와 GA의 차원성에 대해 비교하고자 한다. 주어진 문제 해결을 위한 PSO와 GA에 관련된 파라미터는 표 1과 같다.

표 1. PSO와 GA의 파라미터.

Table 1. Parameters of PSO and GA.

Parameter	PSO	Parameter	GA
Generation	2000	Generation	2000
Swarm size	100	Population size	100
Data type	Real	Data type	Real
$v_{max}$	탐색공간의 20 %	Selection	Roulette
w	(5)	Crossover	One point
[wmin wmax]	[0.4 0.9]	Crossover rate	0.75
$c_1, c_2$	2.0	Mutation	Uniform
		Mutation rate	0.1

표 2. PSO와 GA의 평가를 위한 목적함수.

Table 2. Objective function for evaluation of PSO and GA.

Function	Dimension	Solution space	Objective point	Objective value ( $f_{min}$ )
$f_1$	2	$-2 \leq x_i \leq 2, i=1,2$	(0, -1)	3
$f_2$	3	$-30 \leq x_i \leq 30, i=1,2,3$	$x_i=1, \forall i$	0
$f_3$	6	$-5.12 \leq x_i \leq 5.12, i=1, \dots, 6$	$x_i=0, \forall i$	0
$f_4$	10	$-600 \leq x_i \leq 600, i=1, \dots, 10$	$x_i=100, \forall i$	0

표 3. Goldstein-price 함수에 대한 결과.

Table 3. Results for goldstein-price function.

Classification	PSO	GA	
AVG±STD	3±5.8385e-008	623.53±6123.7	
Best $f_{min}$	2.999997854233	3.000026370227	
Best solution	$x_1$	-0.000013474440	0.000316619873
	$x_2$	-0.999990999699	-1.000000000000

$f_1$  : Goldstein-Price function[19,20]

$$f_1 = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

$f_2$  : Rosenbrock function[12,20-22]

$$f_2 = \sum_{i=2}^3 100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2$$

$f_3$  : Rastigrin function[12,22]

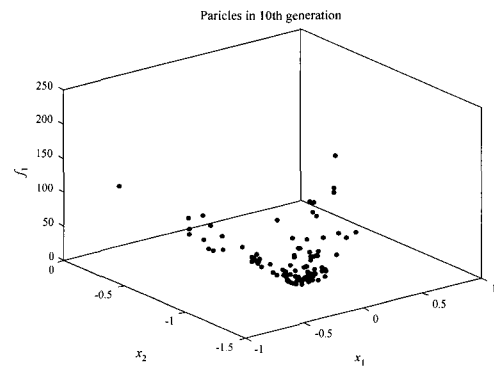
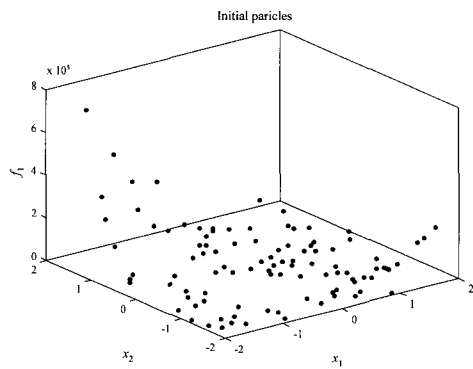
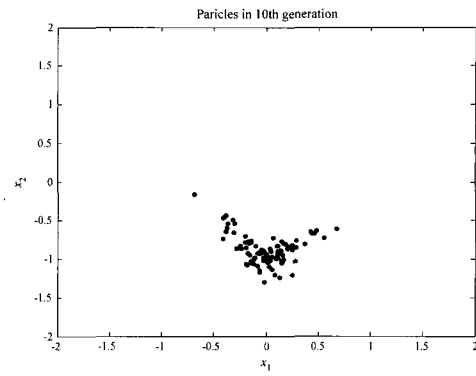
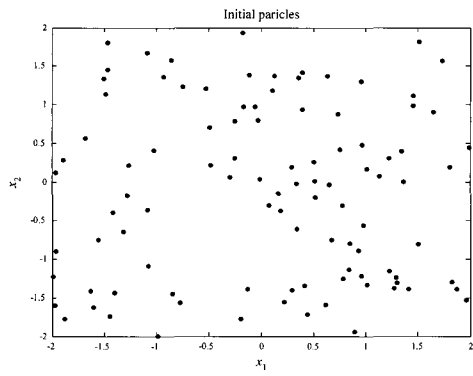
$$f_3 = \sum_{i=1}^6 (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

