실수코딩 유전알고리즘에 관한 연구

A Study on a Real-Coded Genetic Algorithm

진 강 규, 주 상 래
(Gang-Gyoo Jin and Sang-Rae Joo)

Abstract : The increasing technological demands of today call for complex systems, which in turn involve a series of optimization problems with some equality or inequality constraints. In this paper, we presents a real-coded genetic algorithm(RCGA) as an optimization tool which is implemented by three genetic operators based on real coding representation. Through a lot of simulation works, the optimum settings of its control parameters are obtained on the basis of global off-line robustness for use in off-line applications. Two optimization problems are presented to illustrate the usefulness of the RCGA. In case of a constrained problem, a penalty strategy is incorporated to transform the constrained problem into an unconstrained problem by penalizing infeasible solutions.

Keywords : real coding, genetic algorithm, constrained optimization, penalty strategy

I. 서론

실세계의 최적화 문제들은 시스템의 규모가 크고, 변수간의 상호작용이 강하고, 제약(구속)조건이 수반될수록 더욱 복잡해지는 경향이 있다. 이러한 문제들의 탐색공간은 다봉인 경우가 많아 구배에 기초한 기존의 방법을 적용하면 지역해 수렴문제가 제기되고, 도함수를 얻기가 어려운 경우에는 적용 자체가 불가능하게 된다. 이를 해결하기 위한 방법으로 유전알고리즘(genetic algorithms)[1], 모의진화(simulated evolution)[2], 진화전략(evolutionary strategies)[3] 등과 같은 자연현상을 흉내낸 알고리즘들이 개발되어 왔다. 이들은 탐색공간에 대한 사전지식이 없고, 목적함수 외에 보조 정보를 요구하지 않는 장점 때문에 시스템 식별 및 제어, 기계학습, 설비배치, 신경회로망, 신호처리, 생명공학 등 많은 분야에서 성공적으로 이용되고 있다[4][5]. 그 중에서도 유전알고리즘이 차지한 위치는 대단히 크다.

자연선택과 유전학에 기초한 유전알고리즘은 전통적으로 이진코딩 염색체에서 동작해 왔고, 아직도 많은 유전알고리즘에서는 이진코딩을 채용하고 있다. 그러나 사용자가 탐색공간에 대한 사전지식이 없어 큰 정의영역을 택하거나 해의 정밀도를 높이게 되면 염색체 길이는 길어진다. 긴 염색체는 큰 탐색공간을 만들게 되어 계산부담을 증가시키고 경우에 따라서는 탐색을 불가능하게 한다. 한편 복잡한 제약조건이 수반되는 문제에 기존의 이진 연산자들을 적용하면 부적합한 해(infeasible solution)가 발생된다. 이에 대한 해결책으로 변수 표현을 문제공간에 접근시키는 코딩법의 개선, 연산자의 개선, 부적합한 해의 발생과 연관된 복구전략(repairing strategy)과 벌점전략(penalty strategy) 등에 대한 폭 넓은 연구가 수행되고 있다[4][5].

본 연구에서는 이진코딩 유전알고리즘의 이러한 문제점을 극복하고 제약조건을 갖는 최적화 문제 해결에 적합한 하나의 실수코딩 유전알고리즘(real-coded genetic algorithm: RCGA)을 제안한다. 제안된 RCGA에서 염색체는 실수벡터로 표시되고 이를 효율적으로 다룰 수 있도록 유전 연산자를 구현한다. RCGA의 탐색성능은 매개변수들의 선정에 영향을 받기 때문에 적절한 성능평가 환경과 방법으로 최적의 파라미터 영역을 도출한다. RCGA의 효용성은 두 최적화 문제에서 확인된다. 특히, 제약조건을 수반되는 문제의 경우 잠정적인 해가 제약조건을 위반하면 그 위반 정도에 따라 벌점을 부과하도록 벌점전략을 채용하였다.

접수일자 : 1999. 7. 29., 수정완료 : 1999. 12. 22.
진강규 : 한국해양대학교 자동화·정보공학부
주상래 : 코닉시스템 응용 소프트웨어 개발부

II. 유전 연산자

1. 염색체 표현

전통적인 표현법으로서의 이진코딩은 탐색공간이 크고, 고정밀도의 해를 요구하거나, 제약조건이 존재할 때는 어려움을 겪게 된다. 이러한 단점을 보완해줄 수 있는 한 방법이 실수코딩이다. k세대의 실수코딩 염색체(개체) $\mathbf{s}(k)$는 다음과 같이 표시된다.

$$\mathbf{s}(k)= \mathbf{x}^T(k)= (x_1(k)\ x_2(k)\ \cdots\ x_j(k)\ \cdots\ x_n(k)) \quad (1)$$

여기서 $x_j(k)$는 j번째 요소(유전자), $\mathbf{x}(k)\in\mathbb{R}^n$는 해 벡터이고, n은 벡터의 차원이다. 실수코딩을 채용함으로써 염색체의 길이 ℓ는 벡터의 차원 n과 일치하게 되고, 문제공간에서 가까운 두 점은 표현공간에서도 서로 가깝게 되는 특징을 갖는다.

k세대의 집단은 다음과 같이 정의된다.

$$\mathbf{P}(k)= \{\mathbf{s}_1(k)\ \mathbf{s}_2(k)\ \cdots\ \mathbf{s}_i(k)\ \cdots\ \mathbf{s}_N(k)\} \quad (2)$$

여기서 $\mathbf{s}_i(k)= (x_{i1}(k)\ x_{i2}(k)\ \cdots\ x_{ij}(k)\ \cdots\ x_{in}(k))$는 i번째 염색체, $x_{ij}(k)$는 i번째 염색체의 j번째 요소이고, N은 집단 크기로서 세대와는 상관없이 고정된다. 초기집단 $\mathbf{P}(0)$는 유전알고리즘이 가혹한 환경으로부터 해를 찾는 능력을 계량화하도록 무작위법으로 초기화된다. 따라서 i번째 염색체의 j번째 요소 $x_{ij}(0)$는 정의구간 $[x^L_j, x^U_j]$ 내에서 임의로 발생되는 실수값으로 초기화된다.

