

工學碩士 學位論文

실수코딩 유전알고리즘과 단체법을 결합을
하이브리드 탐색법

A Hybrid Search Method Combining a Real-Coded Genetic
Algorithm with a Simplex Search Method

指導教授 陳 康 奎

2002年 2月

韓國海洋大學校 大學院

制御計測工學科

姜 元 龍

本 論 文 을 姜 元 龍 의 工 學 碩 士 學 位 論 文 으 로 認 准 함

委 員 長 工 學 博 士 朴 侏 讚 印

委 員 工 學 博 士 河 潤 秀 印

委 員 工 學 博 士 陳 康 奎 印

2001年 12月 22日

韓 國 海 洋 大 學 校 大 學 院

목 차

Abstract	iii
제 1 장 서 론	1
제 2 장 유전 알고리즘과 단체법	4
2.1 유전알고리즘	4
2.1.1 유전알고리즘의 구조	4
2.1.2 유전알고리즘의 병렬화	12
2.1.3 성능지수와 테스트 환경	15
2.2 단체법	21
2.2.1 탐색 원리	22
2.2.2 Nelder-Mead의 단체법	23
제 3 장 제안한 하이브리드 탐색법	32
3.1 하이브리드 탐색법의 구조	32
3.1.1 이주모델에 기초한 병렬성 구현	33
3.1.2 탐색법의 초기화	35
3.1.3 이주자의 선택 및 이주	35
3.2 하이브리드 탐색법의 파라미터	37
3.2.1 RCGA와 단체법의 파라미터	37
3.2.2 이주율의 결정	37
제 4 장 시뮬레이션	39
4.1 시스템의 파라미터 추정	39

4.2 PID 제어기의 파라미터 동조	44
제 5 장 결 론	49
참고문헌	50

Abstract

One of the main drawbacks in applying genetic algorithms (GAs) to complex problems has been the high computational cost due to their slow convergence rate. This fact gives a difficulty in an attempt to use GAs for function optimization problem. To alleviate this difficulty, an island-model based hybrid search method which combines a real-coded genetic algorithm(RCGA) with a simplex search method is presented. Our motivation for employing the RCGA is to introduce best exploration into the hybrid search method, and the simplex search method is to introduce cost-effective exploitation. In an attempt to make effective use of the exploitation operation of the simplex search method in the proposed search framework, we use a parallel architecture where two algorithms run during the isolation time and exchanges migrants. To demonstrate the superiority of the proposed algorithm, it is compared with an alternative optimization technique, RCGA proposed by Michalewicz in two optimization problems for modeling a system with time delay and tuning the parameters of a PID controller.

제 1 장 서 론

근래에 각종 산업분야에서는 효율을 높이고 비용 등을 절감하는 차원에서 시스템의 최적화가 매우 중요한 문제로 대두되고 있다. 최적화란 시스템에 포함된 주요 파라미터를 매개변수로 하는 목적함수(또는 성능지수)를 최소 또는 최대로 하는 것을 뜻한다. 이런 최적화 문제에 접근하는 기존의 방법들은 목적함수 값 외에도 탐색공간에 대한 사전지식(*a priori* knowledge)을 필요로 하는 경우가 대부분이며, 특히 도함수(구배)를 이용하는 방법은 지역해에 수렴할 가능성이 높다. 이러한 단점을 보완하고자 진화전략(Evolution strategies: ES), 시물레이티드 어닐링(Simulated annealing: SA) 등과 같은 자연현상을 흉내낸 알고리즘들이 개발되어 왔고, 1975년에는 자연계의 적자생존(Natural selection) 원리와 유전학(Genetics)의 개념을 기반으로 하는 확률적 탐색법인 유전알고리즘(Genetic algorithm: GA)이 J. Holland[1] 교수에 의해 제안되었다. GA는 목적함수 값 외에 미분 가능성, 연속성과 같은 탐색공간에 대한 부가적인 정보를 요구하지 않고, 전역탐색 하는 능력을 가지기 때문에 모든 학문 분야에서 최적화문제 해결의 도구로서 폭넓게 응용되고 있으며, 특히 시스템 식별 및 제어, 기계학습, 설비배치, 신경회로망, 신호처리, 생명공학 등에서 성공적으로 이용되고 있다[2][3].

전통적으로 GA는 이진코딩 염색체 상에서 돌아가도록 개발되었고 아직도 이진코딩은 많은 연구자들의 관심의 대상이 되고 있지만, 고정밀, 다차원 문제에 적용할 때에는 몇 가지 단점이 불가피하다. 하나는 고정밀도의 해를 요구하면 염색체 길이가 길어져 매우 큰 탐색공간이 만들어지고 계산부담이 급격히 증가한다는 것이다. 계산부담을 줄이기 위해서는 정밀도를 포기할 수밖에 없다. 다른 하나는 해(문제)공간에서 가까운 두 점이 부호(염색체)공간에서도 항상 가깝도록 사상되지 않는다는 점이다. 즉 해공간 상에서 인접한 점임에도 불구하고 부호공간 상에서 해밍거리가 멀게 되는 해밍절벽(Hamming Cliff) 문제가 발생하여 정밀한 해를 탐색하는 것이 어렵게 된다.

이런 문제들을 해결하기 위해 실수코딩을 채용하거나 다른 탐색 알고리즘과 GA를 접목시키는 연구들이 제안되고 있다[4][5]. 실수코딩 염색체와 특수 연산자로 구현되는 실수코딩 유전알고리즘(Real-Coded Genetic Algorithm: RCGA)은 복호화에 소요되는 연산시간을 절약할 수 있고, 흔히 이진코딩에서 직면하는 해밍절벽 문제를 해결할 수 있으며, 불균등 돌연변이(Non-uniform mutation)를 채용하여 이와 같이 초기세대에는 전역적으로 탐색하고 세대가 경과할수록 국소적으로 탐색하도록 함으로서 정밀도 개선이 가능하다[2]. 하지만 아직도 다른 지역탐색 기법에 비해 고정밀도의 해를 얻는 능력이 부족하다.

한편 GA와 SA의 하이브리드화에 대한 연구가 진행되어 왔다[6-8]. 확률적 등반능력을 가지면서 냉각스케줄(Cooling schedule) 때문에 해가 한 점에서 오랫동안 머물지 않는 SA의 특성을 Sirag과 Weisser[9]는 기존 GA의 돌연변이확률을 지수함수 형태로 변경하는데 사용하였고, Jeong과 Lee[10]는 GA의 돌연변이 연산자로 이용하였다. 그러나 이 방법 또한 SA의 근원적인 속성 때문에 일반적으로 탐색속도가 느린 경향이 있다.

따라서 본 논문에서는 탐색성능과 해의 정밀도를 개선하기 위해 RCGA의 전역탐색능력과 단체법(Simplex Search Method)의 지역탐색능력을 결합한 새로운 형태의 하이브리드 탐색법을 제시한다. 단체법은 n 차원 문제에서 $n+1$ 개의 정점과 이들을 잇는 선분으로 구성되는 단체(Simplex)의 크기를 조정해가며 최적점을 탐색해가는 알고리즘으로서 간단하며 구현이 쉽고, 기억장소가 적어도 되며, 조정할 매개변수가 적어 여러 최적화 문제에 이용되고 있다. 그러나 이 알고리즘은 지역해에 도달하면 이로부터 탈출하는 메커니즘이 없어 전역해를 찾지 못하는 단점을 가진다. 하이브리드 탐색법에서 RCGA와 단체법은 점모델에 기반으로 하는 병렬구조를 갖도록 설계된다. 두 알고리즘은 서로 독립적으로 일정한 시간동안(고립시간) 돌아가다가 서로간에 정보(이주자)를 주고받는다. 여기서 단체법은 RCGA로부터 좋은 개체들을 이주받아 해를 더욱 더 개선하도록 미세탐색을 유도하고, RCGA는 단체법으로부터 집단 내에 성능이 좋고 새로운 유전자를 받아드려 탐색공간을 전역탐색 하게된다.

하이브리드 탐색법의 성능은 각 연산자가 가지는 매개변수의 선정과 밀접한 관계를 가지므로, 광범위한 컴퓨터 시뮬레이션을 통해 최적의 성능을 보이는 매

개변수의 영역을 선정한다. 제안한 탐색법을 두 최적화문제에 적용하여 그 유용성을 확인한다.

본 논문의 내용은 다음과 같이 구성된다. 제 2장은 제안될 하이브리드 탐색법의 기본 요소가 되는 RCGA와 단체법의 원리와 GA의 병렬성에 대해 살펴본다. 제 3장에서는 RCGA와 단체법의 장점을 결합한 하이브리드 탐색법을 제안한다. 섬모델을 기반으로 알고리즘의 병렬성을 구현하고 서로간에 주고받는 개체들의 이주전략 등을 결정한다. 제 4장은 제안한 하이브리드 탐색법의 성능을 두 최적화문제에 적용하여 검증해 보고, 제 5장에서 그 결과를 종합한다.

제 2 장 유전알고리즘과 단체법

이 장에서는 다음 장에서 제안할 하이브리드 탐색법의 기본 요소가 되는 유전 알고리즘과 단체법의 기본 이론과 원리에 대해서 알아본다.

2.1 유전알고리즘

유전알고리즘(Genetic Algorithm: GA)은 유전학과 자연선택 원리를 알고리즘 형태로 구현하여 복잡한 실세계 최적화 문제를 해결하려는 최적화 도구로서 염색체 표현의 다양성, 고급 연산자의 개발 등으로 여러 형태의 알고리즘이 제안되고 있다[1]. 전통적으로 GA는 파라미터 자체를 다루기보다는 알고리즘 상에서 모의진화를 일으키는데 편리하도록 자연 파라미터(Natural parameter)의 형태로 변형하여 사용해 왔고, 이진코딩(Binary coding)을 많이 사용하고 있다.

그러나 이진코딩을 채용하면 탐색공간에 대한 사전지식이 부족하여 큰 정의영역을 선택하거나 해의 정밀도를 높이게 되면 염색체 길이가 길어진다. 긴 염색체는 큰 탐색공간을 만들게 되어 계산부담을 증가시키고 경우에 따라서는 탐색을 불가능하게 한다. 또한 해(문제)공간에서 가까운 두 점이 부호공간에서도 항상 가깝도록 사상되지 않는다는 단점을 가진다. 이러한 문제점을 보완할 수 있도록 실수코딩 염색체와 이를 다룰 수 있는 특수 연산자와 전략을 채용하는 실수코딩 유전알고리즘(Real-coded genetic algorithm: RCGA)이 제안되고 있는데 여기서는 RCGA를 중심으로 그 원리를 알아본다.

2.1.1 유전알고리즘의 구조

그림 2.1은 GAs의 일반적인 구조를 보여준다. 다수의 염색체 개체로 구성되는 초기집단이 생성되면, 목적함수를 기반으로 각 개체의 적합도가 평가되고, 일련

의 재생산, 교배, 돌연변이 연산을 통해 집단이 변경되고, 다시 개체의 적합도가 평가되는 연산을 반복함으로써 집단내 개체들을 최적해 쪽으로 진화해가도록 한다.

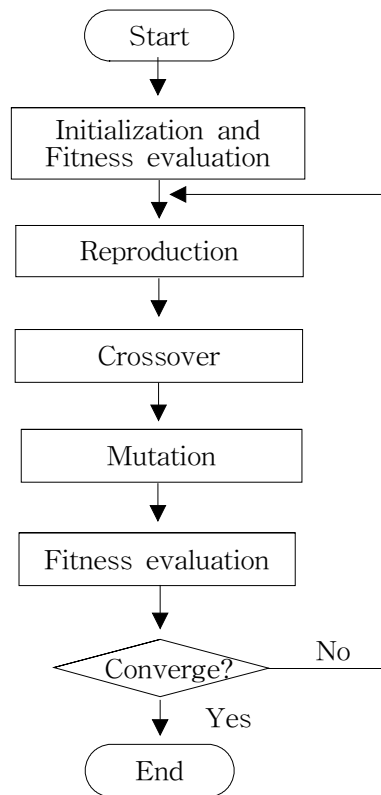


그림 2.1 유전알고리즘의 구조

Fig 2.1 Structure of Genetic Algorithms

가. 염색체의 표현

염색체(Chromosome)의 표현에 대해 알아보기 전에 다음과 같이 최대화 또는 최소화 문제 중 하나로 기술되는 다변수 최적화 문제를 생각하자.

$$\text{최적화 } F(\mathbf{x}) \tag{2.1a}$$

$$\text{탐색영역 } \Omega = \{ \mathbf{x} \mid \mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)} \} \subseteq \mathbb{R}^n \tag{2.1b}$$

여기서 $\mathbf{x}^{(L)}$ 는 하한치 벡터, $\mathbf{x}^{(U)}$ 는 상한치 벡터, n 은 벡터의 차원이다. 전통적으로 이진코딩 유전알고리즘(Binary-coded genetic algorithm: BCGA)의 경우는 \mathbf{x} 를 이진수의 스트링으로 표시하게 되며, 염색체 \mathbf{s} 는 벡터 $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T \in \Omega$ 의 각 원소를 이진수로 표시한 다음 이들을 결합하게 된다. 식 (2.2)는 이진코딩 염색체의 한 예를 보여주고 있다.

$$\begin{array}{cccc} & x_1 & x_2 & \cdots & x_n \\ \mathbf{s} = & (01001 & 01110 & \cdots & 11101) \end{array} \tag{2.2}$$

그러나 앞에서 언급하였듯이 탐색영역을 확대하거나 해의 정밀도를 높이게 되면 BCGA의 염색체 길이가 길어져 계산부담이 증가하고, 경우에 따라서 해밍절벽(Hamming cliff) 문제가 발생되어 비전역 최적해 쪽으로 수렴하는 문제를 일으킨다.

