퍼지 및 지능시스템학회 논문지 2006, Vol. 16, No. 2, pp. 222-227

DNA 컴퓨팅과 진화 모델을 이용하여 Traveling Salesman Problem를 해결하기 위한 DNA 서열 생성 알고리즘

A DNA Sequence Generation Algorithm for Traveling Salesman Problem using DNA Computing with Evolution Model

김은경* · 이상용**
Eun-Gyeong Kim* · Sang-Yong Lee**

* 공주대학교
** 공주대학교 컴퓨터공학부(교신저자)

요 약

현재 막대한 병렬성을 갖는 DNA 컴퓨팅을 이용하여 Traveling Salesman Problem (TSP)를 해결하기 위한 연구가 진행되고 있다. 하지만 기존의 방법은 그래프 문제의 표현에서 DNA의 특성을 고려하지 않아, 실제 생물학적 실험 결과와의 차이가 발생하고 있다. 따라서 DNA의 특성을 반영하고 생물학적 실험 오류를 줄일 수 있는 DNA 서열 생성 알고리즘이 필요하다.

본 논문에서는 DNA 컴퓨팅에 진화 모델의 하나인 DNA 코딩 방법을 적용한 DNA 서열 생성 알고리즘을 제안한다. 제안한 알고리즘은 TSP에 적용하여 기존에 단순 유전자 알고리즘과 비교하였다. 그 결과 제안한 알고리즘은 오류를 최소화한 우수한 서열을 생성하고 생물학적 실험 오류율도 줄일 수 있었다.

Abstract

Recently the research for Traveling Salesman Problem (TSP) using DNA computing with massive parallelism has been. However, there were difficulties in real biological experiments because the conventional method didn't reflect the precise characteristics of DNA when it express graph. Therefore, we need DNA sequence generation algorithm which can reflect DNA features and reduce biological experiment error.

In this paper we proposed a DNA sequence generation algorithm that applied DNA coding method of evolution model to DNA computing. The algorithm was applied to TSP, and compared with a simple genetic algorithm. As a result, the algorithm could generate good sequences which minimize error and reduce the biologic experiment error rate.

Key words : DNA 컴퓨팅, 진화 모델, Traveling Salesman Problem (TSP), DNA 코딩 방법, DNA 서열 생성 알고리즘

1. 서 론

TSP는 다항 시간에 풀리지 않는 NP-complete 문제로써, 이 문제를 해결하기 위해 근사적 알고리즘, 확률적 알고리즘, 유전자 알고리즘 등을 사용하고 있다[1]. 하지만 정점의 수가 커지면 경로를 탐색할 수 있는 경우의 수가 기하급수적으로 증가하여 이러한 알고리즘들로 해결하기 어렵다. 그래서 이를 해결하기 위해 DNA가 갖는 막대한 병렬성과 상보적인 특징을 이용한 DNA 컴퓨팅을 사용하고 있다. DNA 컴퓨팅은 1994년 Adleman 이후에 여러 가지 NP-complete 문제들을 적용하여 DNA 염기 서열의 디자인 및 합성 과정에 대한 결과를 평가하고 있다.

접수일자 : 2006년 1월 16일
완료일자 : 2006년 4월 12일

하지만 DNA 컴퓨팅으로 TSP를 해결할 때 다음과 같은 세가지 문제점들이 발견된다. 첫째, 단순한 합성과 분리 과정만으로 해를 찾기 때문에 많은 시간과 노력이 요구된다. 둘째, 실제 생물학 실험 방법을 연산자로 사용하므로 연산자의 불확실성과 오류의 가능성을 내포하고 있다. 셋째, 그래프 문제를 DNA 코드로 변환할 때, DNA의 특성을 정확하게 반영하지 못하는 문제점을 가지고 있다.

이러한 문제점들을 해결하기 위해 최적의 해결 방법은 아니지만 부분적으로 해결하기 위해 새로운 연산자에 의한 실험 방법의 개발, 생물학 실험 방법의 메커니즘 이해, 유전자 알고리즘을 이용한 모델 등의 연구가 있다[2]. 그리고 잘못된 결합이 적은 DNA의 서열을 생성하기 위해 여러 가지 적합도 함수 또는 전부 탐색[3], 랜덤 탐색[4], 그래프 방법[5] 등의 알고리즘들을 사용하여 DNA 서열 생성을 연구하고 있다.

본 논문에서는 DNA 컴퓨팅과 진화 모델 중 DNA 코딩 방법을 적용한 DNA 서열 생성 알고리즘을 제안한다. 제안

한 알고리즘은 TSP에 적용하여 기존에 단순 유전자 알고리즘과 비교하였다. 그 결과 DNA 서열 생성 알고리즘은 그래프 문제의 표현에서 DNA의 특성을 잘 표현하고, 서열의 길이를 조절할 수 있었다. 또한 생물학적 실험 오류율을 줄임으로써 빠른 시간 내에 최적의 서열을 생성할 수 있다.

