퍼지 및 지능시스템학회 논문지 2003, Vol. 13, No. 1, pp. 37-44

유전알고리즘과 DNA 코딩을 이용한 Numeric 패턴인식

Numeric Pattern Recognition Using Genetic Algorithm and DNA coding

백동화 · 한승수
Dong-Hwa Paek, Seung-Soo Han

NPT Center
명지대학교 정보공학과

요 약

본 논문은 DNA coding 방법과 Genetic Algorithm(GA)을 사용하여 numeric(0~9) 패턴인식 성능을 비교 평가하였다. 이진 스트링의 개체 집단 위에서 모의진화를 일으켜 효율적으로 최적 해를 탐색하는 GA와, 생체 분자인 DNA를 계산의 도구 및 정보 저장도구로 사용하며, Adenine(A), Cytosine(C), Guanine(G), Thymine(T)등의 4가지 염기를 사용하는 DNA coding 방법을 이용하여 numeric 패턴인식을 수행하였다. DNA coding 방법과 GA의 성능을 비교 평가하기 위해서 selection, crossover, mutation 등의 GA연산자를 DNA coding에 동일하게 적용하였다. 실험결과, DNA coding 방법은 GA보다 효과적으로 패턴인식을 수행하였다. GA에 비해 DNA coding 방법의 장점은 스트링의 길이가 가변적이고 해의 중복성을 가지며, 4가지 염기를 이용하기 때문에 해 표현이 다양함을 가지고 있다.

Abstract

In this paper, we investigated the performance of both DNA coding mehtod and Genetic Algorithm(GA) in numeric pattern (from 0 to 9) recognition. The performance of the DNA coding method is compared to the that of the GA. GA searches effectively an optimal solution via the artificial evolution of individual group of binary string using binary coding, while DNA coding method uses four-type bases denoted by Adenine(A), Cytosine(C), Guanine(G) and Thymine(T). To compare the performance of both method, the same genetic operators(crossover and mutation) are applied and the probabilities of crossover and mutation are set the same values. The results show that the DNA coding method has better performance over GA. The reasons for this outstanding performance are multiple candidate solution presentation in one string and variable solution string length.

Key Words : Genetic Algorithm, DNA Coding, Pattern Recognition

1. 서 론

DNA coding 방법과[1][2] Genetic Algorithm(GA)[3]은 자연 생태계의 진화과정에서 관찰된 몇 가지 처리과정 중에서 적자생존(survival of the fittest)의 원리를 컴퓨터 알고리즘과 결합시켜 정립된 최적화(optimization) 알고리즘이다. DNA coding 방법과 유전알고리즘은 자연생태계의 진화 메커니즘을 모방하였는데 실제로 자연계의 진화과정을 모두 밝혀져 있지는 않지만 중요한 몇 가지는 알려져 있어 이러한 진화과장에서 일부 관찰된 것을 사용하였다. 자연생태계의 진화과정에서 일반적으로 인정되고 있는 몇 가지 과정은 다음과 같다.

1) 진화는 문제를 표현하는 염색체에 대해 일어난다.
2) 자연도태(natural selection)는 문제의 최적해가 될 수 있는 염색체와의 관계이며, 최적해가 될 수 있는 염색체는 다음 세대에 많이 전달될 수 있고 그렇지 못한 해는 도태되도록 메커니즘을 구성하는 것이다.
3) 재생산(reproduction) 과정은 우성 염색체를 다음 세대에 전달하는 과정으로 진화가 일어나는 시점이다. 교배는 부모 염색체들을 결합함으로써 매우 다른 염색체의 자손을 만들어 주고, 돌연변이는 부모의 염색체와 자손의 염색체를 다르게 함으로써 새로운 염색체를 만들어 낼 수 있다.
4) DNA coding 방법과 GA는 인공지능 알고리즘처럼 사전 지식을 필요로 하지 않으며, 자연도태와 재생산, 교배, 돌연변이 연산자에 의해서 진화하면서 최적 해를 찾는 알고리즘이다.

DNA coding 방법은 인간의 실제 생체 정보 분자인 DNA를 계산의 도구로 사용하는 알고리즘이다. 실제 생체 분자를 사용함으로써 GA보다 여러 가지 장점들을 가지고 있다. 첫째는 0과 1의 2진수를 사용하는 GA에 비하여 DNA coding 방법은 Adenine(A), Cytosine(C), Guanine(G), Thymine(T)의 4가지 염기를 사용하여 coding하기 때문에 해의 표현이

접수일자 : 2002년 11월 8일
완료일자 : 2003년 2월 1일
본 연구는 과학기술부 및 한국과학재단의 ERC 프로그램을 통한 지원으로 이루어졌으며 이에 감사를 드립니다.

다양하다. 둘째는 DNA coding 방법에서는 coding에 여분이 있으며 또한 중복되어 해를 표현할 수 있다. 셋째는 염색체의 길이가 가변적이다. 이러한 장점들로 인해서 DNA coding 방법은 패턴인식 문제에서 GA보다 효율적으로 해를 찾을 수 있을 것으로 기대된다.