기존의 이진코딩의 문제점을 해결하기 위한 한 방법이 실수코딩 표현법으로서, 실수코딩 염색체는 다음과 같이 표시된다.

$$\mathbf{s} = (x_1 \ x_2 \ \cdots \ x_n) \tag{2.3}$$

실수코딩 염색체의 유전자는 정확히 해 벡터의 원소와 일대일 일치하는 특성을 가진다.

나. 초기집단의 생성

BCGA과 마찬가지로 RCGA도 계속적으로 모의진화를 일으켜 해를 개선해 나

가기 위해 앞에서 정의된 염색체들의 집합을 필요로 한다. N개의 염색체(개체)로 구성되는 k세대의 집단은 다음과 같이 표시된다.

$$\mathbf{P}(k) = \{\mathbf{s}_1(k), \mathbf{s}_2(k), \dots, \mathbf{s}_i(k), \mathbf{s}_N(k)\} \quad (2.4)$$

여기서 $\mathbf{s}_i(k) = (x_{i1}(k) \ x_{i2}(k) \ \dots \ x_{ij}(k) \ \dots \ x_{in}(k))$ 는 i번째 염색체, $x_{ij}(k)$ 는 i번째 염색체의 j번째 원소, N은 집단의 크기이다. N은 세대와는 관계없이 일정한 것으로 간주한다.

본 논문에서는 RCGA가 가혹한 환경에서부터 출발하여 해를 찾는 능력을 확인하기 위하여 초기집단 $\mathbf{P}(0)$ 는 무작위적으로 초기화된다. 따라서 염색체 원소들은 난수발생기를 통해 발생하는 정의영역 $x^{(L)}_j \leq x_{ij}(k) \leq x^{(U)}_j$ ($1 \leq i \leq N, 1 \leq j \leq n$) 내의 실수값으로 설정된다.

다. 적합도 평가

자연계에서 한 생물의 생존능력은 그 환경에 대한 적합도(Fitness)를 반영하듯이 GA에서도 개체의 적합도를 평가하는 과정을 통해 반영된다. GA에서는 기본 유전 연산자를 통해 새로운 집단이 완성될 때마다 개체들의 적합도가 평가되는데 이는 목적함수로부터 계산된다. 환경에 대한 생물의 적합도는 장점을 의미하며 크면 클수록 더 많은 더 많은 보상을 받듯이, GA에서도 적합도가 큰 개체가 더 많은 보상을 받을 수 있도록 배려되어야 한다. 그러기 위해서는 적합도 함수는 최대화 문제 형태로 기술되어야 하고 음의 값을 가져서는 안 된다. 이 때문에 적합도 함수를 얻기 위해서는 목적함수의 적절한 사상(Mapping)이 요구된다.

본 연구에서 사용할 RCGA는 다음 사상을 통해 적합도 값을 계산한다.

$$f(\mathbf{s}) = -F(\mathbf{x}) - v \quad (2.5)$$

여기서 $F: \mathcal{Q} \rightarrow R$ 는 최소화 문제로 가정된 목적함수, $f: S \rightarrow R$ 는 적합도 함수, $\mathbf{s} \in S$ 는 염색체, S 는 염색체 공간, γ 는 항상 $f(\mathbf{s}) \geq 0$ 의 관계를 만족시키는 상수이다. 특히 v 는 성능에 영향을 끼치는 요소로서 부적절하게 선정되면 탐색효율을 저하시킨다. 흔히 탐색 공간 내에서 목적함수의 최소값을 사전에 알 수 없기 때문에 v 는 경험에 의해 충분히 작은 값으로 설정된다. 그러나 세대 진행과 무관하게 항상 고정하게 되면 선택압(Selection pressure)이 떨어지기 때문에 이를 보완 해줄 수 있는 스케일링 윈도우기법(Scaling window scheme)[11]이 필요하다.

라. 유전 연산자

RCGA도 기존의 BCGA과 같이 재생산, 교배, 돌연변이를 기본 유전 연산자(Basic genetic operator)로 채용하게 되는데, 이들은 실수코딩 염색체를 효율적으로 다룰 수 있도록 개선된 것이다.

(1) 재생산 연산자(Reproduction operator)

재생산은 더 적합한 개체가 더 높은 생존기회를 가지는 적자생존(Natural Selection)의 현상을 모방하려는 인위적인 메커니즘이다. 목적함수로부터 각 개체의 적합도가 평가되면 그 크기에 비례하는 선택확률이 부여되고, 이 확률에 따라 선택되어 교배급원(Mating pool)이 형성된다. 일반적으로 적합도가 큰 개체일수록 더 많이 선택될 기회를 갖게 된다.

문헌에는 다양한 재생산을 구현하는 연산자들이 발견되지만 루울렛휠 선택에 기초한 재생산 연산자가 폭넓게 이용되고 있다. 이는 부모세대 개체들의 선택확률에 따라 자손세대를 형성함으로써 몇 가지 단점이 발생된다[12-13]. 그 중 하나는 세대 초기에 초우량 개체가 출현하게 되면 이것이 여러 번 선택되어 유전적 다양성(Genetic diversity)이 한정지워지고 준최적해(Suboptimal solution)로 조기수렴 하게 된다. 이를 보완하기 위해 흔히 집단의 크기는 약 30-200 범위에서 선정되나[14], 다차원 최적화 문제에서는 염색체 길이가 길어져 연산시간이

급격히 증가한다.

루울렛휠 재생산을 채용하는 GA가 작은 집단을 다룰 때 생기는 문제점들을 보완하기 위하여 진 등[15]은 구배와 유사한 재생산 연산자(Gradient-like reproduction operator)를 제안하였고 이 연산자는 다음 6단계를 통해 완성된다.

단계 1: 집단 $P(k-1)$ 내 개체들의 적합도 $f_i(k-1)(1 \leq i \leq N)$ 를 계산하고, 최적의 개체를 찾는다.

$$\begin{aligned} \mathbf{x}_b(k-1) &= \arg \max_{1 \leq i \leq N} [f_i(k-1)] \\ &= [x_{b1}(k-1) \ x_{b2}(k-1) \ \cdots \ x_{bn}(k-1)]^T \end{aligned} \quad (2.6a)$$

$$f_b(k-1) = \max_{1 \leq i \leq N} [f_i(k-1)] (>0) \quad (2.6b)$$

단계 2: $i = 1$

단계 3: $i \leq N$ 의 조건이 만족될 때까지 단계 3-6을 반복한다.

단계 4: 다음 식으로 각 개체에게 새로운 값을 할당한다.

$$x_{ij}(k) = x_{ij}(k-1) + \eta_i \frac{[f_b(k-1) - f_i(k-1)]}{f_b(k-1)} [x_{bj}(k-1) - x_{ij}(k-1)] \quad (1 \leq j \leq n) \quad (2.7)$$

여기서 $x_{ij}(k)$ 는 재생산 후 i 번째 염색체의 j 번째 요소이며, 재생산 계수 η_i 는 평균이 η 이고 분산이 σ^2 인 정규분포 $N(\eta, \sigma^2)$ 를 따르도록 설정되는 난수이다.

단계 5: 재생산된 염색체 $\mathbf{s}_i(k) = (x_{i1}(k) \ x_{i2}(k) \ \cdots \ x_{in}(k))$ 를 교배급원에 복사한다.

단계 6: $i = i+1$

본 논문에서는 이 연산자를 재생산 연산자로 사용한다.

(2) 교배 연산자(Crossover operator)

교배는 자연계 생물들의 유성생식을 알고리즘의 형태로 흉내낸 것으로서 탐색 공간상의 새로운 점을 찾기 위하여 교배는 교배급원으로부터 아버지 염색체 쌍

을 임의로 선택하고, 임의로 선택된 교배점 이후의 유전자들을 서로 교환, 결합하여 자손을 생성한다. 이러한 연산은 생성된 자손의 수가 집단 크기와 같을 때까지 반복된다.

RCGA에서 자주 채용되는 교배 연산자로는 단순교배(Simple crossover)[16], 볼록교배(Convex crossover)[17], 선형교배(Affine crossover)[18] 등이 있다. 본 논문에서는 단순교배와 선형교배를 결합한 연산자를 사용한다. 이 연산자는 교배점의 요소를 일차결합 형태로 변경하고, 교배점 이후의 요소들을 서로 교환하게 된다. 다음 6 단계를 거쳐 완성된다.

단계 1: $i = 1$

단계 2: $i \leq N/2$ 조건이 만족될 때까지 단계 2-6을 반복한다.

단계 3: 교배급원으로부터 아버지 염색체 쌍을 임의로 선택한다.

$$\mathbf{s}_v(k) = (x_{v1}(k) \cdots x_{ci}(k) \ x_{ci+1}(k) \cdots x_{vn}(k)) \quad (2.8a)$$

$$\mathbf{s}_w(k) = (x_{w1}(k) \cdots x_{ci}(k) \ x_{ci+1}(k) \cdots x_{wn}(k)) \quad (2.8b)$$

단, $v, w \in [1, N]$

단계 4: 난수 $r \in [0,1]$ 을 발생시켜 $r \leq P_c$ 이면, 교배점 $c \in [1,n]$ 이 발생되고 교배가 진행된다. 교배점의 두 요소는 다음과 같이 일차결합 형태로 교배되고, 이후의 요소들은 서로 교환된다.

$$\mathbf{s}'_v(k) = (x_{v1}(k) \cdots x'_{vc}(k) \ x_{w,c+1}(k) \cdots x_{wn}(k)) \quad (2.9a)$$

$$\mathbf{s}'_w(k) = (x_{w1}(k) \cdots x'_{vc}(k) \ x_{v,c+1}(k) \cdots x_{vn}(k)) \quad (2.9b)$$

단, $x'_{vc}(k) = \lambda x_{vc}(k) + (1-\lambda)x_{wc}(k) \quad (2.10a)$

$$x'_{vc}(k) = \lambda x_{wc}(k) + (1-\lambda)x_{vc}(k) \quad (2.10b)$$

여기서 λ 는 매개변수로서 0과 1 사이의 난수를 사용한다. 반대로 $r > P_c$ 이면 교배가 일어나지 않고 $\mathbf{s}'_v(k) = \mathbf{s}_v(k)$, $\mathbf{s}'_w(k) = \mathbf{s}_w(k)$ 가 된다.

단계 5: 교배된 염색체를 임시집단에 복사한다.

단계 6: $i = i+1$

교배빈도는 교배확률(P_c)에 따라 달라진다.

(3) 돌연변이 연산자(Mutation operator)

재생산과 교배는 세대가 진행되는 동안 지역탐색(Exploitation)을 통해 해집단을 강하게 해주지만, 속성상 전역탐색(Exploration)하는 능력이 부족하다. 유전알고리즘이 지역해(Local solution)나 사점(Dead corner)에 빠지게 될 경우에는 이로부터 벗어나게 하고, 더 넓은 영역을 탐색할 수 있게 해주는 돌연변이가 필요하다. RCGA는 실수 유전자를 다루기 때문에 돌연변이 연산자는 기존의 BCGA의 연산자와는 다른 형태가 되어야 한다. 본 논문의 RCGA에서는 비균등 돌연변이를 사용한다.

$$x_j = \begin{cases} \tilde{x}_j + \Delta(k, x_j^{(U)} - \tilde{x}_j), & \tau = 0 \text{ 일 때} \\ \tilde{x}_j - \Delta(k, \tilde{x}_j - x_j^{(L)}), & \tau = 1 \text{ 일 때} \end{cases} \quad (2.11a)$$

여기서 v 는 $[0,1]$ 사이의 임의의 난수, T 는 최대의 세대수, b 는 반복횟수에 대한 종속정도를 결정하는 파라미터이다. 함수 $\Delta(k,y)$ 는 세대수 k 가 증가함에 따라 0에 가까워지는 확률이 증가하며, $[0,y]$ 사이의 값을 돌려준다. 한편 돌연변이 발생유무는 전적으로 돌연변이 확률 (P_m)에 따라 결정된다.

