

DNA 코딩을 이용한 multi-modal 함수의 최적점 탐색방법

Global Optimum Searching Technique of Multi-Modal Function Using DNA Coding Method

백동화, 강환일, 김갑일, 한승수*
명지대학교 전기정보제어공학부

Dong Hwa Peak, Hwan Il Kang, Kab Il Kim, and Seung Soo Han*
Division of Electrical and Information Control Engineering
Myongji University

*E-mail : shan@mju.ac.kr

ABSTRACT

DNA computing has been applied to the problem of getting an optimal solution since Adleman's experiment. DNA computing uses strings with various length and four-type bases that makes more useful for finding a global optimal solutions of the complex multi-modal problems.

This paper presents DNA coding method for finding optimal solution of the multi-modal function and compares the efficiency of this method with the genetic algorithms (GA). GA searches effectively an optimal solution via the artificial evolution of individual group of binary string and DNA coding method uses a tool of calculation or information store with DNA molecules and four-type bases denoted by the symbols of A(Ademine), C(Cytosine), G(Guanine) and T(Thymine). The same operators, selection, crossover, mutation, are applied to the both DNA coding algorithm and genetic algorithms. The results show that the DNA based algorithm performs better than GA.

Keywords : DNA computing; GA; Optimization

1. 서론

최근 들어 분자 생물학의 발전으로 인해서 생체 분자를 이용하여 계산을 수행하고자 하는 DNA computing 기법에 대한 연구가 활발해지기 시작했다. 1994년 Adleman의 NP-complete 문제에[1] 대한 DNA computing 실험 이후에 여러 가지의 최적화 문제에 적용되어 왔다. DNA computing 기법은 실제 생체 분자인 DNA를 계산의 도구 및 정보 저장 도구로 사용한다. DNA computing 에서는 A(Adenine),

C(Cytosine), G(Guanine), T(Thymine) 4가지 염기로 정보를 표현한다. 또한 막대한 병렬성을 이용하여 주어진 탐색 공간을 효율적으로 탐색할 수 있다.

GA(Genetic Algorithm)는 Holland의 저서에 서[2] 처음으로 소개되었다. GA는 스트링의 개체 집단 위에서 모의 진화를 일으켜 효율적으로 최적 해를 탐색하는 알고리즘이다. 두 부모의 유전자로부터 그들 자손의 유전자를 형성하는 유성생식과 자연환경에서 일어나는 진화원

리를 흉내내고있다.

본 논문에서는 인위적인 DNA의 개발 메커니즘에 근거를 둔 DNA coding기법과 GA를 사용하여 최적 해를 탐색하는 문제에서 각각을 성능을 조사하였다. 연산자와 그 밖의 조건들은 모두 동일하게 적용하였다.

II. DNA Coding 방법 및 알고리즘

2.1 생물학적 DNA

모든 생명체는 각각 고유의 DNA를 가지고 있다. DNA는 개체의 특성을 발현시키는 유전 코드로서 A, T, G, C 4종류의 염기 배열로 이루어져 있다. 이들 중 세 개의 배열이 한 의미단위를 이루어 해석된다. 이 의미단위를 생물학적인 용어로 codon(유전 정보의 최소단위)이라 한다. 총 64종류의 코돈은 20종류의 아미노산이 된다. 코돈의 64종류의 패턴에 대하여 생성되는 아미노산이 20종류인 이유는 다른 코돈이 같은 아미노산을 만들기도 하기 때문이다. DNA는 RNA로 전사되어 리보솜에서 단백질로 번역된다. 즉 아미노산을 암호화하는 DNA의 배열에 따라 아미노산의 합성순서를 결정하여 여러 종류의 단백질을 만들어낸다. RNA의 단백질로의 번역은 AUG에서 시작되어 UGA에서 번역이 끝난다. DNA에서는 U대신 T를 사용한다. 표1은 RNA 코돈과 생성하는 아미노산에 대한 것을 보여준다.