DNA 코딩 방법의 장점과 더불어 최근 들어 분자 생물학의 발전으로 인해서 생체 분자를 이용하여 계산을 수행하고자 하는 DNA computing 기법에 대한 연구가 활발해지기 시작했다. 1994년 Adleman이 NP-complete 문제인 해밀토니안 경로 문제(Hamiltonian Path Problems: HPP)를 생물학적 과정만으로 해결함으로써 새롭게 DNA computing 기법을 이용한 최적해 문제에 대한 연구가 활발해 졌다[4]. 지금까지의 인공지능에서는 신경망이나 진화 연산처럼 대부분 생물학적 개념만을 이용해서 계산 모델을 만들어 이를 적용하여 왔다. 그러나 DNA computing 기법은 실제 생체 분자인 DNA를 계산의 도구 및 정보 저장 도구로 사용하는 새로운 방법으로 진화 연산과 결합하여 인공지능의 새로운 한 분야가 되었다. 현재까지도 DNA가 가지고 있는 막대한 병렬성을 이용하여 최적화 문제들을 해결하고자 하는 연구들이 많이 진행되고 있으며, DNA 코딩 방법을 이용해서 함수의 최적점을 탐색하는 연구도 이루어지고 있다[5]. GA는 Holland의 저서에서[6] 처음으로 소개되었으며, 스트링의 개체 집단 위에서 모의 진화를 일으켜 효율적으로 최적 해를 탐색하는 알고리즘이다. 두 부모의 유전자로부터 그들 자손의 유전자를 형성하는 유성생식과 자연환경에서 일어나는 진화원리를 흉내낸다. 본 논문에서는 이러한 DNA Computing 방법과 GA를 패턴인식 분야에 적용하려 한다.

패턴인식(Pattern Recognition)이라는 용어는 일반적으로 분류(classification) 또는 패턴묘사(description of objects or patterns)를 의미한다[7]. 패턴 인식 문제들은 자동적으로 개개의 문자들을 확인하거나, 정상 혹은 비정상의 패턴을 알아내는 것이다. 이러한 인지 시스템을 만들기 위해서는 일련의 모델링 데이터들이 필요하며, 이러한 데이터 각각에 대한 패턴들은 이미 알려져 있어야 한다. 통상적인 패턴 인식 시스템에는 광학 문자 인식(optical character recognition; OCR), 구어 인식(speech recognition), 화자 인식(speaker recognition), 지문 인식(fingerprint recognition) 등이 있다. 기존의 패턴인식방법에서는 임의의 패턴을 K개 중의 하나의 클래스로 분류하기 위해, 본질적으로 크게 다음 두 가지로 나눌수 있다. 첫 번째는 기하학적인 또는 통계적인 접근법(geometric or statistical approach)이며[8], 두 번째는 구조적인 또는 구문론적인 접근법(structural or syntactic approach)이다. 통계적인 패턴인식 방법은 각 특성들(features)은 그 패턴 클래스의 확률 밀도 함수에 제약을 받는 것으로 가정한다. 따라서 임의의 패턴 벡터 x가 클래스 Wi에 속한다고 할 때, x는 그 클래스에 대한 조건부 확률을 갖는 임의의 샘플이 된다. 잘 알려진 통계적 결정론(statistical decision theory)이나 판별분석(discriminant analysis)과 같은 방법들은 패턴 클래스들 사이의 경계를 결정한다. 만일 각 클래스의 조건부 확률이 알려져 있다면, 최적의 결정 룰(decision rule)은 베이즈 결정론(Bayes decision theory)에 의해 결정된다. 그러나, 조건부 확률이 알려져 있는 경우 파라미터 결정(parametric decision)이 문제가 된다. 통계적인 패턴 인식 접근법과 비교하여 구조적인 접근법에서 좋은 이점은 분류자체 뿐만 아니라, 패턴에 대한 설명을 제공한다는 것이다. 구조적인 접근법은 패턴은 먼저 일련의 원시 규칙(primitive rule)에 부합하는 명백한 구조를 가지고 있어야 한다. 그러나 노이즈가 심한 패턴들로부터 의미있는 원시(primitive)나 문법 구조를 찾아내는 것은 어렵다.

DNA coding 방법과 GA은 앞에서 설명한 패턴 클래스 사이의 경계를 결정하지 않아도 되며, 패턴인식을 위해서 패턴의 구조를 찾아내지 않고도 패턴인식을 수행할 수 있는 장점이 있다. DNA coding 방법과 GA은 미리 정해놓은 표준패턴에 입력패턴을 받아들여 입력패턴이 될 수 있는 수많은 해.군집을 만들어 표준패턴과 비교를 통하여 해가 될 가능성이 높은 패턴들을 진화시켜 패턴인식을 수행하는 방법으로 효과적으로 패턴인식을 수행할 수 있다.

본 논문에서는 패턴인식에 있어서 이러한 장점을 가지고 있으며 효율적으로 최적해를 탐색하는 DNA coding 방법과 GA을 사용하여 numeric(0~9) 패턴인식 성능을 조사하였다. 연산자 및 각 연산자의 파라미터 등과 같은 조건들은 모두 동일하게 적용하여 각각의 성능을 조사하였다.

2. DNA coding 방법 및 알고리즘

2.1 생물학적 DNA

모든 생명체는 각각 고유의 DNA를 가지고 있다. 그림 1은 DNA는 개체의 특성을 발현시키는 유전코드를 보여주고 있다. DNA는 A, T, G, C 4종류의 염기 배열로 이루어져 있으며, 이 유전코드는 A≡T, G≡C의 수소 결합으로 된 2중 나선구조를 가지고 있다. 그리고 2중 나선은 서로 3′에서 5′로 5′에서 3′으로의 서로 반대 방향으로 상보 결합을 이루고 있다.

