

# Maximal Clique Problem을 해결하기 위한 DNA 코딩 방법을 적용한 DNA 컴퓨팅

김 은 경<sup>†</sup> · 이 상 용<sup>††</sup>

## 요 약

MCP(Maximal Clique Problem)를 해결하기 위해 DNA 컴퓨팅이 사용되고 있다. 그러나 현재의 DNA 컴퓨팅을 MCP에 적용하였을 때, 정점과 간선을 효율적으로 표현할 수 없으며 제한 효소의 잘못된 사용으로 인하여 해를 찾을 수 없는 문제점을 가지고 있다. 본 논문에서는 MCP의 문제점을 해결하기 위해 DNA 컴퓨팅 기법에 DNA 코딩 방법을 적용한 ACO(Algorithm for Code Optimization)를 제안한다. 우리는 ACO를 MCP에 적용하였고, 그 결과 ACO는 Adleman의 DNA 컴퓨팅 알고리즘 보다 가변길이의 DNA 코드를 표현할 수 있으며, 불필요한 정점을 제거한 코드를 생성할 수 있었다. 또한 ACO는 Adleman의 DNA 컴퓨팅 알고리즘 보다 탐색 시간과 생물학적 오류율을 15% 정도 줄임으로써 4배 정도 많은 최종해를 얻을 수 있었다.

## DNA Computing Adopting DNA Coding Method to solve Maximal Clique Problem

Eun-kyoung Kim<sup>†</sup> · Sang-yong Lee<sup>††</sup>

## ABSTRACT

DNA computing has used to solve MCP (Maximal Clique Problem). However, when current DNA computing is applied to MCP, it can't efficiently express vertices and edges and it has a problem that can't look for solutions, by misusing wrong restriction enzyme. In this paper we proposed ACO (Algorithm for Code Optimization) that applies DNA coding method to DNA computing to solve MCP's problem. We applied ACO to MCP and as a result ACO could express DNA codes of variable lengths and generate codes without unnecessary vertices than Adleman's DNA computing algorithm could. In addition, compared to Adleman's DNA computing algorithm, ACO could get about four times as many as Adleman's final solutions by reducing search time and biological error rate by 15%.

**키워드 :** 최대 클리크 문제(Maximal Clique Problem : MCP), DNA 컴퓨팅(DNA Computing), DNA 코딩 방법(DNA Coding Method)

### 1. 서 론

1994년 Adleman 이후 생체분자를 이용하여 계산을 수행하고자 하는 DNA 컴퓨팅에 관한 연구가 활발해지고 있다. DNA 컴퓨팅은 DNA가 갖는 막대한 병렬성과 정보 저장능력, Watson-Crick의 상보결합[3]을 이용하여 조합 최적화 문제를 해결하는데 이용되고 있다[1, 2].

조합 최적화 문제 중 MCP(Maximal Clique Problem)는 HPP(Hamiltonian path problem)의 변형으로, N개의 정점을 갖는 완전 그래프  $G(V)$ 에 대한 서브 그래프  $G(V')$ 를 표현하고,  $G(V-V')$ 의 서브 그래프가 최대 정점을 형성하는 완

전 그래프를 찾는 문제이다.

Ouyang과 Kaplan 등은 처음으로 Adleman의 DNA 컴퓨팅 알고리즘을 이용하여 MCP를 해결하였다[4]. 이들은 MCP를 N개의 정점을 갖는 길이가 N bit인 이진수의 클리크와 상보 그래프(complementary graph)를 표현한후, 결합되지 않은 그래프 정점의 쌍을 상보 그래프를 이용하여 찾는 방법을 제안하였다. 그러나 Ouyang의 방법으로 해를 찾았을 때, 결합에 필요하지 않은 정점이 포함되어 정점과 간선을 효율적으로 생성하지 못하며, 잘못된 제한 효소의 사용으로 해를 찾을 수 없는 문제점을 가지고 있다. 또한 문제의 크기가 커질 경우 MCP를 DNA 컴퓨팅에 적용하기 어렵다.

Noort는 Ouyang의 MCP를 이용하여 노드가 20개인 DNA 컴퓨팅을 실험하였다. Noort는 데이터를 전송하는 시험관들을 통합하여 처리 속도를 향상시켰다. 하지만 DNA를 그래

<sup>†</sup> 준 회원 : 공주대학교 대학원 컴퓨터공학과

<sup>††</sup> 종신회원 : 공주대학교 정보통신공학부 교수

논문접수 : 2003년 4월 19일, 심사완료 : 2003년 11월 11일

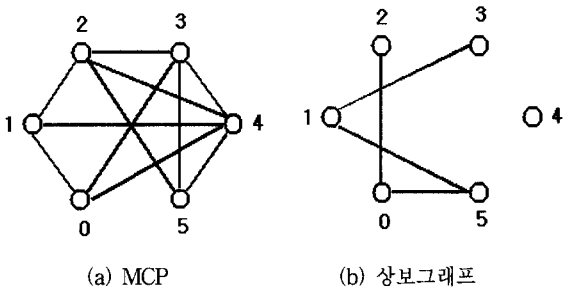
프 문제로 변환하였을때 염기 서열들을 효율적으로 표현하지 못하였고, 잘못된 효소의 사용으로 해를 찾을 수 없는 경우도 발생할 수 있다[5].