제안된 RCGA도 재생산(reproduction), 교배(crossover),

돌연변이(mutation)를 기본 유전 연산자로 채용한다.

2. 재생산 연산자

재생산은 다윈의 적자생존 현상을 모방한 인위적인 메커니즘으로서 적합도값에 기초하여 개체를 선택하고 교배 급원을 구성한다. 이로서 집단은 평균 적합도가 증가하게 되고 점진적으로 해에 접근하게 된다. 재생산을 알고리즘 형태로 구현한 방법으로는 루울렛휠 선택(roulette wheel selection), 토너먼트 선택(tournament selection), 순위에 기초한 선택(rank-based selection) 등이 있다. 이 중에서 루울렛휠 선택이 많이 이용되고 있다. 루울렛휠 선택은 전적으로 부모세대 개체들의 선택확률에 의존하기 때문에, 만약 초기세대에 초우량 개체가 존재하게 되면 이 개체가 집단을 지배하게 되어 유전적 다양성(genetic diversity)이 감소되고 이로 인해 준최적해에 조기수렴 하는 문제점이 발생한다. 따라서 집단의 크기는 30-200 범위에서 유지되는데[7], 만약 이를 다차원 문제에 적용하면 연산부담이 급격히 증가하게 된다.

이를 해결하기 위해 연산자와 탐색전략을 개선하는 문제가 연구되어 왔으며[8,9], 특히 저자는 구배와 유사한 재생산(gradient-like reproduction)을 도입하여 우량 개체의 다중 복제, 재생산 연산 중에 최적 개체의 소멸 등의 문제를 해결하려 하였다[10]. 본 연구에서는 저자가 제안한 이진코딩 재생산 연산자를 보완하여 RCGA에서 구현하였고 연산자의 매개변수를 확률적으로 설정하였다. 재생산은 다음의 3단계를 거쳐 완성된다:

단계 1 : 집단 $\mathbf{P}(k-1)$를 평가하고 최적 개체를 찾는다.

$$\mathbf{s}_b(k-1)= (x_{b1}(k-1)\ x_{b2}(k-1)\ \cdots\ x_{bn}(k-1))= \arg\max_{1\le i\le N}[f_i(k-1)] \tag{3a}$$

$$f_b(k-1)= \max_{1\le i\le N}[f_i(k-1)]\ (>0) \tag{3b}$$

단계 2 : i= 1

단계 3 : i ≤ N이 만족될 때까지 단계 3-6을 반복한다.

단계 4 : (4)를 통해 각 개체에게 새로운 값을 할당한다.

$$x_{ij}(k)= x_{ij}(k-1)+ \eta_i\frac{[f_b(k-1)-f_i(k-1)]}{f_b(k-1)}[x_{bj}(k-1)-x_{ij}(k-1)]\ (1\le j\le n) \tag{4}$$

여기서 $x_{ij}(k)$, $x_{ij}(k-1)$는 각각 재생산 전후의 i번째 개체의 j번째 요소이고, η_i는 양의 계수로서 정규분포 $N(\eta, \sigma^2)$를 따르도록 설정된다. (4)를 벡터형으로 표시하면 다음과 같다.

$$\mathbf{x}_i(k)= (1-\xi_i(k))\mathbf{x}_i(k-1)+\xi_i(k)\mathbf{x}_b(k-1) \tag{5a}$$

단,

$$\xi_i(k)= \eta_i\frac{[f_b(k-1)-f_i(k-1)]}{f_b(k-1)} \tag{5b}$$

단계 5 : 새 개체 $\mathbf{s}_i(k)= (x_{i1}(k)\ x_{i2}(k)\ \cdots\ x_{in}(k))$를 교배 급원에 복제한다.

단계 6 : i= i+1

표 1은 N= 6인 집단에서 수행된 재생산을 예시한 것이다. 여기서 η_i값은 $N(1, 0.3^2)$를 따르는 난수이다. 최적인 5번 염색체는 자동적으로 선택되었고, 2번과 6번은 같지만 서로 다른 위치에 사상되었다. 염색체마다 서로 다른 η_i를 적용함으로써 다중복제를 막을 수 있었다. 재생산 전후의 평균 적합도값을 구해보면 각각 0.2696과 0.3418인데 이것은 집단이 개선되었음을 의미한다.

표 1. 구배에 기초한 재생산의 예.

Table 1. An example of gradient-like reproduction.

i	$s_i(k-1)$	$f_i(k-1)$	η_i	$s_i(k)$	$f_i(k)$	비 고
1	$s_1(k-1)$= (-2.3, 1.5)	0.2507	1.49	$s_1(k)$= (-0.9, 0.0)	0.2579	
2	$s_2(k-1)$= (-1.0,-1.0)	0.1568	1.59	$s_2(k)$= (-0.3,-1.4)	0.1915	
3	$s_3(k-1)$= (0.0, 0.0)	0.2950	0.51	$s_3(k)$= (0.3,-0.5)	0.3318	
4	$s_4(k-1)$= (2.3, 0.5)	0.2673	1.01	$s_4(k)$= (2.1,-0.1)	0.4408	
5	$s_5(k-1)$= (1.5,-2.5)	0.3335	1.35	$s_5(k)$= (1.5,-2.5)	0.3335	$s_b(k-1)$
6	$s_6(k-1)$= (-1.0,-1.0)	0.1568	0.97	$s_6(k)$= (0.3,-1.8)	0.2654	

그림 1은 재생산 전후의 벡터들을 도식적으로 표시한 것이다. 그림에서 보면 재생산 이전의 개체들('×' 표시)은 최적인 5번 쪽으로 움직이는 것을 알 수 있다.