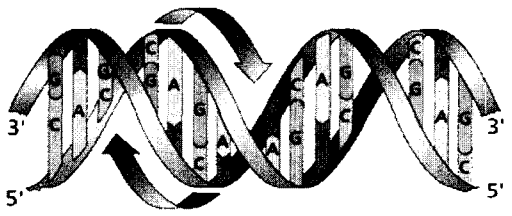


그림 1. 생물학적 DNA 구조
Fig 1. Structure of Biological DNA

표1은 RNA 코돈과 생성하는 아미노산에 대한 정보를 보여준다. A, T, G, C 네 종류의 염기배열 중 세 개의 배열이 한 의미단위를 이루어 해석된다. 이 의미단위를 생물학적인 용어로 코돈(codon)이라 하며, 이는 유전 정보의 최소단위가 된다. 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 합성 순서를 결정하여 여러 종류의 단백질을 만들어낸다. RNA의 단백질로의 번역은 ATG에서 시작되어 TGA에서 번역이 끝난다.

2.2 DNA 알고리즘과 GA

그림 2는 DNA coding기법과 GA의 최적해를 탐색하는 전체적인 알고리즘을 보여주고 있다. DNA coding 방법과 GA은 그림에서처럼 동일한 과정을 수행하여 패턴인식 방법에 적용하였다. 전체적인 알고리즘을 살펴보면 다음과 같다.

1) 문제를 표현하는 초기 해 집단을 random하게 생성한다.
2) 해 집단의 적합도를 구한다.
3) 루울렛 휠 선택자를 구현하여 최종해가 될 가능성이 없는 해들을 도태시키고 가능성이 높은 해들을 보존하여 새로운 자손 집단을 생성한다.
4) 교배 연산자를 수행한다. 교배 연산자는 random하게 발생한 0~1사이의 수가 교배 확률(Pc)보다 작거나 같을 때 수행한다. 교배는 2점 교배를 하며 국소 해에 빠질 위험성을 벗어나기 위해 random하게 교배 점을 선택한다.
5) 돌연변이 연산자를 수행한다. 돌연변이 연산자는 random하게 발생한 0~1사이의 수가 돌연변이 확률(Pm)보다 작거나 같을 때 수행한다. 돌연변이는 모든 코드에 대해서 수행한다.
6) 2에서 4의 과정을 세대수 n이 될 때까지 반복하여 해를 진화시킨다.

표 1. RNA(DNA) 코돈과 생성하는 아미노산
Table 1. RNA(DNA) Codon and Amino Acid

	U		C		A		G		
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu	UCA		UAA	정지	UGA	정지	A
	UUG		UCG		UAG		UGG	Trp	G
C	CUU		CCU	Pro	CAU	His	CGU	Arg	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Gln	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	Lys	AGA	Arg	A
	AUG	Met	ACG		AAG		AGG		G
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glu	GGA		A
	GUG		GCG		GAG		GGG		G

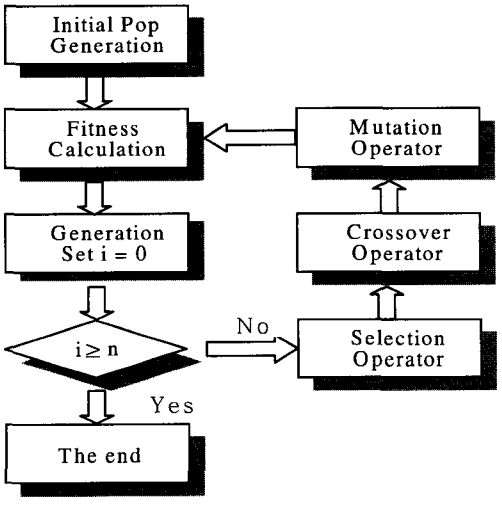


그림 2 DNA 알고리즘 및 GA
Fig. 2. Algorithm used in DNA coding method and GA

2.3 DNA coding 방법

DNA는 A, T, G, C의 4종류의 염기 배열 중 세 개의 배열이 한 의미단위가 되기 때문에 나타낼 수 있는 codon은 총 64종류가 있으며, 이는 다시 20종류의 아미노산이 된다. 아미노산들은 각자의 중요한 의미를 갖기 때문에 염기 배열을 유전정보, 또는 유전 암호라고 한다.

그림 3은 DNA 염색체의 예와 변환 메커니즘을 보여준다. 각각의 codon에 대응하는 아미노산들은 문제 해결을 위한 자신의 역할을 가지고 있다. 염색체에서 문제의 해가 될 수 있는 Gene은 유전코드 ATG에서 시작하고, TAG, TAA, TGA에서 끝난다. 또한 중복 유전자들도 중요한 의미를 갖는다. 그림 4는 한 염색체에서 유전자의 중복을 보여주고 있는데, 하나의 DNA 염색체내에 3개의 Gene이 존재하며 Gene5는 Gene3, Gene4와 중복되어 나타나 있음을 보여주고 있다. 이러한 중복 유전자에 의한 Gene의 표현이 DNA coding에서의 장점중의 하나이며 GA에 비해서 다양한 표현 가능성을 보여준다.

