

GA에 기반을 둔 제한된 수식의 프로그램 발견을 위한 최적 파라미터 조사

황수철*, 조상엽**

요약

유전자 알고리즘을 이용하여 최적화 분야의 문제를 해결할 때 교배율과 돌연변이율과 같은 파라미터는 문제 해결 시스템의 성능에 많은 영향을 준다. 본 연구에서는 제한된 수식이 주어졌을 때 이 수식을 계산할 수 있는 가상 머신의 프로그램을 찾는 문제에서 최적의 교배율과 돌연변이율을 조사하는 방법을 제안한다. 실험에서 열개의 파라미터 조합을 일곱 부류의 수식에 적용하여 해가 발견되기까지의 평균세대수를 비교하였다. 그 결과 네 번째 파라미터 조합이 가장 좋은 결과를 보였다.

Optimal Parameter Investigation for Finding Programs of the Limited Expressions Based on Genetic Algorithms

Su-Chul Hwang*, Sang-Yeop Cho**

ABSTRACT

In order to solve the optimal problems with genetic algorithms, the parameters such as the ratio of the crossover and mutation give an impact to the problem solving systems. This paper proposes the method of investigating the best crossover and mutation parameters in the problem of finding the program of virtual machine that can be calculated them, given the limited expressions. In experiment, we apply the ten combinations of parameters to the seven categories of the expressions and compare the average generations until finding solutions. The result show that the forth case of combination of parameters is the best solution.

Key Words : genetic algorithms, parameter investigations, crossover, mutation, expressions

* 인하공업전문대학 컴퓨터시스템과

** 청운대학교 인터넷학과

· 제1저자(First Author) : 황수철 · 교신저자(Correspondent Author) : 조상엽

· 접수일(2010년 2월 17일), 수정일(1차 : 2010년 3월 19일), 게재확정일(2010년 3월 23일)

I. 서 론

유전자 알고리즘이란 생물의 유전과 진화 메카니즘을 공학적으로 모델화하여 다양한 문제 해결이나 시스템의 학습 등에 응용하려는 최적화 기법이다[1, 2]. 이런 기법을 사용하는 대부분의 연구물들은 주어진 문제에 대한 최적의 솔루션을 찾기 위해서 생물학적 진화 과정을 모방한다. 즉, 주어진 문제에 대한 해답을 소유할 초기 유전자 집단을 만들고 이들 유전자 중에서 우성의 유전자를 선택하여 교배와 돌연변이 등의 연산을 수행함으로써 유전자 자손을 생산해 낸다. 만족할 만한 유전자가 생성될 때까지 이 과정을 반복하여 궁극적으로 원하는 솔루션 찾아낸다. 이때 솔루션을 빠르게 찾아내는데 영향을 주는 주요 요소들은 유전자 선택 방법, 교배율, 돌연변이율 등이 있다.

한편, 문제 비의존 탐색 알고리즘인 유전자 알고리즘은 컴퓨터 제어 시스템, 게임 이론, TSP 문제 등과 같은 분야[3, 4, 5]뿐만 아니라 수학 함수식에 대한 일련의 명령어를 구하는 기호 문제 최적화 분야[6]에서도 매우 유용하다.

앞서 언급했듯이 유전자 알고리즘을 이용하여 최적화 분야의 문제를 해결할 때 교배율과 돌연변이율 같은 파라미터는 문제 해결 시스템의 성능에 많은 영향을 준다. 본 연구에서는 제한된 수식이 주어졌을 때, 수식을 계산할 수 있는 유전자 알고리즘 적용의 스택 가상 머신[7]에서 프로그램(일련의 명령어)을 구하는데 최적의 교배율과 돌연변이율이 무엇인지 실험을 통해서 찾고자 하였다.

용하여 가상 머신으로 하여금 그 수식의 결과 값을 계산해 내는 일련의 명령어 즉, 프로그램을 구하는 범주로 하여 이 수식 리스트 부류에 가장 적합한 교배율과 돌연변이율을 무엇인지를 찾고자 한다.

이 실험에서는 표 1처럼 각 수식은 문제 해결에 대한 복잡도를 줄이기 위해서 항이 3개 이하 이면서 계수를 포함하지 않으며, 3개 이내의 변수나 상수를 사용하면서 동시에 각 변수가 지니는 차수의 전체 합이 8 이하가 되도록 제한하였다.

