2002년도 대한전기학회 하계학술대회 논문집 2002.7.10-12

Manabe형의 일반화에 관한 자바 구현

강환일, 김갑일, 한승수, *강환수
명지대학교 전기정보제어공학부, *동양공업전문대학 전자상거래과

The Java Implementation of the generalization of the Manabe Standard Fo

Hwan Il Kang, Kab Il Kim, Seung Soo Han, *Hwan Soo Kang
Division of Electrical & Information Control Eng, *Dept. of E-Commerce, Dongyang Technical College

Abstract - 진화 알고리즘은 생물의 유전적 진화 과정을 이용한 새로운 문제 해결의 방안으로 결정론적 방법으로 해결하지 못한 난제에 적합한 알고리즘으로 알려져 있다. 본 논문에서는 진화 알고리즘의 연구를 기반으로 전달함수 출력 파형 검출을 위한 기법에서 이용되고 있는 런지-커타(Runge-Kutta) 방법에서의 상미분 방정식의 해를 구하는 기법에서 유전 알고리즘을 이용하여 그 결과를 찾아본다. 본 논문에서의 구현은 자바 언어를 이용하며, 자바 언어를 적용한 구현 방법과 유전 알고리즘의 효율적 기법을 제시한다.

1. 서 론

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.[1,2,3,4,5]

본 논문에서는 단위 계단응답에 의한 전달함수 출력 파형 검출에서의 상승 시간과 정정 시간을 구하는 문제의 기법인 Runge-Kutta 방법에서의 상비분 방정식의 해를 구하는 기법에서 유전 알고리즘을 이용한다. 즉 본 논문에서 구현한 유전 알고리즘은 자바 언어를 사용한 그래픽 사용자 인터페이스를 이용하여 주요 패라미터를 쉽게 수정이 가능하도록 하며, 유전 알고리즘의 진행 과정을 파악할 수 있도록 하여, 유전 알고리즘을 쉽게 이용할 수 있도록 한다.

2. 런지 커타 기법

단위계단응답에 의한 전달함수의 출력 파형을 검출하고자 한다. 이때 파형에서 상승시간과 정정 시간을 구할 수 있다. 여기서 상승시간은 정상상태의 최종 값의 10%에서 90%까지 도달하는데 걸리는 시간을 뜻한다. 그리고 정정 시간은 입력 파형이 인가된 시각부터 입력 파형의 최종 값의 98%와 102%사이의 범주 안에 파형이 들어오는 첫 시각까지의 시간을 측정하는 것이다. 단위 계단응답에 의한 전달함수의 출력 파형을 검출하기 위해 Runge-Kutta 방법[6]을 이용하였다. Runge-Kutta 방법은 상미분방정식의 해를 구할 수 있다. 이때 입력은 단위 계단입력을 이용한다. 만약 전달함수가

$G(s)=\dfrac{b_0}{a_4s^4+a_3s^3+a_2s^2+a_1s+a_0}$ 이라면 이를 상미분방정식으로 바꾸어 표현하면

$a_4y''''+a_3y'''+a_2y''+a_1y'+a_0y=b_0u$

여기서 $x_1=y$

$x_2=y'$

$x_3=y''$

$x_4=y'''$

라고 치환하면 다음의 4개의 일차 미분방정식의 시스템으로 표현된다.

$\dot{x}_1=x_2$

$\dot{x}_2=x_3$

$\dot{x}_3=x_4$

이 되며

$\dot{x}_4=-\dfrac{a_0}{a_4}x_1-\dfrac{a_1}{a_4}x_2-\dfrac{a_2}{a_4}x_3-\dfrac{a_3}{a_4}x_4+\dfrac{b_0}{a_4}u$

이것을 선형시스템으로 변형하면

$\dot{x}=Ax+Bu$

$x^T=[x_1,x_2,x_3,x_4]$

$$A=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -\dfrac{a_0}{a_4} & -\dfrac{a_1}{a_4} & -\dfrac{a_2}{a_4} & -\dfrac{a_3}{a_4}\end{bmatrix}$$

$$B=\begin{bmatrix}0\\0\\0\\\dfrac{b_0}{a_4}\end{bmatrix}u(u=1)$$ 되는데 여기서 $$Y=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}$$ 라 놓으면 다음 식

$\dfrac{dY}{dt}=AY+B$이 되며 Runge-Kutta 방법은

$k_1=AY+B$

$k_2=A(Y+k_1h/2)+B$

$k_3=A(Y+k_2h/2)+B$

$k_4=A(Y+k_3h)+B$

$Y(t+h)=Y(t)+h/6(k_1+2k_2+2k_3+k_4)$

$$Y(t+h)=\begin{bmatrix}y(t+h)\\y'(t+h)\\y''(t+h)\\y'''(t+h)\end{bmatrix}$$ 이 된다.