$f_4$  : Griewank function[12,20,22]

$$f_4 = \frac{1}{4000} \sum_{i=1}^{10} (x_i - 100)^2 - \prod_{i=1}^{10} \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$$

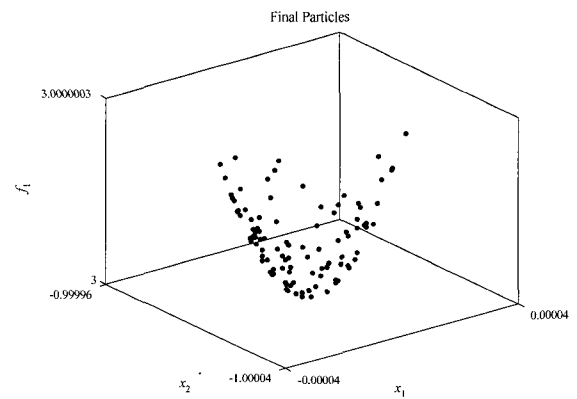
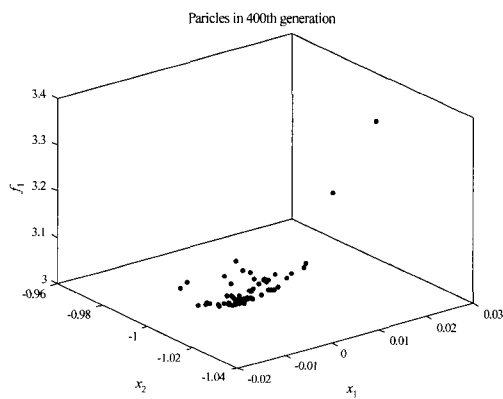
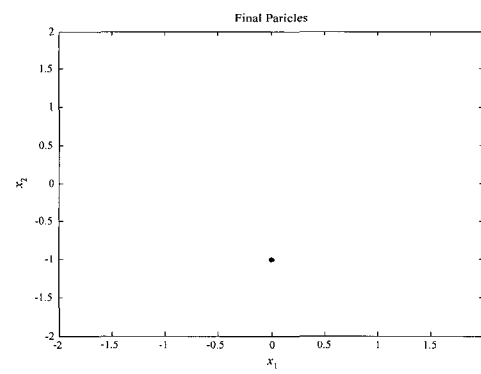
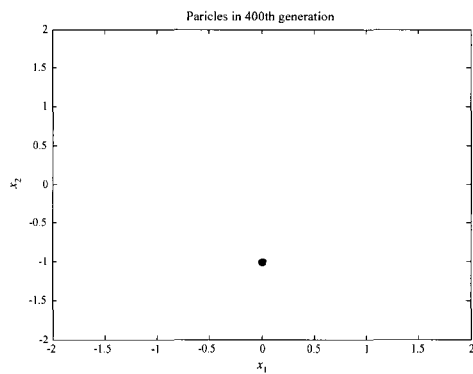
본 논문에서는 PSO와 GA의 비교를 위해 표 2와 같은 비선형 함수를 이용하며, 이들 함수는 최적화 알고리즘의 평가 함수로 자주 이용된다.

표 3은  $f_1$  함수(Goldstein-Price function)에 대한 결과를 보여준다. 표에서 AVG와 STD는 최종 세대의 군집(swarm과 population)에 대한 개체들이 가지는 목적함수의 평균과 표



(a) Initial generation swarm (AVG±STD : 44691±1.0194e+05)

(b) 10th generation swarm (AVG±STD : 26.792±29.764)



(c) 400th generation swarm (AVG±STD : 3.0141±0.0436)

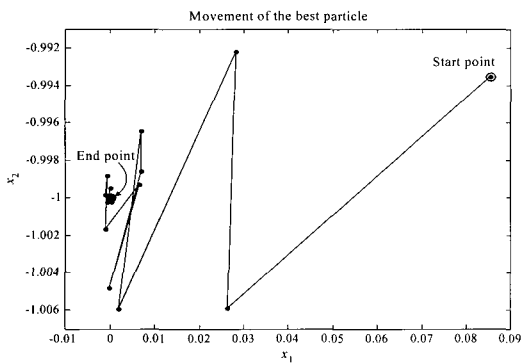
(d) Final generation swarm (AVG±STD : 3±5.8385e-08)

그림 1. 목적함수에 대한 swarm의 이동.