표 1. 수식 리스트
Table 1. List of Expressions

수식번호	수식
exp01	$f = x \cdot y + y^2 + z$
exp02	$f = x^3 + y^2 + z$
exp03	$f = x^5 + 4$
exp04	$f = x^4 + y^4$
exp05	$f = x^2 + x \cdot y^2 + z^2$
exp06	$f = x^2 \cdot y + x \cdot y^2 + z^2$
exp07	$f = x^8$

2.2 스택 가상 머신

가상머신은 레지스터의 사용 없이 하나의 스택으로 구성된 0-주소 형태이며, 표 2처럼 5개의 간단한 명령어 집합을 갖고서 일련의 프로그램을 처리한다. 사용되는 명령어 집합은 표 2와 같다. 이 명령어는 간단하지만 다양한 함수식을 해결하는데 사용될 수 있다. 예들 들어, 스택 TOP에 임의의 값이 있다고 할 때 그 원소의 제곱 값($f = x^2$)을 구하고자 한다면 스택 머신은 다음과 같은 일련의 명령어를 실행할 수 있다.

II. 이론적 배경

2.1 문제 영역

본 연구에서는 추구하고자 하는 문제 영역은 표 1과 같은 부류의 식이 주어졌을 때 유전자 알고리즘을 적

명령어	—	DUP	MUL
stack	x	x x	x*x

이 일련의 명령(프로그램)은 스택의 TOP에 있는 값을 스택에 복사해 넣고 TOP에 있는 2개의 원소를 서로 곱하는 기능을 수행한다.

표 2. 명령어 집합
Table 2. a set of instructions

명령어	기능
DUP	스택의 TOP을 복사 (stack: 2 ==> 2 2)
SWAP	스택의 TOP에 있는 2개의 원소를 서로 교환 (stack: 2 5 ==> 5 2)
MUL	스택의 TOP에 있는 2개의 원소를 서로 곱함 (stack: 2 5 ==> 10)
ADD	스택의 TOP에 있는 2개의 원소를 서로 더함 (stack: 2 5 ==> 7)
OVER	스택의 2 번째 항목을 복사 (stack: 2 5 ==> 2 5 2)
NOP	No Operation

2.3 유전자 알고리즘을 적용한 스택 가상머신

일반적인 유전자 알고리즘은 그림 1과 같은 구조를 갖는다. 먼저 문제 영역에 해당하는 솔루션을 인코딩하여 염색체로 구성된 유전자 모집단을 생성한 후 각 염색체를 대상으로 적합도 계산한다.

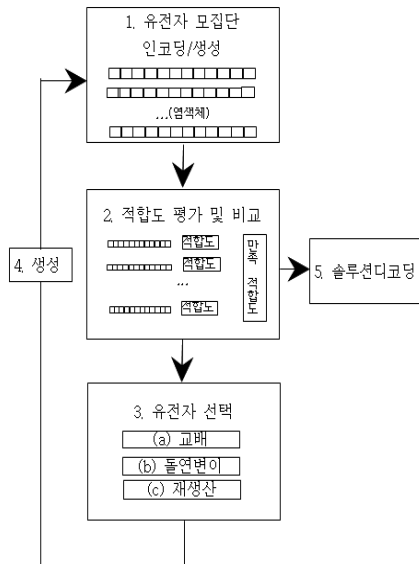


그림 1. 유전알고리즘
Fig. 1. genetic algorithm

그리고 적합도 값이 솔루션에 얼마나 근접했는지 비교하여 주어진 조건을 만족하지 않는다면 적합도 값이 큰 유전자들을 부모로 선택되게 하여, 지정한 교배율과 돌연변이율을 적용해 새로운 유전자 자식 집단을 생성해 낸다. 이 과정을 적합도 값이 특정 조건을 만족할 때 까지 반복 수행함으로써 주어진 문제에 대한 솔루션을 발견하고 이를 디코딩함으로써 우리가 원하는 답을 구한다.