본 논문에서는 이와 같은 문제점을 해결하기 위하여 DNA 컴퓨팅 알고리즘에 DNA 코딩 방법을 적용한 ACO(Algorithm for Code Optimization)를 제안한다. ACO는 효율적인 정점과 간선을 생성하며, 결합에 필요하지 않은 정점들을 제거한다. 또한 잘못된 제한 효소의 사용을 막을 수 있기 때문에 최단 거리의 경로를 빠른 시간 내에 발견할 수 있다.

2. 관련 연구

2.1 MCP

MCP(Maximal Clique Problem)는 N개의 정점을 갖는 완전 그래프 G(V)에 대한 서브 그래프 G(V')를 표현하고, G(V-V')의 서브 그래프가 최대 정점을 형성하는 완전 그래프를 찾는 문제이다[6].



(그림 1) MCP 그래프

(그림 1)은 Ouyang이 해결한 MCP 그래프로 (그림 1)(a)는 최대 클리크 문제(Maximal Clique Problem), (그림 1)(b)는 상보그래프(complementary graph)를 나타낸 것이다. 상보그래프는 주어진 그래프에 포함되지 않는 간선으로 생성한 그래프이다. 이 문제에서 최대 클리크는 {2, 3, 4, 5}이며, <표 1>은 Ouyang이 MCP를 해결한 알고리즘이다.

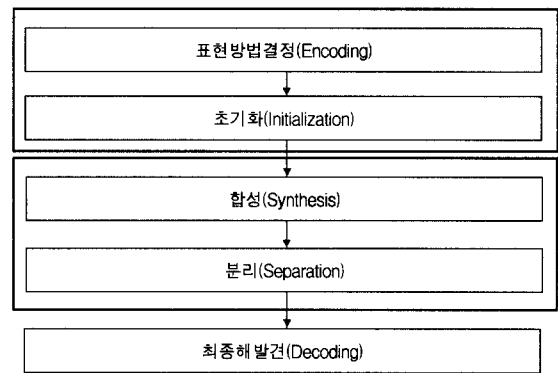
<표 1> Ouyang의 Maximal Clique Problem 해결을 위한 알고리즘

<p>step 1 : N개의 정점을 갖는 그래프를 길이가 N bit인 이진수로 클리크를 표현한다. (1 : 정점 포함, 0 : 정점 포함하지 않음)</p> <p>step 2 : 상보그래프를 이용하여 간선에 의해 결합되지 않은 그래프 정점의 쌍을 찾는다.</p> <p>step 3 : 상보그래프에 포함되는 그래프의 정점을 삭제한다.</p> <p>step 4 : 남은 조합 중에 1을 가장 많이 포함하는 클리크를 최대클리크로 한다.</p>
---

2.2 DNA 컴퓨팅

DNA 컴퓨팅은 DNA나 RNA와 같은 실제 생체 분자를

계산의 도구로 사용하는 새로운 계산 방법이다. DNA는 4개의 염기인 A(Adenine), T(Thymine), C(Cytosine), G(Guanine)로 정보를 표현하며, 방향을 갖는 이중 나선형 구조로 꼬여져 있다. 이들 염기에는 대용량 데이터를 저장할 수 있는 메모리 기능을 가지고 있으며, 정해진 규칙에 의해 상호 보완적인 방식의 Watson-Crick 결합을 하고 있다[3]. 그리고 복잡한 염기 조합의 패턴은 하나의 유전 정보를 담고 있으며, 인체내에서 자연 발생하는 효소에 의해 읽혀지고 있다. 이러한 효소는 생물학 실험 방법들과 함께 DNA 컴퓨팅의 연산자로 사용되고 있다.



(그림 2) Adleman의 DNA 컴퓨팅 알고리즘

(그림 2)는 Adleman의 DNA 컴퓨팅 알고리즘을 나타낸 것이다. Adleman은 주어진 문제를 DNA 코드로 표현하고, 생성된 DNA 코드를 단 한번의 합성과 분리 과정을 거쳐 조합 최적화 문제를 해결하였다[1].

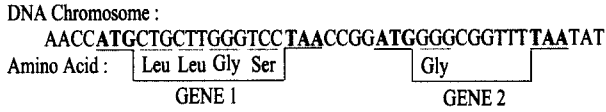
이러한 DNA 컴퓨팅은 매우 낮은 에너지로 작동되기 때문에 많은 에너지가 필요없다. 그리고 나노 수준의 막대한 병렬성을 이용하여 NP-complete에 효과적인 접근이 가능하게 되었으며, 계산 속도와 정보의 저장 및 처리 효율에서도 우수함을 보이고 있다[1, 4, 5].