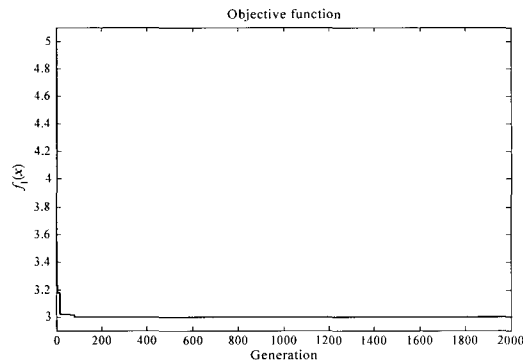
Fig. 1. Movement of swarm for the objective function.

준편차를 나타낸다. PSO는 100개의 개체들이 목표점을 향해 모여 있음을 나타내며, GA의 개체들은 흩어져 있는 모습을 보여준다. 이것은 개체들의 상호관계를 나타낸다. PSO의 경우 개체들이 정보공유를 통해 한 목표점을 향하고 있는 반면, GA의 개체들은 상호 독립적인 관계로 존재한다. 'Best solution'은 PSO와 GA에 의해 생성된 최적의 해를 나타내며, 이들에 의한 목적함수의 결과는 'Best  $f_{min}$ ' 값이다. 표 3의 결과로부터 두 알고리즘은 최종목표 점에 도달하였음을 알 수 있다. PSO에서 좋은 해의 지식정보는 모든 particle에 의해 유지되지만 GA는 개체군의 변화와 함께 이전 지식정보는 파괴된다. 물론 엘리트 전략에 의해 유지되기는 하지만 모든 유용한 개체정보를 가지고 있지 못함[19]을 표 3에서 보여주고 있다.

그림 1은 PSO의 particle들이 목표점을 향해 가는 모습을 세대별로 보여주고 있다. 여기서 particle은  $pbest$ 의 개체들이다. 그림 1의 (c)와 (d)에서 알 수 있듯이 최적의 위치정



(a) Movements of  $gbest$



(b) Optimized process

그림 2. 목적함수에 대한 PSO에 의한 최적화.

Fig. 2. Optimization of PSO for the objective function.

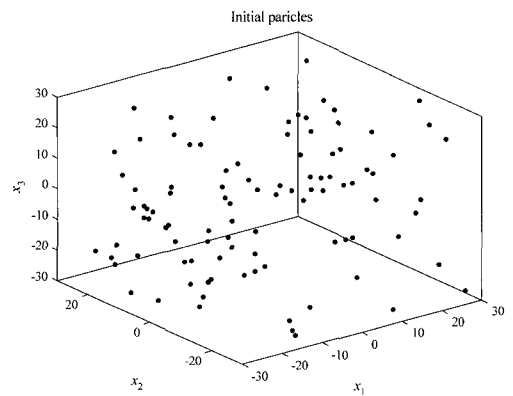
표 4. Rosenbrock 함수에 대한 결과.

Table 4. Results for rosenbrock function.

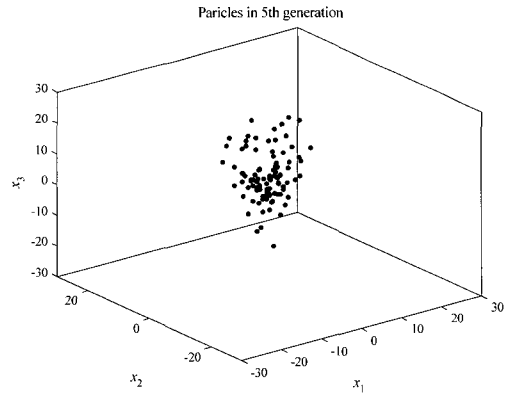
Classification	PSO	GA	
AVG±STD	0.021708±0.1206	13114±65086	
Best $f_{min}$	0.000000000076	8.446863174438	
Best solution	$x_1$	0.999996244907	1.933593750000
	$x_2$	0.999992311001	3.691406250000
	$x_3$	0.999984622002	13.593750000000

보를 공유함으로써 particle들은 하나의 목표점을 가지며, 속도(velocity) 연산자에 의해 개체의 위치를 조절해 간다. 그림 2는 global best particle ( $gbest$ )의 움직임과 이 particle에 의한 목표점 도달과정을 보여준다.