이러한 유전자 알고리즘을 이용하여 주어진 문제를 해결할 때 선택방법, 교배점, 교배율, 돌연변이율 등과 같은 여러 파라미터들이 중요한 역할을 한다. 본 연구에서는 룰렛 선택 방법과 단일 교배점을 사용했을 때 어떤 조합의 교배율과 돌연변이율이 보다 좋은지를 모색하고자 하였다.

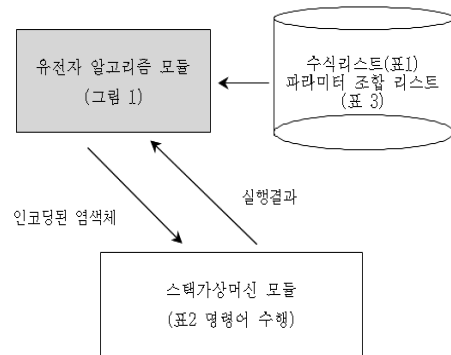


그림 2. 스택 가상머신
Fig. 2. stack virtual machine

그림 2는 위에서 언급한 유전자 알고리즘을 수식에 대한 스택가상머신 프로그램에 구하데 적용한 전체적인 구성도이다. 처음에는 한 수식에 파라미터 조합 중 하나인 교배율과 돌연변이를 설정하며, 그 다음 이 값을 가지고 유전자 알고리즘이 적용된 스택가상머신을 실행한다. 즉, 유전자 알고리즘의 모집단에서 적합도 평가를 하여 만족되는 솔루션이 없다면 교배 및 돌연변이, 재생산 과정을 거쳐서 새로운 유전자 모집단을

만들고, 이 모집단내의 인코딩된 염색체를 스택가상머신 모듈에서 실행되도록 한 후에 다시 적합도 평가를 실행한다.

가. 솔루션 인코딩

본 연구에서는 인코딩시 염색체 크기를 10으로 하여 표1의 제한된 수식에 대한 프로그램을 구할 수 있도록 하였다. 염색체의 크기를 10 이상으로 한다면 복잡한 수식에 대한 프로그램을 구할 수 있지만 시간이 다소 소요된다.

스택에서 사용되는 명령어는 표 3처럼 숫자로 인코딩되며, 랜덤 함수를 통해 크기가 10인 유전자 모집단이 생성된다.

예를 들어, 그림 3처럼 일련의 명령어 [DUP SWAP MUL ADD ADD SWAP NOP MUL ADD SWAP]는

0 1 2 3 3 1 5 2 3 1인 염색체 솔루션

으로 인코딩된다.

표 3. 명령어와 인코딩

Table 3. instructions and encoding

명령어	인코딩
DUP	0x00
SWAP	0x01
MUL	0x02
ADD	0x03
OVER	0x04
NOP	0x05

나. 적합도 평가

스택가상머신을 통해 건네진 염색체에 대한 적합도 값 평가를 위해서 3층 구조의 평가 방식을 이용하였다. 즉, 스택 가상머신이 오류 없이 수행되었고 1개의 값만이 스택에 남았는지 그리고 스택 가상머신의 실행 결과가 주어진 수식에 대한 기댓값과 일치하는지 비교하였다.

만약 3 개의 값이 부합된다면 실행된 염색체는 주어진 수식에 대한 솔루션이므로 디코딩하여 일련의 명령어 즉, 프로그램과 함께 솔루션이 발견된 유전자 세대 값을 얻게 된다.

다. 선택

유전자 선택은 적합도 값을 근거로 염색체를 미래 유전자 모집단을 구성하는 것으로, 최적의 염색체만을 사용한다면 솔루션 공간의 다양성 부족으로 인해 매우 제한적일 수 있다. 반면에, 임의로 선택 작업이 이루어진다면 최적의 솔루션을 보장할 수 없게 된다.

최적의 솔루션을 구하기 위한 다양한 선택 방법이 있지만 본 연구에서는 솔루션의 다양성과 최적의 솔루션을 고려하여 염색체의 적합도가 클수록 다음 세대에 전파될 가능성이 높은 룰렛-휠 선택 방법을 사용하였다.

라. 재조합

선택 과정을 거친 염색체들은 교배와 돌연변이 연산이 적용되어 새로운 유전자 모집단으로 생성된다. 교배에는 단일 교배점과 다중 교배점이 있는데, 본 연구에서는 단일 교배점을 사용하였으며, 교배율 0.75~0.95 사이의 값을 적용하였다.