3. 런지 커타 기법의 유전 알고리즘 구현

본 논문에서는 런지 커타 기법의 문제 해결을 위한 유전 알고리즘의 개발을 자바 언어를 이용하여 구현하였다. 본 논문에서 구현한 유전 알고리즘 패키지는 RungeKuttaGA라 지칭하며, 이는 일반 응용 프로그램과 애플릿으로 모두 사용 가능하다. 개발 프로그램의 주요 클래스를 살펴보면, 유전 알고리즘의 내용을 구현하고 있는 Ga.class, Chromosome.class, Generatuion.class를 구성하며, 런지 커타의 주요 알고리즘은 RungeKutta.class에서 구현한다. 프로그램의 그래픽 사용자 인터페이스 역할을 담당하는 RungeKuttaWin.class에서는 런지 커타 기법에 유전자 알고리즘을 적용한 최상의 결과를 그리는 RKCanvas.class와 유전 알고리즘의 각 세대의 품질의 진화 과정을 그리는 GaCanvas.calss를 갖는다.

본 논문에서 구현한 프로그램의 인터페이스를 살펴보면서 구현 알고리즘을 알아보자. 아래 그림에서 왼쪽 부분은 런지 커타의 입력 값과 유전 알고리즘의 입력 값을 수정할 수 있는 입력 부분과 유전 알고리즘의 의한 출력 결과를 문자로 보이는 결과 출력부분으로 구성된다. 오른쪽 상단에는 결과의 최상 염색체를 출력하는 부분과 런지 커타의 결과를 그려주는 부분이 있고, 오른쪽 하단에는 유전 알고리즘에서 각 세대의 평균 품질과 누적된

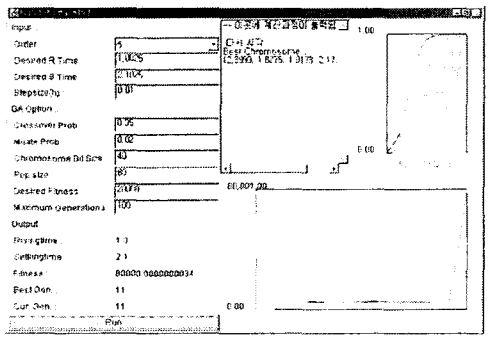

그림 1 프로그램 구성

최고 품질을 그려주는 캔버스로 구성된다.

본 논문에서 구현한 유전 연산 알고리즘에서 교차 연산에서는 일점 교차를 이용하며, 변이 연산에서는 염색체를 구성하는 모든 비트에 대하여 변이 확률보다 낮은 경우 변이를 일으키도록 한다. 선택 연산은 룰렛휠 선택 연산을 수행하며, 염색체의 표현은 이진수로 나타낸다. 각 염색체의 이진수는 범주의 최소 값의 차이를 염색체의 비트 크기의 이진수로 나타낼 수 있도록 한다. 실험시에 염색체의 크기를 조정하여 보다 정밀한 염색체의 값을 표현할 수 있도록 한다.

염색체의 구성은 차수에 따라 구성되는 감마의 값을 표현하는데, n차이면 n-1개의 감마 값을 갖으며, 이 감마의 순서는 의미가 있다. 또한 교차도 같은 순서의 감마끼리 연산을 행한다. 목적함수는 다음과 같다.

1 / ((ST-Desired ST)^ 2 + (RT - Desired RT)^ 2

ST : 실험 결과인 정정 시간
RT : 실험 결과인 상승 시간
Desired ST : 실험에서 원하는 정정 시간
Desired RT : 실험에서 원하는 상승 시간

또한 실험자가 최대 세대수와 최고 목적함수 값을 입력할 수 있도록 하였으며, 이 두 가지 조건 중 하나를 만족하면 유전 알고리즘은 그 수행을 종료한다. 진화 과정의 품질을 그리는 캔버스에서는 각 세대의 평균 품질과 누적된 최고 품질을 표현하고 있으며, 이를 꺽은 선형태로 그리는데, 항상 상위 선이 최고 품질이고 하위 선이 평균 품질을 나타낸다. 실험 상미분방정식의 차수는 4차에서 8차까지 5가지를 실험 할 수 있도록 하였으며, 이 중 한 값을 선택하도록 콤보박스를 이용한다. 선택 된 차수에 따라 기준 정정시간과 상승시간이 설정되는데 이 값은 원하는 경우 수정이 가능하다. 실험 결과의 문자 출력은 최고 목적함수값의 염색체와 목적함수 값, 최고 세대수, 상승시간, 정정시간을 표현한다.