2.3 DNA 코딩 방법

DNA 코딩 방법은 1995년에 Yoshikawa가 처음으로 제시한 변형된 형태의 유전자 알고리즘으로, DNA를 이용한 선택, 재생, 교배, 돌연변이 연산자를 사용한다[7, 8]. 일반적인 유전자 알고리즘은 0, 1의 이진수로 염색체를 표현하는 반면 DNA 코딩 방법은 4가지 염기를 사용하여 표현한다. 그럼으로써 긴 길이의 염기 서열이 아닌 적은 수의 아미노산 서열을 사용하여 다양한 해를 표현할 수 있다.

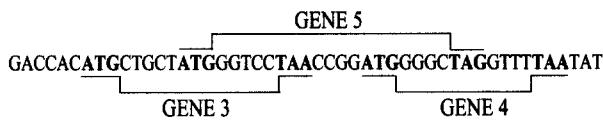
DNA가 하나의 아미노산으로 해석되기 위해서는 3개의 염기서열이 필요하며, 이것을 생물학적인 용어로 코돈(codon)이라고 한다. 이것을 아미노산 번역표에 따라 총 64가지의 아미노산으로 해석된다. 하지만 중복된 것을 제외한

20개의 아미노산만으로 해석된다.



(그림 3) DNA 염색체의 번역의 예

(그림 3)에서 보는 것과 같이 염기서열은 start 코돈인 ATG에서 시작하여 stop 코돈인 TGA(TAA, TAG)에서 아미노산 번역이 끝나며, 코돈을 아미노산으로 해석함으로써 짧은 DNA 코드에서도 많은 정보를 얻을 수 있다.

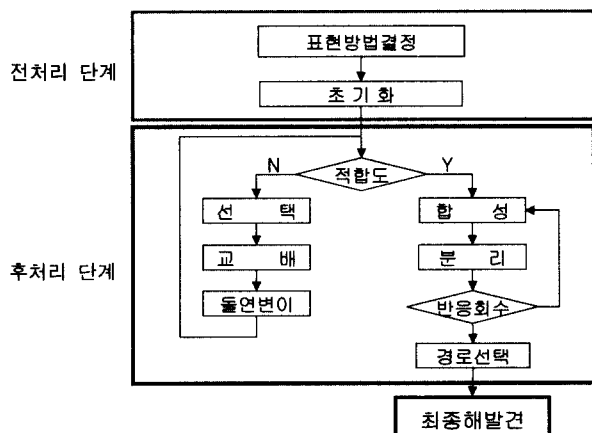


(그림 4) 유전자 중복의 예

DNA 코딩 방법의 특징을 살펴보면 첫 번째로, (그림 4)에서 보는 것처럼 염색체의 중복을 효율적으로 표현할 수 있다는 것이다. 두 번째로, 하나의 아미노산을 만드는 코돈이 여러 개이므로 지식 표현이 쉽다. 세 번째로, 교차점이 임의로 주어지기 때문에 염색체의 길이가 가변적이라는 것이다. 이러한 특징들은 염색체의 기능과 동작을 더욱 생물학적으로 가깝게 모델링한다.

### 3. ACO

본 논문에서는 조합 최적화 문제인 MCP를 풀기 위해 Ouyang의 DNA 컴퓨팅 알고리즘을 개선하고, DNA 코딩 방법을 적용하여 정점과 간선을 생성하였다. 그리고 DNA 코딩 방법의 연산자를 이용하여 합성과 분리 과정을 반복 처리하므로써 해를 찾는 ACO를 제안하였다.



(그림 5) ACO의 전체 흐름도

ACO는 MCP 문제에 대하여 DNA의 특징을 최대한 반영한 정점과 간선을 생성하며, 결합에 필요하지 않은 정점은 특정부위를 끊는 제한효소를 사용하여 제거함으로써 효율적으로 생성할 수 있다. 그리고 비효율적인 탐색으로 인한 시간과 노력이 낭비되는 것을 줄일 수 있으며, 반복적인 합성과 분리 과정을 통해 생물학적 실험 오류율을 최소화함으로써 많은 해를 탐색할 수 있다.

(그림 5)는 ACO의 전체 흐름도를 표현한 것으로 전처리 단계와 후처리 단계로 구분된다. 먼저 ACO의 전처리 단계를 살펴보면, 표현 방법 결정 과정과 초기화 과정으로 나뉜다. 표현 방법 결정 과정에서는 주어진 염기 서열을 DNA 분자의 특성을 잘 반영할 수 있도록 DNA 코딩 방법을 이용하여 가변길이의 정점과 제한 효소를 포함한 간선을 생성한다. 정점과 간선은 바로 표현할 수 없으며 이들을 DNA 염기 서열로 변환하기 위해서는 (그림 6)과 같은 프로시저를 이용한다. 먼저 간선을 생성하기 위하여 모든 DNA 코드에 대해 start codon(ATG)의 위치를 파악하고, i번째 위치와 (i+1)번째 위치 전까지의 사이를 간선으로 표현한다. 단, DNA code의 처음 위치가 start codon으로 시작되지 않을 경우 DNA 코드의 처음 부분부터 i번째 위치 전까지를 간선으로 한다.