마. 엘리트 전략

RCGA에서 채용하는 구배와 유사한 재생산은 최적개체의 생존을 보장해 주지만 이후의 교배와 돌연변이 연산을 거치는 동안 파괴될 수도 있다. 최적 개체의 소멸은 좋은 유전자를 잃게되는 요인이 되므로 다음 세대로 소멸되지 않고 전달되는 것을 보장해 주는 엘리트 전략(Elitist strategy)이 필요하다. 엘리트전략은 이전 세대의 최적 개체를 저장하고 있다가 만약 현 세대의 마지막 단계에서 파괴, 소멸되면 저장된 최적 개체를 현 세대의 가장 약한 개체와 교환해 준다. 그

림 2.2는 이를 보여주고 있다. 이 엘리트 전략은 일반적으로 유전알고리즘의 성능을 개선해 주는 것으로 알려져 있다.

```

Real_CodedGeneticAlgorithmwithElitism()
{
    k= 0;
    Initialize P(k);
    Evaluate P(k);
    While(not termination condition)
    {
        k= k+1;
        Reproduce P(k) from P(k-1);
        Crossover P(k);
        Mutate P(k);
        Evaluate P(k);
        Apply elitism;
    }
}

```

그림 2.2 엘리트 전략을 채택하는 RCGA
 Fig 2.2 RCGA employing the elitist strategy

2.1.2 유전알고리즘의 병렬화

GA를 적용할 때 고려되어야 할 중요한 두 요소는 집단의 다양성 유지와 계산 부담이다. 이 둘 사이에 균형을 유지하기 위해서는 적절한 타협이 필요하다. 작은 집단을 취하면 연산시간은 짧아지지만 해를 탐색하지 못하는 경우가 발생된다. 이를 개선할 목적으로 큰 집단을 취하면 다양성의 확대로 더 좋은 해를 찾을 수 있지만 계산량이 급격히 증가한다. 계산량은 또 입력 변수의 규모에 비례

해서 늘어난다. 이러한 문제는 다수의 프로세서를 사용하여 계산량을 분산시키거나 GA의 구조를 변경하여 병렬화가 원활하도록 함으로써 해결이 가능하다. 병렬성의 필요성은 자연환경에서 종들이 분화집단을 유지하며 생존해가는 것으로부터도 유추할 수 있다. 어떤 GA가 하나의 집단에서 모의진화가 일어나고, 하나의 프로세서가 모든 연산을 담당한다면 이를 직렬 유전알고리즘(Serial genetic algorithm)라 한다. 만약 연산의 일부를 다수의 프로세서가 분담하여 동시에 실행하거나, 아니면 다수의 프로세서가 각자 직렬 유전알고리즘을 독립적으로 실행하고 주기적으로 이들간에 정보를 교환하게 되면 탐색효율 개선이 가능한데, 이를 병렬 유전알고리즘(Parallel genetic algorithm: PGA)이라 한다.

GA의 병렬화는 세 가지 형태로 달성될 수 있는데, 전역모델, 이주모델, 확산모델이 그것이다[19-25]. 전역모델은 전체 집단을 단일 육종 단위로 취급하고 GA 안에 있는 본래의 알고리즘적인 병행을 이용함을 목적으로 한다. 이주모델은 집단을 수 개의 부집단(Subpopulation) 또는 지역집단(Local population)으로 나누고, 각 집단은 기존의 직렬 유전알고리즘 제어 아래 독립적으로 다른 진화 과정을 겪게 되는 육종 단위로 취급된다. 좋은 유전자가 증식되는 것을 돕기 위하여 개체들은 시시각각으로 지역집단 사이를 이주한다. 이러한 형태의 유전알고리즘은 Coarse-grained으로 간주된다. 확산모델은 각 개체를 하나의 분리된 육종 단위로 간주하고 한정된 영역 내에서 선택된 개체들과 교배하게 된다. 확산 유전알고리즘은 보통 Fine-grained으로 간주된다.

가. 전역모델(Global model)

환경안에서 개체들의 적합성이 개별적으로 평가되듯이 GA에서도 개체들의 적합도 평가 작업이 개별적으로 행해지고 있기 때문에 병렬처리 시스템에 적용할 수 있다. 흔히 개체의 적합도 평가 작업이 연산시간의 대부분을 소모하는 경향이 있다. 만약 개체 평가에 병렬연산을 적용하면 연산부담을 상당히 줄일 수 있다. 또한 교배와 돌연변이 역시 선택된 개체들에 대해 병렬처리될 수 있다. 이와

같은 GA가 가지는 본성적인 병렬성이 이용된다. 전역모델에서는 하나의 전역 집단이 유지되고 여기에 전통적인 GA 연산자들이 수행된다. 적합도 할당과 선택이 전역 집단에서 이행되면 선택된 개체들의 교배와 돌연변이, 함수의 평가가 병렬적으로 수행된다.

나. 이주모델(Migration model)

이주모델은 큰 집단을 많은 소규모, 다소 덜 격리된 부집단(Semi-isolated subpopulation) 또는 격리된 부집단(Isolated subpopulation)으로 나눔으로써 지리적인 집단 분포, 즉 자연계에서 섬모델을 흉내내고 있다. 각 부집단은 개체들을 지역적으로 진화시키기 위하여 각자 지역경쟁과 재생산 법칙을 사용하는 분리된 육종 단위가 된다. 이 모델은 각 지역집단에 하나의 프로세서가 할당된다. 각 집단은 서로 다른 집단과 독립적으로 병렬적으로 정해진 세대, 즉 고립시간(Isolation time) 동안 돌아간다. 각 프로세서는 서로 다른 초기 집단을 가지고 시작하기 때문에 유전적 부동은 이 집단들을 다른 방향으로 몰고 가게 된다. 각각은 지역 최소점 방향으로 동작할 수 있다. 문제는 어떻게 하면 이러한 유전적 부동을 줄이면서 안정한 뎀(Deme)이 다른 적합도 피크치 주위에 형성되게 하고, 전체 집단을 지역해로부터 전역해쪽으로 움직이게 하는가 이다.

소집단 내에서 일어날 수 있는 유전적 부동에 의한 유전적 다양성이 감소되는 것을 막아주기 위하여 집단 사이에 유전자 교류(Gene flow)가 충분하게 일어나도록 이주자를 서로 주고받는다. 이 때 일어나는 문제로는 하나의 부집단이 어떤 다른 집단과 교류할 것인지, 무엇을 교류할 것인지, 얼마나 자주 교류할 것인지 등이 있다. 주요 관점들은 요약하면 다음과 같다[26-29]:

- 이주할 집단의 결정
- 이주빈도(Migration frequency) 또는 이주주기의 결정
- 이주율(Migration rate) 또는 이주할 개체들의 수

- 이주할 개체들의 선택
- 교체될 개체들의 선택

2.1.3 성능 지표와 테스트 환경

GA는 확률적인 배경에 기반하고 있기 때문에 탐색환경에 따라 달라질 수밖에 없기 때문에 어떤 환경에서 테스트할 것인지를 결정하는 것은 매우 중요하다. 많은 연구자들로부터 폭 넓게 수용되고 있는 환경은 최소화문제 내에서 구축된 De Jong[30]의 5개의 테스트 함수이다.

가. 성능지수

GA의 성능은 적합한 성능평가 방법을 통해 측정된다. 탐색환경(Search Environment)과 탐색전략(Search strategy)이 정해진 상황에서 GA의 유효성을 정량화하기 위해서 채택되는 성능평가 방법으로는 여러 가지가 있을 수 있겠으나 다음의 방법들이 자주 이용되고 있다.

정의 (최적성능) : 탐색환경 e 에서 사용된 탐색전략 s 의 k 세대 최적성능(Best performance)은 집단 내 최적 개체의 목적함수 값으로 정의된다.

$$\hat{F}_e(s, k) = \underset{1 \leq i \leq N}{\text{best}} \{F_i(k)\} \quad (2.12)$$

여기서 “best”는 최적을 의미하는 연산자로서 최대화 문제에서는 “maximum”, 최소화 문제에서는 “minimum”이 된다.

□

정의 (지금까지의 최적성능) : 탐색환경 e 에서 사용된 탐색전략 s 의 지금까지

의 최적성능은 초기세대부터 k 세대까지 집단 중에서 발견된 최적 개체의 목적 함수 값으로 정의된다.

$$F_e^*(s, k) = \underset{1 \leq j \leq N}{\text{best}} \{ \hat{F}_e(s, j) \} \quad (2.13)$$

앞서 정의된 성능평가 기반으로 초기세대부터 T 세대까지의 평균성능을 구할 수 있는데, 지역 오프라인 성능(off-line performance)은 다음과 같이 정의된다 [30].

□

정의 (지역 오프라인 성능) : 탐색환경 e 에서 탐색전략 s 의 지역 오프라인 성능(local off-line performance)은 초기세대부터 T 세대까지 최적 목적함수 값의 가중치 평균값으로 정의된다.

$$u_e^*(s) = \sum_{k=1}^T w_k F_e^*(s, k) / \sum_{k=1}^T w_k \quad (2.14)$$

□

정의 (전역 오프라인 강인성) : 전체 탐색환경 E 에서 탐색전략 s 의 전역 오프라인 강인성(global off-line robustness)은 지역 오프라인 성능의 가중치 평균값으로 정의된다.

$$U_E^*(s) = \sum_{e \in E} w_e u_e^*(s) / \sum_{e \in E} w_e \quad (2.15)$$

□

지역 오프라인 성능은 오프라인 환경에서 유전알고리즘의 적응 능력을 측정할 때 사용된다.

주어진 테스트 환경에서 제안한 탐색법의 성능을 측정하기 위해서는 적절한 평가가 요구되는데, 이 평가를 위하여 De Jong이 제안한 오프라인 성능 측정법을 이용한다.

나. 테스트 함수

De Jong의 테스트 함수는 연속/불연속, 볼록/비볼록, 저차/고차, 이차형식/비이차형식, 단봉/다봉, 결정적/확률적 탐색특성이 내포되도록 선택되었기 때문에 아직도 파라미터 선정 시 많이 이용되고 있다. 여기서는 De Jong의 테스트 함수를 보완한 5개의 테스트 함수를 사용한다.

이 함수는 연속, 볼록, 10차, $10 < x_i \leq 10$ 의 특성을 가지며, $\mathbf{x}=[0 \ 0 \ 0]^T$ 에서 최솟값을 가짐을 보인다. 그림은 이 함수의 2차원 반전 그래프다.

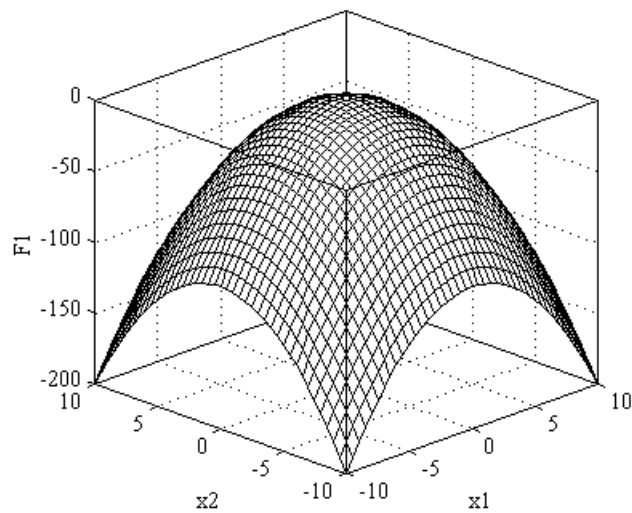


그림 2.3 테스트 함수 F_1 (2차원 반전)

Fig 2.3 Test Function F_1 (Two-dimensional and inverted)

$$F_2(\mathbf{x}) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2, \quad -10 \leq x_i \leq 10 \quad (2.17)$$

이 함수는 연속, 비볼록, 단봉, 저차원의 특성을 가지며, $\mathbf{x}=[1 \ 1]^T$ 에서 최소값 0을 가진다. $x_2 = x_1^2$ 을 따라 대단히 깊은 골짜기가 형성되므로 해를 얻기가 대단히 까다로운 문제중의 하나이다. 그림은 이 함수의 반전된 모습을 보여주고 있다.

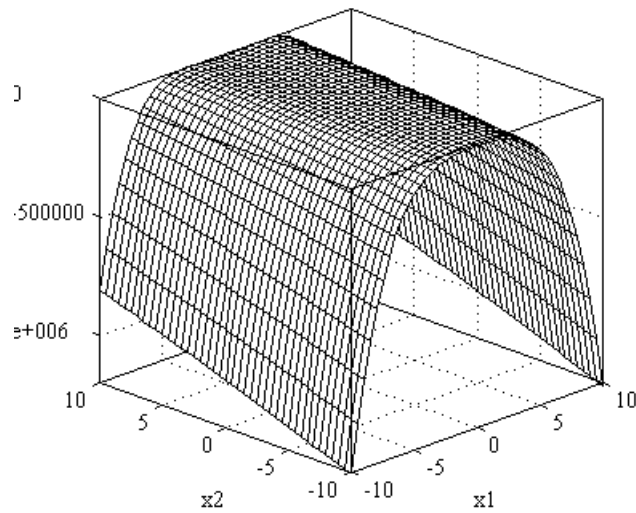


그림 2.4 테스트 함수 F_2 (반전)

Fig 2.4 Test Function F_2 (Inverted)

$$F_3(\mathbf{x}) = \sum_{i=1}^5 [x_i], \quad -5.12 \leq x_i \leq 5.12 \quad (2.18)$$

이 함수는 제안한 유전알고리즘이 불연속구간을 탐색할 수 있는지를 테스트하기 위해 선택된 함수로 불연속, 비볼록, 단봉의 탐색특성을 가지며, $\mathbf{x} = [-5.12 \ -5.12 \ -5.12 \ -5.12]^T$ 에서 최소값 -30 을 가진다. 연산자 $[x]$ 는 x 를 초과하지 않는 최대치 정수를 의미한다. 그림은 이 함수의 2차원 형태를 그린 것이다.