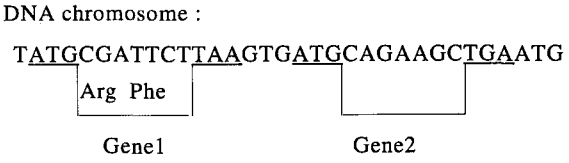


그림 3. DNA coding 및 해석의 예
Fig 3. Example of DNA Coding and Interpretation

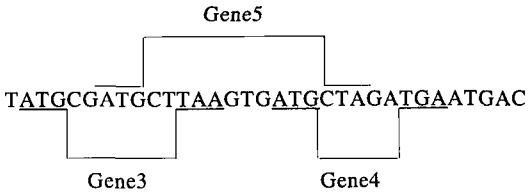
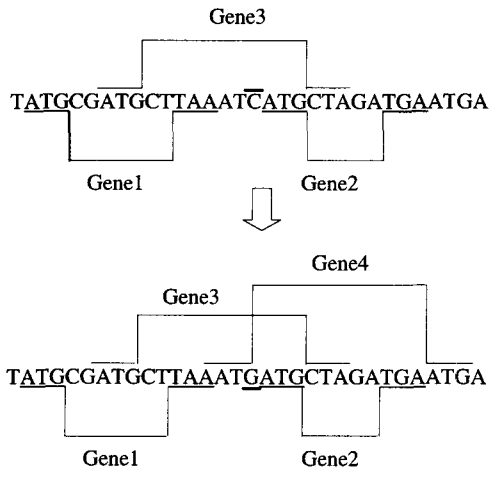


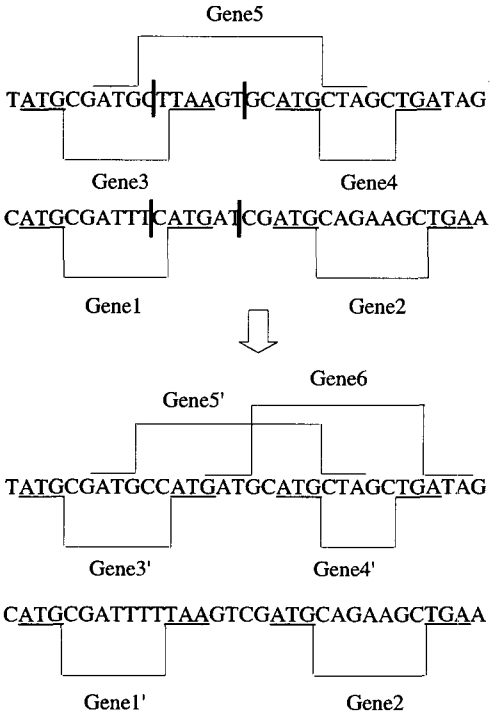
그림 4. 유전자의 중복의 예
Fig 4. Example of Overlapping of Genes

그림 5는 GA에서 사용하는 돌연변이와 교배 연산자를 DNA coding 스트링에 적용한 것을 보여주고 있다. 그림 5(a)는 돌연변이의 예로, 돌연변이 확률에 따라 돌연변이 연산자에 의해서 C가 G로 바뀌었으며 이와 같은 돌연변이 결과로 Gene4가 새로 생겼음을 알 수 있다. 돌연변이는 random하게 A, T, G, C중 하나로 바뀌게 된다. 그림 5(b)는 교배의 예를 보여준다. 교배는 주어진 교배확률에 의해서 발생하며 본 논문에서는 이점 교배 (two-point crossover)를 하여 두 부모 염색체 안에 분산되어 있는 유전정보를 결합하지 못하는 일점 교배(one-point crossover)의 단점을 보완하도록 했다. 결과로 Gene6이 새로 생겼음을 알 수 있다.

DNA coding 방법은 앞에서 살펴본 것과 같이 다음과 같은 장점들을 가지고 있다. 첫째는 0과 1의 2진수를 사용하는 GA에 비하여 DNA coding 방법은 A, T, G, C의 4가지 염기를 사용하여 coding하기 때문에 해의 표현이 다양하다. 둘째는 DNA coding 방법에서는 coding에 여분이 있으며 또한 중복되어 해를 표현할 수 있다. 셋째는 염색체의 길이가 가변적이다.



(a) 돌연변이
(a) Mutation



(b) 교배
(b) Crossover

그림 5. 돌연변이와 교배의 예
Fig 5. Example of Mutation and Crossover

3. DNA coding 방법과 GA을 이용한 패턴인식

3.1 DNA coding 방법과 GA를 이용한 패턴인식 방법

Numeric 패턴매칭이란 주어진 패턴의 대응관계를 찾는 문제로서, 유사 패턴의 위치, 크기, 회전각도, 등이 확실하지 않은 만큼 간단한 문제가 아니다. 입력패턴과 가장 유사한 표준패턴을 찾기 위해서 본 논문에서는 입력패턴의 위치, 크기, 회전각도를 DNA coding 방법과 GA의 염색체 코드로 사용하였다.