한편 돌연변이율은 기존 염색체의 특성 계승을 고려하여 0.01과 0.05로 설정하였다.

마. 스택가상머신

스택가상머신은 인코딩된 염색체를 디코딩하여 해당되는 명령을 실행 한다 이때, 수식의 매개변수로 건네받은 값이 사용된다. 가상머신 실행 결과 스택은 1 개의 계산 결과 값을 갖거나 1개 이상의 값을 보유할 수 있다.

이 값들은 유전자 알고리즘 모듈로 전달되어 적합도 평가에서 솔루션을 발견하는데 활용된다.

III. 실험

3.1 실험 환경 및 방법

하드웨어의 특성을 최대한 줄이기 위해서 (프로그램에 랜덤함수 사용하기 때문) 동일한 PC인 CPU 펜티엄 2.53MHz, 윈도우 xp 상에서 Visual C++ 개발 환경의 종료 없이 실험하였다.

실험 방법으로는 앞서 언급한 그림 2의 구성도 상에서 표 1의 각각의 수식에 대해 표 4의 파라미터 조합 10개를 적용하였으며, 객관성 있는 결과를 얻기 위해서 각 수식에 적용된 교배율과 돌연변이율을 가지고 알고리즘 모듈과 가상머신 모듈을 10회씩 실행하였다. 그림 3은 이 과정을 보인 것이다.

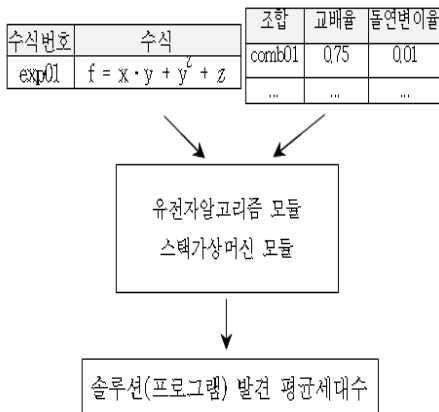


그림 3. 실험 과정

Fig. 3. processing of experiment

이 과정을 매년 실행한 후에는 수식에 대한 솔루션 즉 수식에 대한 프로그램(일련의 명령어)과 함께 솔루션을 발견한 세대수가 구해지는데, 본 연구에서는 한 수식에 대해서 10회 실행했을 경우 솔루션을 발견한 평균 세대수를 가지고 다른 조합 및 수식들과 비교 분석하였다.

매번 솔루션이 구해지지 않는 경우는 없었으나, 표 1에서 제시한 수식보다 복잡해지거나 염색체의 크기

를 작게 할 경우 솔루션이 구해지지 않는 경우도 발생하였다.

본 실험에서 사용한 교배율과 돌연변이율은 기존 연구에서 사용한 범주의 값으로 교배율은 0.5 단위로 5개를 설정하였고, 돌연변이율은 0.01과 0.05로 2개를 설정하였다. 표 4는 이들 값으로 구성되어 각 수식에 적용할 교배율과 돌연변이율에 대한 파라미터 조합 리스트를 보인 것이다.

표 4. 파라미터 조합 리스트

Table 4. list of parameter combination

조합	교배율	돌연변이율
comb01	0.75	0.01
comb02	0.75	0.05
comb03	0.80	0.01
comb04	0.80	0.05
comb05	0.85	0.01
comb06	0.85	0.05
comb07	0.90	0.01
comb08	0.90	0.05
comb09	0.95	0.01
comb10	0.95	0.05

3.2 실험 결과

표 5는 실험 결과로서 서로 다른 교배율과 돌연변이율을 각 수식에 대해 적용했을 때 솔루션을 발견한 평균 세대수를 보인 것이며, 그림 4는 이를 그래프로 나타낸 것이다.

예를 들어 표 5는 수식1($f = x \cdot y + y^2 + z$)에 조합 comb01(교배율:0.75, 돌연변이율:0.01)을 적용해 10회 실행했을 때 솔루션을 발견한 평균 세대가 15.7이다. 한편 조합 comb01을 수식1에서 수식7까지 적용했을 때 수식 3($f = x5 + 4$)이 솔루션을 가장 빠르게 발견하며, 수식 5($f = x2 + x \cdot y2 + z2$)의 솔루션을 가장 늦게 발견함을 알 수 있다.