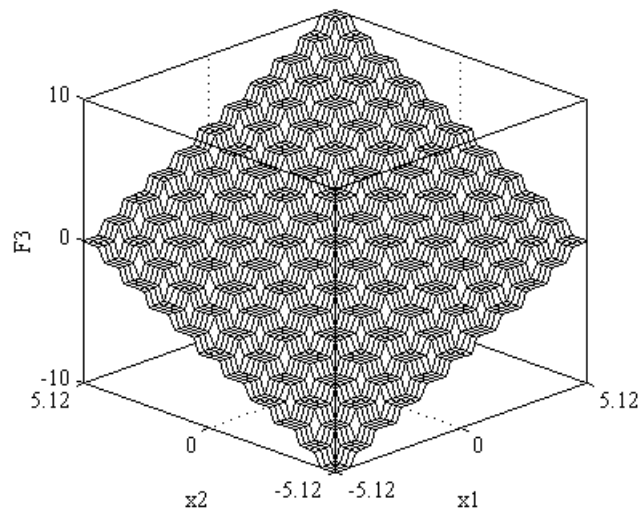


그림 2.5 테스트 함수 F_3 (2차원 반전)

Fig 2.5 Test Function F_3 (Two-dimensional and inverted)

$$F_4(\mathbf{x}) = \frac{1}{0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}}, \quad -70 \leq x_i \leq 70 \quad (2.19)$$

이 함수는 연속, 비볼록, 다봉의 탐색특성을 보인다. 25개의 지역해를 가지고, $\mathbf{x}=[32 \ 32]^T$ 에서 전역 최소값 1을 가진다. 그림은 이 함수의 반전된 모습을 보이고 있다.

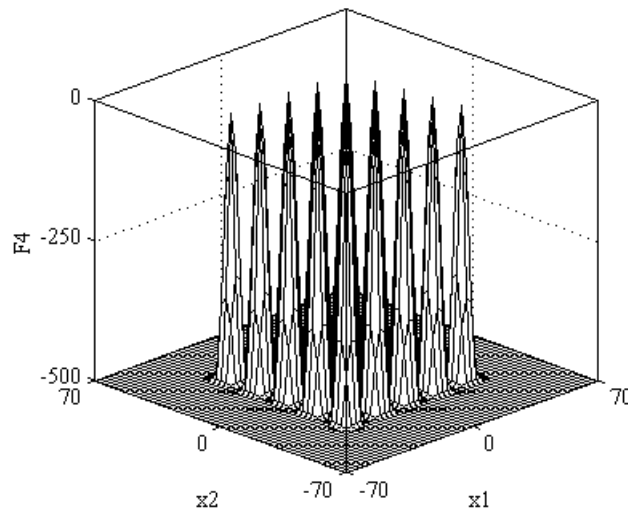


그림 2.6 테스트 함수 F_4 (반전)

Fig 2.6 Test Function F_4 (Inverted)

$$F_5(\mathbf{x}) = 0.5 + \frac{(\sin^2 \sqrt{(x_1^2 + x_2^2)} - 0.5)}{(1.0 + 0.001 \times (x_1^2 + x_2^2))^2}, \quad -10 \leq x_i \leq 10 \quad (2.20)$$

이 함수는 연속, 볼록, 다봉의 탐색특성을 가지고 있고, 특성상 지역해에 빠질 가능성이 높다. $\mathbf{x}=[0 \ 0]^T$ 에서 전역해 0을 갖는다.

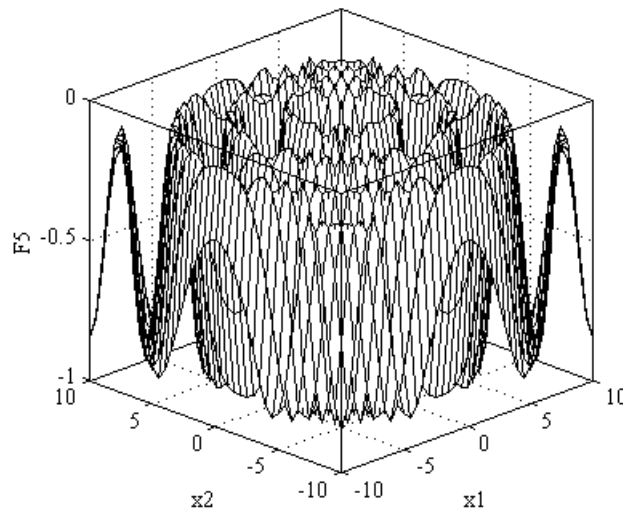


그림 2.7 테스트 함수 F_5

Fig 2.7 Test Function F_5

2.2 단체법

다차원 공간의 탐색법 중 하나인 단체법(Simplex search method)은 현재의 데이터 집합을 이용하여 정해진 탐색의 방향으로 전개해가면서 지역 탐색하는 기법으로서 1962년 Spendly 등[31]에 의해서 처음으로 소개되었고, Nelder와

Mead[32]에 의해 더욱 보완되었다. 단체법은 알고리즘이 간단하여 구현이 쉽고, 해를 찾는데 구배 정보를 필요치 않고 함수 값만을 이용하기 때문에 많이 이용된다.

2.2.1 탐색원리

단체법은 탐색공간의 차원 n 보다 하나 더 많은 $n+1$ 의 정점(Vertex)의 집합 $\{x_1, x_2, \dots, x_{n+1}\}$ 을 적절히 전개함으로써 최적해를 탐색한다. 이 때 $n+1$ 개의 정점과 이들을 연결하는 선분으로 형성된 기하학적 형태를 단체(Simplex)라 한다. 따라서 1차원의 경우에는 단체는 선이 되고, 2차원의 경우는 삼각형, 3차원의 경우는 사면체(Tetrahedron)가 된다.

단체법의 탐색은 초기 정점으로부터 시작하고, 대응되는 함수값의 정보를 이용한다. 매 반복 시 단체는 성능이 가장 나쁜 점(최소화 문제의 경우 함수값이 가장 큰 정점)을 교체할 새로운 점을 생성한다. 성능이 가장 나쁜 점을 x_w 라 하면 x_w 는 나머지 점들의 중심점 x_M 건너편 쪽으로 반전에 의해 새로이 생성된다.

$$x_R = x_M + (x_M - x_w)$$

$$\text{단, } x_M = \frac{1}{n} \sum_{\substack{i=1 \\ i \neq w}}^{n+1} x_i \quad (2.21)$$

따라서 새로운 단체는 $\{x_1, x_2, \dots, x_{w-1}, x_{w+1}, \dots, x_{n+1}, x_R\}$ 로 재설정 된다. 이러한 자리바꿈은 최적점으로 수렴할 때까지 계속적으로 반복된다. 새로운 점은 현존하는 점들의 선형결합으로 생성되고 이 때 반사, 확장 및 축소 세 가지 연산이 사용된다. 그림 2.5는 2차원 단체의 움직임 예를 보여주고 있다.

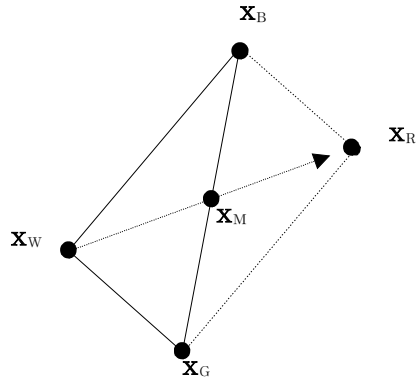


그림2.8 이차원 단체의 예
 Fig 2.8 Example of 2-dimensional simplex

그림에서 첨자 B(Best)는 최고의 성능, G(Good)는 좋은 성능, W(Worst)는 최악의 성능을 가지는 점을 의미하며, \mathbf{x}_M 는 \mathbf{x}_B 와 \mathbf{x}_G 의 중심점이다. \mathbf{x}_W 는 \mathbf{x}_M 을 가로질러 새로운 단체의 \mathbf{x}_R (Reflect)를 생성한다. 결국 단체는 최소값 쪽으로 공간의 아래쪽으로 이동하게 된다.

2.2.2 Nelder-Mead의 단체법

Nelder-Mead의 단체법은 Spendly[31]의 알고리즘을 수정한 것으로서 그림 2.9는 Nelder-Mead의 단체법의 흐름도를 보여주고 있다.

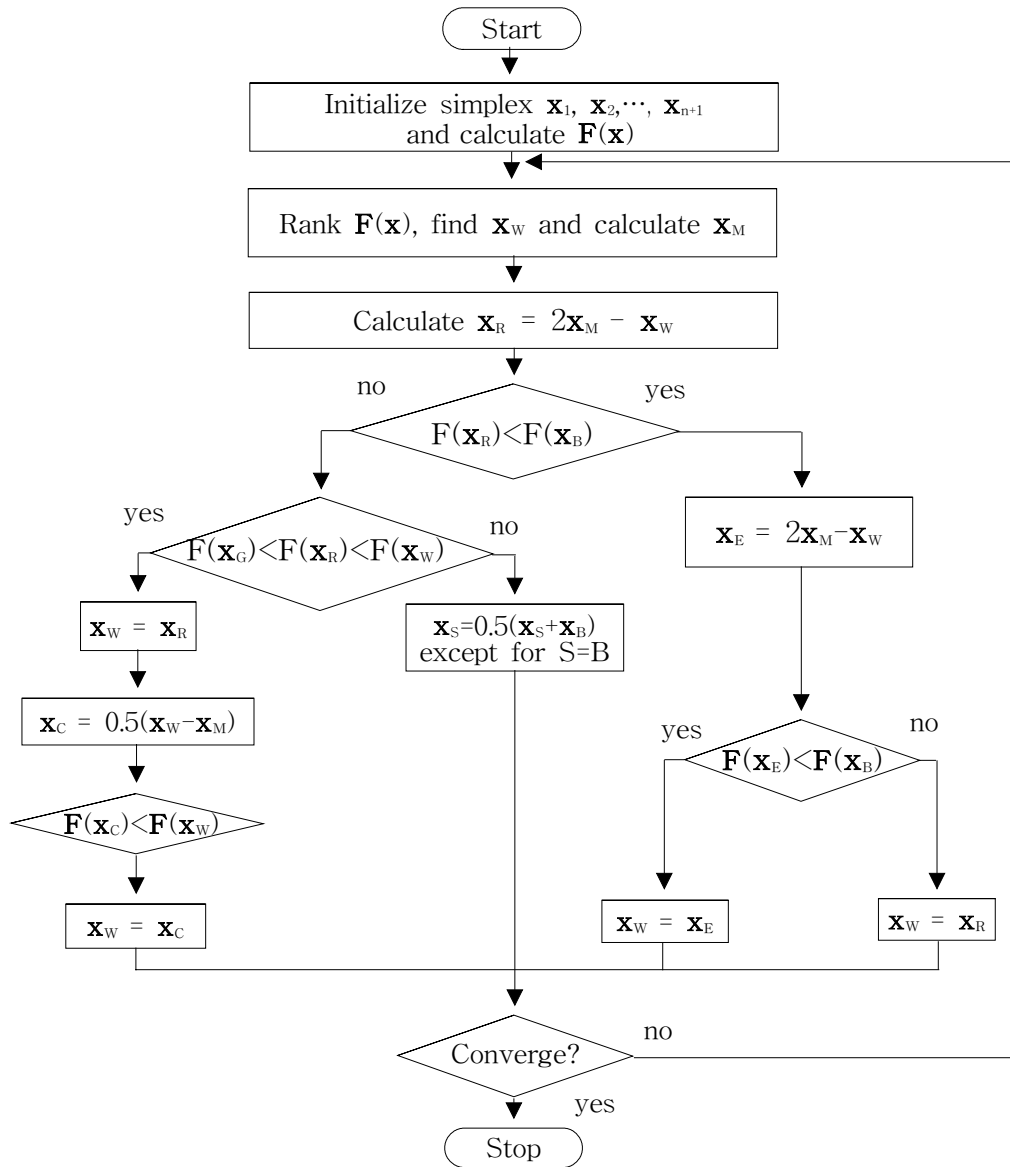


그림 2.9 Nelder-Mead 단체법의 흐름도

Fig 2.9 Flow chart of the Nelder-Mead simplex method

Nelder-Mead의 단체법은 다음의 세 가지 이유로 최적화 문제에 자주 이용된다 [33].

- 원리가 간단하고 프로그래밍 하기가 쉽다.
- 알고리즘이 수행된 후 단지 몇 번의 반복과정으로도 두드러지게 개선된 근사해(또는 최적점에 가까워진 점)에 접근한다.
- 1회 반복과정에 적어도 차원 n 만큼의 목적함수를 계산하는 방법들은 부단히 시간 소모적이다. 만일 이러한 방법들이 성공적이라면 이들 중에 Nelder-Mead 단체법이 함수평가를 가장 적게 한다.

정점들에서의 함수값의 크기를 비교하여 단체의 크기를 조정해 가며 최적점을 탐색하는 Nelder-Mead 단체법은 다음과 같은 원리로 동작한다. 여기서는 도식적으로 표현하는 것이 가능하도록 2-차원 함수 $F(\mathbf{x})$ 의 최소화 문제를 생각해 보자.

가. 초기화

일반적으로 n -차원의 초기 단체를 형성하기 위해서는 $n+1$ 개의 점이 선택되어야 하지만 2차원의 경우는 삼각형을 이루는 세 벡터 $\mathbf{x}_i(i=1,2,3)$ 가 주어지면 된다. 이 점에서 함수값 $F(\mathbf{x}_i)$ 이 구해지면 그의 크기에 따라 정렬된다. 여기서는 이해가 쉽도록 다음과 같이 가장 좋은 점은 첨자 B, 두 번째로 좋은 점은 첨자 G, 가장 나쁜 점은 첨자 W로 표시하기로 한다.

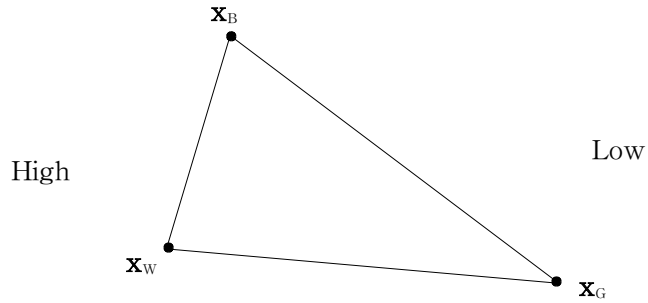


그림 2.10 초기 단계의 단체
Fig 2.10 Simplex at the beginning step

따라서

$$F(\mathbf{x}_B) \leq F(\mathbf{x}_G) \leq F(\mathbf{x}_W) \quad (2.22)$$

가 성립한다.