DNA 코딩 방법과 GA을 이용하여 패턴인식 과정에서 필요한 입력패턴을 정확히 표현하기 위하여 이용할 특징을 결정하는 특징공간의 설정 단계와 각 특징값을 구하여 입력패턴을 특징공간의 한 점으로 사상시켜주는 특징 추출 단계를 생략할 수 있다. DNA coding 방법과 GA은 미리 정해놓은 표준패턴에 입력패턴을 받아들여 입력패턴의 위치, 크기, 회전각도 등을 변화시켜가며 수많은 해 군집을 만들어 표준패턴과 비교를 통하여 해가 될 가능성이 높은 패턴들을 진화시켜 패턴인식을 수행하는 방법이다.

그림 6은 DNA coding 방법과 GA을 이용한 패턴 인식 방법의 전체적인 개요를 보여준다.

1) 미지의 패턴을 입력한다.
2) Coding 과정에서는 입력패턴에 대하여 DNA 알고리즘과, GA를 이용하여 전처리를 수행하다. 전처리 과정에서 모든 입력패턴의 회전, 확대, 축소, 이동 등의 연산을 수행한다.
3) 전처리 된 입력패턴과 표준 패턴을 비교하여 적합도를 계산하는 패턴 매칭을 수행함으로서 인식결과를 얻게 된다.

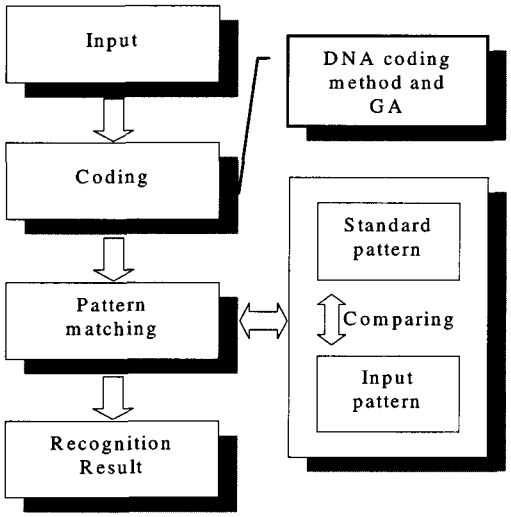


그림 6. 패턴 인식 개요
Fig. 6. Outline of Pattern Recognition

3.2 DNA coding 방법과 GA의 전처리 수행 기법

DNA Computing 방법과 GA의 코드들은 패턴 매칭을 수행하기 위해서 이동, 회전, 확대, 축소의 의미를 가지는 파라미터를 사용한다. 입력 패턴이 표준 패턴과 가장 잘 매칭되는 회전, 확대, 축소, 이동 파라미터 값들을 찾는 것이 목적이며, 이를 위해 각 파라미터들을 2진 스트링으로 표현한다. 그림 9는 GA에서 사용한 길이가 21 bit인 이진 스트링의 각각 코드의 의미를 보여준다. 회전에는 7 bit를, X축, Y축 확대/축소에는 각각 2 bit를, 좌/우 이동에는 각각 3 bit를, 그리고 상/하 이동에는 각각 2 bit를 할당하였다. 각각의 알고리즘으로 찾은 스트링 코드 정보를 통해 입력패턴에 대해 이동, 회전, 확대, 축소를 수행한 후의 입력패턴과 이미 정하여 놓은 표준패턴을 비교하여 패턴 매칭을 수행한다.

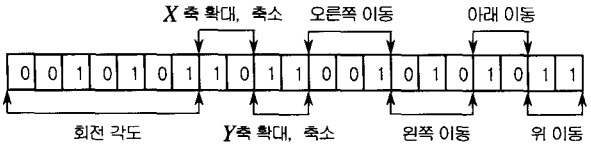


그림 7. 전처리과정에서 패턴인식을 위한 코드들의 의미 (GA String)

Fig. 7. Meaning of Codes for Pattern Recognition in Preprocessing(GA String)

DNA coding 방법에서도 GA와 같은 전처리 과정을 수행하기 위해서 염색체내에서 하나의 해가 되는 Gene들을 길이 21인 2진수로 변환을 시켜준다. 그림 8은 Gene이 2진수로 변환되는 과정을 보여준다. Gene 값은 Gene이 포함하는 codon들에 대응되는 아미노산의 코드정보의 합으로 구할 수 있다. 표 2는 아미노산에 해당하는 코드 정보를 보여주고 있다. 시작 codon과 종료 codon은 4에 대응되도록 하였다. DNA 스트링의 길이가 300일 때 포함될 수 있는 아미노산은 3개의 코드로 구성되어 있기 때문에 100개(=300÷3)이며, 첫 번째 아미노산이 시작 codon이고 마지막 아미노산이 끝을 나타내는 codon일 경우 최대 98개(=100-2)의 아미노산을 갖는 스트링이 된다. 이때 아미노산 스트링이 가질 수 있는 최대 값은 모든 아미노산이 19의 값을 갖는 Gly 일 때이며 이때의 값은 19×98 = 1862 가 된다. DNA 스트링이 가질 수 있는 최소값은 모든 아미노산이 0의 값을 갖는 Phe 일 때이며 이때의 값은 0 이 된다. 이러한 과정을 통해서 Gene의 값은 21 bit의 GA 스트링인 0과 221 - 1 사이의 특정한 수에 대응되고 이 정수를 2진수로 변환해서 전처리 과정을 수행하게 된다. 식 (1)은 Gene의 값을 0~221 - 1사이의 값으로 mapping 시켜주는 함수이다. 예를 들어 그림 3에서 Gene1의 아미노산은 18과 0에 대응되므로 Gene1의 값은 18 + 0 = 18 이 되며, 이것을 식(1)에 대입하여 계산한 f(x)값을 2진수로 변환하면 '000000100111100110001' 이 된다.

