

## 유전 알고리즘 기반의 함수 최적화를 위한 자바 패키지 개발에 관한 연구

강 환 수(姜 煥 秀), 동양공업전문대학교 전산경영기술공학부

전화 : (02) 2610-1941 / 팩스 : (02) 2610-1859

강 환 일(姜 煥 一), 명지대학교 전기정보 제어공학부

전화 : (0335) 330-6476 / 팩스 : (0335) 321-0271

송 영 기(姜 煥 一), 현대정보기술

전화 : (02) 3483-4210 / 팩스 : (02) 3483-4199

### A Study on the Development Java Package for Function Optimization based on Genetic Algorithms

Kang Hwan Soo, School of Computer and Information Management Engineering,  
Dongyang Technical College, E-mail : hskang@dongyang.ac.kr

Hwan Il Kang, Division of Electrical & Information Control Eng., Myongji Univ.,  
E-mail: hwan@wh.myongji.ac.kr

Young Gi Song, Research institute of Hyundai Information Technology Co. Ltd.,  
E-mail: bysie@shinbiro.com

#### Abstract

Many human inventions were inspired by nature. The artificial neural network is one example. Another example is Genetic Algorithms(GA). GAs search by simulating evolution, starting from an initial set of solutions or hypotheses, and generating successive "generations" of solutions. This particular branch of AI was inspired by the way living things evolved into more successful organisms in nature.

To simulate the process of GA in a computer, we must simulate many times according to varying many GA parameters. This paper describes the implementation of Java Package for efficient applications on Genetic Algorithms, called "JavaGA". The JavaGA used as a application program as well as applet provides graphical user interface of assigning major GA parameters.

#### I. 서론

60년대부터 시작된 유전 알고리즘은 90년대에 들어와 그 관심이 높아지고 있으며, 결정론적인 방법으로 해결하지 못한 난제에 적합한 알고리즘으로 그 응용 분야를 확장하고 있다. 컴퓨터 과학의 대표적인 난제군인 NP-Complete 문제군의 최적화 버전들이 좋은 응용 대상의 예이다.

본 논문에서는 이러한 유전자 알고리즘을 사용한 일반적인 문제에 바로 이용할 수 있도록 자바 클래스 패키지를 개발하였다. 즉 본 논문에서 구현한 유전 알고리즘은 자바 언어를 사용한 그래픽 사용자 인터페이스를 이용하여 주요 파라미터를 쉽게 수정이 가능하며, 유전 알고리즘의 진행과정을 파악할 수 있도록 하여, 유전 알고리즘을 쉽게 이용할 수 있도록 하였다. 또한 이 유전 알고리즘 프로그램은 인터넷 상의 월드와이드 웹에서도 이용할 수 있도록 구현하였다.

#### II. 유전 알고리즘

생물학에서 유전 물질은 DNA, 즉 염색체로 구성된 다. 개체들은 교차에 의해 염색체를 부분 결합하고 돌

연변이에 의해 미소한 변화가 된 새로운 염색체를 가진 새로운 개체들을 만들어내고, 개체들은 환경에 적응하기 유리한 정도에 따라 선택적으로 번성한다. 유전 알고리즘의 기본 구조는 이러한 생물의 진화 과정을 문제 해결 과정으로 옮겨 놓은 것이다.

문제 해결상의 임의의 해는 유전 알고리즘이 이해하는 형태로 표현되어 염색체라 부른다. 여러 개의 염색체가 구성되어 하나의 개체가 형성되고, 이러한 개체의 모음이 하나의 사회를 이루듯이 유전 알고리즘에서는 대부분의 경우 정해진 수의 염색체 집단을 운영하는데 이를 해집단(population)이라 한다. 유전 알고리즘에서는 염색체상의 각 인자인 유전자(Gene)가 최소 단위가 된다.

```

n 개의 초기 염색체 생성;
repeat {
  for i = 1 to k {
    두 염색체 p1, p2 선택;
    offspringi = crossover(p1, p2);
    offspringi = mutation(offspringi);
  }
  offspring1, ..., offspringk 을 population내의
  k 개의 염색체와 대치;
} until(정지 조건 만족);
남은 해 중 최상의 염색체를 return;
    
```

그림 1 전형적 유전 알고리즘

그림 1은 전형적인 유전 알고리즘의 구조를 보인다. 유전 알고리즘은 대부분 정해진 수의 해로 구성되는 해집단을 갖는다. 이 그림에서 해집단의 해의 수는 n이다. 먼저 n개의 해를 임의로 생성한다. 이 n개의 해집단으로부터 k개의 새로운 해를 만들어 내는데 각각의 해는 선택(selection), 교차(crossover), 변이(mutation)의 단계를 거쳐 만들어지며, 이 결과인 k개의 해는 해집단 내의 k개의 해와 대치된다. 이러한 과정을 임의의 정지 조건이 만족될 때까지 수행한 후 해집단에 남은 해 중 가장 좋은 해를 답으로 삼는다.