나. 좋은 변의 중점

삼각형을 구성하는 과정에서 B와 G를 잇는 선분의 중점을 사용한다. 중점은 다음과 같이 두 벡터를 평균하여 구한다.

$$\mathbf{x}_M = \frac{\mathbf{x}_B + \mathbf{x}_G}{2} \quad (2.23)$$

다. 수렴판정 조건

Reflection 반복과정은 n+1개의 점에서의 함수값들의 표준편차가 정해진 값보다 작게 되었을 때 중지한다.

$$SSE = \sqrt{\frac{1}{n+1} \sum_i (F(x_i) - \bar{F})^2} \quad (2.24)$$

$$\text{단, } \bar{F} = \frac{\sum_{j=1}^{n+1} F(x_j)}{(n+1)}$$

라. 반전

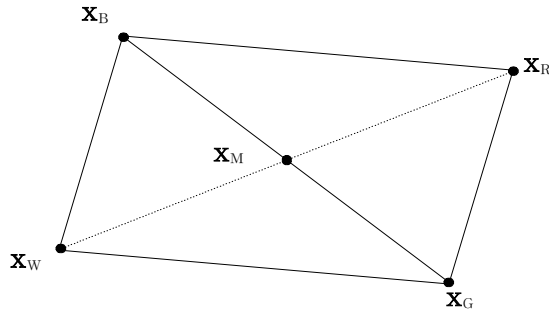


그림 2.11 반전
Fig 2.11 Reflection

반복과정이 시작되면 각 극점에서의 함수값들 중에서 가장 높은 점, 두 번째 높은 점, 그리고 가장 낮은 점을 확인하여 각각 \mathbf{x}_W , \mathbf{x}_G , \mathbf{x}_B 라 놓고 이점들에 대응하는 함수값들을 $F(\mathbf{x}_W)$, $F(\mathbf{x}_G)$, $F(\mathbf{x}_B)$ 로 결정한다. 가장 높은 점 \mathbf{x}_W 를 제외한 나머지 극점들의 기하학적 중심점(Centroid) \mathbf{x}_M 를 계산하여 \mathbf{x}_W 를 \mathbf{x}_M 방향으로 새로운 극점 \mathbf{x}_R 를 생성시킨다. 이러한 과정의 반전은 다음의 식에 의하여 수행된다.

$$\mathbf{x}_R = (1+\alpha)\mathbf{x}_M - \mathbf{x}_W \quad (2.25)$$

여기서 α 는 반전계수라 한다.

마. 반전 · 확대 · 축소 · 수축 과정

- 반전수용

반전된 \mathbf{x}_R 에서의 함수값을 $F(\mathbf{x}_R)$ 라 했을 때 만일 $F(\mathbf{x}_B) \leq F(\mathbf{x}_R) \leq F(\mathbf{x}_G)$ 이라면 \mathbf{x}_W 를 \mathbf{x}_R 로 대체한 후 새로운 반복과정을 시작한다.

- 확대시도

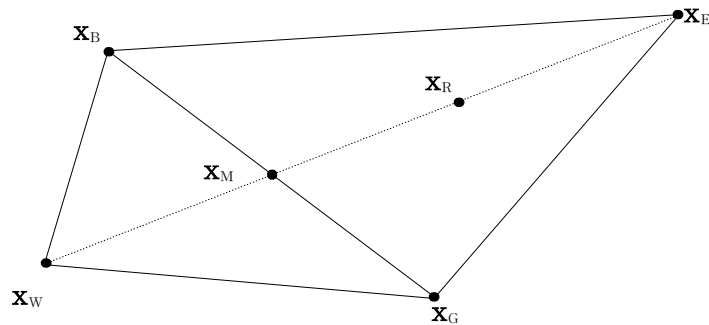


그림 2.12 반전과 확장

Fig 2.12 Reflection and Expansion

만일 $F(\mathbf{x}_R) < F(\mathbf{x}_B)$ 이라면 같은 방향으로 확장하여도 좋은 결과값이 예상되므로 다음의 식에 의하여 확장된 점을 구한다.

$$\mathbf{x}_E = \nu \mathbf{x}_R + (1-\nu) \mathbf{x}_M \tag{2.26}$$

확장된 극점 \mathbf{x}_E 에서의 함수값을 $F(\mathbf{x}_E)$ 라 했을 때, 만일 $F(\mathbf{x}_E) < F(\mathbf{x}_B)$ 이라면 확장된 점을 수용한다. 즉 \mathbf{x}_W 를 \mathbf{x}_E 로 대체한 후 다시 새로운 반복과정을 시작한다. 그러나 $F(\mathbf{x}_R) \geq F(\mathbf{x}_B)$ 라면 확장을 취소하고 이전의 반전을 수용한다.

- 축소 시도

만일 반전된 점의 함수값이 $F(\mathbf{x}_w)$ 보다 크다면, 즉 $F(\mathbf{x}_R) > F(\mathbf{x}_w)$ 이라면, 축소를 시도한다. 축소는 다음의 식에 의하여 수행이 되고 β 를 축소계수라 한다.

$$\mathbf{x}_c = \beta\mathbf{x}_w + (1-\beta)\mathbf{x}_M \quad (2.27)$$

만일 $F(\mathbf{x}_c) \leq F(\mathbf{x}_w)$ 이라면 축소를 수용하고 계속해서 반복과정을 수행한다.

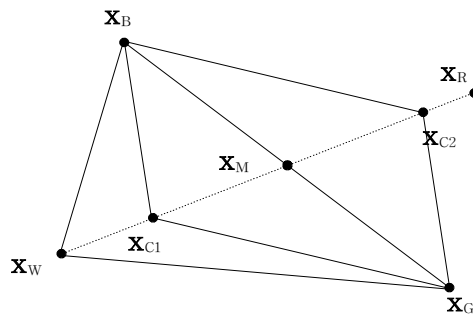


그림 2.13 축소
Fig 2.13 Contraction

- 수축(shrink)

축소된 새로운 극점의 함수값 $F(\mathbf{x}_c)$ 가 $F(\mathbf{x}_w)$ 보다 크다면, 즉 $F(\mathbf{x}_c) > F(\mathbf{x}_w)$, 축소 시도는 거부되고 단체는 \mathbf{x}_s 방향으로 수축을 한다. \mathbf{x}_s 을 제외한 나머지 극점들은 수축계수 δ 에 의하여 수축정도가 결정된다.

$$\mathbf{x}_s = \delta\mathbf{x}_s + (1-\delta)\mathbf{x}_B \quad (2.28)$$

수축된 극점들의 함수값을 계산한 후 다시 반복과정을 수행한다.
 \mathbf{x}_C 에서의 함수값이 \mathbf{x}_W 에서의 함수값보다 작지 않으면, 점 \mathbf{x}_G 와 \mathbf{x}_W 를 \mathbf{x}_B 쪽 방향으로 수축해야 한다(그림 2.14 참조). 그리고 나서, 점 \mathbf{x}_G 를 \mathbf{x}_M 으로 바꾸고, 점 \mathbf{x}_W 를 \mathbf{x}_S 로 바꾼다. 여기서 \mathbf{x}_S 는 \mathbf{x}_B 와 \mathbf{x}_W 를 잇는 선분의 중점이다.

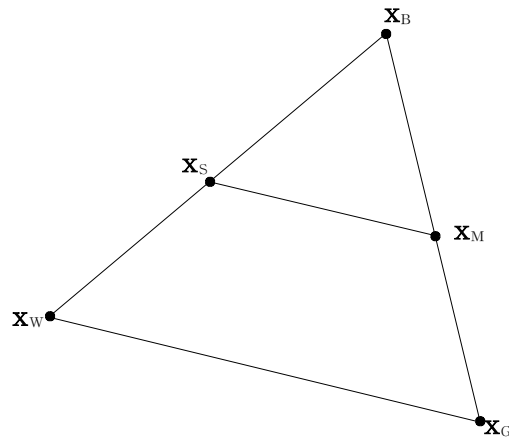


그림 2.14 \mathbf{x}_B 쪽 방향으로의 삼각형 수축
 Fig 2.14 Triangle shrink to direction \mathbf{x}_B

어떤 알고리즘이 계산량의 측면에서 효율적이기 위해서는 함수값을 필요로 할 때만 계산하도록 해야한다. Nelder-Mead 방법에서는 각 단계마다 새로운 꼭지점을 구하여 W를 대체한다. 따라서 새로운 꼭지점을 찾게 되면, 더 이상 조사할 필요가 없으므로, 반복 단계를 끝낸다. 2차원 문제에 대한 논리적 결정과정을 그림 2.15에 자세하게 나타내었다.

```

Nelder_MeadSimplexSearchMethod()
{
  Initialize parameters;
  While(not termination condition)
  {
    Compute  $\mathbf{x}_R = (1+\alpha)\mathbf{x}_M - \mathbf{x}_W$  and  $F(\mathbf{x}_R)$ ;
    if  $F(\mathbf{x}_R) < F(\mathbf{x}_G)$ 
      if  $F(\mathbf{x}_B) < F(\mathbf{x}_R)$  then
        Replace  $\mathbf{x}_W$  with  $\mathbf{x}_R$ ;
      else
        Compute  $\mathbf{x}_E = \nu\mathbf{x}_R + (1-\nu)\mathbf{x}_M$  and  $F(\mathbf{x}_E)$ ;
        if  $F(\mathbf{x}_E) < F(\mathbf{x}_B)$  then
          Replace  $\mathbf{x}_W$  with  $\mathbf{x}_E$ ;
        else
          Replace  $\mathbf{x}_W$  with  $\mathbf{x}_R$ ;
        endif
      endif
    else
      if  $F(\mathbf{x}_R) < F(\mathbf{x}_W)$  then
        Replace  $\mathbf{x}_W$  with  $\mathbf{x}_R$ ;
      else
        Compute  $\mathbf{x}_C = (\mathbf{x}_W + \mathbf{x}_M)/2$  and  $F(\mathbf{x}_C)$ ;
        if  $F(\mathbf{x}_C) < F(\mathbf{x}_W)$  then
          Replace  $\mathbf{x}_W$  with  $\mathbf{x}_C$ ;
        else
          Compute  $\mathbf{x}_S = \delta\mathbf{x}_S + (1-\delta)\mathbf{x}_B$  and  $F(\mathbf{x}_S)$ ;
          Replace  $\mathbf{x}_W$  with  $\mathbf{x}_S$ ;
          Replace  $\mathbf{x}_G$  with  $\mathbf{x}_M$ ;
        endif
      endif
    endif
  }
}

```

그림 2.15 Nelder-Mead 탐색법의 논리적 결정과정

Fig 2.15 Logical process of the Nelder-Mead simplex method

제 3 장 제안한 하이브리드 탐색법

GA는 병렬처리가 가능하고 전역 탐색능력이 뛰어난 장점을 가지고 있는 반면 세대가 거듭됨에 따라 개체군 내에 존재하는 모든 개체들이 서로 닮아가면, 유전적 다양성이 결핍되어 더 이상 진화하지 못하는 경우가 발생할 수 있다. 다시 말하면, 개체들의 유전형이 모두 비슷해지거나 같아져서 유전자 변이를 일으키지 못하고, 이로 인해 탐색능력을 상실하여 최적해에 도달하지 못하게 된다.

이러한 문제의 해결을 위해 개체군의 크기와 돌연변이 확률을 높여서 다양성을 유지시키는 방법을 사용하였다. 그러나 이 방법들은 일정수준의 성능향상을 가져왔지만, 수렴시간과 계산시간 등을 고려한 전체적인 성능은 기존의 방법과 비교하여 크게 향상된 바가 없기 때문에 실질적인 문제에는 적합하지 않다.

한편, 지역탐색법 중의 하나인 단체법은 전역탐색의 역할이 부족하고 탐색도중 지역해에 빠지는 문제점이 발생된다. 본 연구에서는 RCGA의 장점인 전역탐색 기능과 단체법의 장점인 지역탐색 기능을 조합함으로써 해의 정밀도를 높이고, 수렴속도를 개선하는 병렬 구조를 갖는 하이브리드 탐색법을 제안한다. 제안한 알고리즘의 탐색성능은 매개변수들의 선정에 영향을 받기 때문에 적절한 성능평가 환경과 방법으로 최적의 매개변수를 도출한다.

3.1 하이브리드 탐색법의 구조

앞장에서 GA의 병렬성에 대해 살펴보았다. 일반적으로 PGA는 다봉의 탐색공간을 가지거나 다수의 목적함수를 최적화하는 문제에는 직렬 유전알고리즘 보다 더 효과적이며, 목적함수를 평가하는 수가 적음에도 불구하고 직렬 유전알고리즘에 비해 더 빨리 해를 찾을 수 있으며, 비록 단일 CPU 컴퓨터 상에서 PGA를 직렬로 구현한 것도 직렬 유전알고리즘 보다 성능이 더 좋은 것으로 알려져 있다[34-36]. 기존의 직렬 유전알고리즘과 비교하여 PGA의 특징은 다음과 같다:

- PGA가 자연진화의 생물학적 모방에 더 가깝다.
- 일반적으로 직렬 유전알고리즘에 비해 최적해를 찾는 데 함수의 평가를 덜 요구한다.
- 유전적 부동을 극복하기 위하여 다중 최적해를 찾는 잠재력을 가지고 있다.
- 강인성이 더 높다.
- PGA의 구현은 최적해를 찾는 데 그의 선형적으로 처리속도를 높이기 위하여 다른 병렬 구조를 효율적으로 이용하기가 적합하다.