표 5. 솔루션 발견까지 세대수

Table 5. average generations until finding solution

조합	수식1	수식2	수식3	수식4	수식5	수식6	수식7
comb01	15.7	365.8	9.2	3152.7	4787.9	769.6	13.5
comb02	19	357.1	16.1	4833.8	3272.6	1028.2	4.5
comb03	7	407.8	3.6	4386.2	4437.9	369.8	14.3
comb04	16.8	268.4	9	2370.5	4131	1036.2	7.9
comb05	17.6	210.3	702	2143	4320.6	566.3	16.2
comb06	12.9	469.7	12.8	2177.1	8053.9	1048.7	15
comb07	8.9	281.3	12.1	3608	3404.7	553.9	16.2
comb08	13	273.8	9.9	4005.2	5842.9	665.5	8.3
comb09	15.5	516.8	13.5	7671	5562.8	612.3	24
comb10	16.2	289.5	10.3	3564.6	6277.7	1288.6	14.7

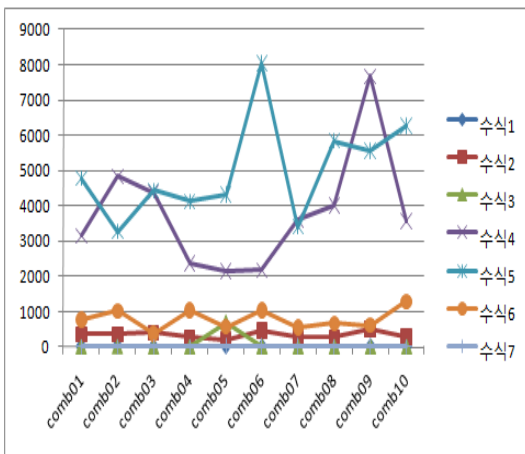


그림 4. 표 5에 대한 그래프

Fig. 4. graph for Table 5

표 6은 파라미터 조합인 교배율과 돌연변이율을 각 수식에 대해 적용했을 때 솔루션을 빨리 발견한 평균 세대수를 순서를 보인 것으로, 파라미터 조합별 7개 수식의 솔루션 발견 평균 세대 순서를 의미하기도 한다.

표 6. 솔루션 발견까지 세대수(교배율과 돌연변이율 적용)

Table 6. average generations until finding solution (crossover and mutation apply)

조합	수식1	수식2	수식3	수식4	수식5	수식6	수식7	평균
comb04	16.8	268.4	9	2370.5	4131	1036.2	7.9	1120.0
comb07	8.9	281.3	12.1	3608	3404.7	553.9	16.2	1126.4
comb05	17.6	<u>210.3</u>	702	<u>2143</u>	4320.6	566.3	16.2	1139.4
comb01	15.7	365.8	9.2	3152.7	4787.9	769.6	13.5	1302.1
comb02	19	357.1	16.1	4833.8	3272.6	1028.2	<u>4.5</u>	1361.6
comb03	<u>7</u>	407.8	<u>3.6</u>	4386.2	4437.9	<u>369.8</u>	14.3	1375.2
comb08	13	273.8	9.9	4005.2	5842.9	665.5	8.3	1545.5
comb10	16.2	289.5	10.3	3564.6	6277.7	1288.6	14.7	1637.4
comb06	12.9	469.7	12.8	2177.1	8053.9	1048.7	15	1684.3
comb09	15.5	516.8	13.5	7671	5562.8	612.3	24	2059.4

그림 5는 이를 그래프로 표현한 것이다. 여기서 파라미터 조합이 comb04(교배율:0.8, 돌연변이율: 0.05) 일 때 본 논문에서 제시한 7개 유형의 수식에 대해 가장 빠른 세대로 솔루션을 발견했으며, 파라미터 조합 comb07(교배율:0.9, 돌연변이율:0.05)이 그 뒤를 이었음을 보여 준다.