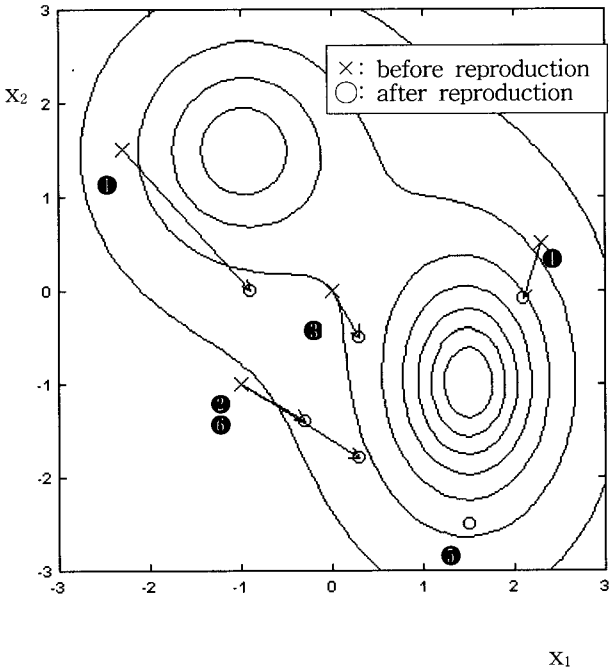


그림 1. 재생산 전후의 벡터 위치.

Fig. 1. Vector locations before and after reproduction.

(4) 또는 (5)로 기술되는 재생산은 η_i가 0과 2사이에서 유지될 때 안정하게 되므로[10], η_i가 이 범위 밖의 값이면 난수발생기에 의해 새로운 값이 발생된다. 비록 η_i가 이

범위내에 있더라도 경우에 따라서는 부적합한(정의영역 밖으로 사상) 개체들을 생산할 수 있는데 이 때는 복구 알고리즘이나 벌점전략을 사용하여야 한다. 본 논문에서는 정의영역을 벗어나면 적합한 해가 발생될 때까지 η_i의 초기 설정값을 20%씩 감하면서 (4)를 다시 연산하도록 하였다. (5)에서도 알 수 있듯이 $\xi_i(k)$가 0과 1사이의 값이면 생성되는 개체는 항상 적합하게 된다.

3. 교배 연산자

자연계 생물들의 유성생식을 흉내낸 교배 연산은 탐색공간상의 새로운 점을 찾기 위하여 교배급원으로부터 부모 염색체 쌍을 임의로 선택하고 임의로 선택된 부스트링을 서로 교환해줌으로써 자손을 생성한다. 실수코딩 염색체의 구조는 이진코딩의 그것과는 다르게 되므로 기존의 교배 연산자를 적용하기가 어렵고 특수한 형태를 필요로 한다. 자주 이용되는 연산자로는 단순교배(simple crossover), 산술적교배(arithmetical crossover)[4], BLX-α교배(BLX-α crossover)[11] 등이 있다. 제안된 교배 연산자에서는 교배점의 요소는 일차결합 형태로 변경되고, 교배점 이후의 요소들은 서로 교환된다.

단계 1 : i= 0

단계 2 : i < N 조건이 만족될 때까지 단계 2-6을 반복한다.

단계 3 : 부모 염색체 쌍을 임의로 선택한다.

$$s_v(k)= (x_{v1}(k)\cdots x_{vj}(k)\ x_{vj+1}(k)\cdots x_{vn}(k)) \quad (6a)$$

$$s_w(k)= (x_{w1}(k)\cdots x_{wj}(k)\ x_{wj+1}(k)\cdots x_{wn}(k)) \quad (6b)$$

단, v, w∈ [1, N]

단계 4 : 난수 r∈ [0, 1]을 발생시켜 $r \le P_c$이면, 교배점 c∈ [1, n]을 발생시키고 교배가 이행된다. 교배점의 두 요소는 일차결합 되고, 이후 요소들은 서로 교환된다.

$$s'_v(k)= (x_{v1}(k)\cdots x'_{vc}(k)\ x_{w,c+1}(k)\cdots x_{wn}(k)) \quad (7a)$$

$$s'_w(k)= (x_{w1}(k)\cdots x'_{wc}(k)\ x_{v,c+1}(k)\cdots x_{vn}(k)) \quad (7b)$$

$$\text{단, } x'_{vc}(k)= \lambda x_{vc}(k)+(1-\lambda)x_{wc}(k) \quad (7c)$$

$$x'_{wc}(k)= \lambda x_{wc}(k)+(1-\lambda)x_{vc}(k) \quad (7d)$$

이고 λ는 0과 1 사이의 난수이다.

반대로 $r > P_c$이면 선택된 부모가 자손이 된다. 즉, $s'_v(k)= s_v(k)$, $s'_w(k)= s_w(k)$이 된다.

단계 5 : 교배된 자손들을 임시집단에 복제한다.

단계 6 : i= i+2

(7)에서도 알 수 있듯이 교배를 통해 탐색되는 영역은 매개변수 λ값에 따라 달라진다. 본 연구에서는 λ의 값을 선정함에 있어서 0과 1사이의 볼록(convex) 구간으로 한정하였지만, 어파인(affine) 구간까지 확장하면 탐색이 확대된다[5]. 이 때도 복구 알고리즘이나 벌점전략이 필요하다.

4. 돌연변이 연산자

재생산과 교배는 세대가 진행되는 동안 지역탐색(exploitation)함으로서 집단을 강하게 해주지만, 속성상 전역탐색(exploration)하는 능력이 부족하다. 유전알고리즘이 지역해나 사점(dead corner)에 빠지게 될 때 이로부터 벗어나게 하고, 더 넓은 영역을 탐색할 수 있도록 돌연변이가 필요하다. RCGAs의 경우에도 돌연변이가 필요하며, 이 경우에는 기존의 돌연변이와는 다른 형태가 되어야 한다. 제안된 RCGA에는 균등 돌연변이[4]가 이용된다. 돌연변이는 다음 5단계를 통해 완결된다.

단계 1 : i= 0

단계 2 : i < N·n 조건이 만족될 때까지 단계 2-5를 반복한다.

단계 3 : 염색체에서 유전자 하나를 선택한다(순차적으로).

$$s_v(k)= (x_{v1}(k)\cdots x_{vj}(k)\cdots x_{vn}(k)),\ v\in[1,N] \quad (8)$$

단계 4 : 난수 r∈ [0, 1]을 발생시켜 $r \le P_m$이면 돌연변이를 일으킨다.

$$s'_v(k)= (x_{v1}(k)\cdots x'_{vj}(k)\cdots x_{vn}(k)) \quad (9)$$

단, $x'_{vj}(k)$는 $x_j(k)$의 정의영역 내에서 임의로 발생되는 난수이다.