따라서 본 연구에서는 RCGA의 전역탐색과 단체법의 지역탐색 장점을 결합하여 탐색성과 정밀도를 개선하도록 알고리즘의 병렬성을 이용한다. 본 논문에서 제안된 하이브리드 탐색법은 섬 모델(Island model)의 구조[29]를 취한다. 일반적으로 섬 모델은 전체 개체군을 여러 개의 부개체군으로 분할하여 각 부개체군들을 독립적으로 진화시키며, 정해진 세대가 지나면 각 부개체군에서 임의의 개체들을 선택하여 다른 부개체군으로 이주시키는 병렬구조를 가지고 있다. 각 부개체군들이 서로 다른 유전자형으로 수렴해감으로써 탐색 공간을 분할하여 탐색하는 효과를 얻을 수 있으며, 또한 단체법의 지역탐색 능력을 이용할 수 있다는 장점을 가지고 있다.

따라서 두 알고리즘은 서로 다른 파라미터 집단 또는 정점들을 사용하여 독립적으로 정해진 세대, 즉 고립시간(Isolation time) 동안 돌아간다. 집단 사이에 유전자 교류(gene flow)가 충분하게 일어나도록 이주자를 서로 주고받는다.

3.1.1 이주모델에 기초한 병렬성 구현

제안한 하이브리드 탐색법의 구조는 이주모델(섬모델)을 기반으로 하기 때문에 RCGA와 단체법이 병렬로 결합된 것으로 생각할 수 있으며, 두 알고리즘이 각각 고립시간 동안 독립적으로 진행해 가다가 서로간에 이주자를 교환하며, 이

러한 과정은 주기적으로 일어난다. 그러나 본 연구에서는 독립된 프로세서를 사용하여 완벽한 병렬성을 구현하는 것이 어려워 두 알고리즘을 단일 CPU 컴퓨터 상에서 직렬로 구현하는 의사 병렬 유전알고리즘(Pseudo parallel genetic algorithm)을 제안하였다. 여기서 RCGA는 2장에서 언급한 재생산, 교배, 돌연변이 연산자를 사용하고, 엘리트전략과 스케일링윈도 전략을 사용하였다. 단체법으로는 Nelder-Mead의 것을 사용하였다. 그림 3.1은 하이브리드 탐색법의 구조를 보여주고 있다.

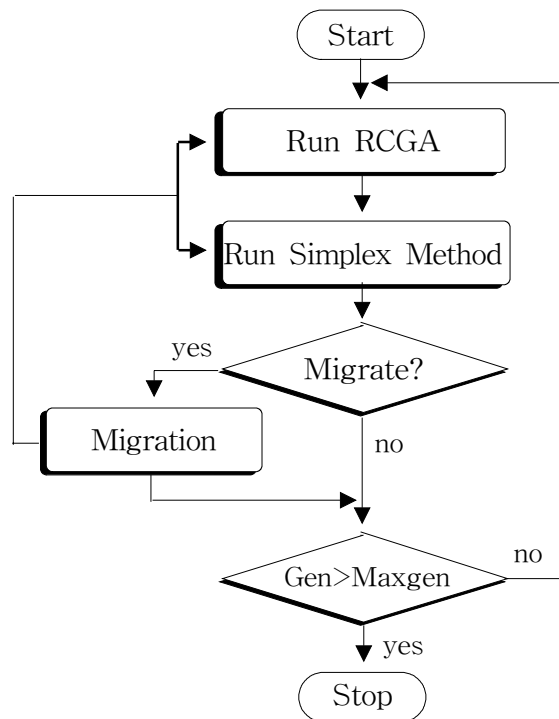


그림 3.1 하이브리드 탐색법의 구조

Fig 3.1 Structure of the hybrid search method

3.1.2 탐색법의 초기화

제안한 하이브리드 탐색법에서 RCGA가 계속적으로 모의진화를 일으켜 해를 개선해 나가기 위해서는 앞서 정의된 염색체 개체들의 집합을 필요로 하며, 집단은 N 개의 염색체로 구성된다. 본 논문에서는 RCGA가 가혹한 환경에서부터 출발하여 해를 찾는 능력을 확인하기 위하여 초기집단 $\mathbf{P}(0)$ 는 무작위법으로 초기화된다. 따라서 염색체 유전인자(요소)들은 정의영역 $x^{(L)}_{ij} \leq x_{ij}(k) \leq x^{(U)}_{ij}$ ($1 \leq i \leq N, 1 \leq j \leq n$) 내에서 발생하는 실수 난수값으로 설정된다. RCGA와 독립적으로 동작되는 단체법 또한 $n+1$ 개의 초기 정점을 필요로 하기 때문에 이들도 정의영역 내에서 무작위적으로 초기화된다.

3.1.3 이주자의 선택 및 이주

본 연구에서 제안한 하이브리드 탐색법의 구조는 두 알고리즘이 일정한 기간 동안 각자 독립적으로 해를 찾다가, 매 이주율(M_r)마다 선택된 이주자(Migrant)를 서로 이주시키는 형식을 채택하였다. 이주는 3 단계로 이루어진다. 첫번째 단계에서는 하나의 알고리즘에서 이주시킬 개체를 선택하고, 두번째 단계에서의 이들과 교체할 개체를 이주시킬 알고리즘에서 선택하고, 마지막으로 교체하게 된다.

식 (2.1)의 최소화 문제를 다시 생각하면, 이주 개체의 선택과 이주는 다음과 같은 방법으로 진행된다.

가. 단체법에서 RCGA로 이주

단계 1: 단체법에서 성능이 가장 좋은 개체(이주자) 하나를 선택한다.

$$\mathbf{x}_b(k) = \arg \min_{1 \leq i \leq n+1} [F_i(k)]$$

여기서 $\mathbf{x}_b(k)$ 는 k 세대에서의 성능이 가장 좋은 정점으로서 첨자 $b \in [1, n+1]$ 는 best를 의미하며, $F_i(k) = F(\mathbf{x}_i(k))$ 는 i 번째 개체의 목적함수 값이다. 최대화 문제의 경우에는 min은 max로 교체된다.

단계 2: RCGA 집단 중에 성능이 가장 나쁜 개체를 선택한다.

$$\mathbf{s}_w(k) = \arg \min_{1 \leq i \leq N} [f_i(k)]$$

여기서 $\mathbf{s}_w(k)$ 는 k 세대에서의 성능이 가장 나쁜 개체로서 첨자 $w \in [1, N]$ 는 worst를 의미하며, $f_i(k) = f(\mathbf{s}_i(k))$ 는 i 번째 개체의 적합도 값이다.

단계 3: 이주자를 RCGA의 집단 중에 성능이 가장 나쁜 개체와 교체한다.

$$\mathbf{s}_w(k) = \mathbf{x}_b^T(k)$$

여기서 T 는 벡터의 전치를 의미한다.

나. RCGA에서 단체법으로 이주

단계 1 : RCGA의 집단 중에 성능이 좋은 순서로 M_n 개의 이주자를 선택한다.

$$j = \text{rank}(\mathbf{s}_i(k)), i, j \in [1, N]$$

$$f(\mathbf{s}_j(k)) \geq f(\mathbf{s}_{j+1}(k)), j \in [1, N-1]$$

여기서 j 는 개체의 적합도 크기에 따라 정해지는 순위(Rank)이고, 이주자들은 $j \geq M_n$ 인 순위를 가진다.

단계 2 : 단체법중에서 성능이 나쁜 순서로 M_n 개의 개체를 선택한다.

$$j = \text{rank}(\mathbf{x}_i(k)), i, j \in [1, n+1]$$
$$F(\mathbf{x}_j(k)) \leq F(\mathbf{x}_{j+1}(k)), j \in [1, n]$$

단계 3 : M_n 개의 이주자들을 단체법에서 성능이 나쁜 개체와 교체한다. 만약 $M_n \geq n+1$ 이면 단체법의 개체들은 전부 교체되게 된다.

3.2 하이브리드 탐색법의 파라미터

3.2.1 RCGA와 단체법의 파라미터

본 논문에서 사용한 RCGA의 제어 파라미터로는 집단크기, 재생산 계수, 교배 확률, 돌연변이 확률이 있고 각각 $N=20$, $\eta=1.7$, $P_c=0.8$, $P_m=0.3$ 으로 선정하였다. 스케일링 윈도는 1을 채용하였다. 한편 단체법의 계수는 Nelder-Mead[32]의 결과에 따라 반전 계수, 확장계수, 축소계수, 수축계수를 각각 $\alpha=1.0$, $\nu=2.0$, $\beta=0.5$, $\delta=0.5$ 로 하였다.

3.2.2 이주율의 결정

PPGA 구현 시 각 집단의 크기, 이주율, 고립시간, 개체 이주방법, 이주할 개체와 교체될 개체들의 선택 등 몇 가지 사항들이 고려되어야 한다. 일반적으로 이들은 경험과 실험을 기초로 결정된다. 이주율이 너무 높으면 부집단들의 유전적 다양성이 줄어들고, 반대로 너무 낮으면 지역해로 조기 수렴해가는 문제점이 발생한다. 본 연구에서는 2차원 이상의 최적화 문제만을 고려하였기 때문에 이주 개체수를 $M_n = 3$ 으로 고정하였다.

최적 파라미터의 선정은 주어진 테스트 환경에서의 성능 평가 방법을 이용하여 전역 오프라인 강인성을 계산하는 일련의 작업을 요구한다. 가능한 모든 파라미터 영역에서 찾는 것은 현실적으로 어렵기 때문에 이주율을 $M_r \in \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$ 로 10단계로 하여 탐색하였다. 계산되는 성능은 초기집단의 선택에 민감하므로 서로 다른 씨드로 알고리즘을 5번 실행하고 그 결과를 평균하여 그 영향을 최소화하였다. 그림은 M_r 을 축으로 하여 $\bar{U}_E^*(s)$ 를 그린 것으로, $M_r = 4$ 때의 성능이 좋음을 알 수 있다.

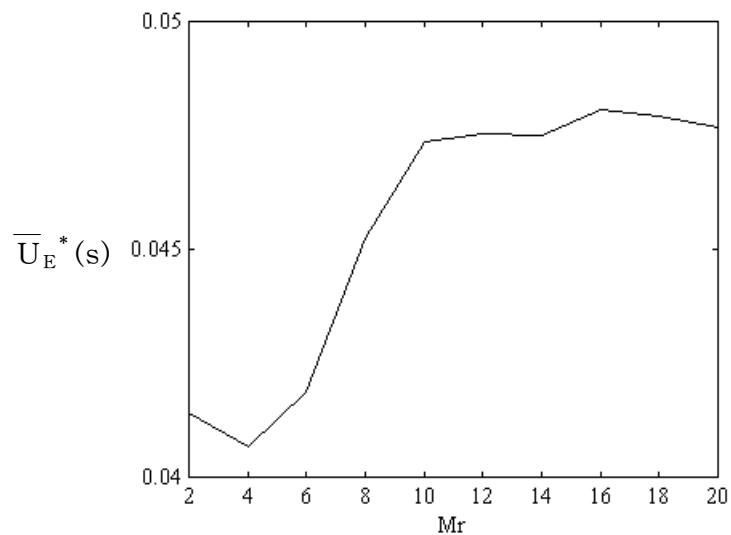


그림 3.2 이주율에 대한 전역 오프라인 강인성

Fig 3.2 Global off-line robustness vs the migration rate

제 4 장 시뮬레이션

제안된 하이브리드 탐색법의 성능평가를 위해 두 다변수 비선형 최적화 문제를 고려하였다. 하나는 연속시간 시스템의 파라미터를 추정하는 문제이고, 다른 하나는 PID 제어기의 파라미터를 동조하는 문제이다. 하이브리드 탐색법과 그 성능평가를 비교할 목적으로 Michalewicz가 제안한 RCGA[2]를 사용하였고 두 알고리즘에 채용한 연산자, 탐색전략과 파라미터는 다음 표와 같다.

표4.1 성능 비교용 두 탐색법

Table 4.1 Two search methods for performance comparison

탐색법		연산자/탐색전략	파라미터 값
제안된 하이브리드 탐색법	RCGA	구배와 유사한 재생산	$\eta = 1.7$
		수정단순교배	$P_c = 0.8$
		동적돌연변이	$P_m = 0.3$
		엘리트 전략	채용
		스케일링 윈도우 기법	$W_s = 1$
	단체법	반전	$\alpha = 1.5$
		확장	$\beta = 2.0$
		축소	$\gamma = 0.5$
		수축	$\delta = 0.5$
Michalewicz의 RCGA		루울렛휠 선택에 기초한 재생산	-
		산술교배	$P_c = 0.8$
		동적돌연변이	$P_m = 0.3$
		엘리트 전략	채용
		스케일링 윈도우 기법	$W_s = 1$

4.1 시스템의 파라미터 추정

먼저 연속시간 플랜트의 축소차수 모델을 얻는 문제를 고려하였다. 일반적으로 산업현장에서 사용되고 있는 제어기는 플랜트 모델에 기초하여 설계되는 경

우가 많다. 플랜트의 입력신호의 주파수 성분이 충분할 경우에는, 고차의 모델을 이용하면 모델화되지 않은 운동의 영향을 충분히 줄일 수 있어서 제어기의 성능 개선이 가능하지만 반대로 제어기 설계가 복잡해지는 경향이 있다. 이 때문에 흔히 축소차수 모델을 이용하며, 축소차수 모델을 기반으로 저차의 제어기를 설계하면 고차에 비해 상대적으로 소프트웨어의 오류수정이 쉽고, 계산량도 적어지는 장점이 있다.