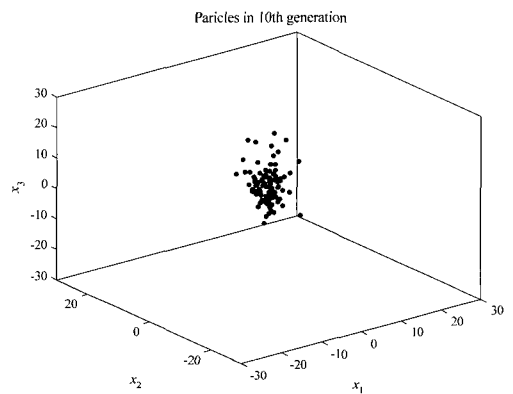
표 4은  $f_2$  함수(Rosenbrock function)에 대한 결과를 보여준다. 평균(AVG)과 표준편차(STD)로부터 PSO와 GA 개체들의 상호관계를 판단 할 수 있다. 이것은 표 3의 결과에서도 나타나는 특징으로 PSO의 particle들은 swarm안에서 정보를 공유함으로써 상호 발전적인 협력관계를 가지지만 GA의 개체들은 그들의 군집 안에서 독립적으로 존재하며 상호배타적 특징을 가진다. 표 4의 결과는 PSO가 GA보다는 최종 목표점에 잘 도달하였음을 보여준다.



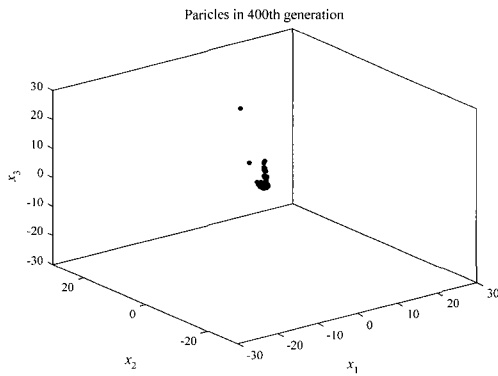
(a) Initial generation swarm ( $3.2049e+07 \pm 2.9331e+07$ )



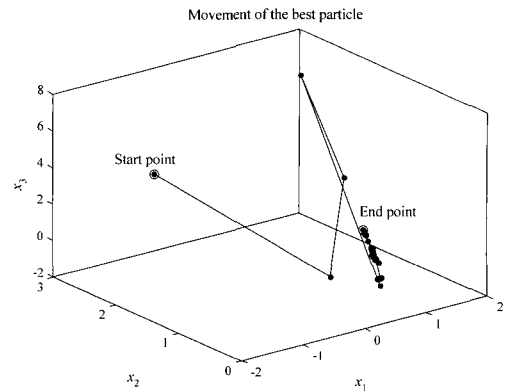
(b) 5th generation swarm ( $73059 \pm 1.3927e+05$ )



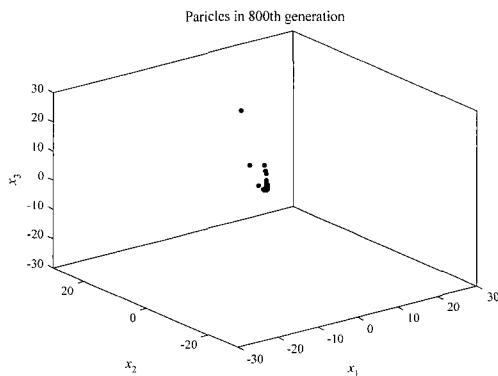
(c) 10th generation swarm ( $15177 \pm 38465$ )



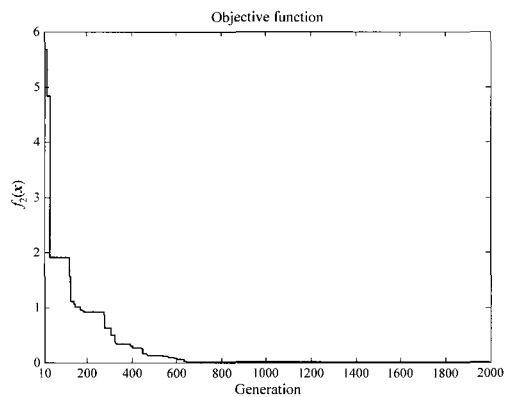
(d) 400th generation swarm (13.192±17.66)