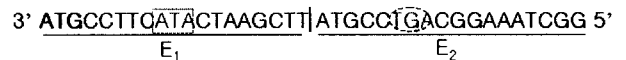
```

Initialize (E = Edges,
           DNA_seq = (DNA sequence),
           S_codon = Start codon("ATG"))

Begin
  If Middle(DNA_seq, 1, 3) = S_codon Then
    i = 0
    Call Edge_Make(E, DNA_seq, i)
  Else
    i = 1
    Call Edge_Make(E, DNA_seq, i)
  End if
End

Sub Edge_Make(Edge_t_No, DNA, Temp_Start_No)
  For L = 1 to Length (DNA) Step 3
    codon = Middle(DNA, i, 3)
    If codon = S_codon Then i = i + 1
    If i > Edge_t_No Then Exit for
    E(i) = E(i) + codon
  Next
End Sub
    
```

(그림 6) 간선 표현 프로시저



(그림 7) 간선 표현의 예

(그림 7)은 (그림 6)에 따라  $E_1$ 과  $E_2$ 를 생성한 예이다.

```

Initialize( N = Nodes,
           DNA_seq = (DNA sequence),
           S_codon = Start codon("AT*", Not ATG")
           E_codon = End codon("TAA, TGA, TAG") )

Begin
  For i = 1 to N
    Sn = InString(E(i), S_codon)
    E(i) = Middle(E(i), Sn)
    En = InString(E(i+1), E_codon)
    If E = 0 Then
      V(i) = V(i) + _
      Middle(E(i+1), 1, Integer(Length(E(i+1))/2))
    Else
      V(i) = V(i) + Middle(E(i+1), 1, En)
    End If
    V(i) = CE(i) + V(i) + CE(i)

  Call Com_Model(V(i))

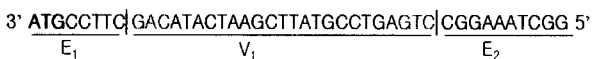
  If i mod 2 then
    V(i) = E(i) + V(i) + E(i+1)
  Else
    V(i) =  $\frac{E(i+1) + V(i) + E(i)}{2}$ 
  End If
Next
End
    
```

(그림 8) 가치 염기 서열 표현 드로시저

이렇게 표현된 간선에 의해 그 정점의 포함 여부를 결정하는 가치 염기 서열(value sequence)을 모든 DNA 코드에 대하여 (그림 8)에 따라 생성된다. 먼저 간선  $E_i$ 에서 처음으로 나타나는 AT\*(ATT, ATC, ATA)를  $V_i$ 로 지정하고,  $E_{i+1}$ 에서 처음으로 나타나는 stop codon인 TAA, TGA, TAG를  $V_{i+1}$ 로 지정하여 두 간선 사이의 가치 염기 서열로 표현한다. 단, stop codon이 없을 경우에는  $E_{i+1}$ 의 염기 서열 1/2bp(base pair)를 가치 염기 서열로 한다. 가치 염기 서열 생성 후 두 번째는 제한효소를 가치 염기 서열의 앞과 뒤에 결합시켜 완전 그래프에 포함되지 않는 정점을 특이적으로 잘라 연결시키지 않는다. 세 번째는  $i$ 가 짝수이면  $E_i V_i E_{i+1}$ 로 정점을 표현하며,  $i$ 가 홀수이면 상보 서열인  $\overline{E_{i+1} V_i E_i}$ 으로 정점을 표현한다.



(그림 9) 가치 염기 서열 표현의 예



(그림 10) 제한 효소를 포함한 정점 표현의 예

(그림 9)는 (그림 8)에 따라 간선 사이에 가치 염기서열

을 포함한 예이고, (그림 10)은 (그림 9)에서 제한효소를 포함한  $i$ 가 짝수일때의 정점의 표현 예이다. 위와 같은 방법으로 정점과 간선을 생성한 후 DNA 코드를 아미노산 번역 표에 따라 아미노산 코드로 번역한다. 이처럼 표현 방법 결정 과정이 끝나면 다음 과정인 초기화 과정에서 결정한 표현 방법을 반영한 DNA 코드들을 생성한다.

ACO의 후처리단계를 살펴보면 적합도 평가를 하여 적합도를 만족하지 않을 경우, DNA 코딩 방법의 연산자를 이용하여 적합도를 재평가한다. 그리고 만족하는 경우 반응횟수만큼 합성과 분리 과정을 거쳐 우수한 경로를 선택하여 최종해로 한다.