단계 5 : i= i+1

5. 엘리트 전략

RCGA의 재생산에서는 최적 개체가 반드시 선택되지만 교배와 돌연변이를 거치는 동안 파괴될 수 있다. 최적 개체의 소멸은 좋은 유전자를 잃게 되므로 한 세대에서 다음 세대로 최적 개체의 생존을 보장해 주는 엘리트전략(elitist strategy)[6]이 필요하다. 이 전략은 이전 세대의 최적 개체를 저장하고 있다가 현 세대의 마지막 단계에서 이 개체가 소멸되면 저장된 개체를 현 세대의 최악 개체와 교환해 준다.

III. 제어 파라미터의 설정

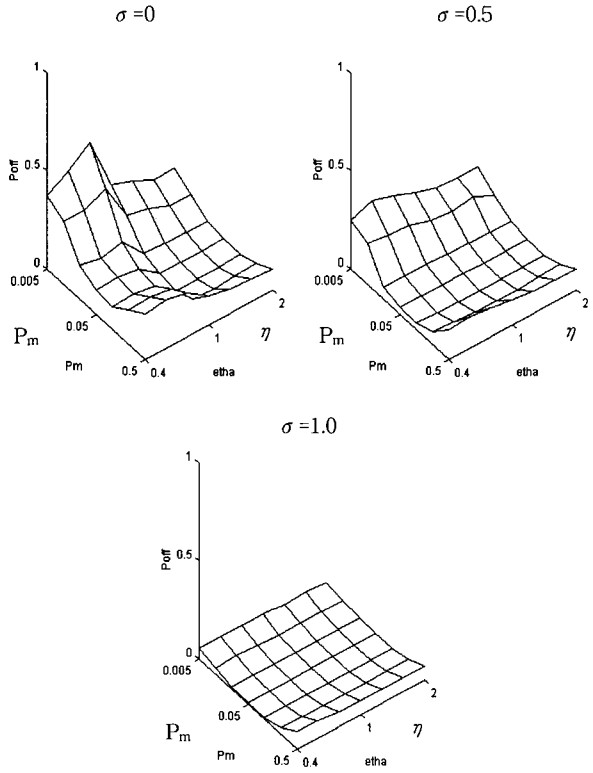
여러 다른 알고리즘과 같이 RCGA도 해의 정밀도와 탐색성능에 영향을 미치는 매개변수, 즉 제어 파라미터(control parameter)를 가지는데, 집단크기(N), 재생산 계수(η_i), 교배확률(P_c), 돌연변이확률(P_m) 등이 여기에 해당된다. 주어진 환경에서 최적의 성능을 확보하기 위해서는 이들의 적절한 설정은 대단히 중요하다. 현재까지 이들을 설정하는 체계적인 방법은 없으나 경험과 실험에 의존함으로서 어느 정도 합당한 결과를 유도해 내고 있다[10][12][13]. 경험과 실험을 바탕으로 하는 방법은 적절한 테스트 환경(테스트 함수)과 성능측정 수단을 필요로 한다.

1. 테스트 환경과 성능측정 방법

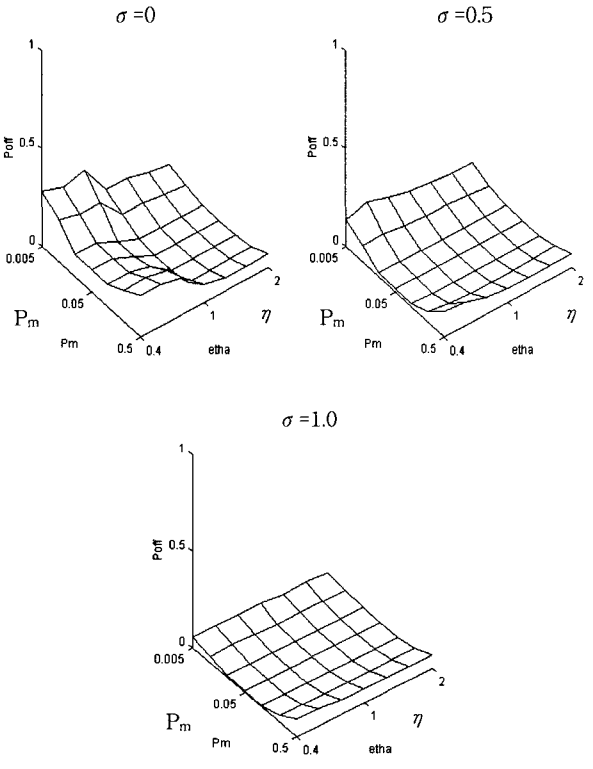
테스트 환경으로는 De Jong의 벤치마크 함수(5개의 최소화 문제)를 사용하였고 특히 네 번째 함수의 경우에는 확률적 성질은 고려하지 않았다. 주어진 테스트 환경에서 유전알고리즘의 성능은 적절한 평가척도를 통해 측정되는데 De Jong의 오프라인 성능측정 방법이 이용된다[6].