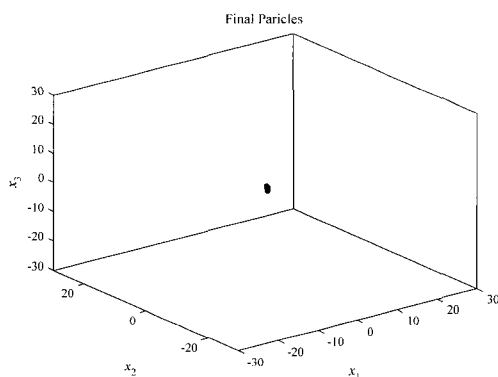
(a) Movements of *gbest*



(e) 800th generation swarm (2.5684±8.3378)



(b) Optimization proce



(f) Final generation swarm (0.021708±0.1206)

그림 3. 목적함수에 대한 swarm의 이동.  
Fig. 3. Movement of swarm for the objective function.

그림 3은 PSO의 particle들이 목표점을 향해 가는 모습을 세대별로 보여주고 있으며, 그림 4는 *gbest*의 움직임과 이 particle에 의한 목표점 도달과정을 나타낸다.

$f_2$  함수(Rosenbrock function)에 대한 PSO와 GA의 최적화 과정을 그림 5에서 비교하여 보여준다. GA는 빠른 수렴을 보이지만 목표점에 도달하지 못한 반면, PSO는 600세대에서 수렴특징을 나타내며, 최적해의 탐색에 성공하였음을 보여준다. 이것은 PSO의 특징을 나타내는 것으로 PSO는 GA의 교배연산이나 돌연변이 연산과는 달리 관성하중에 의해 수렴 특징을 조절함으로써 수렴의 침체문제를 극복할 수 있는 기능을 가진다[12].

그림 4. 목적함수에 대한 PSO에 의한 최적화.  
Fig. 4. Optimization of PSO for the objective function.

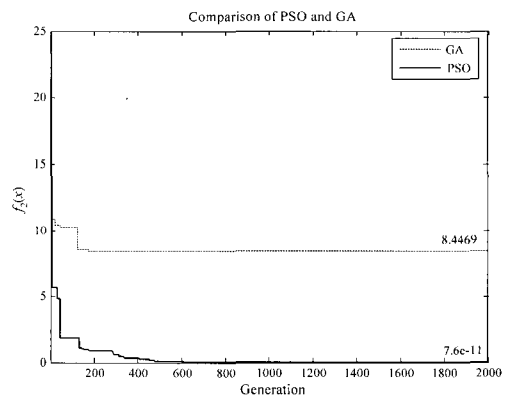


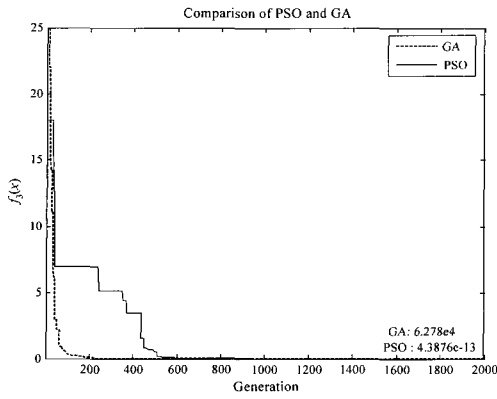
그림 5. PSO와 GA의 최적화 비교.  
Fig. 5. Comparison of optimizations for PSO and GA.

표 5. Rastigrin와 griewank 함수에 대한 결과.

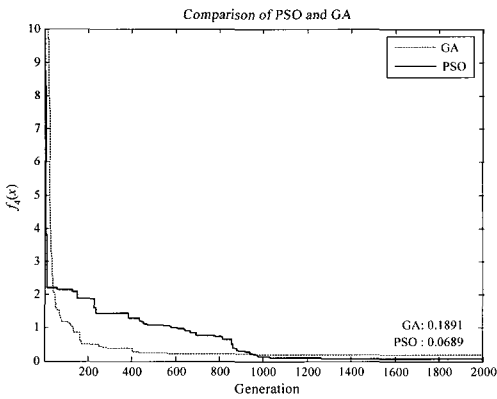
Table 5. Results for rastigrin and griewank functions.