<표 2> 각 아미노산에 부여된 코드

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

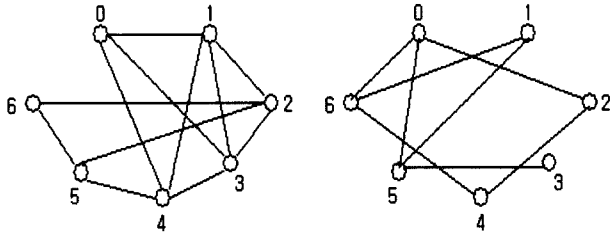
본 논문에서 적합도 평가는 <표 2>의 아미노산 코드에 값을 식 (1)의 비례 선택법(roulette wheel)으로 계산한다. 교배는 균일 교배 방법을 사용하는데 간선 염기 서열에서 제한효소의 인식부분을 특정 위치에 삽입하여 개체군을 형성하였기 때문에 인식 부분에 변동이 없도록 하기 위함이다. 돌연변이는 일점 돌연변이를 사용하며, 간선 염기 서열에서 제한효소인식 부분에 돌연변이가 일어나지 않도록 제한을 두어 세대수 만큼 반복한다.

$$f(x) = x + k | \sin(32x) |, 0 \leq x < \pi, k : \text{아미노산상수} \quad (1)$$

이렇게 하여 높은 적합도를 갖는 우수한 코드를 선택하고 주어진 반응횟수만큼 합성과 분리 과정을 거친다. 이 분리 과정에서 해가 될 가능성이 없는 것은 항체 친화력 반응, PCR(Polymerase Chain Reaction), 젤 전기 영동법 등과 같은 생물학적 연산자를 이용하여 미리 제거한다. 마지막으로 다시 한번 PCR을 이용하여 특정 부위의 서열을 증폭시킨다. 그리고 젤 전기 영동법으로 일정 길이의 염기 배열만을 추출하고, 항체 친화력 반응을 이용하여 그래프의 모든 정점을 한번만 방문한 경로를 선택하여 최종해로 한다.

#### 4. 실험 및 분석

(그림 11)에서 주어진 정점 7개인 MCP를 대상으로 ACO와 Adleman의 DNA 컴퓨팅 알고리즘을 비교 평가하였다.



(a) Maximal clique problem (b) Complementary graph  
(그림 11) MCP의 예제 그래프

실험은 P4, 1Ghx, RAM 256M의 PC에서 C언어를 사용하였고, 실험에서 사용된 파라미터들은 <표 3>과 같이 설정하였다. 원래 Adleman의 DNA 컴퓨팅 알고리즘은 1회의 단순한 합성과 분리 과정으로 이루어지지만, 본 실험에서는 ACO와 동일하게 반응횟수와 반복횟수를 곱한 총 반응횟수를 1000으로 설정하였다. 그리고 ACO는 DNA 코드가 가변 길이, Adleman의 DNA 컴퓨팅 알고리즘에서는 DNA 코드가 최소 10bp, 최대 20bp 사이의 고정길이를 설정하였다.

<표 3> DNA 컴퓨팅에 사용한 파라미터들

변수	ACO	Adleman의 DNA 컴퓨팅 알고리즘
집단 크기	100	100
세대수	100	100
교배 연산 비율	0.5	0.5
돌연변이 연산 비율	0.1	0.1
간선 표현 코드 길이	가변길이	10bp(최소), 20bp(최대)
가치 표현 코드 길이	가변길이	6bp
총 반응횟수	반복횟수	10
	반응횟수	100
생물학 실험 오류확률	0.01	0.01

<표 4> 제한 효소

제한 효소	인식 염기 서열
Aat II	GACGTC
Bgl I	AGATCT
EcoR I	GAATTC
Hind III	AAGCTT
Pst I	CTGCAG
Spe I	ACTAGT
Ssp I	AATAAT

MCP를 해결하기 위해 실험에 사용된 제한효소(restriction enzyme)는 <표 4>와 같으며, 생성된 염기 서열들 중에서 상보 그래프에 포함된 간선들의 서열을 삭제하기 위

해 사용된다. 만약 제한효소 인식 부분에 오류 즉, 하나의 염기 서열에 인식 부분이 여러 곳에 존재하는 경우나, 제한효소가 인식 부분을 정확히 인식하지 못하는 경우에는 re-pair enzyme를 이용하여 DNA 서열 중 특정 염기로 교체하여 인식하도록 하였다.

<표 5> ACO의 평균 적합도

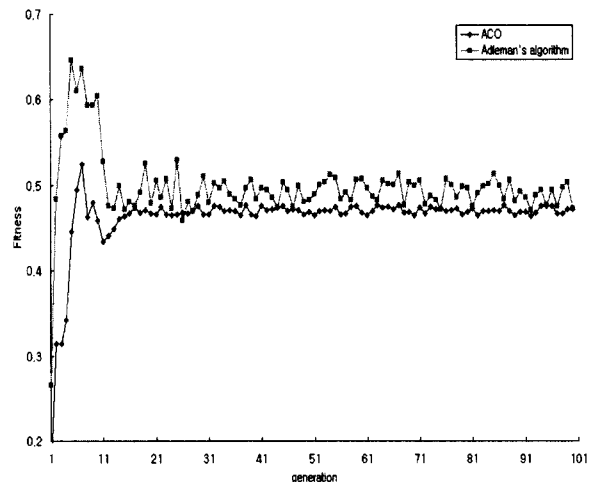
구분	ACO	Adleman의 DNA 컴퓨팅 알고리즘
평균 적합도	0.49859	0.426646

<표 5>에서 보는 것과 같이 ACO는 Adleman의 DNA 컴퓨팅 알고리즘보다 약 17% 높은 적합도를 보이며, 정점 7개에 대한 최대 클리크를 탐색하였다.

