

PSO(Particle Swarm Optimization) 탐색과정의 가시화 툴 (Visualization Tool of searching process of Particle Swarm Optimization)

유 명 련(Myung-Ryun Yoo), 김 현 철(Hyun-Chul Kim)

요 약

복잡한 문제들의 근사해를 구하기 위하여 최근 다양한 방법들이 소개되고 있다. 이러한 방법들은 주로 금속의 서랭(Annealing)에 의해 금속분자의 에너지가 최저점에 도달하는 과정을 모의실험한 최적화 기법(Simulated Annealing), 생물의 적자생존(Survival of Fittest)과정을 이용한 최적화 기법인 유전적 알고리즘(Genetic Algorithm)등 물리적 현상이나 생물·생명에 관련된 모의를 최적화 문제에 응용한 방법들이다.

최근에 소개된 Particle Swarm Optimization(PSO)는 주로 조류나 어류등의 생물의 무리가 각각의 개체가 가지고 있는 정보를 공유해가며 먹이를 찾아가는 과정을 모의한 기법이다. 하지만, 이 기법의 탐색과정이 명확하게 밝혀져 있지 않다.

본 논문에서는 PSO의 탐색과정을 가시화하는 것을 목적으로 한다. 탐색과정을 가시화하는 작업을 통해 그 탐색능력을 시각적으로 파악하는 것이 가능하며 기법에 관한 이해를 돕고 교육적 효과도 기대 가능하다.

Abstract

To solve the large scale optimization problem approximately, various approaches have been introduced. They are mainly based on recent research advancement of simulations for evolutions, flocking, annealing, and interactions among organisms on artificial environments. The typical ones are simulated annealing(SA), artificial neural network(ANN), genetic algorithms(GA), tabu search(TS), etc.

Recently the particle swarm optimization(PSO) has been introduced. The PSO simulates the process of birds flocking or fish schooling for food, as with the information of each agent is share by other agents. The PSO technique has been applied to various optimization problems of which variables are continuous. However, there are seldom trials for visualization of searching process.

This paper proposes a new visualization tool for searching process particle swarm optimization(PSO) algorithm. The proposed tool is effective for understanding the searching process of PSO method and educational for students.

Key words : PSO, 탐색과정, 가시화

I. 서론

1980년대에 금속의 서랭(Annealing)에 의해 금속분자의 에너지가 최저점에 도달하는 과정을 모의실험한 최적화 기법(Simulated Annealing), 생물의 적자생존(Survival of Fittest)과정을 이용한 최적화 기법인 유전적 알고리즘(Genetic Algorithm)등 물리적 현상이나 생물·생명에 관련된 모의를 최적화 문제에 응용한 기법들이 주목을 받기 시작했다[1, 2]. 이러한 기법들은 모던 휴리스틱스(Modern Heuristics)기법이라 불리며, 문제의 구조에 관계없이 매우 다양하고 복잡한 문제들의 근사해를 구하는

것이 가능한 특징을 가지고 있다. 이러한 이유로 이 기법들은 향후 문제 해결 능력의 향상과 새로운 발전이 기대된다고 하겠다.

최근에, 모던 휴리스틱스 기법의 하나로 Particle Swarm Optimization(PSO)가 제안되었다. 이 기법은 주로 조류나 어류등의 생물의 무리가 각각의 개체가 가지고 있는 정보를 공유해가며 먹이를 찾아가는 과정을 모의한 기법이다[3]. 하지만, 해를 탐색하는 과정이 불분명하다[4]. 해를 탐색하는 과정을 가시화하는 일은 전통적인 수리적 방법으로 해결이 어려운 local optimum이 많은 문제에서 PSO의 적응성 등 해법의 특징과 성능을 이해

하는데 도움이 되며, 이는 해법의 탐색능력을 높일 수 있는 방법을 찾는 데 도움이 될 수 있다. 또한 가시화하는 작업을 통해 해법 이해에 대한 교육적 효과를 기대할 수 있다.

본 연구에서는 PSO의 탐색과정을 가시화하는 것을 목적으로 한다. PSO가 2변수의 연속형 비선형함수(non linear function)를 대상으로 하는 최소화문제(minimization problem)에서 어떠한 해의 탐색과정을 보이는가에 관한 가시화 결과를 보고하는 것을 목적으로 한다.