표 1 RNA(DNA) 코돈과 생성하는 아미노산

	U	C	A	G							
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U		
	UUC		UCC		UAC		UGC		C		
	UUA		UCA		UAA		정지		UGA	정지	A
	UUG		UCG		UAG		정지		UGG	Trp	G
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U		
	CUC		CCC		CAC		CGC		C		
	CUA		CCA		CAA		CGA		A		
	CUG		CCG		CAG		CGG		G		
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U		
	AUC		ACC		AAC		AGC		C		
	AUA		ACA		AAA		AGA		A		
	AUG		ACG		AAG		AGG		G		
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U		
	GUC		GCC		GAC		GGC		C		
	GUA		GCA		GAA		GGA		A		
	GUG		GCG		GAG		GGG		G		

2.2 DNA Coding 기법과 GA 알고리즘

그림 1은 DNA coding기법과 GA의 최적해를 탐색하는 전체적인 알고리즘을 보여주

고 있다. 전체적인 알고리즘을 살펴보면 다음과 같다.

- 1) 문제를 표현하는 초기 해 집단을 random하게 생성한다.
- 2) 초기 해 집단의 적합도를 구한다.
- 3) 룰렛 휠 선택자를 구현하여 최종 해가 될 가능성이 없는 해들을 삭제하고 가능성이 높은 해들만 보존하여 해를 진화시킨다.
- 4) 교배와 돌연변이 연산자를 수행한다. 교배는 2점 교배를 하고 국소 해에 빠질 위험성을 벗어나기 위해 random하게 교배 점을 선택하였으며, 돌연변이는 모든 코드에 대하여 행하였다. 교배와 돌연변이는 모두 주어진 확률값에 의해 행해진다.
- 5) Stop 조건을 만족하면 정지한다.

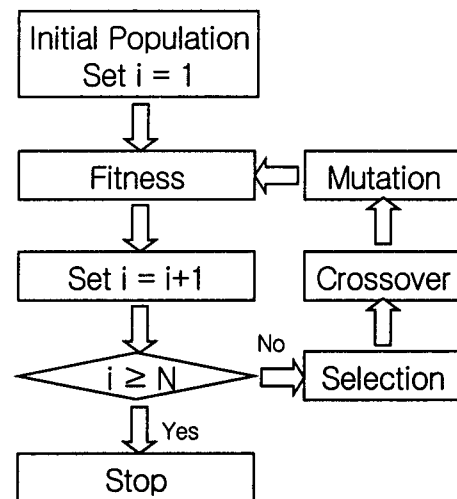


그림 1 DNA Coding과 GA에서의 사용 알고리즘

2.3 DNA, GA의 Coding 방법

DNA coding은 A, T, G, C의 4종류의 염기 배열의 차이로 나타낼 수 있는 총 64종류의 codon은 20종류의 아미노산이 된다. 아미노산들은 각자의 중요한 의미를 가진다. 그래서 염기 배열을 유전정보, 또는 유전 암호라고 한다. 유전코드는 ATG에서 시작하고 TGA에서 끝난다.

그림 2는 DNA 염색체의 예와 변환 메커니즘을 보여준다. 유전자는 시작 codon ATG에서 시작하고 종료 codon TGA에서 끝난다. 각각의 codon에 대응하는 아미노산들은 문제 해결을 위한 자신의 역할을 갖는다. 또한 중복 유전자들은 중요한 의미를 갖는다. 그림 3은 한 염색체에서 유전자의 중복을 보여주고 있다. 이러한 중복 유전자에 의한 연산이 DNA coding에서의 장점중의 하나이다.

그림 4는 교배와 돌연변이의 예를 보여준다. 교배는 이점 교배를 하여 두 부모 염색체 안에

분산되어 있는 어떤 유전정보를 결합하지 못하는 일점 교배의 단점을 보완해 준다. 그림 4(b) 돌연변이의 예로 돌연변이에 의해서 C가 G로 바뀌었다. 돌연변이 결과, gene 7이 새로 생겼다. 돌연변이는 random하게 A, T, G, C중 하나로 바뀌게 하였다.