$$f(x) = \frac{Gene(2^{21}-1)}{1862} \qquad (1)$$

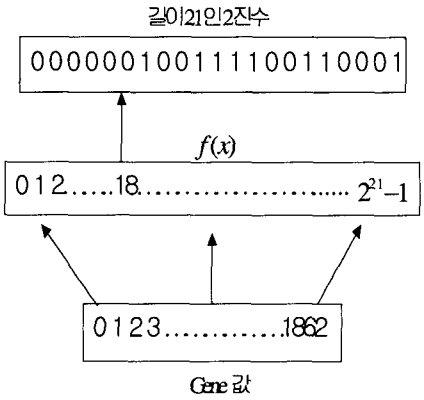


그림 8. Gene을 2진수로 변환

Fig. 8. Gene Conversion into Binary String

표 2. 각 아미노산에 부여된 코드표 2

Table 2. Code for Each Amino Acid

Phe	0	Ser	5	His	10	Glu	15
Leu	1	Pro	6	Gln	11	Cys	16
Ile	2	Thr	7	Asn	12	Trp	17
Met	3	Ala	8	Lys	13	Arg	18
Val	4	Tyr	9	Asp	14	Gly	19

4. 실험 및 결과

4.1 모의실험

본 연구에서는 DNA coding 방법과 GA를 이용하여 0~9의 numeric 패턴을 인식하는 성능을 각각의 방법으로 수행하여 비교하였다. 두 알고리즘에서 사용한 연산자 및 각 연산자의 파라미터 등과 같은 조건들은 모두 동일하게 적용하였다.

실험에서 사용한 표준패턴과 입력패턴은 16×16의 2차원 배열을 사용하였다. 그림 9(a)는 패턴 매칭을 위해 이미지로부터 선분의 밝기를 세 단계로 표현한 표준패턴의 예를 보여주고 있다. 표준패턴의 경우 효율적인 적응도 계산을 위하여 선분의 밝기를 1, 2, 5 세 단계로 세분화 시켰다. 그림 9(b)는 입력패턴의 예를 보여주고 있다.

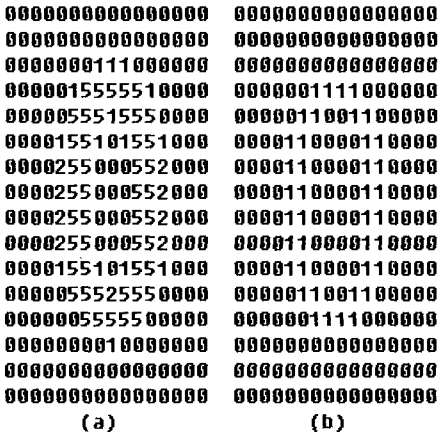


그림 9 (a) 표준 패턴 (b) 입력패턴

Fig. 9. (a) Standard Pattern (b) Input Pattern

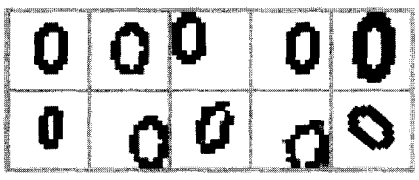


그림 10. 0에 대한 여러 가지 입력패턴의 예

Fig. 10. Examples of Various Input Pattern For 0

여러 가지 입력패턴에 대한 패턴인식 성능을 알아보기 위해서 표준 패턴을 회전, 확대, 축소, 이동, 그리고 이동한 후 회전시킨 이미지를 입력 패턴으로 사용하였다. 입력 패턴 이

미지의 회전 각도는 ±10도, ±20도, ±30도, ±40도로 회전시켰으며, 확대는 1.2배, 축소는 0.8배의 입력패턴을 사용하였다. 그림 10은 0에 대한 여러 가지 입력패턴의 예를 보여준다.

그림 11은 길이가 21인 이진 스트링에서 회전, 이동, 확대, 축소 연산자들의 값을 구하기 위한 파라미터 값들을 보여준다. 입력패턴의 회전각도 θ는 그림 7에서의 회전에 해당되는 0 과 1의 코드 정보와 그림 11에서 회전 파라미터 값과의 곱의 합으로서 구할 수 있다. 확대와 이동 값도 회전과 같이 0과 1의 코드 정보와 파라미터 값과의 합으로서 구한다.