본 연구논문의 2장에서 PSO의 개요와 특징, 그리고 알고리즘에 관해 설명한다. 3장에서는 PSO의 탐색과정을 가시화하는 방법에 관해 기술한다. 제 4장에서는 PSO의 가시화의 결과와 그 결과로 알 수 있는 사항들과 탐색능력의 검증에 관해 기술한다. 제 5장에서 본 논문의 결론을 기술한다.

II. Particle Swarm Optimization

2.1 PSO의 정의

1995년에 진화형 계산기법의 일종으로 PSO가 J.Kennedy와 R.Eberhart에 의해 제안되었다[4]. PSO는 종래의 새나 물고기 무리의 움직임에 관한 연구로부터 유도되었다. 다시말해, 이러한 무리가 먹이를 찾아가는 과정에서 무리 전체가 정보를 공유한다는 가설과 무리 내부의 에이전트(Agent; PSO에서 무리 내의 각 개체를 지칭)가 지금까지의 자기의 경험과 무리전체에 공유되어 있는 정보를 기초로 하여 행동한다는 개념을 최적화 과정에 도입한 기법이라 할 수 있다.

이 기법은 다음과 같은 장점이 있다.

(1) 알고리즘이 단순하고, 계산 시간이 짧으며 대량의 메모리가 필요없다.

(2) 연속형과 비연속형의 문제 양쪽에 적용 가능하다.

PSO에서 각 에이전트는 각각이 지금까지의 탐색 중 최량의 목적함수 $F(pbest_k)$ 를 기억하고 있다. 또한 각 에이전트는 전체의 에이전트가 이제까지의 탐색과정에서 발견한 해 중에 최량의 해, 다시말해 집단에서 발견한 해 중에 최량의 목적함수값 $F(gbest)$ 와 그 해의 위치 벡터 $gbest$ 의 정보를 공유한다. 각 에이전트는 현재의 위치 벡터와 속도벡터, 그리고 $pbest$, $gbest$ 를 이용해서 식(2.1)에 의해 이동을 하게 된다. 또한 각 에이전트의 위치벡터의 수정은 현재의 위치와 수정된 속도를 이용해서 식(2.2)와 같이 행해진다.

$$v_k^{next} = a_1 v_k^{now} + a_2 rand \times (pbest_k - s_k^{now}) + a_3 rand \times (gbest - s_k^{now}) \quad (k = 1, 2, \dots, N) \quad (2.1)$$

$$s_k^{next} = s_k^{now} + v_k^{next} \quad (k = 1, 2, \dots, N) \quad (2.2)$$

- 단, v_k^{now} : 현재 에이전트 k 의 속도벡터
- s_k^{now} : 현재 에이전트 k 의 위치벡터
- v_k^{next} : 수정된 에이전트 k 의 속도벡터
- s_k^{next} : 수정된 에이전트 k 의 위치벡터
- k : 에이전트 번호
- N : 에이전트의 총 수
- a_1, a_2, a_3 : 가중치 계수
- $pbest_k$: 에이전트 k 가 지금까지의 탐색 중 발견한 최량의 위치벡터
- $gbest$: 전체 에이전트가 지금까지의 탐색 중 발견한 최량의 위치벡터

전술한 개념을 바탕으로 2차원공간에서의 어떤 에이전트의 속도와 위치의 변경에 관한 개념을 그림 2.1에 나타내었다. 단, 편의상 가중치계수 a_1, a_2, a_3 는 모두 1로 동등한 가중치를 갖는 것으로 한다. 아래 그림에서 k 번째 에이전트의 현재위치 s_k^{now} 와 그 속도 v_k^{now} 가 주어졌으므로 한다.