고려한 플랜트는 식 (4.1)과 같이 시간지연을 갖는 이차 시스템이고, 추정될 모델은 그 축소차수 모델이다.

플랜트 :

$$G_P(s) = \frac{e^{-2s}}{(1+10s)(1+s)} \quad (4.1)$$

모델 :

$$G_M(s) = \frac{ke^{-Ls}}{1+\tau s} \quad (4.2)$$

식 (4.2)는 과감쇠(over-damped) 성질을 갖는 산업용 프로세스의 대표적인 모델로서 시간지연을 갖는 일차 시스템이다. 여기서 k 는 정상이득, τ 는 시정수, L 은 시간지연을 의미한다.

모델의 파라미터 추정은 그림 4.1과 같이 플랜트와 모델을 병렬로 연결하고 같은 입력을 가할 때, 모델의 동특성이 시스템과 일치하도록 적응 알고리즘이 모델의 파라미터를 조절하게 된다.

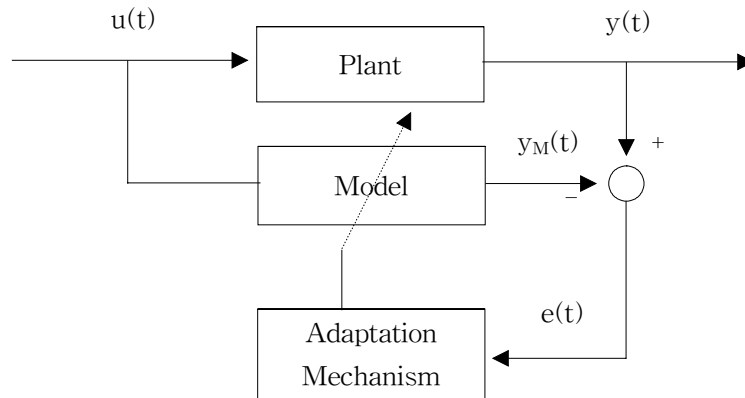


그림 4.1 파라미터 추정

Fig 4.1 Parameter estimation

이 때 두 출력값의 절대치 오차의 적분을 목적함수로 사용하였고, 이 목적함수 값이 최소가 되도록 파라미터를 추정한다.

$$J(\phi) = \int_0^{t_f} |y - y_M| dt \quad (4.3)$$

여기서 $\Psi = [k \ \tau \ L]^T$ 는 추정되는 파라미터 벡터이고, t_f 는 적분 상한값으로서 이후의 적분값이 무시되어도 좋을 정도로 충분히 큰 값으로 설계자에 의해 적절히 선택되어진다.

결국 주어진 문제는 비선형 다변수 최적화 문제로서, 그 탐색공간은 다봉일 수 있고, 도함수를 구하는 것이 어렵기 때문에 기존의 방법으로는 전역해를 구하기가 어렵다. 따라서 제안한 탐색법과 Michalewicz의 RCGA를 목적함수 값이 최소가 되도록 모델의 파라미터를 조절하는데 이용한다. 이 때 파라미터의 탐색 범위는 $0 \leq k, \tau, L \leq 20$ 으로 하였다.

아래 표 4.2는 독립된 씨드로 두 탐색법을 각각 5번 돌려 50, 100, 300, 500 세

대에서 “지금까지의 최적성능”을 얻어 평균과 표준편차를 구한 것이고, 표 4.3 또한 같은 세대에서 식 (4.4)의 Euclidean norm을 구한 것이다.

$$\|\mathbf{x}(k) - \mathbf{x}(\infty)\| = \sqrt{\sum_{i=1}^n (x_i(k) - x_i(\infty))^2} \quad (4.4)$$

여기서 $\mathbf{x}(\infty)$ 는 세대가 충분히 지난 후 두 알고리즘에 의해 탐색된 해의 평균치를 구한 것이다. 표 4.2는 두 알고리즘의 상대적인 탐색성능을 보이기 위한 것이며, 표 4.3은 추정된 파라미터들의 정밀도를 비교하기 위한 것이다.

표 4.2 지금까지의 최적성능의 평균과 표준편차

Table 4.2 Means and deviations of the so-far-best objective function value

탐색법 \ 세대수	50	100	300	500
Proposed Method	0.1259±0.2997×10 ⁻¹	0.1004±0.1166×10 ⁻¹	0.0869±0.1000×10 ⁻⁵	0.0869±0.1000×10 ⁻⁶
Michalewicz의 RCGA	0.8637±0.4704	0.6801±0.4687	0.1799±0.6060×10 ⁻¹	0.1219±0.1336×10 ⁻¹

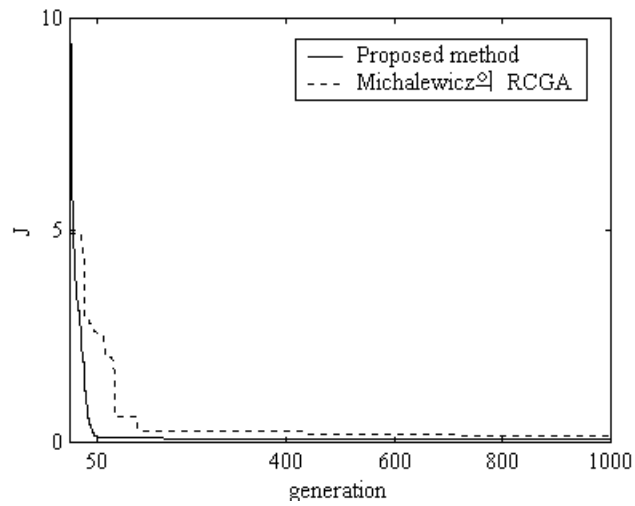
표 4.3 추정된 파라미터의 Euclidean norm

Table 4.3 Euclidean norm of estimated parameters

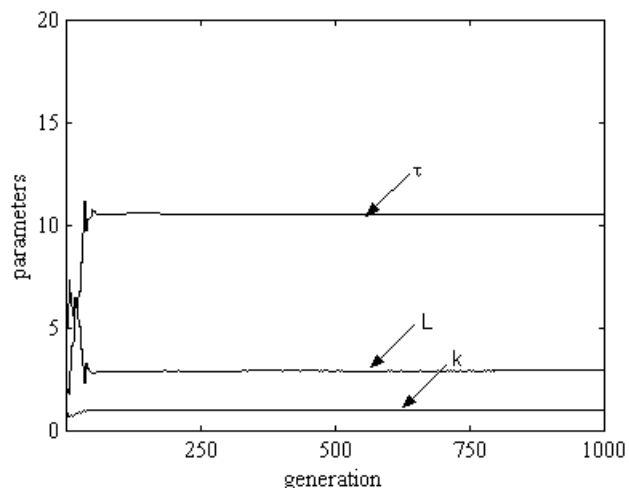
탐색법 \ 세대수	50	100	300	500
Proposed Method	0.2384±0.1621	0.1015±0.6926×10 ⁻¹	0.0106±0.1011×10 ⁻¹	0.0092±0.5563×10 ⁻²
Michalewicz의 RCGA	1.6932±1.2651	1.1374±0.6067	0.3430±0.1051	0.2046±0.7888×10 ⁻¹

표에서 보면 제안된 방법이 Michalewicz의 RCGA보다 탐색성능이 빠르고, 약 500 세대에서 소숫점 3자리 이하의 정밀도를 유지하지만 Michalewicz의 RCGA는 500세대가 경과함에도 불구하고 소숫점 1자리 이하의 정밀도를 유지하

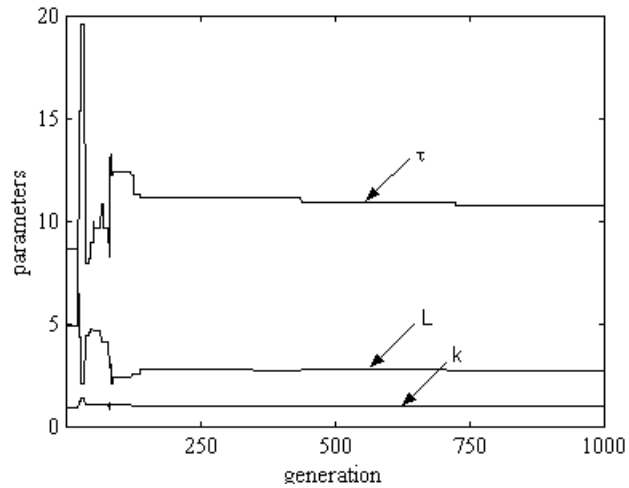
고 있다. 그림 4.2는 제안한 방법과 Michalewicz의 RCGA의 파라미터 탐색과정을 보여주는 한 예로서 그림 4.2(a)는 두 탐색법의 “지금까지의 최적성능”을 비교한 것이고 그림 4.2(b)와 (c)는 각각의 파라미터를 그린 것이다.



(a) 목적함수



(b) 파라미터(제안한 방법)



(c)파라미터(Michalewicz의 RCGA)

그림 4.2 진화적 파라미터 추정

Fig 4.2 Evolutionary parameter estimation

그림 4.2도 제안한 방법이 Michalewicz의 RCGA보다 빨리 탐색함을 쉽게 확인할 수 있다.

4.2 PID 제어기의 파라미터 동조

다음은 하이브리드 탐색법을 PID 제어기의 파라미터 동조문제에 적용하여 그 성능을 확인하였다. 그림 4.3과 같은 PID 제어시스템을 고려하였다. 여기서 제어 대상으로 앞서 추정된 모델을 사용하였고, 밸브의 동작한계, 증폭기의 토르크 제한 등 현실성이 있는 제어문제를 다루기 위해 제어대상 직전에 그림과 같은 비선형 요소를 추가하였다.

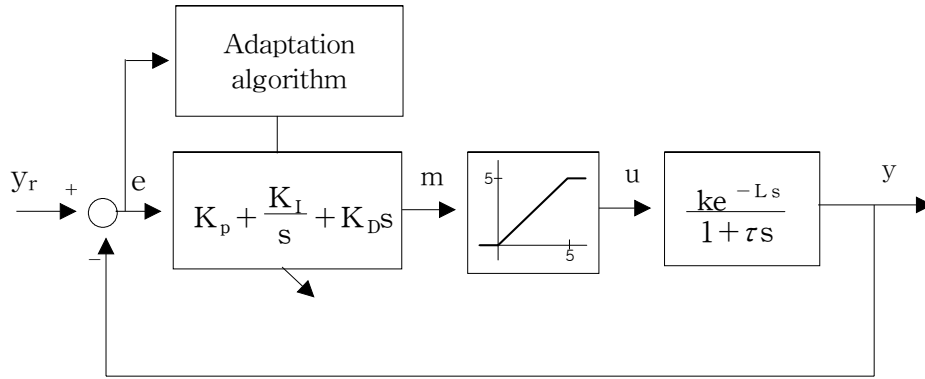


그림 4.4 PID 제어기의 파라미터 동조
 그림 4.4 Parameter tuning of the PID controller

플랜트:

$$G_M(s) = \frac{ke^{-Ls}}{1 + \tau s} \quad (4.5)$$

단, $k = 1.0013$, $\tau = 10.5169$, $L = 2.9354$

비선형 요소:

$$u = \begin{cases} 5, & m > 5 \\ m, & 0 < m \leq 5 \\ 0, & m < 0 \end{cases}$$

이때 PID 제어기 동조에 사용된 목적함수는 다음과 같다.

$$J(\phi) = \int_0^{t_f} |y_r - y| dt \quad (4.6)$$

단, $\Psi = [K_p \ K_I \ K_D]^T$ 는 동조되는 파라미터 벡터이고, t_f 는 적분 상한값이다. 결국 주어진 문제는 시스템의 방정식을 만족하면서 목적함수를 최소로 하는 Ψ 를 구하는 문제로 귀착된다. 탐색영역은 $0 \leq K_p, K_I, K_D \leq 10$ 로 하였다.

표 4.4와 표 4.5도 두 방법을 이용한 성능 결과를 비교하였다. 마찬가지로 탐색속도도 개선되고, 파라미터의 정밀도도 개선할 수 있음을 확인할 수 있다.