### III. 자바 패키지 개발

본 논문에서는 유전 알고리즘의 효과적인 활용을 위하여 자바 언어를 이용하여 유전 알고리즘을 구현하였다. 본 논문에서 구현한 유전 알고리즘 패키지는 JavaGA라 지칭하며, 이는 일반 응용 프로그램과 애플릿으로 모두 사용 가능하다. 본 장에서는 이 JavaGA의 효용성을 설명하기 위하여 JavaGA를 구성하는 클래스의 설명을 유전 알고리즘 구현에서 제시하고, 이 JavaGA를 이용한 실험결과를 JavaGA 활용결과에서

살펴보도록 한다.

본 논문에서는 유전 알고리즘을 쉽게 활용할 수 있는 자바 패키지의 구현에 그 목적을 두었다. 본 논문에서 구현한 JavaGA는 일반 응용프로그램과 애플릿으로 동시에 사용될 수 있도록 하였으며, 특히 유전 알고리즘을 이용한 문제의 해를 찾기 위하여 여러 번의 시행을 돕도록 유전 알고리즘 매개변수의 수정이 용이하도록 하였다. 본 JavaGA에서 구현한 클래스는 모두 11개로 구성된다.

본 JavaGA에서는 현재 선택 방식은 룰렛 휠 방식을 이용하고 있으며, 염색체의 유전자 표현은 이진수 표현(Binary Encoding)과 십진수 표현 방법을 모두 제공한다. 염색체를 구성하는 유전자의 수는 사용자 인터페이스를 통하여 쉽게 수정이 가능하다. 교차 연산은 일점(one-point) 교차를 이용하며, 변이 방법은 전형적 변이 방법으로, 모든 염색체의 유전자마다 난수를 발생하여 이 난수가 지정된 변이 확률보다 작으면 새로운 난수로 유전자의 값을 변이시키는 방법을 이용한다. 이 때 이용되는 변이 확률은 화면에서 쉽게 수정이 가능하다. 각 세대에서 교차연산과 변이연산에 의하여 새로이 생성된 염색체는 이전 세대의 모든 염색체를 대체하는 방법인 세대형 대체 방법을 사용한다.

JavaGA가 구성하는 클래스는 모두 11개로 다음과 같은 계층구조를 갖는다. 클래스의 계층구조는 간단하며, 클래스 GA와 ResultsCanvas가 클래스 AppCanvas를 부모로 상속받으며, 클래스 AppCanvas는 부모로 클래스 Canvas를 상속받는다. 클래스 RunGA는 애플릿으로도 이용할 수 있도록 클래스 Applet을 부모로 하며, 주요 유전자 패러미터를 입력하는 스크롤 콤포넌트를 구성하기 위한 클래스 ParamPanel은 클래스 Panel을 부모로 한다. 나머지 클래스는 모두 클래스 Object를 상속받는 클래스이다.

상위 클래스	상위클래스	클래스명
Canvas	AppCanvas	GA
Canvas	AppCanvas	ResultsCanvas
	Applet	RunGA
	Panel	ParamPanel
		Generation
		Chromosome
		RealRandom
		Results
		EvalFunc
		GAFuncGraph

그림 2 JavaGA class 계층구조

본 논문에서 구현한 JavaGA가 구성하는 클래스를

살펴보면, 염색체를 표현한 클래스 Chromosome, 각 세대를 표현한 Generation 등 모두 11개이다. 다음은 JavaGA의 11개 클래스의 기능 설명이다.