<표 6> 반응시간에 대한 최종해의 탐색 결과

반응시간(s)	실험횟수	해를 발견한 횟수		최종해의 개수	
		ACO	Adleman's	ACO	Adleman's
10 <sup>4</sup>	20	23	12	11	3
5×10 <sup>3</sup>	20	11	4	4	1
10 <sup>3</sup>	20	11	0	1	0

또한 ACO는 <표 6>에서 보는 바와 같이 Adleman의 DNA 컴퓨팅 알고리즘보다 약 4배정도 많은 최종해를 찾을 수 있었다. 그리고 제한효소의 적절한 사용으로 상보 그래프에 포함되지 않은 간선들을 삭제할 수 있어서, 낮은 생물학적 실험 오류율을 갖을 수 있어 빠른 시간에 우수한 해를 생성하였다.



(그림 12) MCP의 세대별 적합도

(그림 12)는 각 세대별 평균 적합도를 나타낸 그래프이다. x축은 세대수를 나타내며, y축은 적합도를 나타낸다. 그래프에서 보는 것과 같이 ACO는 15세대 이후 고른 적합도를 보이고 있다. 또한 13세대에서 20세대사이에서는 처음으로 서브그래프가 완전 그래프인 최대 클릭을 탐색할 수 있었다. 하지만 Aldeman의 DNA 컴퓨팅 알고리즘은 불규칙한 적합도를 보이며 51세대에서 78세대사이에서 처음으로 최대 클릭을 탐색할 수 있었다.

<표 7>은 실제 실험에 사용한 ACO의 간선 염기 서열과 가치염기 서열을 나타냈고, <표 8>은 Adleman의 간선 염기 서열과 가치 염기 서열을 나타냈다. 그리고 <표 9>와 <표 10>은 각각 ACO의 가변길이의 정점 코드와 Adleman의 고정길이의 정점 코드인 즉, 최소 염기 서열 10bp와 최대 염기 서열 20bp에서 정점 염기 서열을 나타낸다. (그림 11)에서 최대 클릭은 {2, 3, 4, 5} 또는 E0 -> V0 -> E1 -> V1 -> E2 -> E3 -> E4 -> E5 -> E6이며, 실험에 ACO를 적용한 결과 해를 염기 서열로 표시하면 다음과 같다.

GCTTGGGTAAGGGATCCTAACCTT

- > ATCCTCAATAA
- > AAGATGTAGCCCTCATAGGGTGCCAAG
- > ATTAAGTGA
- > CTTATGGCATTTTAGATCCCCGGGACT
- > AGTATGTTTAACGATTCCAGTAACGCA
- > TGCATGTAGCGTTTCCCGATAGGATCGAGG
- > CCTATGCTAGCTTCTAAATAGTCTC
- > GAGATGCTAACGGATCTCCC

<표 7> 실험에 사용한 ACO의 간선과 가치 염기 서열

순서	ACO의 간선 염기 서열 코드	가치 염기 서열 코드
0	GCTTGGGTAAGGGATCCTAACCTT	ATCCTCAATAA
1	AAGATGTAGCCCTCATAGGGTGCCAAG	ATTAAGTGA
2	CTTATGGCATTTTAGATCCCCGGGACT	ATAAGCTGGAATAA
3	AGTATGTTTAACGATTCCAGTAACGCA	ATCGTGGCATAG
4	TGCATGTAGCGTTTCCCGATAGGATCGAGG	ATTTGGTCTGA
5	CCTATGCTAGCTTCTAAATAGTCTC	ATAACGTAG
6	GAGATGCTAACGGATCTCCC	ATCATGTGTGA

<표 8> 실험에 사용한 Adleman의 간선과 가치 염기 서열

순서	코드 길이	Adleman의 간선 염기 서열 코드	가치 염기 서열 코드
0	10bp	GTACCAATGGCTT	AGCGCG
	20bp	GTCCTAATTGAGTCCGCTCTT	AGCGCG
1	10bp	AAGATGCCGGCCTAAG	GCCTGG
	20bp	AAGAATCCTATCGCCTGAACTCAAG	GCCTGG
2	10bp	CCTTATTATCGTAACT	CCGGGC
	20bp	CCTCCTAGTCTATCTGTAACCCACT	CCGGGC
3	10bp	AGTGGTTCGGTTGCA	TTCCAA
	20bp	AGTCGGAATCCAGCATACTGGTGA	TTCCAA
4	10bp	TGCATATGCAAGTAGG	GTTCTG
	20bp	TGCTCCTAATTGTCCCTGGTACAGG	GTTCTG
5	10bp	CCTCCGGAATATCTC	TGTGTG
	20bp	CCTCAACTTTACCTTGTGATCTCTC	TGTGTG
6	10bp	GAGAGTCAGTCCT	AATTCC
	20bp	GAGGATACTAGCCTCTCTAACCC	AATTCC