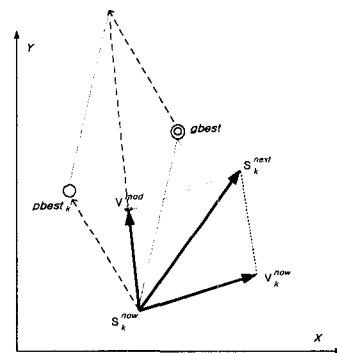


그림 2.1 에이전트의 속도와 위치 수정의 개념

예를 들어, 이 에이전트가 “○”으로 표시된 위치를 $pbest_k$, “◎”로 표시된 위치를 $gbest$ 로 하여 기억하고 있을 때, 에이전트는 다음과 같이 속도를 변경한다; 에이전트 k 는 현재 향하고 있는 방향으로부터 이제까지 자신이 탐색한 해중에 제일 우수한 해 $pbest_k$ 와 집단 전체가 발견한 해중에 제일 우수한 해인 $gbest$ 가 있는 방향으로 자신의 방향을 변경하고자 한다. 이 변경하고자하는 행동은 v_k^{mod} 벡터로 표시 가능하다(v_k^{mod} 는 식 2.1에서 우변의 제2항과 제3항의 합). 즉, 에이전트 k 의 속도의 변경은 현재의 속도 v_k^{now} 와 v_k^{mod} 를 더하는 것으로 이루어지며, 속도를 변경한 후의 에이전트 k 의 새로운 벡터 v_k^{next} 로 표현된다. 이상과 같이 변경한 v_k^{next} 에 따라서,

에이전트 k 는 새로운 위치 s_k^{next} 로 자신의 위치를 변경한다[4, 5].

위에 기술한 PSO의 탐색은 다음과 같은 특징이 있다.

- (1) PSO는 유전적 알고리즘(Genetic Algorithm)등과 같이 복수의 탐색점을 가지며 각 탐색점의 $pbest$ 와 집단의 $gbest$ 를 이용하여 각 탐색점을 확률적으로 변경시켜가는 것에 의해 전역적인 최적해(global optimum solution, 최량해)를 발견하는 방법이다.
- (2) 이제까지의 속도를 유지하는 대역적 탐색(식 2.1의 우변 1항)과 $pbest$, $gbest$ 를 이용하여 그에 가까이 접근하는 국소탐색(식 2.1의 우변 2,3항)의 균형을 잘 이루어 나갈수 있는 기법이다.
- (3) R. Eberhart[4]등에 의해 제안된 원래의 PSO의 변경량은 연속치이고, 연속형 변수를 포함하는 문제에 적용 가능하나, 이 변량을 이산치로 하는 것도 가능하며, 일반적인 이산형 변수를 포함하는 문제에 적용 가능하도록 확장이 가능하다.
- (4) 연속형의 변수와 이산형의 변수가 혼합되어있는 경우에도 전체적인 집단의 탐색에는 모순이 없다.
- (5) PSO의 개념은 원래 2차원 공간에서 고안되었으나 n 차원 공간으로의 확장에도 문제점은 없다.

2.2 PSO의 알고리즘

PSO의 알고리즘은 다음과 같이 단계적으로 기술할 수 있다.

- (step1) 전체 에이전트에 대해 초기 위치 벡터 s_k 와 속도 벡터 v_k 를 난수를 이용해 설정한다.
- (step2) 전체의 에이전트에 대해 초기의 위치 벡터 s_k 를 이제까지의 최량해의 위치 $pbest$ 로 한다. 이때 그 중 가장 우수한 $pbest_k$ 를 전체 에이전트에 관한 최량의 벡터 $gbest$ 로 한다.
- (step3) 에이전트 k 의 속도 벡터 v_k 를 식(2.1)을 이용하여 갱신한다.
- (step4) 에이전트 k 의 위치 벡터 s_k 를 식(2.2)을 이용하여 갱신한다.
- (step5) 에이전트 k 에 관해 현재의 위치에서 목적함수 값 $F(s_k)$ 가 $F(pbest)$ 보다 우수한 목적함수 값을 가지면 $pbest_k = s_k$ 로 한다.
- (step6) step 3에서 step 5까지의 처리를 모든 에이전트에 관해서 실시하였다면 step 7로, 그렇지 않으면 step 3으로 간다.
- (step7) 전체 에이전트에 대해서 $F(pbest_k)$ 가 $F(gbest)$ 보다 우수한 목적함수 값을 가지면 $gbest = pbest_k$ 로 한다.

(step8) 규정된 반복횟수에 도달하면 탐색을 종료하고, 그렇지 않으면 Step 3으로 진행한다.