2. 관련 연구

2.1 DNA 컴퓨팅

DNA 컴퓨팅은 실제 생체 분자인 DNA나 RNA와 같은 살아 있는 세포를 응용한 것으로, 화학적으로 합성된 DNA를 계산의 수단 및 정보 저장 매체로 사용한다. DNA는 4개의 염기인 A (Adenine), T (Thymine), C (Cytosine), G (Guanine)가 2중 나선 구조로 구성되어 있다. 이들 염기에 대용량 데이터를 저장할 수 있는 메모리 기능을 가지고 있으며, 정해진 규칙에 의해 상호 보완적인 방식의 Watson-Crick 결합을 하고 있다. 그리고 복잡한 염기 조합의 패턴은 하나의 유전 정보를 담고 있으며, 인체내에서 자연 발생하는 효소에 의해 읽혀지고 있다. 효소는 생물학 실험 방법들과 함께 DNA 컴퓨팅의 연산자로 사용되고 있다. 따라서 일반 컴퓨터의 연산자를 사용하지 않고, Melting과 Annealing, Ligation, Polymerase Chain Reaction(PCR), Enzyme reaction, Gel Electrophoresis, Antibody Affinity 등의 여러 가지 실험 과정들을 연산자로 사용한다.

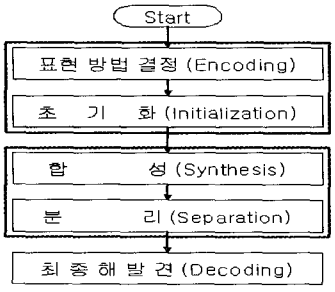

그림 1. Adleman의 DNA 컴퓨팅 알고리즘
Fig. 1. Adleman's DNA computing algorithm

기본적인 DNA 컴퓨팅은 Adleman의 Hamiltonian Path Problem (HPP)를 해결한 모델이며, [그림 1]은 Adleman의 DNA 컴퓨팅 알고리즘을 나타낸 것이다. Adleman은 HPP를 해결하기 위해 시작 정점에서 도착 정점의 모든 정점을 정확히 한번만 포함하는 경로를 찾기 위해 모든 가능한 해답을 DNA 코드로 표현하고, 단 한번의 합성과 분리 과정을 통해 후보해를 생성한 다음, 찾는 해가 있는지 검사하여 답을 제시하였다[6].

DNA 컴퓨팅 알고리즘은 Adleman의 방법을 기초로 하여 전부 탐색[3], 랜덤 탐색[4], 그래프 방법[5] 등의 다양한 알고리즘으로 NP-complete 문제를 해결하고 있다. 이러한 알고리즘은 다양한 생화학적 특성과 실험 조건들을 반영하여 DNA의 서열을 생성할 수 있다.

2.2 진화 모델

진화 모델은 자연세계의 진화 과정을 컴퓨터상에서 시뮬레이션 함으로써 복잡한 실세계의 문제를 해결하고자 하는 계산 모델이다. 진화 모델 알고리즘은 구조가 간단하고 방법이 일반적이어서 응용범위가 매우 넓으며, 특히 적응적 탐색과 학습 및 최적화를 통한 공학적인 문제의 해결에 많이 이용되고 있다. 종류로는 유전자 알고리즘, 진화전략, 진화 프로그래밍, 유전자 프로그래밍, DNA 코딩 방법이 있다.

진화 모델의 하나인 DNA 코딩 방법은 1995년 Yoshikawa가 제안한 변형된 형태의 유전자 알고리즘이다[7]. 일반적인 유전자 알고리즘은 0, 1의 이진수를 사용하지만, DNA 코딩 방법은 A(Adenine), G(Guanine), T(Thymine), C(Cytosine)의 4진수를 사용하여 선택, 재생, 교배, 돌연변이 연산을 한다. 그리고 [그림 2]와 같이 A, T, G, C 중 3개의 염기(코돈 ; codon)가 하나의 의미 단위인 아미노산을 지정할 수 있으며, 그 수는 중복을 제외한 20가지가 있다.

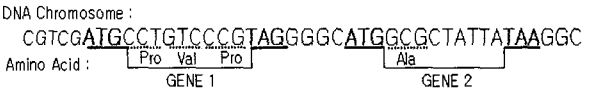

그림 2. DNA 염색체의 번역 예
Fig. 2. Example of DNA chromosome translation

DNA 코딩 방법의 특징을 살펴보면, 염색체의 중복을 효율적으로 표현할 수 있다. 그리고 하나의 아미노산을 만드는 코돈이 여러 개이므로 지식 표현이 쉬우며, 교차점이 임의로 주어지기 때문에 염색체의 길이가 가변적이다. 그러므로 더욱 더 생물학적으로 가깝게 모델링 할 수 있다.