$$u_e^*(s, N_G)= \frac{1}{N_G}\sum_{k=1}^{N_G} F_e^*(k) \quad (10a)$$

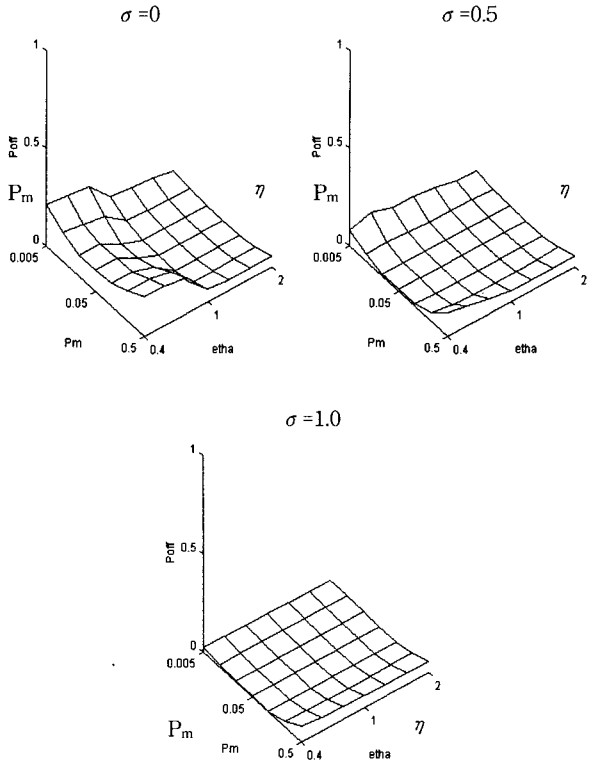
$$\text{단, } F_e^*(k)= \underset{1\le i\le k}{\text{best}}\ F_e(i) \quad (10b)$$



(a) N= 10



(b) N= 20



(c) N= 30

그림 2. 전역 오프라인 강인성.
Fig. 2. Global off-line robustness.

(10)에서 s는 탐색전략, e는 테스트 환경, $F_e^*(k)$는 k세대까지의 최적의 함수값, N_G는 모의실험에 사용된 최대 세대수를 뜻한다. 이 성능측정 방법은 주어진 환경에서 지정된 세대동안 최적 함수값들의 평균을 구하게 된다.

(10)은 국부적인 테스트 환경에서 얻게 되는 평가척도이다. 따라서 전체 테스트 환경(E)에 적용할 수 있는 평가척도로 전역 오프라인 강인성(global off-line robustness) $\overline{U}_E^*(s, N_G)$을 정의하며 이는 정규화 오프라인 성능 $\overline{u}_e^*(s, N_G)$로부터 계산된다[10][12].

$$\overline{U}_E^*(s, N_G) = 100[1 - \underset{e \in E}{\text{ave}} \{ \overline{u}_e^*(s, N_G) \}] \quad (11)$$

여기서 ave{ · }는 평균을 의미하는 연산자이고, 정규화 오프라인 성능은 다음과 같이 계산된다[10][12].

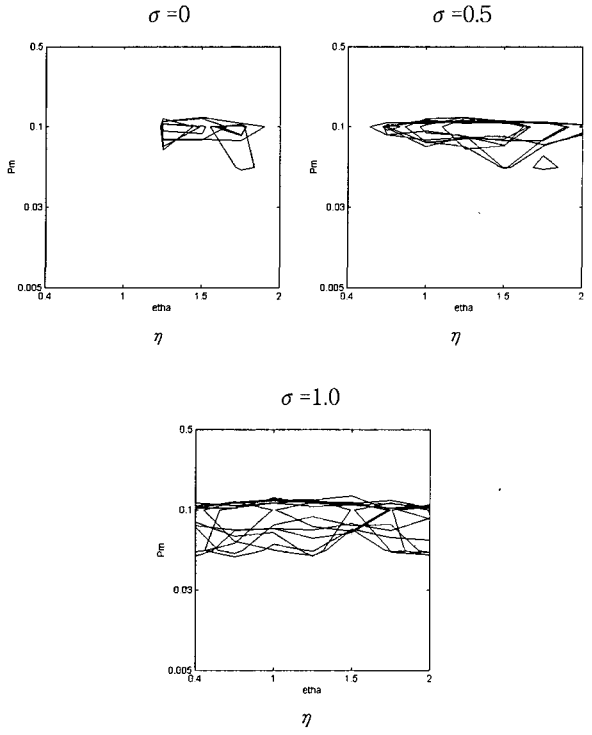
$$\overline{u}_e^*(s, N_G) = \frac{u_e^*(s, N_G) - u_e^*(s, N_G)_{min}}{u_e^*(rs, N_G) - u_e^*(s, N_G)_{min}} \quad (12)$$

(12)에서 $u_e^*(s, N_G)_{min}$은 탐색성능 중에서 최소값을 뜻하며, $u_e^*(rs, N_G)$는 무작위 탐색법의 오프라인 성능을 나타낸다.

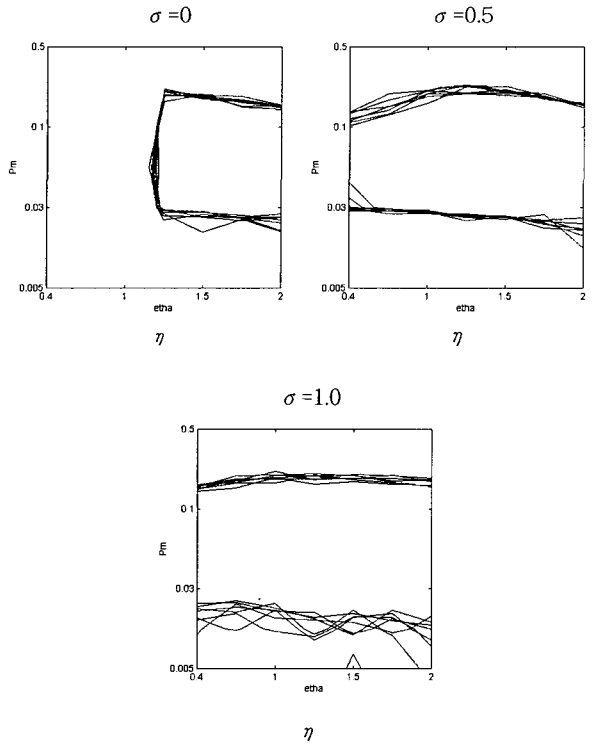
2. 최적 제어 파라미터

최적 제어 파라미터의 도출은 주어진 테스트 환경에서 성능평가 방법을 이용하여 전역 오프라인 강인성을 계산하는 일련의 작업을 요구한다. 가능한 모든 제어 파라미터 영역에서 이를 계산하는 것은 현실적으로 대단히 어렵기