Classification		PSO	GA
$f_3$	AVG±STD	3.66e-013±1.2e-013	0.59752±2.9548
	Best $f_{min}$	4.3876e-013	0.000627821893
$f_4$	AVG±STD	0.10633±0.10563	0.31718±0.51485
	Best $f_{min}$	0.068920336664	0.189074456692





(a) Rastigrin function



(b) Griewank function

그림 6. PSO와 GA의 최적화 비교.

Fig. 6. Comparison of optimizations for PSO and GA.

$f_3$ (Rastigrin)과  $f_4$ (Griewank)에 대한 결과 PSO 결과는 표 5와 같고 그림 6은 이들의 탐색과정을 보여준다. GA는 PSO에 비해 빠른 수렴특징을 보이지만 최적 해에는 미치지 못한 결과를 나타내고 있으며, 반면 PSO는 수렴특징을 조절하면서 최종 목표점에 도달하였다.

지금까지 몇 종류의 함수들의 통해 PSO의 특징들을 살펴보았다. 결과에서 보여주는 것처럼 수치예제들은 particle의 움직임을 통해 정보공유와 수렴조절의 특징을 GA와 비교하여 잘 나타내고 있다.

이번 예제에서는 PSO와 GA가 차원이 증가하는 경우, 즉, 고차원에서는 어떠한 특징을 가지고 있는지 살펴보도록 하자. 차원의 증가에 따른 특징을 비교하기 위해 아래 식과 같은 Schwefel 함수[22]를 이용한다.

$$f_5 = - \sum_{i=1}^n (x_i \sin(\sqrt{|x_i|})), -500 \leq x_i \leq 500$$

여기서  $n=3, 5, 10, 15, 20, 25, 30, 35, 40, 50, 100$  차원이다.

표 6은  $f_5$ (Schwefel)에 대한 결과를 보여준다. PSO와 GA의 개체들은 고차원일수록 목표점을 중심으로 넓게 분포되고 있지만, 앞의 예제에서 보여준 결과와 마찬가지로 PSO의 정보공유의 특징이 잘 나타나고 있다. 특히 개체군의 평균치는 고차원에서 확연히 구별되고 있다.

표 6. Schwefel 함수에 대한 결과.

Table 6. Results for schwefel functions.

Dimension	Classification	PSO	GA
3	AVG±STD	-1256.9±3.5183e-05	-46.505±324.65
	Best $f_{min}$	-1256.9	-936.6
5	AVG±STD	-1976.5±3.2829e-05	-5.9351±438.19
	Best $f_{min}$	-1976.5	-1317.7
10	AVG±STD	-3953±6.3427e-05	4.7071±626.4
	Best $f_{min}$	-3952.9	-2010.0
15	AVG±STD	-5214.2±0.00013885	-101.51±660.5
	Best $f_{min}$	-5214.2	-2809.8
20	AVG±STD	-7075.3±0.00015698	-48.456±890.28
	Best $f_{min}$	-7075.3	-3250.6
25	AVG±STD	-8356.3±0.00023118	-135.6±1039.2
	Best $f_{min}$	-8356.3	-3792.4
30	AVG±STD	-9702.7±0.00036603	172.88±1117.2
	Best $f_{min}$	-9702.6	-4538.3
35	AVG±STD	-11142±0.00038662	-247.42±1137.7
	Best $f_{min}$	-11141.6	-4514.7
40	AVG±STD	-12578±0.00056063	-251.86±1407.7
	Best $f_{min}$	-12577.7	-5056.5
50	AVG±STD	-14959±0.00080976	-78.718±1416.5
	Best $f_{min}$	-14959.0	-5778.8
100	AVG±STD	-25819±49.747	-126.03±1987.4
	Best $f_{min}$	-25858.1	-7808.2

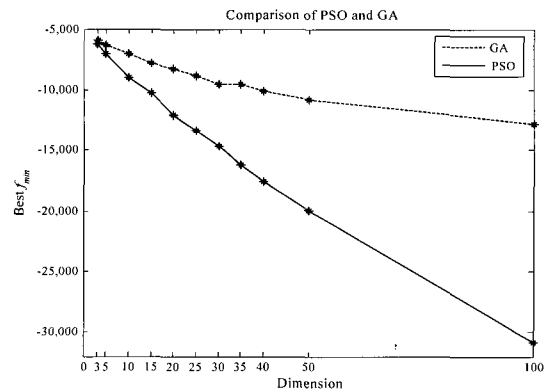


그림 7. PSO와 GA의 차원성 비교.