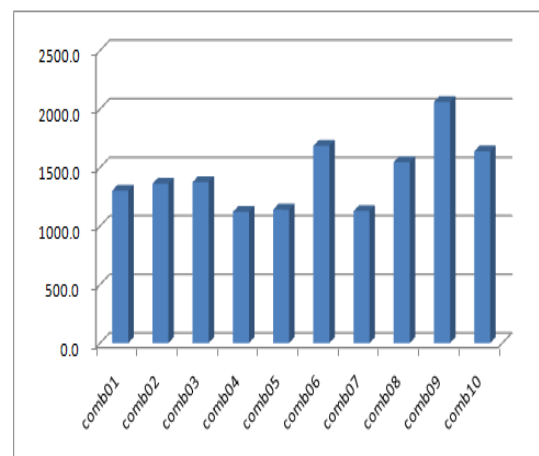


그림 5. 표 6에 대한 그래프

Fig. 5. graph for Table 6

한편, 표 7과 그림 6은 9가지 파라미터 조합 모두 수식에 적용했을 때 각 수식별 솔루션 발견 평균 세대수를 보인 것이다.

표 7. 솔루션 발견까지 세대수(모든 파라미터 적용)
Table 7. average generations until finding solution
(all parameter apply)

수식	수식1	수식2	수식3	수식4	수식5	수식6	수식7
평균 세대	15.8	382.3	88.7	4212.5	5565.8	882.1	15.0

이 자료는 수식 5와 수식 4의 경우가 솔루션을 발견하는데 많은 세대가 걸렸음을 보여 주며, 수식 1, 3, 7은 빠른 세대에서 솔루션이 찾아 졌음을 보여 준다.

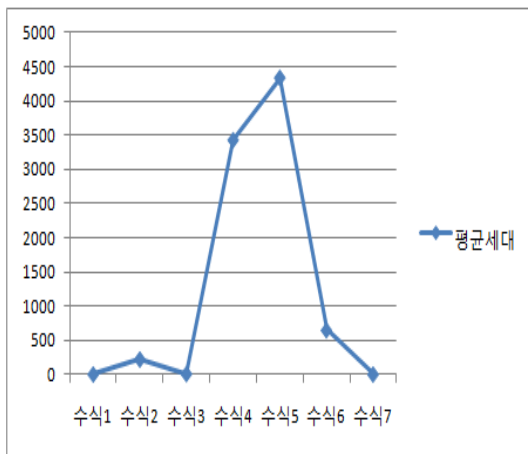


그림 6. 표 7에 대한 그래프
Fig. 6. graph for Table 7

IV. 실험 결과 분석

이번 실험을 통해서 비록 다른 파라미터 조합인 comb07(교배율:0.9, 돌연변이율: 0.01), comp05(교배율:0.85, 돌연변이율: 0.01)와는 다소 근소한

차이가 있지만, 파라미터 조합 comb04(교배율:0.8, 돌연변이율: 0.05)가 주어진 수식 전체에 대해서는 솔루션 발견 평균 세대 1120.0으로 최적의 교배율 및 돌연변이율을 알 수 있다.

아울러 표 6의 파라미터 조합별 7개 수식의 솔루션 발견 평균 세대와 표 7의 9가지 파라미터 조합에 대한 수식별 솔루션 발견 평균 세대의 전체 평균(1250.9)과 비교해 보았을 때 앞서 언급한 파라미터 조합 comb04(1120.0), comb07(1126.4), comb05(1139.4)이 우수함을 보였다.

한편, 표 6에서 밑줄 친 세대 값은 수식별 가장 우수한 교배율과 돌연변이율 조합을 의미하는데, 여기서 전체 주어진 수식에 대해서 최적의 교배율 및 돌연변이 조합인 comb04와 comb07이 제외되었다. 이렇듯 개별 수식과 전체 수식에 대해 차이를 보이는 것은 수식 4~6과 같은 유형이 다른 수식들에 비해 솔루션 발견 세대 값이 크기 때문이다.

V. 결 론

본 연구에서는 제한된 수식이 주어졌을 때 이 수식을 계산할 수 있는 가상 머신의 프로그램(일련의 명령어)을 찾는 최적화 문제에서 최적의 교배율과 돌연변이율이 무엇인지 실험을 통해서 구하였다.