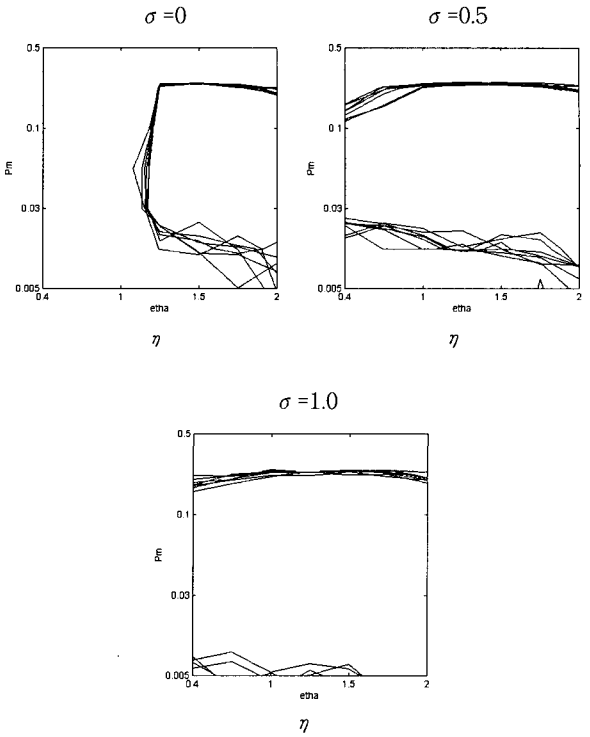
때문에 여기서는 제한된 범위만을 고려하였다. 따라서 N ∈ {10, 20, 30}는 3단계, η_i의 설정에 필요한 평균 η ∈ {0.4, 0.7, 1.0, 1.3, 1.5, 1.7, 2.0}는 7단계, 표준편차 σ ∈{0, 0.5, 1.0}는 3단계로 하였다. 한편 P_c∈{0.4, 0.5, 0.6, 0.7, 0.8,



(a) N= 10



(b) N= 20



(c) N= 30

그림 3. 제어 파라미터의 최적 영역(98%).
Fig. 3. Optimal region of the control parameters(98%).

0.9, 1.0}는 7단계, P_m∈{0.005, 0.01, 0.03, 0.05, 0.1, 0.3, 0.5}은 7단계로 하였다. 계산되는 성능은 초기집단의 선택에 민감하게 되므로, 서로 다른 씨드(seed)로 알고리즘을 10번 실행하고 그 결과를 평균하여 영향을 최소화하였다.

그림 2는 η과 P_m을 축으로 $\overline{U}_E^*(s, N_G)$을 그린 것으로서 P_c에 대해서 평균한 것이다. 가로축의 서로 다른 그림은 σ를 매개변수로 한 경우이며, 세로축의 그림들은 N을 매개변수로 한 것이다. 이를 바탕으로 그림 3은 98%의 강인성을 가지는 집합을 그린 것이다. 서로 다른 등고선들은 P_c의 변경에 따른 그림이다. 그림에서 최적의 제어 파라미터는 등고선을 포함한 내부 영역에 존재하며 안쪽으로 갈수록 강인성이 커짐을 유추할 수 있다.

N은 작을수록 연산부담이 줄지만 유전적 부동 현상의 원인을 제공하고, 어느 범위 내에서 클수록 탐색성능은 개선되나 비례해서 연산시간이 길어진다. 이런 이유로 수렴속도와 연산부담 사이에서 적절히 타협된다. 그림 3에서 N이 클수록 등고선 면적이 넓어지는 것은 최적 영역이 확대됨을 뜻하고, 이는 탐색성능이 개선됨을 의미한다.

정규분포 $N(\eta, \sigma^2)$를 따르도록 발생되는 η_i 또한 성능에 영향을 미치는 변수로서 η_i가 너무 작으면 탐색이 침체되고, 너무 크면 탐색이 불안정하게 된다. 그림에서도 σ 값이 작을 때는 최적의 η은 특정 범위내에 존재하고, 클수록 그 범위가 넓어짐을 알 수 있다. σ = 0인 경우에는 N과는 관계없이 η = 1.2-1.9 사이에서 최적 구간이 발견되는

데 앞서 수행한 연구 결과[10]와 유사성을 보인다. σ 값이 클수록 영역이 넓어지는 것은 탐색의 확대로 얻은 결과이나 이로 인해 부적합한 영역을 탐색하는 기회가 증대되었다. 비록 복구전략을 채용하게 된다 하더라도 σ>0.8은 바람직스럽지 못함을 경험적으로 알 수 있었다.

일반적으로 P_c는 낮으면 새로운 개체 발생이 적게 되어 탐색이 침체되고, 반대로 높으면 더 나은 개체를 찾는 것보다 더 빠른 속도로 좋은 개체를 파괴하는 경향이 있다. 그림에서 보면 선택된 단계내의 P_c는 N과 σ가 작을 때 그 영향이 확대되는 것 같았으나 커지면 그다지 민감하지 못하였다.

유전알고리즘에 도입되는 전역탐색의 양은 대부분 P_m에 의해 조절되는데 낮게 설정하면 지역탐색에서 벗어나기가 어렵고, 반대로 너무 높게 설정하면 불안정한 탐색을 주게 된다. 기대한 바와 같이 최적의 P_m은 특정 범위 내에 존재함을 확인할 수 있다. N이 클수록 그 영역이 확대되는 것은 당연한 결과라 할 수 있다.

Ⅳ. 시뮬레이션

제안된 RCGA의 성능을 확인하기 위하여 두 최적화 문제를 선택하고 시뮬레이션을 실시하였다. 제어 파라미터로 N= 10, η = 1.5, σ = 0.5, P_c= 0.5, P_m= 0.07을 사용하였다. 특히 제약조건이 수반되는 경우에는 비제약성 최적화 문제로 변환하기 위해 벌점전략을 사용하였다.

1. 예제 1

다음 함수의 최소점을 찾는 문제가 고려되었다[4].