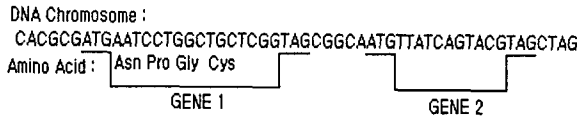


그림 2 염색체의 변환 메커니즘

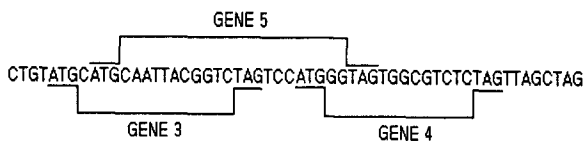
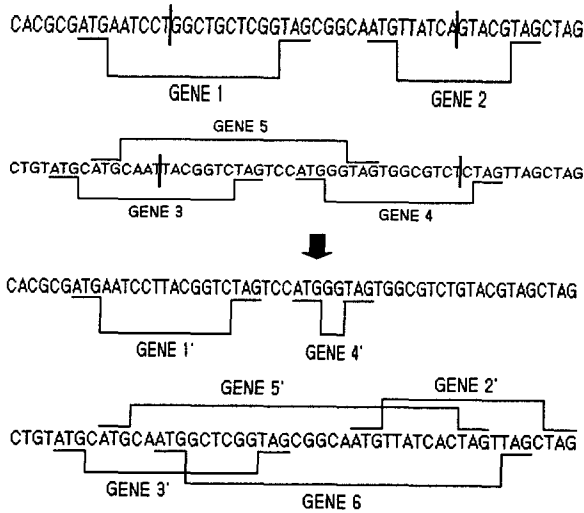
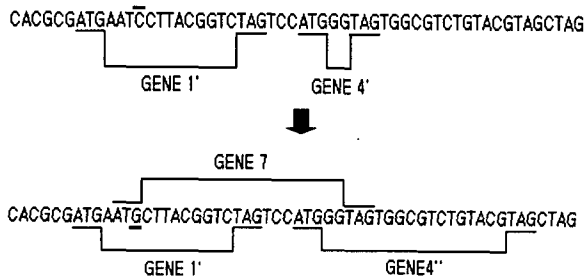


그림 3 유전자의 중복



(a) 교배



(b) 돌연변이

그림 4 교배와 돌연변이

III. 실험 및 결과

3.1 모의 실험

DNA coding 기법과 GA의 성능을 비교하기 위해서 다음과 같은 multi-modal 함수를 사용하였다.

$$f(x) = x + |\sin(32 \times x)|$$

$$0 \leq x \leq \pi$$

주어진 문제에서 x 의 범위 내에서 $f(x)$ 가 최대가 되는 점을 DNA coding 방법과 GA로 찾아서 각각의 성능을 조사하였다.

DNA coding에 있어서 코돈은 각자의 의미를 가지고 있다. 하나의 시작 코돈을 가지고 진화를 하는 것은 염색체를 비효율적으로 사용하는 것이 된다. 그래서 표 2와 같이 시작 codon을 ATG 대신 AT*로 즉 4종류로 지정하고 종료 코돈은 없이 하였다.

표 2 코돈의 의미

Phe	0	Pro	4	His	4	Glu	7
Leu	1	Thr	5	Gln	5	Cys	8
Ile	0	Ala	6	Asn	6	Trp	0
Met	2	Tyr	2	Lys	7	Arg	8
Ser	3	정지	無	Asp	7	Gly	9

본 논문에서는 표 3과 같이 DNA coding 방법과 GA의 조건을 같게 하였다. 각각의 세대수는 50세대로 하였으며 집단 크기는 60, crossover 확률은 0.8%, mutation 확률은 0.1%로 주었다.

표 3 실험 조건

	DNA	GA
세대수	50	50
집단 크기	60	60
염색체 길이	300	20
Crossover 확률	0.8%	0.8%
Mutation 확률	0.1%	0.1%

3.2 결과 및 고찰

표 4는 DNA coding 방법과 GA를 이용했을 경우에 각각 50 세대 후에 찾은 최적해와 그때의 함수 값을 보여주고 있다. 그림 5는 DNA coding 방법을 사용했을 경우와 GA를 사용했을 경우, 각 세대에서의 최적해(Objbest)와 각 세대에서 해의 평균(Objave)을 나타내고 있다.