2.3 가중치 계수의 설정

식(2.1)의 가중치계수 a_1 은 에이전트의 속도 수정량을 변경하는 파라미터의 하나이다. 본 논문에서 가중치계수 a_1 은 식(2.3)에서와 같이 초기치와 종료값을 설정하여 반복횟수가 진행됨에 따라 감소하도록 설정한다.

$$a_1 = W_{max} - \frac{iter \times (W_{max} - W_{min})}{iter_{max}} \quad (2.3)$$

단, W_{max} : 가중치 계수 a_1 의 초기값

W_{min} : 가중치 계수 a_1 의 종료값

$iter$: 현재의 반복 횟수

또한, 가중치계수 a_2 , a_3 는 2.0으로 설정한다. 이것은 식(2.1)의 제 2항 및 제 3항의 계수가 0에서 1까지의 난수와 가중치계수의 곱으로 되어있기 때문에 그 평균이 1이 되도록 하기 위한 것이다[5, 6].

2.4 제약 위반시의 대처법

PSO를 가시화할 때 표시가 가능한 해공간(해 탐색 영역)에는 한계가 있기 때문에, 이에 기초하는 실행가능 영역이 존재하게 된다. 그러나, 식(2.2)에 의해 위치 벡터 s_k (해)의 갱신시(Step5), 새로운 해에 해당하는 위치벡터가 이 해 공간으로부터 벗어날 가능성이 있다. 본 논문에서는 그림 2.2에 나타난 대처방법들 중 한가지를 선택한다[7, 8].

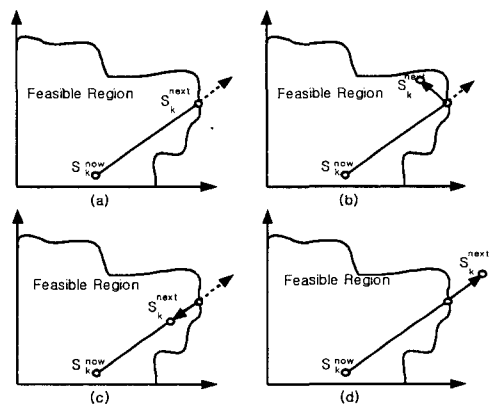


그림 2.2 제약일탈시의 대처 방법
((a)고정, (b)반사, (c)반전 (d)투과)

그림 2.2의 (a), (b), (c), (d)는 다음을 의미한다.

(a) 고정처리

위치벡터 s_k 를 속도벡터 v_k 와 해공간의 경계와의 교차점에 고정시킴

- (b) 반사처리
해공간의 경계를 반사면으로하여 속도 v_k 벡터가 영역을 벗어난 만큼 반사시킴
- (c) 반전처리
속도 벡터 v_k 와 해공간의 경계와의 교차점을 시점으로하여 속도벡터 v_k 가 영역을 벗어난 양만큼 v_k 의 역방향으로 반전시킴
- (d) 투과처리
가상적으로 해공간영역을 허용함

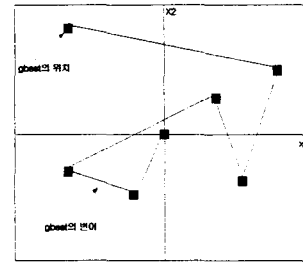


그림 3.2 gbest변이의 가시화

III. 가시화방법

탐색과정의 가시화를 위한 틀은 JAVA 애플릿으로 설계하였다. 컴퓨터의 기종, 운영체제 및 장소에 구애받지 않고 인터넷상에서 탐색과정을 볼 수 있도록 제공하기 위함이다.

PSO의 해 탐색과정을 가시화하는 화면은 에이전트의 움직임을 가시화하는 해공간, 목적함수의 그래프 표시를 위한 그래프공간, 현재의 해 탐색상태를 보여주는 파라메타를 출력하는 공간, 또한 파라메타 입력을 위한 텍스트 필드와 처리 제어를 행하는 버튼등의 도구 부분으로 구성한다. 각각을 다음 절에서 설명한다.

3.1 에이전트 움직임의 가시화

해공간을 탐색하는 하나의 에이전트 k를 가시화 화면에 한 점(x_{1k}, x_{2k})으로 플롯(plot)함으로써 복수 에이전트의 움직임(해 탐색과정)을 가시화 한다. 또한, 현재의 위치에서 가장 우수한 목적함수 값을 갖는 에이전트를 다른 에이전트와 구별하기 위해 다른 색으로 플롯한다. 그 에이전트의 목적함수 값이 현재까지의 최우수 해 이상의 양질의 값이라면, 역시 구별을 위해 색을 변경한다. 그림 3.1은 가시화 한 에이전트의 움직임을 나타낸다.