클래스명	기능
GA	유전 알고리즘을 총괄하는 클래스로 유전 알고리즘에 필요한 매개변수를 지정하도록 하며, 직접 결과를 표시하는 기능을 담당
Generation	각 세대의 염색체를 관리하며 다음 세대를 위한 선택, 교차, 변이 연산을 시행하여 새로운 세대의 염색체를 생성
Chromosome	염색체의 기능을 담당하는 클래스로 유전자의 표현은 이진법과 십진법을 모두 표현가능
RunGA	일반 응용프로그램이나 애플릿으로 실행 가능한 실행 클래스
Results	각 세대의 생성 결과를 자료구조로 표현
ResultsCanvas	각 세대의 생성 결과를 캔버스에 그리는 기능을 담당
ParamPanel	유전 알고리즘의 주요 파라미터를 표현하는 패널
AppCanvas	캔버스의 효과적인 그림을 위한 클래스
RealRandom	다양한 난수를 발생시키는 클래스
EvalFunc	함수의 최대값을 구하는 문제에 이용하는 함수 표현 클래스
GAFuncGraph	캔버스에 그려질 각종 함수의 그리기를 위한 클래스

그림 3 JavaGA class 설명

JavaGA는 일반 응용프로그램 및 애플릿으로 사용 가능하며, Java SDK1.2 이상의 환경이 설정된 모든 시스템에서 활용이 가능하다.

#### IV. 함수 최적화 적용

본 논문에서 구현한 JavaGA를 이용한 활용 결과를 살펴보기 위하여 유전 알고리즘에서 전통적인 응용 분야 중의 하나인 함수 최적화 문제를 살펴보고자 한다. 함수 최적화는 임의의 함수가 주어지고 이 함수 값을 최대화 또는 최소화하는 변수 값(들)을 찾는 문제이다.

예를 들면  $f = \sum_{i=1}^n -x_i \cdot \sin(\sqrt{|x_i|})$ ,  $-500 \leq x_i \leq 500$ 에 대해 함수  $f$ 를 최소화하는  $x_i$  값들을 찾는 문제와 같은 것이다. 이 문제에서 변수의 개수가 5 개라고 하면 한 해는 아래와 같이 단순히 5 개의 변수 값을 나열하는 실수 염색체로 표현할 수 있다.

-20.67	120.23	0.02	-19.21	488.31
--------	--------	------	--------	--------

JavaGA는 염색체의 유전자의 표현을 전통적인 이진

수와 일상의 십진수로 표현하는 방법을 모두 제공하는데, 다음 예에서 두 가지 방법을 이용한 함수 최적화의 문제를 해결해 보자. 그 첫 문제로  $f(x) = 2x^2 - x + 8$  함수의 최대 값을 구하는 문제이다. 변수  $x$ 의 범위는  $-100 \leq x \leq 100$  인 경우, 유전자의 표현을 이진수로 구성하고, 유전 알고리즘에 이용되는 각종 파라미터를 다음과 같이 지정하여 JavaGA를 이용하여 해를 구해 보았다.

population size = 130  
 chromosome length = 30  
 max # of generation = 80

유전 알고리즘의 연산자인 교차는 일점 교차로 하고 확률은 prob = 0.65로 지정하였으며, 연산자 변이는 기본(각 Gene에서 발생한 난수가 아래의 확률보다 작으면 새로운 난수를 발생하여 지정하는 방법)으로 확률은 prob = 0.01로 설정한 결과 다음과 같은 결과가 도출되었다.

함수 값 및 세대수 : 20,107.937, 61 세대  
 x 값 : (-100)

다음은 실험 결과의 유전 세대의 진행에 따른 품질의 변화를 나타낸 그림이다.

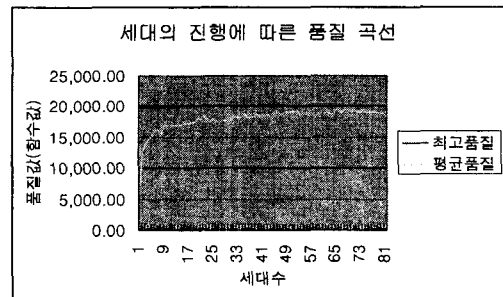


그림 4 함수 최적화 문제의 품질 곡선 결과 1

다음 그림은 본 프로그램의 결과인 화면을 나타낸다. 위에서 설정한 유전 알고리즘의 주요 파라미터가 지정된 것을 볼 수 있다.