X 축은 세대수를 나타내며 Y 축은 $f(x)$ 의 값을 나타낸다. 표 4에 의하면 DNA coding 방법에서는 4.0929, GA에서는 4.0909의 최대 값을 찾았으며, 이는 DNA coding 방법에 의한 해가 GA에 의한 해보다 0.04%의 극히 작은 성능이

향상되었음을 보여주고 있다. 하지만 그림 5를 보면 최적해가 GA에서는 20세대 이상이 지나야만 하지만, DNA coding 방법에서는 초기세대에서부터 최적해를 구할 수 있음을 보여주고 있다.

표 4 DNA coding 방법과 GA에 의한 최적값

	DNA	GA
최적해(x)	3.093413	3.095528
f(x)의 값	4.092991	4.090853

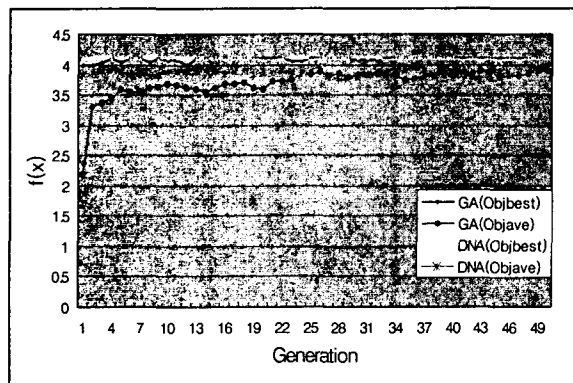


그림 5 각 세대별 f(x)값의 변화

IV. 결 론

본 연구에서는 최적해를 구하는 문제에 있어서 DNA coding 방법과 GA의 성능을 비교하여 보았다. 두가지 방법 모두 최적해를 효과적으로 찾았으며, DNA coding 방법에 의한 결과가 약간의 성능 향상이 있음을 알 수 있었다. 여분의 특성과 유전자 중복의 허용성 때문에 지식의 표현이 다양한 DNA coding 방법에 의한 방법에서는 주어진 multi-modal 함수의 최적값을 GA에 비하여 매우 적은 세대에 찾는 우수한 결과를 보여주었다.

DNA coding 방법은 DNA 분자의 막대한 병렬성과 여러 gene가 하나의 염색체내에 있을 수 있다는 장점 때문에 복잡한 문제에 적용하였을 경우 우수한 결과를 얻을 수 있을 것으로 예상된다.

향후 연구에서는 시작 codon을 지정함에 있어서의 비율 결정, 염기의 종류의 연구와 패턴 인식에 있어서의 DNA coding 방법의 적용방법과 효율성에 대한 연구가 이루어져야 할 것이다.

V. 참고문헌

[1] Leonard M. Adleman, "Molecular

Computation of Solutions To Combinatorial Problems", *Science*, pp. 159-171, 1996

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975

[3] Tomohiro Yoshikawa, Takeshi Furuhashi and Yoshiaki Uchikawa, "The Effects of Combination of DNA Coding Method with Pseudo-Bacterial GA," *Proc. IEEE Int. Conf. Evolutionary Computation*, Indianapolis, IN, USA, pp. 285-290, April, 1997

[4] Gheorghe Paun, Grzegorz Rozenberg, Arto Salomaa, *DNA Computing-New computing Paradigms*, Springer, Berlin, July 1998

[5] M. Amos, *DNA Computing*, Ph.D. thesis, The University of Warwick, UK, September 1997

[6] Brian Hayes, "The Invention of The Genetic Code," *American Scientist*, January-February, 1998

[7] R. Deaton et. al, "A DNA Based Implementation of an Evolutionary Search for Good Encodings for DNA Computation," *Proc. IEEE Int. Conf. Evolution Computation*, Indianapolis, IN, USA, pp. 267-271, April, 1997

[8] Piotr Wasiewicz, Tomasz Janczak, J. Mulaka, "The Inference via DNA Computing," *IEEE*, pp. 988-993, 1999

[9] Sungyong Yun et. al, "Acquisition of Fuzzy Rules Using DNA Coding Method," *한국퍼지 및 지능시스템학회 '98 춘계학술대회 학술발표 논문집*, vol. 8, no. 1, pp.16-19, 1998

[10] Dong-Wook Lee, Kwee-Bo Sim, "A Characteristics of Cellual Automata Neural Systems," *한국 퍼지 및 지능시스템 학회 추계학술대회 논문집*, pp. 267-272, 1998. 11월