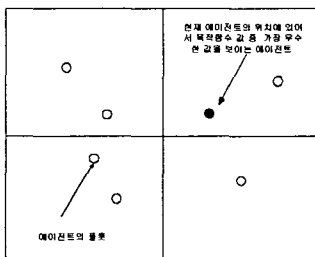


그림 3.1 에이전트 움직임의 가시화 (에이전트 수 7)

규정된 반복횟수의 종료시에는 그림 3.2와 같은 gbest의 궤적을 표시한다.

3.2 그래프 표시

에이전트의 움직임을 표시하는 것과 동시에, 현재의 해 탐색과정의 목적함수 값의 변화를 알수 있도록 그 추이를 표시하는 그래프를 가시화 화면에 나타낸다. 그래프 공간에 있어 종축이 목적함수 값을 표시하며 횡축이 반복횟수를 표시한다. 또한, 그래프 공간의 종축으로 표현되는 목적함수 값은 최소값을 0.0으로 하고, 최대 값은 초기의 목적함수 값으로 한다. 같은 방법으로 횡축으로 표현되는 반복횟수는 최소치를 0으로 하고 최대치는 규정된 반복횟수로 한다. 그림 3.3에 사용될 그래프 공간의 개요를 나타내었다. 또한 다음 두 종류의 그래프가 표시 가능하다.

- (1) 현재까지의 최량의 목적함수값
현재까지의 반복 중에 최량의 목적함수 값, $F(gbest)$ 의 추이를 표시한다. 이는 현재의 값보다 양호한 목적함수 값이 얻어질 때만 새로 갱신되는 값이므로 이 그래프는 수평 혹은 오른쪽으로 기우는 형태가 된다.
- (2) 현재 에이전트의 위치
현재 에이전트의 위치에서 최량의 목적함수값 $Min\{F(s_k)\}$ 을 표시한다. 이는 반복횟수 마다 갱신된다.

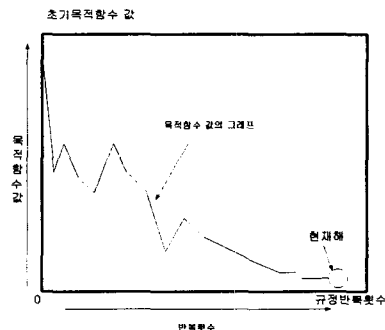


그림 3.3 목적함수 값의 그래프 표시

3.3 파라메터의 출력

반복횟수별로 현재의 처리상태를 표시하는 파라메타를 가시화 화면에 출력한다. 출력하는 파라메타는 현재의

반복횟수, 현재 에이전트의 위치에서의 최량의 목적함수 값 $Min\{F(s_k)\}$, 현재까지의 최우수 목적함수 값 $F(gbest)$, 그리고 그 위치 벡터 $gbest$ 로서, 규정반복횟수 내에서 대역적인 최적해가 얻어진 경우는 해당 반복횟수를 출력한다.

3.4 tool 부분

해법의 탐색과정 및 각 도구의 기능과 이해도를 높이기 위해 처리를 제어하는 다음과 같은 도구들을 가시화 화면에 추가한다.

(1) 파라미터의 입력

파라미터의 변화에 의해 해 탐색과정의 상위를 확인가능하도록 파라미터의 입력 영역을 둔다. 입력 파라미터로는 에이전트수, 규정반복횟수, 에이전트 속도벡터 성분의 상한과 하한값, 해 공간의 제약범위, 에이전트의 속도 수정에 있어서의 가중치계수 $a_1(Wmax, Wmin)$, a_2 , a_3 이다.

(2) 설정의 변경

현재와 다른 설정이 가능하도록 설정을 변경시킬 수 있는 영역을 추가한다. 이 영역을 통해 입력 가능한 설정은, 에이전트의 화면출력의 빈도, 에이전트의 위치벡터가 해의 제약범위를 벗어날 경우의 처리방법, 그래프 표시의 변경, 에이전트의 속도 수정에 사용되는 난수의 인수(seed) 설정 여부 등이다.

(3) 처리의 제어

사용자의 조작으로 처리 개시 또는 종료를 행할 수 있도록 그 기능을 가진 버튼을 추가한다. 또한, 해 탐색과정을 일시 정지시켜 관찰이 가능토록 하기위해 일시 정지 기능을 추가한다.