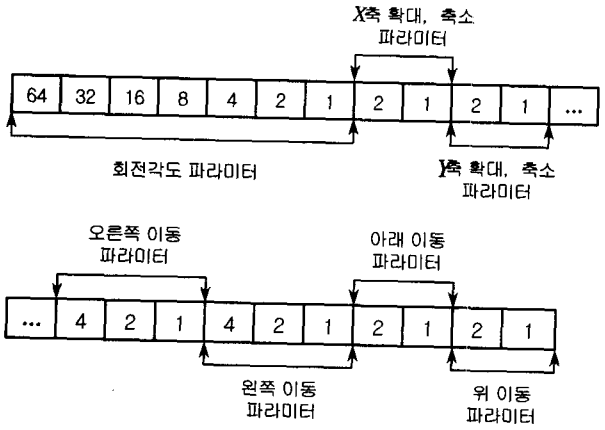


그림 11. 연산자들(회전, 이동, 확대 축소)의 값을 구하기 위한 파라미터

Fig. 11. Parameters for Computing the Values of Rotation, Move and Enlargement Operators

식 (2)는 회전각도 θ를 계산하는 방정식이다. i_v는 그림 7에서 각 코드의 정보(0 혹은 1)를 나타내고, x_{iv}는 그림 11에서 해당 인덱스의 파라미터 값을 나타낸다. θ값은 0~391의 범위를 가진다.

$$\theta = 3 \times \sum_{i=1}^{7} i_v \times x_{iv} \tag{2}$$

입력패턴에 대한 회전은 식 (2)에서 구한 θ값을 이용하여 식 (3)에 의해서 이루어진다. Xang와 Yang는 회전각도를 나타내며 r은 radian값을 나타낸다. X축과 Y축에 대한 확대는 MTABLE의 인덱스에 따라 0.9배, 1.0배, 1.1배, 1.2배 확대된다. 그림 7에서의 X축과 Y축의 확대를 나타내는 0과 1의 코드정보와 그림 11에서 확대 파라미터 값과의 곱의 합으로서 MTABLE인덱스를 계산한다. 표 3은 MTABLE의 인덱스에 따른 확대 비율을 나타낸다. pattern[y][x]에 대한 좌우와 상하 이동은 식 (4)에 의해서 구할 수 있다. MTABLE[xax]와 MTABLE[yax]는 X축과 Y축의 확대 비율을 나타내고, CENTERX와 CENTERY는 패턴에서의 기준좌표(7, 7)을 나타내며, x, y는 pattern[y][x]에서의 배열 인덱스이다. 그리고 lsh와 rsh는 각각 왼쪽과 오른쪽 이동 파라미터로부터 구한 값이고, upsh와 downsh는 각각 위와 아래 이동 파라미터로부터 얻은 값이다. 식 (4-1)과 (4-2)에서 구한 값 x2와 y2를 이용하여, 좌우 이동 값 x1과 상하 이동 값 y1을 구할 수 있다.

표 3. M_{TABLE}의 인덱스에 따른 확대 비율

Table 3. Enlargement Ratio by M_{TABLE} Index

M_{TABLE} 의 인덱스	확대 비율
0	0.9
1	1.0
2	1.1
3	1.2

$$\begin{aligned} X_{ang} &= (\cos(r) - \sin(r)) \\ Y_{ang} &= (\sin(r) + \cos(r)) \\ r &= \theta \times 3.141592/180.0 \end{aligned} \tag{3}$$

$$\begin{aligned} x2 = {} & ((M_{TABLE}[x_{ax}] \times (x+1)) \\ & + (CENTER_X \times M_{TABLE}[x_{ax}] \\ & - CENTER_X) - 1) \end{aligned} \tag{4-1}$$

$$\begin{aligned} y2 = {} & ((M_{TABLE}[y_{ax}] * (y+1)) \\ & + (CENTER_X * M_{TABLE}[y_{ax}] \\ & - CENTER_X) - 1) \end{aligned} \tag{4-2}$$

$$\begin{aligned} x1 &= x2 - l_{sh} + r_{sh} \\ y1 &= y2 - up_{sh} + down_{sh} \end{aligned} \tag{4-2}$$

표 4. 모의실험에 사용된 파라미터

Table 4. Parameters Used in Simulation

	DNA	GA
세대수	100	100
집단 크기	100	100
염색체 길이	300	21
Crossover 확률	0.7	0.7
Mutation 확률	0.1	0.1

적합도는 16×16의 표준 패턴과 회전, 이동, 확대, 축소 등의 연산을 수행한 입력패턴과 각 픽셀의 곱의 합으로 구한다. 표 4는 모의실험에 사용된 파라미터 값들을 보여준다. DNA coding 방법과 GA에서 사용된 값들은 성능비교를 위해서 세대수와 각 세대에서의 집단의 크기는 100으로 하였으며, 교배 확률은 0.7, 돌연변이 확률은 0.1로 동일한 값을 사용하였다. 표준 패턴은 10개(0~9)로 구성하였고, 테스트하기 위해 23가지의 입력패턴을 사용하였으며, 정확한 결과를 위해서 입력패턴에 대해서 10회씩 반복 실험하였다.

4.2 실험결과 및 고찰

표 5는 표준패턴을 이동, 회전, 확대, 축소 등을 행한 입력패턴에 대한 패턴 인식률을 나타내고 있다. 회전 입력 이미

지는 표준패턴을 회전시킨 이미지와, 이동을 행한 후 회전시킨 이미지를 포함하고 있다. 이동한 입력이미지의 인식률은 표준패턴에 대해 좌·위 이동, 우·아래 이동, 좌·아래 이동, 우·위 이동한 패턴의 평균 인식률을 나타낸다. DNA coding 방법은 GA보다 전체적으로 인식률이 높게 나타났으며, 모든 입력 패턴에 대한 평균인식률은 7.9% 우수하게 나타났다. 표준패턴을 아무 변형 없이 그대로 입력패턴으로 사용했을 때 DNA coding 방법은 91.0%의 인식률을 보였고, GA는 80.0%의 인식률을 보였다.