표 4.4 지금까지의 최적성능의 평균과 표준편차

Table 4.4 Means and deviations of the so-far-best objective function value

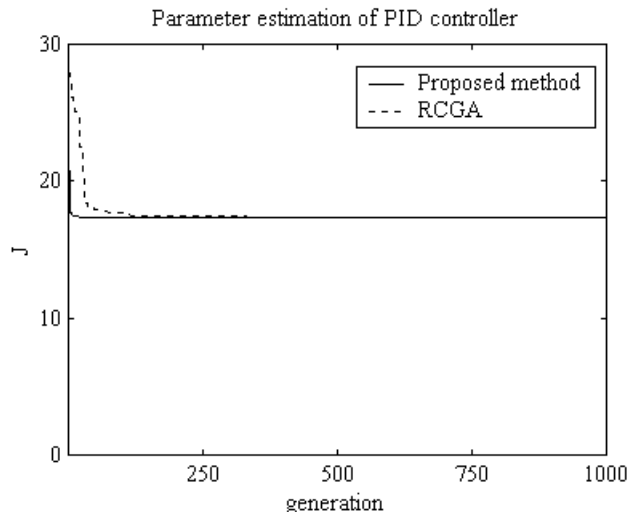
세대수 \ 탐색법	50	100	300	500
Proposed Method	$17.4615 \pm 0.2400 \times 10^{-1}$	$17.4615 \pm 0.2400 \times 10^{-1}$	$17.4527 \pm 0.1959 \times 10^{-1}$	$17.4527 \pm 0.1959 \times 10^{-1}$
Michalewicz의 RCGA	18.8184 ± 0.9583	17.9452 ± 0.4077	$17.5305 \pm 0.3442 \times 10^{-1}$	$17.5258 \pm 0.3109 \times 10^{-1}$

표 4.5 추정된 파라미터의 Euclidean norm

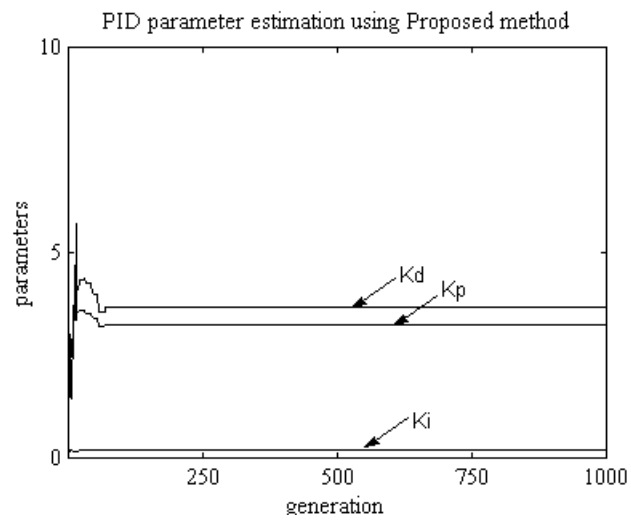
Table 4.5 Euclidean norm of estimated parameters

세대수 \ 탐색법	50	100	300	500
Proposed Method	$0.3723 \pm 0.1000 \times 10^{-1}$	$0.0057 \pm 0.100 \times 10^{-3}$	$0.0003 \pm 0.100 \times 10^{-4}$	0.0000 ± 0.0000
Michalewicz의 RCGA	2.6866 ± 0.2501	1.3776 ± 0.2300	$1.1201 \pm 0.100 \times 10^{-1}$	$0.7529 \pm 0.100 \times 10^{-2}$

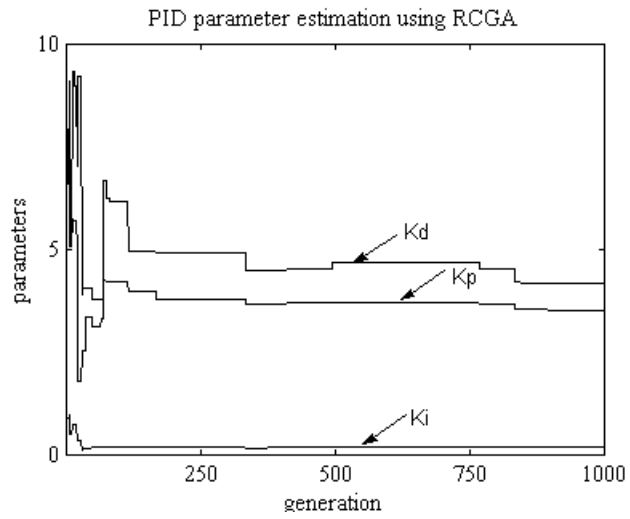
그림 4.5 또한 제안한 방법과 Michalewicz의 RCGA의 파라미터 탐색과정을 보여주는 한 예로서 앞에서 얻은 결과와 유사함을 알 수 있다.



(a) 목적함수



(b) 파라미터(제안한 방법)



(c) 파라미터(Michalewicz의 RCGA)

그림 4.5 PID제어기의 파라미터 동조

Fig 4.5 Parameter tuning of the PID controller

제 5 장 결 론

일반적으로 단일점 탐색법은 정밀탐색이 가능하나 탐색공간의 기울기 정보를 필요로 하거나, 지역 최적해에 수렴할 가능성이 있으며, 지역해에 빠지면 이로부터 탈출할 방법이 없다는 단점을 가지고 있다. 한편, 유전알고리즘은 전역탐색하는 능력을 가진 반면 지역탐색하는 능력이 떨어지며, 세대가 반복됨에 따라 개체들의 유전자형이 닮게 되어 탐색에 정체현상이 나타난다. 이러한 현상으로 인해 탐색능력을 상실하거나 지역해로 조기수렴하는 문제점이 발생된다.

따라서 본 논문에서는 단체법의 정밀탐색 능력과 유전알고리즘의 전역탐색 능력을 서로 결합하여 섬모델을 기반으로 하는 병렬 구조의 하이브리드 탐색법을 제안하였다. 제안한 하이브리드 탐색법에서는 두 알고리즘이 각각 고립시간 동안 독립적으로 진행해 가다가 서로간에 이주자를 교환하며, 이러한 절차는 주기적으로 일어난다. 이렇게 함으로써 탐색능력을 개선할 수 있었고 또한 해의 정밀도를 높일 수 있었다. 제안한 방법을 시스템의 모델링, PID 제어기의 파라미터를 동조하는 문제에 적용한 결과 만족할 만한 결과를 얻을 수 있었다.

차후로 다른 알고리즘과의 병렬 구현과, 보다 폭 넓은 테스트 환경에서 그 성능을 확인해보는 체계적인 연구가 필요하다고 생각된다.

참 고 문 헌

- [1] J. H. Holland, *Adaptation in natural and artificial systems*, The University of Michigan Press, Michigan, 1975
- [2] Z. Michalewicz, *Genetic algorithm + data structures = evolution program*, Springer-Verlag, Inc., Heidelberg, Berlin, 1996.
- [3] M. Gen and R. Cheng, *Genetic algorithms and engineering design*, John -Wiley & Sons, Inc., N.Y., 1997.
- [4] G. Jin and S. Joo, "A Study on a Real-coded Genetic Algorithm", *Journal of Control Automation and Systems Engineering*, Vol. 6, No. 4, pp. 268-275, 2000
- [5] R. Young and A. Reel, "A Hybrid Genetic Algorithm for a Logic Problem", *Proceedings of the 9th European Conference on Artificial Intelligence*, pp. 744--746, 1990.
- [6] K. Kurbel, B. Schneider and K. Singh, "*Parallelization of Hybrid Simulated Annealing and Genetic Algorithm for Short-term Production Scheduling*", in *Proceedings of International Conference on Intelligent, Knowledge and Integration for Manufacturing*, Nanjing, China, March 28-31, 1995 (to appear).
- [7] K. C. Tan, Y. Li, D.J. Murray-Smith and K.C. Sharman, "*System Identification and Linearisation Using Genetic Algorithms with Simulated Annealing*," *Proc. First IEE/IEEE Int. Conf. on GA in Eng. Syst.:*

- Innovations and Appl. , pp. 164--169, 1995.
- [8] Yun Li, Kay Chen Tan, Kim Chwee Ng, and David J. Murray-Smith
 “Performance Based Linear Control System Design by Genetic Evolution with Simulated Annealing,” Proc. 34th IEEE CDC. New Orleans. 1995, pp. 731-736
- [9] Sirag, D. J. and P. T. Weisser, *Toward a unified thermodynamic genetic operator*, in Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms, Cambridge, MA, 1987, Lawrence Elbaum publ., Hillsdale, NJ, 1987.
- [10] I. K. Jeong and J. J. Lee. *Adaptive simulated annealing genetic algorithm for system identification*. Eng. Applic. Artificial Intell., 9(5) : 523--532, 1996. 129
- [11] J. J. Grefenstette, “Optimization of Control Parameter for Genetic Algorithms,” IEEE Trans. Syst., Man, Cybern., Vol. SMC-16, No.1, pp 122-128,1986
- [12] J. C. Potts, T. D. Giddens and S. B. Yadav, “The development and evaluation of an improved genetic algorithm based on migration and artificial selection,” *IEEE Trans. Syst., Man, Cybern.*, Vol, No. 1, pp. 73-86, 1994
- [13] M. Gen, R. Cheng, *Genetic algorithms and engineering design*, John-Wiley & Sons, Inc., N.Y., 1997
- [14] K. Krishnakumar, “Micro-genetic algorithms for stationary and non-stationary function optimization”, *SPIE, Intelligent Control and Adaptive Systems*, Vol. 1196, pp. 289-296, 1989
- [15] 김길태, 최형식, 진강규, “유전알고리즘을 이용한 크레인 시스템의 최적제어”, 한국박용학회지, 제22권, 제4호, pp.80-86, 1998
- [16] L. Davis, *Handbook of genetic algorithms*, Van Norstrand Reinhold,

N.Y., 1991

- [17] J. D. Schaffer et al., "A Study of control parameters affecting online performance of genetic algorithms for function optimization," *Proc. 3rd Int. Conf. on Genetic Algorithms and Their Applications*, Arlington, VA, pp. 51-60, 1989
- [18] G. A. Manson, *Time optimal control methods arising from the study of overhead cranes*, PhD. Thesis Univ. of Strathclyde, Glasgow, UK, 1997
- [19] H. Muhlenbein, M. Schomisch and J. Born, "The Parallel Genetic Algorithm as Function Optimizer," *Parallel computing*, Vol. 17, pp. 619-632
- [20] P. Spiessens and B. Manderick, "A Massively Parallel Genetic Algorithm: Implementation and First Analysis," *Proc. 4th Int. Conf. on Genetic Algorithms*, Morgan Kaufman, San Mateo, CA, pp. 279-286, 1991
- [21] H. Tamaki and Y. Nishikawa, "A Parallel Genetic Algorithm based on a Neighborhood Model and Its Application to the Jobshop Scheduling," *Parallel Problem Solving From Nature 2*, R. Manner and B. Maderick(Eds), North-Holland, pp. 573-582, 1992
- [22] A. J. Chipperfield and P. J. Fleming, "Parallel Genetic Algorithms: A Survey," *Research Report No. 518*, Department of Automatic Control and Systems Engineering, University of Sheffield, UK, 1994
- [23] A. Kapsalis, G. D. Smith, and V. J. Rayward-Smith, "A Unified Paradigm for Parallel Genetic Algorithms," *Proc. AISB Workshop on Evolutionary Computing*, T. C. Fogarty(Ed.), Springer-Verlag, Leeds, UK, Vol, 865, pp. 131-149, 1994
- [24] M. Schwehm, "Massively Parallel Genetic Algorithms," *Procc. Int. Conf. MPP'94*, Delft, Netherlands, pp. 505-512, 1994
- [25] A. J. Chipperfield, P. J. Fleming and A. Y. Zomaya, *Parallel Genetic*

- Algorithms, Parallel & Distributed Computing Handbook, McGraw-Hill, 1996
- [26] K. Deb and D. E. Goldberg. "An Investigation of Niche and Species Formation in Genetic Function Optimization," Proc. 3rd Int. Conf. on Genetic Algorithms, 1989
- [27] R. J. Collins and D. R. Jefferson. "Selection in massively parallel genetic algorithms," Proc. 4th Int. Conf. on Genetic Algorithms, Morgan Kaufmann, 1991
- [28] C. Ryan, "Niche and Species Formation in Genetic Algorithms," Practical Handbook of Genetic Algorithms: Applications, L. Chambers(Ed.), Vol.1, pp. 57-74, 1995
- [29] 정인수, 권기호, 홍석봉, 김유석, "유전자 알고리즘의 다양성 향상을 위한 Island Model의 성능 개선에 관한 연구," 제 5 회 인공지능, 신경망, 및 퍼지 시스템 종합학술대회 발표논문집, pp. 199-202, 1996
- [30] K. A. De Jong, *An analysis of the behavior of a class of genetic adaptation systems*, PhD. Dissertation, Univ. of Michigan, Ann Arbor, Michigan, 1975
- [31] W. Spendly, G. R. Hext, and F. R. Himsforth, "Sequential application of simplex designs in optimization and evolutionary operation," *Technometrics*, vol. 4, pp. 441-461, 1962.
- [32] Nelder, J. A. and R. Mead, "A Simplex Method for Function Minimization," *Computer J.*, 7(1965), 308-313.
- [33] J. C. Lagarias. J. A. Reeds. M. H. Wright and P. E. Wright, "Convergence properties of the Nelder-Mead simplex algorithm in low dimensions," *SIAM Journal on Optimization*, Vol. 8, 1998
- [34] R. Tanese, "Distributed Genetic Algorithms," Proc. 3rd Int. Conf. on Genetic Algorithms, J. D. Schaffer(ed.), Morgan Kaufmann Publishers,

San Mateo, pp. 434-439, 1989

- [35] R. Lohmann, "Applicatin of Evolution Strategy in Parallel Populations," Parallel Problem Solving from Nature, Lecture Notes in Computer Science, H,-P, Schwefel and R. Manner(Eds), Springer-Verlag, Vol. 496, pp.198-208, 1991
- [36] G. Rudolph, "Global Optimization by Means of Distributed Evolution Strategies," Parallel Problem Solving from Nature, Lecture Notes in Computer Science, H,-P, Schwefel and R. Manner(Eds), Springer-Verlag, Vol. 496, pp. 209-213, 1991
- [37] 진강규, 유전알고리즘과 그 응용, 교우사, 2000