4. 유전 알고리즘 실험

본 논문에서 구현한 기법의 결과를 살펴보자. 다음은 차수가 4인 런지 커타에서의 요구되는 상승시간과 정정시간이 각각 0.9976초와 2.165초인 경우이며, 간격 시간은 0.01초인 경우의 실험 결과이다. 유전 알고리즘의 각종 패라미터를 다음과 같이 지정하여 실험하였다.

population size = 100
chromosome length = 40(bits)
max # of generation = 100
Desired Fitness = 10000

유전 알고리즘의 연산자인 교차는 일점 교차 방식을 취하고 확률은 prob = 0.35로 지정하였으며, 연산자 변이는 확률은 prob = 0.02로 설정한 입력에 대하여 다음과 같은 결과가 도출되었다.

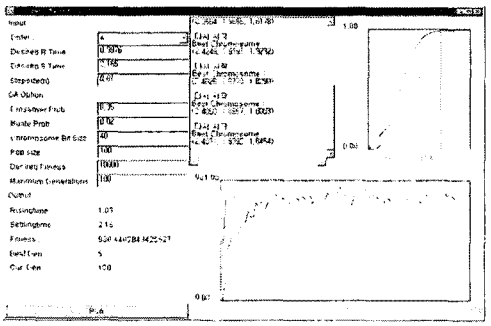

그림 2 차수 4인 경우의 실험 결과

위 실험에서 구한 가장 좋은 세대는 5번째 세대이며 감마 값은 각각 (2.4081, 1.9392, 1.8454)이며, 이를 사용하여 구한 상승시간과 정정시간은 1.03초와 2.16초로 적합도 값은 930.44이다.

다음은 차수가 7인 런지 커타에서의 요구되는 상승시간과 정정시간이 각각 1.0059초와 2.0832초인 경우이며, 간격 시간은 0.01초인 경우의 실험 결과이다. 유전 알고리즘의 각 종 패라미터를 다음과 같이 지정하여 실험하였다.

population size = 200
chromosome length = 50(bits)
max # of generation = 50
Desired Fitness = 50000

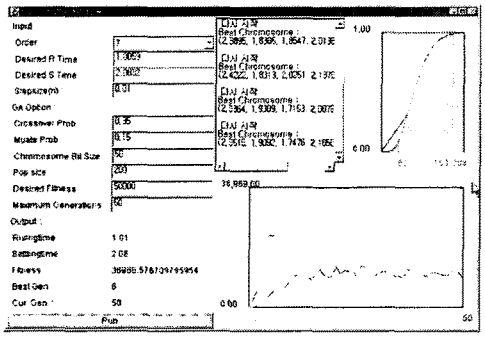

그림 3 차수 7인 경우의 실험 결과

유전 알고리즘의 연산자인 교차는 일점 교차 방식을 취하고 확률은 prob = 0.35로 지정하였으며, 연산자 변이는 확률 0.15로 입력하여 다음과 같은 결과가 도출되었다. 본 실험에서 구한 가장 좋은 세대는 9번재 세대이며 염색체 구성 값은 각각 (2.3515, 1.9092, 1.7476, 2.1858, 2.1565, 3.2412)이며, 이를 사용하여 구한 상승시간과 정정시간은 1.01초와 2.08초로 적합도 값은 36968.57이다.

5. 결 론

본 논문에서는 전달함수 출력 파형 검출에서의 상승시간과 정정 시간을 구하는 문제의 기법인 Runge-Kutta 방법에서의 상비분 방정식의 해를 구하는 기법에서 유전 알고리즘을 이용하였다. 이의 구현을 위하여 자바 언어를 이용하였으며, 유전 알고리즘은 자바 언어를 사용한 그래픽 사용자 인터페이스를 이용하여 주요 패라미터를 쉽게 수정이 가능하도록 하며, 유전 알고리즘의 진행과정을 파악할 수 있도록 하여, 유전 알고리즘을 쉽게 이용할 수 있도록 하였다.

현재 구현한 알고리즘은 유전 알고리즘에서 기본이 되는 알고리즘을 이용하고 있으며, 이를 적용한 실험에서 원하는 결과는 처음에 선택한 염색체의 많은 의존성을 갖고 있다. 이를 개선한 다양한 유전 연산 기법의 이용을 통한 안정적인 결과를 가져올 수 있는 개선된 알고리즘의 연구를 향후 계획한다.

[참 고 문 헌]

[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.

[6] S. C. Chapra & R. P. Cande, Numerical methods for engineers, pp.695--704, WCB/McGraw-Hill Company, Singapore, 1998.