Fig. 7. Comparison of dimensionalities for PSO and GA.

고차원에 대한 최적 해의 탐색능력은 GA에서 현저히 떨어지고 있음을 그림 7을 통해 알 수 있다. 저차원에서는 PSO와 비슷한 결과를 보이지만 차원의 증가에 대해 PSO의 탐색능력은 꾸준한 반면, GA는 그렇지 못하다. 그림 7은 두 알고리즘의 고차원에서 탐색 능력의 차이를 뚜렷하게 나타내고 있다

IV. 결론

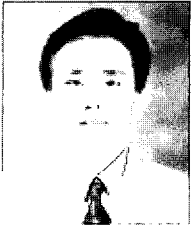
본 논문에서는 생물학적 원리를 이용한 방법 중 대표적인 방법인 PSO(Particle Swarm Optimization)와 GA(Genetic Algorithm)에 대한 특징과 차원성을 실험을 통해 비교하여 다루었다. PSO와 GA의 공통된 특징으로는 생물학적 원리

를 이용한 알고리즘이라는 것과 주어진 문제의 해결을 위해 목적함수를 이용하며, 확실적인 최적화 알고리즘, 개체군을 기반으로 동시에 여러 지점에서 최적지점을 찾아 나간다. 또한 임의의 지점을 검사함으로써, 해 공간 전체를 탐색할 수 있는 능력을 가지고 있다. 그러나 각 알고리즘에 적용된 생물학적 특징은 확연히 다르게 나타난다. 그것은 사회적 행동양식과 자연선택이다. 뿐만 아니라 이들 특징에 기반 한 알고리즘 측면에서도 몇 가지 다른 특징을 찾을 수 있었다. 그것은 알고리즘을 운영하는 연산자와 수렴을 조절하는 능력이다. PSO는 GA에 비해 적은 수의 연산자에 의해 구동되며, 관성하중에 의해 수렴성을 조절함으로써 최적 해에 접근하였다. 수렴침체는 본질적으로 같은 코드를 처리할 때 발생되는데, GA의 경우 이를 극복하지 못하는 문제점을 보였다. 또 한 가지 PSO의 특징은 최적 해에 대한 정보를 각 개체들이 상호 공유를 한다는 것이다. 즉 개체군 안에서 정보를 공유함으로써 상호 발전적인 협력관계를 가진다. 따라서 GA와 같이 엘리트 전략에 의해 유지되는 하나의 유용한 정보만을 제공하는 것이 아니라 개체군 전체가 유용한 위치정보를 제공하며 해에 대한 방향을 제시해준다. 이러한 특징들은 실험을 통해 확인할 수 있었으며, 특히 고차원으로 갈수록 두드러진 차이점을 보였다.

PSO는 실험을 통해 확인된 특징들과 간단한 개념, 쉬운 구현, 빠른 수렴 때문에 오늘날 많은 주목을 받게 되었고 다양한 영역에서 적용을 위한 시도가 이루어지고 있다. PSO는 GA와의 공통되고 구별된 특징을 이용한 합성과 함께 응용범위의 확대와 각 분야에서 두드러진 결과들을 나타낼 것이다.