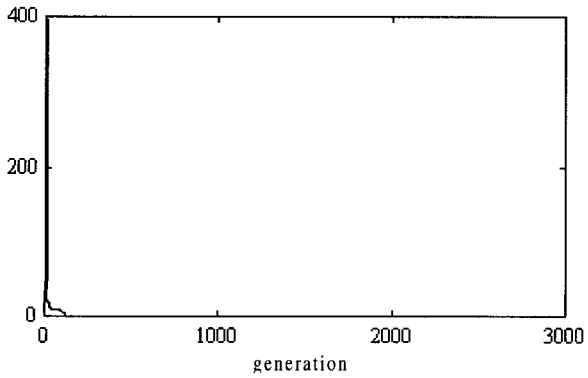
최소화 :

$$F_1(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1) \quad (13a)$$

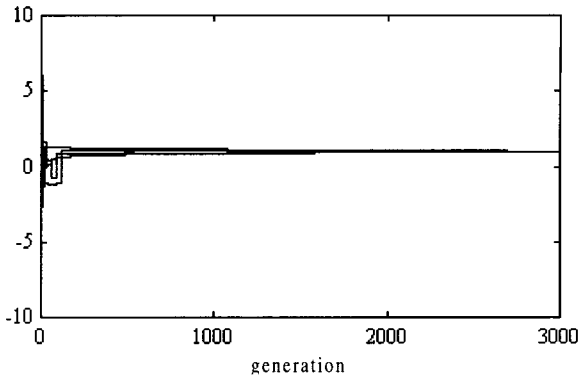
영역 제한조건 :

$$-10 \le x_i \le 10,\ i \in [1, 4] \quad (13b)$$

이 함수는 $\mathbf{x}$= $[1\ 1\ 1\ 1]^T$에서 전역해를 가지며 함수값은 0이다. 탐색영역으로는 영역 제한조건을 사용하였다. 그림 4는 RCGA가 약 2800세대에서 근사해 $\hat{\mathbf{x}}$= [0.993 0.986 1.007 1.014]^{T}를 찾고 있는 것을 보여주고 이 때의 함수값은 0.181×10^{-3}이다.



(a) 목적함수



(b) 해 벡터

그림 4. 함수의 최적화.

Fig. 4. Function optimization.

2. 예제 2

다음은 PI 제어기의 파라미터를 최적 동조하는 문제가 고려되었다.

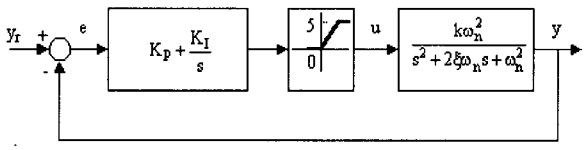


그림 5. PI 제어시스템.

Fig. 5. PI control system.

플랜트 파라미터는 ζ = 0.9, ω_n= 1, k= 1이고 다음과 같은 제약조건이 고려되었다.

$$g_1(\mathbf{x}) = 10 - M_p(\mathbf{x}) \ge 0 \quad (14a)$$

$$g_2(\mathbf{x}) = 1.1 - t_r(\mathbf{x}) \ge 0 \quad (14b)$$

$$g_3(\mathbf{x}) = 5.5 - t_s(\mathbf{x}) \ge 0 \quad (14c)$$

여기서 $\mathbf{x}$=$[K_P\ K_I]^T \in \mathbb{R}^2$이고 M_p, t_r, t_s는 각각 오버슈트, 도달시간, 정정시간(5%)이다. 시스템의 응답이 이 조건에 부합하면서 최적의 응답특성을 갖도록 RCGA는 K_P와 K_I를 조정하는데, 이때 사용된 목적함수는 다음과 같다.

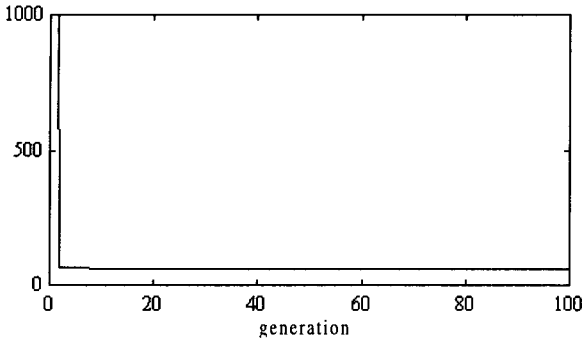
$$J(x) = \int_0^{t_f} (30e^2 + u^2)\ dt \quad (15)$$

여기서 t_f는 이후의 적분값이 무시되어도 좋을 정도로 충분히 큰 값으로 선택된다. 결국 주어진 문제는 시스템의 방정식과 제약조건을 동시에 만족하면서 목적함수를 최소로 하는 $\mathbf{x}$를 구하는 문제로 귀착된다. 모의진화 도중 잠정적인 해가 (14)의 제약조건을 위반 할 경우에 그 위반 정도에 따라 벌점을 부과하여 제약조건이 없는 최적화 문제로 변환하였다.

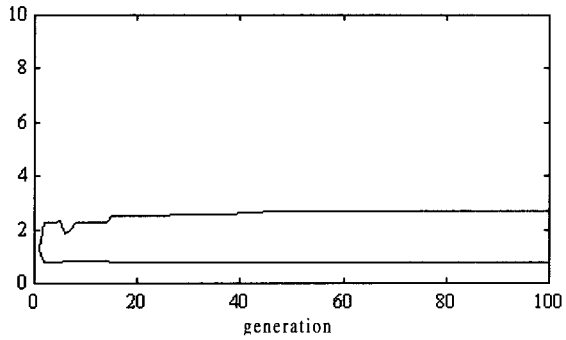
$$F(\mathbf{x}) = J(\mathbf{x}) + P(\mathbf{x}) \quad (16a)$$

여기서 P : $\mathbb{R}^2 \rightarrow \mathbb{R}$는 벌점함수이고 다음 조건을 만족한다.

$$P(\mathbf{x})=\begin{cases} 0 & , \mathbf{x}\text{가 적합한 해일 경우} \\ \sum_{j=1}^{3} w_j g_j^2(\mathbf{x}) & , \mathbf{x}\text{가 부적합한 해일 경우} \end{cases} \quad (16b)$$