표 5. 표준패턴, 이동, 회전, 확대, 축소 패턴에 대한 패턴인식률

Table 5. Pattern Recognition Ratio for Standard, Move, Rotation, Enlargement, Reduction Patterns

입력 이미지 \ 알고리즘	패턴 인식률	
	DNA	GA
표준 패턴	91.0 %	80.0 %
이동	78.5 %	66.3 %
±10도 회전	66.8 %	62.3 %
±20도 회전	68.0 %	61.8 %
±30도 회전	67.0 %	59.5 %
±40도 회전	64.3 %	59.3 %
0.8배 확대	80.0 %	75.0 %
1.2배 확대	82.0 %	70.0 %
모든 입력 패턴에 대한 평균 인식률	70.9 %	63.0 %

표 5에서 볼 수 있듯이 단순 이동 또는 확대한 패턴에 대한 인식률은 DNA와 GA 모두에게서 비교적 높은 인식률을 보여주고 있으나, 회전과 이동을 동시에 행한 패턴에 대한 인식률은 상대적으로 낮은 수치를 보여주고 있다. 특히 회전각도가 클수록 GA의 패턴 인식률이 크게 떨어졌는데, 이에 반해서 DNA coding 방법을 사용했을 경우에는 GA를 사용했을 때 보다 인식률의 감소가 둔화됨을 보여주고 있다. 이는 복잡한 경우일수록 DNA coding 방법에 의한 패턴인식 성능이 우수함을 나타내 준다고 할 수 있다. 이러한 결과는 DNA 코딩 방법이 가지고 있는 해의 표현이 다양해 해가 될 수 있는 입력패턴이 다양해지고, coding에 여분이 있으며 중복되어 해를 표현할 수 있어 표준패턴과 비교할 수 있는 입력패턴, 즉 해가 될 수 있는 입력패턴이 많아져 좋은 결과를 얻을 수 있었던 것으로 생각된다. 그러나 DNA 코딩 방법은 염색체의 길이가 GA보다 길어서 연산 시간이 많이 소요되는 단점이 있다.

4. 결 론

본 연구에서는 numeric 패턴인식 문제에 있어서 동일한 연산자를 사용하여 DNA coding 방법과 GA의 성능을 비교하여 보았다. 실험 결과에서 DNA coding 방법이 GA보다 효과적으로 패턴인식을 하였다. DNA coding 방법은 DNA 분자의 막대한 병렬성과 여러 가지의 해가 하나의 염색체내에 있을 수 있다는 장점 때문에 복잡한 문제에 적용하였을 경우 우수한 결과를 얻을 수 있을 것으로 예상된다.

향후 연구에서는 교배 위치에 따른 탐색 성능의 관계, 본 논문에서 사용한 연산자 외 다른 연산자의 적용, 염기의 종류의 연구와 여러 가지의 패턴 인식에 있어서의 DNA coding 방법의 적용방법과 효율성에 대한 연구가 이루어져야 할 것이다.

참 고 문 헌

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

[2] M. Amos, "DNA Computing", *PhD. thesis*, The University of Warwick, UK, September 1997

[3] 김용호, 전홍태, "Genetic Algorithm 응용기술", *한국통신학회지*, Vol. 9, No. 11, pp. 809-817

[4] Leonard M. Adleman, "Molecular Computation of Solutions To Combinatorial Problems", *Science*, pp. 159-171, 1996

[5] 백동화, 강환일, 김갑일, 한승수, "DNA 코딩과 진화연산을 이용한 함수의 최적점 탐색방법," *퍼지 및 지능시스템학회 논문지*, vol. 11, No. 6, pp. 538-542, 2001

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

[7] Richard O. Duda and Peter E. Hart, Pattern Classification and Scene Analysis, *A wiley-Interscience Publication*, 1973

[8] K. Fukunaga, Introduction to Statistical Pattern Recognition, *Academic Press*, N.Y., 1990

[9] Tomohiro Yoshikawa, Takeshi Furuhashi and Yoshiki 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

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

[11] 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

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

[13] Surapong Auwatanamongkol. Pattern recognition using Genetic Algorithm, *IEEE* pp. 822-828, 2000

저 자 소 개

백동화 (Paek, Dong-Hwa)

2001년 : 대구 가톨릭대학교 자동차 전자공학과 졸업

2001 ~ 현재 : 명지대학교 전기 정보공학 과 석사 과정

관심분야 : 신경회로망, 유전알고리즘, DNA Computing, Pattern Recognition.

Phone : 016-786-1663

E-mail : fog0577@hanmail.net

한승수 (Han, Seung-Soo)

1986년 : 연세대학교 전기공학과 졸업

1988년 : 연세대학교 전기공학과 졸업(석사)

1996년 : 조지아공대 전기 및 컴퓨터 공학과 졸업(박사)

2000년 ~ 현재 : 명지대학교 정보공학과 부교수

관심분야 : 신경회로망, 유전알고리즘, DNA Computing, Pattern Recognition, 정보보호

Phone : 031-330-6345

Fax : 031-321-0271

E-mail : shan@mju.ac.kr