<표 9> 실험에 사용한 ACO의 정점 염기 서열

DNA 구성	ACO의 정점 염기 서열 코드
$E_0 \vee_0^0 E_1$	GCTTGGGTAAGGGATCCTAACCTTATCCTC AATAAAAGATGTAGCCCTCATAGGGTGCCA AG
$E_0 \vee_0^1 E_1$	GCTTGGGTAAGGGATCCTAACCTTAAGATG TAGCCCTCATAGGGTGCCAAG
$E_2 \vee_1^0 E_1$	AGTCCCGGGATCTAAAATGCCATAAGTCA CTTAATCTTGGCACCTATGAGGGCTACAT CTT
$E_2 \vee_1^1 E_1$	AGTCCCGGGATCTAAAATGCCATAAGCTT GGCACCTATGAGGGCTACATCTT
$E_2 \vee_2^0 E_3$	CTTATGGCATTTTAGATCCCCGGGACTATA AGCTGGAATAAAGTATGTTAACGATTCC AGTAACGCA
$E_2 \vee_2^1 E_3$	CTTATGGCATTTTAGATCCCCGGGACTAGT ATGTTAACGATTCCAGTAACGCA
$E_4 \vee_3^0 E_3$	CCTCGATCCTATCGGGAAACGCTACATGCA CTATGCCACGATTGCGTTACTGGAATCGTT AAACATACT
$E_4 \vee_3^1 E_3$	CCTCGATCCTATCGGGAAACGCTACATGC ATGCGTTACTGGAATCGTTAAACATACT
$E_4 \vee_4^0 E_5$	TGCATGTAGCGTTTCCCGATAGGATCGAG GATTTGGTCTGACCTATGCTAGCTTCTAA ATAGTCTC
$E_4 \vee_4^1 E_5$	TGCATGTAGCGTTTCCCGATAGGATCGAG GCCTATGCTAGCTTCTAAATAGTCTC
$E_6 \vee_5^0 E_5$	CGGGGAGATCCGTTAGCATCTCTACGTTA TGAGACTATTTAGAAGCTAGCATAGG
$E_6 \vee_5^1 E_5$	CGGGGAGATCCGTTAGCATCTCGAGACTAT TTAGAAGCTAGCATAGG

〈표 10〉 실험에 사용한 Adleman의 정점 염기 서열

DNA 구성	코드 길이	Adleman의 정점 염기 서열 코드
$E_0 \vee_0^0 E_1$	10bp	GTACCAATGGCTTAGCGCGAAGATGCC GGCCTAAG
	20bp	GTCCTAATTGAGTCCCGCCTCTTAGCG CGAAGVAATCCTATCGCCTTGAAC TCAAG
$E_0 \vee_0^1 E_1$	10bp	GTACCAATGGCTTAAGATGCCGGCCTA AG
	20bp	GTCCTAATTGAGTCCCGCCTCTTAAGV AATCCTATCGCCTTGAACCTCAAG
$E_2 \vee_1^0 E_1$	10bp	AGTTACGATAATAAGGCCAGGCCTTA GGCCGGCATCTT
	20bp	AGTGGGTTACAGATAGGACTAGGAGGC CAGGCCTTGAGTTCAAGCGATAGGAT TCTT
$E_2 \vee_1^1 E_1$	10bp	AGTTACGATAATAAGGCCTTAGGCCGG CATCTT
	20bp	AGTGGGTTACAGATAGGACTAGGAGGC TTGAGTTCAAGCGATAGGATCTT
$E_2 \vee_2^0 E_3$	10bp	CCTTATTATCGTAACTCCGGGCAGTGG TTCCGGTTGCA
	20bp	CCTCCTAGTCTATCTGTAACCCACTC CGGGCAGTCGCGAATCCAGCATACTGG TGCA
$E_2 \vee_2^1 E_3$	10bp	CCTTATTATCGTAACTAGTGGTTCCGG TTGCA
	20bp	CCTCCTAGTCTATCTGTAACCCACTA GTCCGGAATCCAGCATACTGGTGCA
$E_4 \vee_3^0 E_3$	10bp	CCTACTTGCATATGCATTTGGAATGCA AGGCCAACCT
	20bp	CCTGTACCAGGGGACAATTAGGAGCAT TCCAACTGTACCAGGGGACAATTAGG AGCA
$E_4 \vee_3^1 E_3$	10bp	CCTACTTGCATATGCATGCAAGGCCAA CCTACT
	20bp	CCTGTACCAGGGGACAATTAGGAGCAC CTGTACCAGGGGACAATTAGGAGCA
$E_4 \vee_4^0 E_5$	10bp	TGCATATGCAAGTAGGGTTCTGCCTTC CGGAATATCTC
	20bp	TGTCCTAATTGTCCCCTGGTACAGGG TTCTGCCTCAACTTTACCTTGTGATCT CCTC
$E_4 \vee_4^1 E_5$	10bp	TGCATATGCAAGTAGGCCTTCCGGAAT ATCTC
	20bp	TGTCCTAATTGTCCCCTGGTACAGGC CTCAACTTTACCTTGTGATCTCCTC
$E_6 \vee_5^0 E_5$	10bp	AGGAGGACTGACTCTCCACACAGAGAT ATTCCGGA
	20bp	GGGTTAGAGAGGCTAGTATCTCCACA CAGAGGATCACAAAGGTAAGTTGA GG
$E_6 \vee_5^1 E_5$	10bp	AGGAGGACTGACTCTCGAGATATTCCG GA
	20bp	GGGTTAGAGAGGCTAGTATCTCGAGG AGATCACAAAGGTAAGTTGAGG