(a) 목적함수

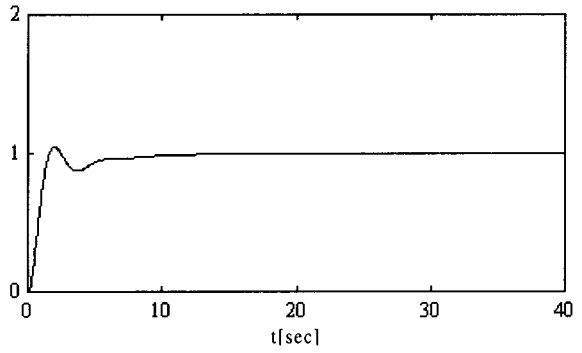


(b) 파라미터

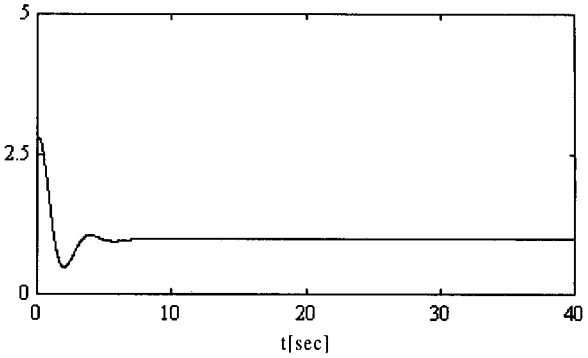
그림 6. PI 제어기의 최적 파라미터 동조.
Fig. 6. Optimal parameter tuning of the PI controller.

여기서 벌점상수로는 w_1= 100, w_2= 10, w_3= 100이 선택되었다.

그림 5는 탐색영역을 $0 \le K_P, K_I \le 10$으로 하였을 때의 파라미터의 동조 과정을 보여주고 있다. RCGA는 약 45세대에서 최적 해 $\hat{K}_P$= 2.723, $\hat{K}_I$= 0.769를 찾고 있고 이 때의 목적함수값은 61.877이다.



(a) 출력



(b) 제어입력

그림 7. PI 제어시스템의 계단 응답.
Fig. 7. Step response of the PI control system.

그림 7은 최적으로 동조된 PI 제어시스템의 계단응답을 그린 것으로 M_p= 5.0[%], t_r= 1.03[sec], t_s= 5.5[sec]이 되어 앞의 설계사양을 모두 만족한다.

V. 결론

본 연구에서는 제약조건이 수반되는 복잡한 최적화 문제를 해결하는 한 도구로서 RCGA를 제안하였다. 유전 연산자를 개선하였고, 최적 제어 파라미터를 얻기 위하여 전역 오프라인 강인성을 평가하는 광범위한 시뮬레이션을 실시하였다. 그 결과 문제와 환경에 따라 적절히 선택할 수 있는 최적 영역을 도출하였다. RCGA의 성능을 확인하기 위하여 다변수 함수의 최소화 문제와 PI 제어기의 최적 동조 문제에 적용하였고, 특히 제약조건이 수반되는 경우에는 벌점전략을 채용하였다. 시뮬레이션 결과 만족스러운 결과를 얻을 수 있었다.

참고문헌

[1] J. H. Holland, Adaptation in natural and artificial systems, The University of Michigan Press, Michigan, 1975.

[2] L. J. Fogel, "Extending communication and control through simulated evolution, bioengineering- an engineering view", *Proc. Symp. Engineering Significance of the Biological Sciences*, G. Bugliarello(ed.), San Francisco Press, Inc., San Francisco, pp. 286-304, 1968.

[3] H. P. Schwefel, Numerical optimization for computer models, John Wiley, Chichester, UK, 1981.

[4] Z. Michalewicz, Genetic algorithm + data structures = evolution programs, Springer-Verlag, Inc., Heidelberg, Berlin, 1996.

[5] M. Gen and R. Cheng, Genetic algorithms and engineering design, John-Wiley & Sons, Inc., N.Y., 1997.

[6] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptation systems", *PhD. Disser-*

tation, The University of Michigan, Ann Arbor, Michigan, 1975.

[7] D. E. Goldberg, Genetic algorithms in search, optimization and machine learning, Addison-Wesley Publishing Co. Inc., N. Y., 1989.

[8] D. E. Goldberg, "Sizing populations for serial and parallel genetic algorithms", *Proc. 3rd Int. Conf. on Genetic Algorithms and Their Applications*, Arlington, VA, pp.70-79, 1989.

[9] K. Krishnakumar, "Micro-genetic algorithms for stationary and non-stationary function optimization", *SPIE, Intelligent Control and Adaptive Systems*, Vol. 1196, pp. 289-296, 1989.

[10] G. Jin, "Intelligent fuzzy logic of processes with time delays", *PhD Thesis*, Univ. of Wales, Cardiff, UK, 1995.

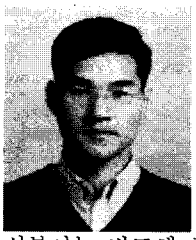
[11] L. J. Eshelman, R. A. Caruana, and J. D. Schaffer, "Bases in the crossover landscape", *Proc. 3rd Int. Conf. on Genetic Algorithms*, J. Schaffer(Ed.), Morgan Kaufmann Publishers, LA, pp.10-19, 1989.

[12] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms", *IEEE Trans.* Syst. Man Cybern., Vol. SMC-16, pp. 122-128, 1986.

[13] 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.

진 강 규

1953 년 10월 12일생. 1977년 한국해양대학교 기관학과(공학사). 1985년 Florida Institute of Technology 전기·전자·컴퓨터공학과(공학석사). 1996년 University of Wales, Cardiff 전기·전자·시스템공학과(공학박사). 1981년~현재 한국해양대학교 자동화·정보공학부 교수. 관심분야는 퍼지제어, 유전알고리즘.



주 상 래

1971년 11월 16일생. 1995년 한국해양대학교 제어계측공학과 졸업(공학사). 1999년 한국해양대학교 제어계측공학과(공학석사). 1999년~현재 코닉시스템 응용 소프트웨어 개발부 연구원. 관심분야는 반도체 공정 제어, 최적제어, 유전알고리즘.