이를 위해, 7가지 부류의 수식에 파라미터 조합 10개를 적용했을 때 솔루션을 발견하는 평균 세대수를 비교하여 보았다. 그 결과, comb07(교배율:0.9, 돌연변이율: 0.01), comb05(교배율:0.85, 돌연변이율: 0.01)와는 다소 근소한 차이가 있지만, 파라미터 조합 comb04(교배율:0.8, 돌연변이율: 0.05)가 일반적으로 최적의 파라미터 조합임을 확인하였다.

한편, 부수적으로 수식별로 우수한 교배율 및 돌연변이율도 확인하였다. 표 8은 이번 실험의 결과를 요약한 것이다.

표 8. 실험 결과 요약
Table 8. experiment summary

수식 유형	수식별 적용시 최적 파라미터 값	전체 수식 적용시 최적 파라미터 값
수식1 $f = xy + y^2 + z$	교배율: 0.8 돌연변이율: 0.01	교배율: 0.8 돌연변이율: 0.05
수식2 $f = x^3 + y^2 + z$	교배율: 0.85 돌연변이율: 0.01	
수식3 $f = x^5 + 4$	교배율: 0.8 돌연변이율: 0.01	또는 교배율: 0.9
수식4 $f = x^4 + y^4$	교배율: 0.85 돌연변이율: 0.01	
수식5 $f = x^2 + xy^2 + z^2$	교배율: 0.75 돌연변이율: 0.05	돌연변이율: 0.01
수식6 $f = x^2y + xy^2 + z^2$	교배율: 0.8 돌연변이율: 0.01	
수식7 $f = x^8$	교배율: 0.75 돌연변이율: 0.05	

이번 실험은 제한된 수식과 조합을 가지고 단일 교배점과 룰렛-휠 선택 기반으로 진행되었다. 추후 보다 일반적인 수식과 다양한 파라미터 조합을 가지고 다중 교배점에서 최적 파라미터를 구해내는 연구의 수행이 요구되며, 아울러 객주어진 수식에 대해 파라미터 조합을 객관성 있게 찾아내는 방법론이 함께 연구되어야 하겠다.

참고문헌

- [1] D. E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [2] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1993.
- [3] R. Hollstein, *Artificial Genetic Adaption in Computer Control Systems*, Ph.D diss, University of Michigan, 1971.
- [4] A. Mark, B. Sendhoff, and H. Wersing, "A Decision Making

Framework for Game Playing Using Evolutionary Optimization and Learning," in *The 2004 Congress on Evolutionary Computation*, Vol. 1, pp.373 - 380, June 19-23, 2004.

- [5] J. D. Wei, and D. T. Lee, "A New Approach to the Traveling Salesman Problem Using Genetic Algorithm with Priority Encoding," in *The 2004 IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 1457-1464, June 19-23, 2004.
- [6] M. T. Jones, *AI Application Programming*, Charles River media, 2005.
- [7] K. Stoffel, and L. Spector, "High-Performance, Stack-Based Genetic Programming", in *Proceedings of the First Annual Conference on Genetic Programming*, Stanford, USA, pp.224-229, July 28 - 31, 1996.
- [8] 이수철 외, "구간값 모호집합 사이의 유사도 측정", *한국 지식정보기술학회 논문지*, 제4권 제2호, pp.15-22, 2009.



황수철(Su-Chul Hwang)

1986년 중앙대학교 전자계산학과 학사
1988년 중앙대학교 대학원 전산과 석사
1993년 중앙대학교 대학원 전산과 박사
2004.8~2005.7 CSUS 방문교수

1991년~현재 인하공업전문대학 컴퓨터시스템과 교수
※ 관심분야: 인공지능, 지능형시스템, 인터넷/모바일 응용



조상엽(Sang-Yeop Cho)

1986년 한남대학교 전자계산학과(공학사)
1988년 중앙대학교 전자계산학과(이학석사)
1993년 중앙대학교 전자계산학과(이학박사)
1993년~1994년 중앙대학교 컴퓨터소프트웨어
연구소 객원연구원

1995년~현재 청운대학교 인터넷학과 교수
※ 관심분야: 인공지능, 퍼지이론, 페트리넷 응용, 인터넷