(4) 목적함수의 선택

화면상의 선택 박스의 항목에서 최소화를 대상으로 하는 목적함수들이 선택가능하다. 선택가능한 함수는 Schaffer $f6$ 함수, Griewank 함수, Sphere 함수, Rastrigrin 함수 등 5가지 비선형 함수이다.

(5) 목적함수의 3차원 그래프 표시

해 공간이 목적함수별로 변화하는 모양이 구체적으로 이해 가능하도록 목적함수 값을 높이 축으로 하여 3차원 그래프를 표시하는 다이얼로그 버튼을 둔다.

(6) 표시속도의 조절

화면상의 스크롤 바를 통해 표시속도를 인터랙티브한 방식으로 조절 가능하다. 이 기능으로 탐색과정을 보다 구체적으로 관찰하며 속도를 조절할 수 있게 된다.

3.5 가시화 화면의 구성

가시화 방법을 그림 3.4의 가시화 화면으로 구성한다.

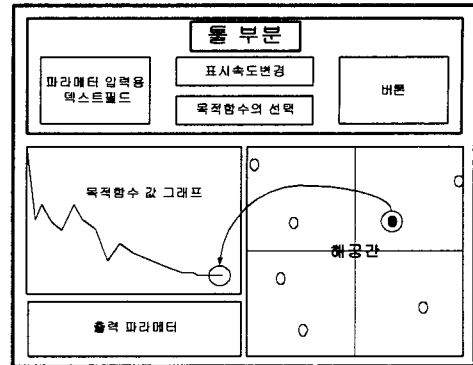


그림 3.4 가시화 화면의 구성
(그래프표시 $Min\{F(s_k)\}$)

IV. 계산 예

4.1 대상 비선형함수

PSO의 탐색과정을 가시화하기 위하여 2변수의 연속 비선형 함수인 식(4.1)과 같은 Schaffer $f6$ 함수 최소화를 목적으로 하는 예제를 다룬다.

$$F(x) = 1 - f6 = 1 - \left(0.5 + \frac{0.5 - \sin^2 \sqrt{x_1^2 + x_2^2}}{(1 + 0.001(x_1^2 + x_2^2))^2} \right) \quad (4.1)$$

이 함수의 대역적인 최적해는 $x_g = (x_1, x_2) = (0, 0)$ 으로 최소화 목적함수 값은 $F(x_g) = 0$ 이다. 그림 4.1에 이 함수의 그래프를 표시한다[8].

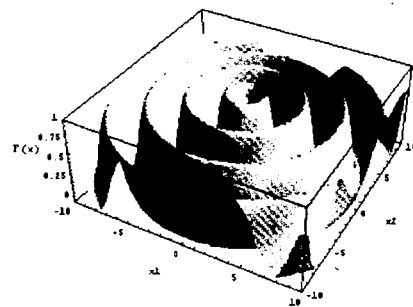


그림 4.1 Schaffer $f6$ 함수의 그래프

4.2 PSO의 가시화 결과

그림 4.2에 3장의 가시화 방법을 이용하여 개발한 프로그램으로 PSO의 탐색과정을 가시화한 예를 나타내었다. 파라미터는 에이전트 20, 규정반복횟수 500, 해의 제약조건범위 $-100 \leq x_1, x_2 \leq 100$, 제약위반시의 대처법은 반사, 속도벡터 0, 성분크기의 범위는 $-100 \leq v_k \leq 100$,

$W_{max}=8.0$, $W_{min}=0.4$, $a_2=2.0$, $a_3=2.0$ 으로 하고, 그래프 표시는 $Min(F(s_k))$ 로 한다. 이 예에서는 그림 4.2(a)에서와 같이 에이전트의 초기위치에서 부터 시작하여 해 탐색과정을 거쳐 그림 4.2(b)와 같이 다수의 에이전트가 이 목적함수의 대역적 최적해인 원점 $(x_1, x_2) = (0.0, 0.0)$ 혹은 그 근방에 도달하고 있으며 결과로 최소 목적함수값 0.0을 얻어낼수 있었다.