## 5. 결 론

본 연구에서는 MCP에 대한 기존 DNA 컴퓨팅의 문제점을 분석하고, 이를 해결하기 위해 DNA 컴퓨팅에 DNA 코딩 방법을 적용한 ACO를 제안하였다.

ACO는 가변길이의 DNA 코드를 생성할 수 있으며, 불필요한 정점을 제한 효소를 사용하여 각 세대에서 제거하였고, 짧은 코드를 가지고도 넓은 범위의 코드를 표현할 수 있었다. 또한 서열 탐색 시간과 생물학적 오류율이 Adleman의 DNA 컴퓨팅 보다 15%를 줄임으로써, 최종해의 수는 4배 정도 증가하였다.

향후 과제로는 더 많은 정점을 갖는 MCP에 대한 실험과 NCBI에 등록된 DNA 염기 서열을 적용한 연구가 필요할 것이다.

## 참 고 문 헌

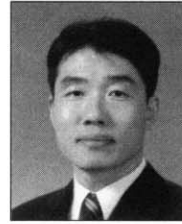
- [1] L. M., Adleman, "Molecular computation of solutions to combinatorial problems," Science, 266, pp.1021-1024, 1994.
- [2] M. Amos, "Theoretical and experimental DNA computation," Bull, European Assoc, for Theor, Computer Science, 67, pp.125-138, 1999.
- [3] J. D. Watson, M. Gliman, J. Wikowski, M. Zoller, Recombinant DNA, 2nd Ed., Scientific American Books, New York, 1992.
- [4] Q. Ouyang, P. D. Kaplan, S. Liu, A. Libchaber, "DNA Solution of the Maximal Clique Problem," Science 278, pp. 446-449, 1997.
- [5] D. van Noort, F.-U. Gast, J. S. McCaskill, "DNA Computing in Microreactors," submitted to Lecture Notes in Computer Science, DNA 7, Florida, 2001.
- [6] M. R. Garey, D. S. Johnson, Computers and Intractability : A Guide to the Theory of NP-completeness, W. H. Freeman and Company, San Francisco, p.211 1979.
- [7] T. Yoshikawa, T. Furuhashi, Y. Uchidawa, "Acquisition of Fuzzy Rules of Constructing Intelligent Systems using Genetic Algorithm based on DNA Coding Method," Proceedings of International Joint Conference of CFS/IFIS/SO FT '95 on Fuzzy Theory and Applications.
- [8] T. Yoshikawa, T. Furuhashi, Y. Uchidawa, "The Effect of Combination of DNA Coding Method with Pseudo-Bacterial GA," Proceeding of the 1997 IEEE International InterMag. 97 Magnetics Conference 1997.



**김 은 경**

e-mail : rotnrwk@kongju.ac.kr  
2001년 공주대학교 화학과(학사)  
2003년 공주대학교 대학원 컴퓨터공학과  
(공학석사)  
2003년~현재 공주대학교 대학원 컴퓨터  
공학과 박사과정

관심분야 : 바이오인포메틱스, 인공생명, DNA 컴퓨팅, 유전자  
알고리즘 등



**이 상 용**

e-mail : sylee@kongju.ac.kr  
1984년 중앙대학교 전자계산학과(공학사)  
1988년 일본동경대학대학원 총합이공학  
연구과(공학석사)  
1988년~1989년 일본 NEC 중앙연구소  
연구원

1993년 중앙대학교 일반대학원 전자계산학과(공학박사)  
1996년~1997년 University of Central Florida 방문교수  
1993년~현재 공주대학교 정보통신공학부 교수  
관심분야 : 인공지능, 에이전트, 컴퓨터게임, 바이오인포매틱스