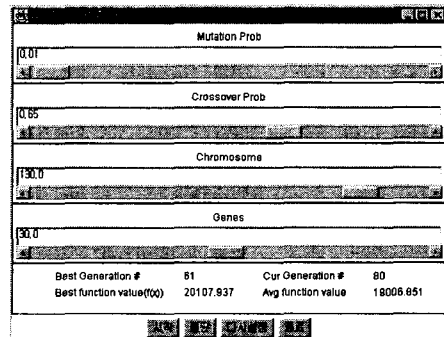


그림 5 함수 최적화 문제의 결과 1

다음 문제는  $x_i$  는 실수이며,  $i$  는 1, 2, 3인 경우, 합수  $f = \sum_{i=1}^3 (x_i^2)$ ,  $-100 \leq x_i \leq 100$  의 최대 값을 구하는 문제이다. 유전자 표현 방식은 십진수 표현을 하였으며 유전 알고리즘에 이용되는 각종 파라미터를 다음과 같이 지정하여 JavaGA를 이용하여 해를 구해 보았다.

population size = 150  
 chromosome length = 3(Real encoding에 의해 각  $x_i$  값이 Genes을 구성)  
 max # of generation = 80

유전 알고리즘의 연산자인 교차는 일점 교차(이진수의 방법과 동일하게 난수의 의해 발생된 위치에서 각각의 염색체를 교차하여 새로운 염색체를 생성하는 방법) 방식을 취하고 확률은 prob = 0.75로 지정하였으며, 연산자 변이는 기본(각 Gene에서 발생한 난수가 아래의 확률보다 작으면 새로운 난수를 발생하여 지정하는 방법)으로 확률은 prob = 0.035로 설정한 결과 다음과 같은 결과가 도출되었다.

합수 값 및 세대수 : 29800.921, 74 세대  
 $x$  값 : (100.00, 100.00, -99.00)

다음에 위 함수 최적화 문제의 실험 결과의 유전 세대의 진행에 따른 품질의 변화를 나타낸 그림이다.

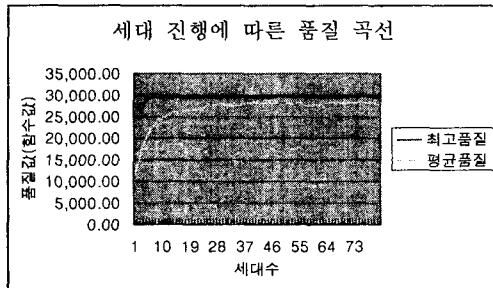


그림 6 함수 최적화 문제의 품질 곡선 결과 2

그림 7에서는 위 함수의 결과를 나타내는 JavaGA의 모습을 보이고 있다. 최대 80회의 세대를 거치는 동안 74세대에서 가장 최대 값을 나타내는 것을 볼 수 있다.

본 논문에서 제시한 JavaGA는 일반적인 문제를 유전 알고리즘으로 해결하는데 좋은 결과를 도출할 뿐 아니라 교차 연산에 이용되는 교차 확률과 변이 연산의 변이 확률 세대의 인구수 및 염색체를 구성하는 유전자의 수를 화면에서 지정하여 실험할 수 있는 그래픽 사용자 인터페이스를 제공한다.

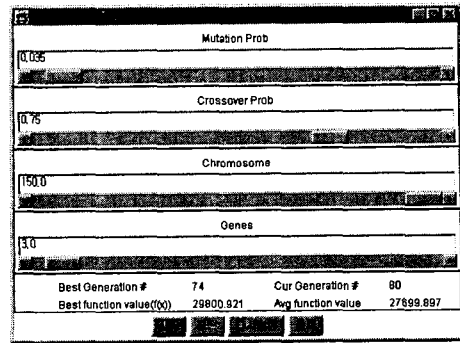


그림 7 함수 최적화 문제의 결과 2

## V. 결론

본 논문에서 구현한 JavaGA는 유전 알고리즘 전형적 방법을 교차 및 변이의 확률, 세대의 염색체 수, 염색체를 구성하는 유전자의 수를 그래픽 사용자 인터페이스를 통하여 쉽게 수정이 가능하여, 여러 번 실험을 통하여 결과를 도출하는 유전 알고리즘의 실험에 적합한 프로그램이다. 이 패키지를 발전시켜 유전 알고리즘에 이용되는 다양한 분야를 지원할 수 있도록 발전시키도록 하겠다. 특히 각 유전 알고리즘에 이용되는 각종 연산자의 다양한 여러 방법을 우선적으로 활용할 수 있도록 할 것이다.

감사의 글  
 본 연구는 한국과학재단 특정기초연구(1999-2-302-013-3) 지원으로 수행되었음.

## 참고문헌(또는 Reference)

- [1] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.
- [2] Z. Michalewicz, Genetic Algorithm + Data Structure = Evolution Program, Springer, 1992
- [3] David E. Goldberg, Genetic Algorithms in search, Optimization & Machine Learning, Addison-Wesley, 1989
- [4] Melanie Mitchell, An Intriduction To Genetic Algorithms, MIT Press, 1997
- [5] J. Koza, Genetic Programming, MIT Press, 1992.