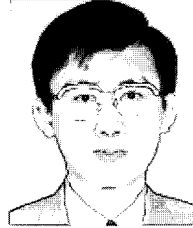
#### 참고문헌

- [1] D. E. Goldberg, *Genetic Algorithms in search, Optimization & Machine Learning*, Addison-wesley, 1989.
- [2] K. A. De Jong, "Are genetic algorithms function optimizers?", In R. Manner and B. Manderick, editors, *Parallel Problem Solving from Nature 2*, North-Holland, Amsterdam, 1992.
- [3] Z. Michalewicz, *Genetic Algorithms+Data Structure =Evolution Programs*, Springer-Verlag, 1992.
- [4] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge, MA: MIT Press, 1992.
- [5] H. G. Beyer, *The Theory of Evolution Strategies*, Berlin, Germany: Springer-Verlag, 2001.
- [6] H. G. Beyer and H. P. Schwefel, Evolution strategies: A comprehensive introduction, *Nat. Comput.*, vol. 1, no. 1, pp. 3-52, 2002.
- [7] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, Piscataway, NJ: IEEE Press, 1996.
- [8] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. IEEE Int. Conf. Neural Networks*, vol. IV, pp. 1942-1948, 1995.
- [9] M. A. Abido, Optimal Design of Power-System Stabilizers Using Particle Swarm Optimization, *IEEE Trans. Energy Conversion*, vol. 17, no. 3, pp. 406-413, 2002.
- [10] Z. L. Gaing, A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System, *IEEE Trans. Energy Conversion*, vol. 19, no. 2, pp. 384-391, 2004.
- [11] K. E. Parsopoulos and M. N. Vrahatis, On the Computation of All Global Minimizers Through Particle Swarm Optimization, *IEEE Trans. Evolutionary Computation*, vol. 8, no. 3, pp. 211-224, 2004.
- [12] J. Robinson and Y. Rahmat-Samii, Particle Swarm Optimization in Electromagnetics, *IEEE Trans. Antennas and Propagation*, vol. 52, no. 2, pp. 397-407, 2004.
- [13] A. Salman, I. Ahmad, and S. Al-Madani, Particle swarm optimization for task assignment problem, *Microprocessors and Microsystems*, vol. 26, no., pp. 363-371, 2002.
- [14] J. Kennedy, The particle swarm: Social adaptation of knowledge, *Proc. IEEE Int. Conf. Evolutionary Comput.*, pp. 303-308, 1997.
- [15] S. Mostaghim and J. Teich, Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO), *Proc. IEEE 2003 Swarm Intelligence Symp.*, pp. 26-3, 2003.
- [16] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, Particle swarm optimization for minimax problems, *Proc. IEEE 2002 Congr. Evolutionary Computation*, pp. 1582-1587, 2002.
- [17] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, Particle swarm optimization for integer programming, *Proc. IEEE 2002 Congr. Evolutionary Computation*, pp. 1576-1581, 2002.
- [18] R. Eberhart and Y. Shi, Comparison between genetic algorithms and particle swarm optimization, *LNCS-Evolutionary Programming VII*, vol. 1447, pp. 611-616, 1998.
- [19] B. Liu, L. Wang, Y. H. Jin, F. Tang and D. X. Huang, Improved particle swarm optimization combined with chaos, *Chaos, Solitons & Fractals*, vol. 25, no. 5, pp. 1261-1271, 2005.
- [20] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and L. M. Wang, An improved GA and a novel PSO-GA-based hybrid algorithm, *Information Processing Letters*, vol. 93, pp. 255-261, 2005.
- [21] B. Brandstatter and U. Baumgartner, Particle Swarm Optimization-Mass-Spring System Analogon, *IEEE Trans. Magnetics*, vol. 38, no. 2, pp. 997-1000, 2002.
- [22] S. He, Q. H. Wu, J. Y. Wen, J. R. Saunders, and R. C. Paton, A particle swarm optimizer with passive congregation, *Bio-Systems*, vol. 78, pp. 135-147, 2004.



**박 병 준**

1998년 원광대 제어계측공학과 졸업. 2000년 동 대학원 석사. 2003년 동 대학원 박사. 현재 캐나다 Alberta 대학 Post-Doc. 관심분야는 시스템자동화, 퍼지이론 및 뉴로퍼지 네트워크 응용, 지능형 모델링 및 계산지능.



**오 성 권**

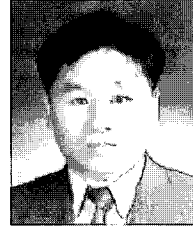
1981년 연세대학교 전기공학과 졸업. 1983년 동 대학원 석사. 1993년 동 대학원 박사. 1993년~2004년 원광대학교 전 기전자 및 정보공학부교수. 2005년~현재 수원대학교 전기공학과 교수. 관심분야는 퍼지 이론 및 신경회로망 응용, 지능형 모델링 및 제어.



**김 용 수**

1981년 연세대학교 전기공학과 졸업. 1983년 KAIST 전기 및 전자공학과공학석사. 1993년 Texas Tech Univ. 전기공학과 공학박사. 1995년~현재 대전대학교 컴퓨터공학부 부교수. 관심분야는 신경회로망, 퍼지 논리, 패턴인식,

영상처리, 침입탐지 시스템.



**안 태 천**

1980년 연세대학교 전기공학과 졸업. 1980년 동 대학원 석사. 1986년 동 대학원 박사. 1982년~현재 원광대학교 전기전자 및 정보공학부 교수. 관심분야는 지능형 모델링, 계측 및 제어 시스템, 계산지능, 디지털 제어.