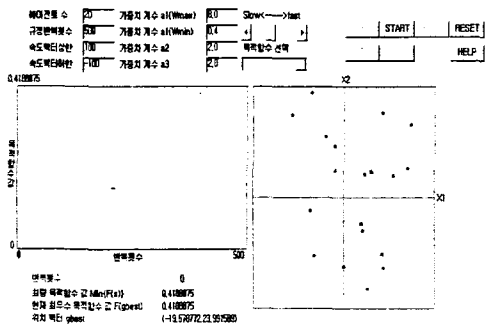


그림 4.2(a) PSO의 가시화 결과 (초기치)

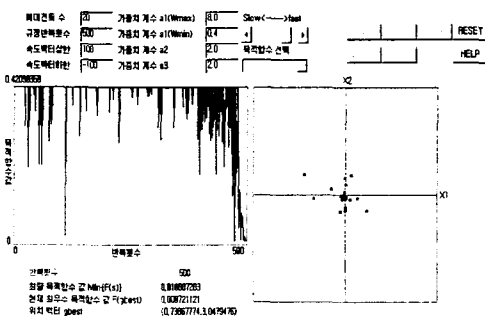


그림 4.2(b) PSO의 가시화 결과 (결과치)

V. 결론

최근에, 조류나 어류 등의 생물의 무리가 각각의 개체가 가지고 있는 정보를 공유해가며 먹이를 찾아가는 과정을 모의한 기법의 하나로 Particle Swarm Optimization(PSO)기법이 제안되어 다양한 응용문제에 그사용이 늘어가고 있으나 이 기법의 탐색능력 등 검증되어야 할 점이 있다. 해 탐색 과정을 가시화 하는 작업은 탐색 능력을 시각적으로 파악하도록 하여 탐색 방법과 능력에 대한 이해를 돕고 또한 교육적인 효과도 기대 가능하다.

본 논문에서는 최근 제안된 모던 휴리스틱(Modern Heuristic) 기법인 Particle Swarm Optimization의 소개와

해의 탐색과정을 가시화 하는 방법을 제안하며, 그 구현수단을 개발하고, 프로그램을 실행하는 일로 PSO의 탐색과정을 가시화하였다. 향후 목적함수에 맞는 파라미터의 조합을 검토함으로써 PSO의 해 탐색과정에 어떠한 변화가 생기든지 관찰하고, 해 탐색능력을 극대화할수 있는 파라미터의 조합을 발견해 권고하고자 한다.

접수일자 : 2002. 9. 13 수정완료 : 2002. 10. 20

참고문헌

- [1] H. Kim et al, "An algorithm for Thermal Unit Maintenance Scheduling Through GA, SA and TS", IEEE Power Engineering Society, pp. 291-295, 1996
- [2] Goldberg, D. E. "Genetic Algorithms in search, Optimization, and Machine Learning", Reading, MA:Addison-Wesley, 1989
- [3] Heppner. F, and U. Grenader, "A Stochastic Nonlinear Model for Coordinated Bird Flocks", AAAS Publications, Washington, DC, 1990
- [4] J. Kennedy and R.Eberhart, "Particle swarm Optimization", Proc of IEEE International Conference on Neural Networks, VOL. IV, pp. 51, Perth, Australia, 1995
- [5] Y.Shi and R.Eberhart, "Parameter Selection in Particle Swarm Optimization", Annual Conference on Evolutionary Programming, pp. 93, San Diego, USA, 1998
- [6] Joseph O'Neil (武藤健志 역) : 獨習, 翔永社,1999
- [7] H. Yoshita, K.Kawata, Y.Hukuyama, Y.Nakanishi, "A Particle Swarm Optimization For Reactive Power and Voltage Control Considering Voltage Stability", Proc. on Intelligent System Application to Power Systems, pp. 103, Rio de Janeiro, Brazil, 1999
- [8] Andrew M. Assad, Norman H. Packard, "Emergent Colonization in Artificial Ecology", Proc. on 5th European Conference on Artificial Life, pp. 37, Paris, 1999



유 명 련 (Myung-Ryun Yoo)

正會員

1994 안동대학교 컴퓨터공학과
(공학사)

1996 포항공과대학교 정보통신대
학원(공학석사)

2002 영남대학교 대학원 컴퓨터 공학과(공학박사)

1996 ~ 현재 안동정보대학 인터넷정보과 교수

관심분야 : 운영체제, 멀티미디어



김 현 철(Hyun-Chul Kim)

正會員

1984 숭실대학교 전자계산학과(학사)

1993 숭실대학교 정보과학대학원(석사)

1996 일본 이바라키대학교 대학원
시스템공학과(박사)

2000 ~ 현재 경주대학교 컴퓨터 전자공학부 교수

관심분야 : 시스템공학