2.3 Traveling Salesman Problem (TSP)

TSP는 n개의 도시와 도시간의 거리가 주어질 때, 출발 도시에서 시작하여 모든 도시를 단 한번만 방문하고 원래의 출발지로 되돌아오는 최단 길이의 경로를 찾는 문제이다. 다시 말하면 각각의 도시들에 대한 이동 경로가 순열로 주어질 때, 각 도시와 다음 도시와의 유클리드 거리의 합이 최소가 되는 경로를 선택하는 것이다. TSP는 그래프의 정점의 수가 커지면 경로를 탐색할 수 있는 경우의 수가 기하급수적으로 증가한다. 또한 컴퓨터로 계산 할 수 있는 시간 복잡도는 적어도 O((n-1)!)로 표현할 수 있다. 따라서 그래프의 크기가 커지면 해결하기 어려운 문제점을 갖고 있다.

이를 해결하기 위해 다양한 연구들이 시도되고 있다. 대표적으로 분기 한정법 (Branch-and-Bound) 알고리즘이나 동적 프로그래밍 (Dynamic Programming) 알고리즘과 같은 최적해를 구하는 방법이 있고, 확률적 탐색 휴리스틱 (probability search heuristic)에 근거해서 근사해를 구하는 유전자 알고리즘 등 다양한 방법들이 시도되고 있다. 또한 Adleman의 DNA 컴퓨팅 알고리즘을 기반으로 TSP의 간선에 일정한 가중치를 적용하여 DNA 코드를 생성하는 규칙을 제안한 Narayanan과 Zorbalas의 연구를 볼 수 있다[8].

3. DNA 서열 생성 알고리즘

DNA 서열 생성 알고리즘은 [그림 3]과 같이 전처리 과정과 후처리 과정으로 구분된다. 이 알고리즘은 TSP를 DNA 컴퓨팅으로 해결할 때, 발생했던 문제점들을 해결하기 위해 DNA 컴퓨팅의 기본적인 모델인 Adleman의 DNA 컴퓨팅 알고리즘과 기존의 DNA 컴퓨팅에서 사용되지 않았던 진화 모델의 하나인 DNA 코딩 방법을 적용하여 개선한 것이다.

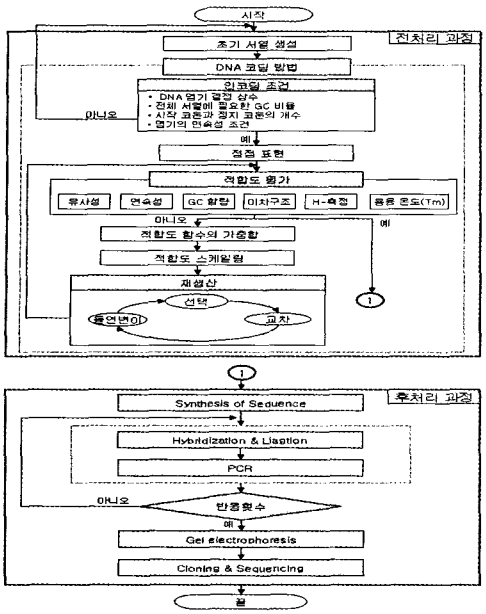

그림 3. DNA 서열 생성 알고리즘의 흐름도
Fig. 3. Flowchart of DNA sequence generation algorithm

3.1 전처리 과정

1) Init_Seq Module

초기 DNA 서열의 생성은 문제에 사용할 긴 서열을 Randomized Block Design (RBD)을 적용하여 질량이 서로 다른 염기를 적합하게 배열할 수 있도록 하는 DNA 염기 결정 상수와 최대 발생 빈도와 최소 발생 빈도로 나타낸 전체 서열에 필요한 GC 비율을 설정하였다. 그리고 정점을 긴 DNA 서열로부터 분리하는데 사용하기 위한 시작 코돈과 정지 코돈의 개수와 동일한 염기의 발생을 막기 위한 염기의 연속성 조건을 함께 설정하도록 하였다. 이렇게 설정된 조건들이 DNA 염기인 A, T, G, C의 고른 분포와 DNA의 다양한 특징을 고려한 <식1>을 사용하였다.

$$seq = d \times int(RAND() \times GC_{ratio} \times Codon \times Con + 0.5) \quad <1>$$

2) Vertex_Sev Module

위에서 생성된 긴 DNA 서열을 DNA 코딩 방법을 사용하여 시작 코돈인 ATG 앞에서 잘라 가변 길이의 정점을 생성한다.

3) Fitness Module

위 과정으로 결정된 정점 서열은 오류의 발생 가능성이 적은 아래의 개선된 여러 가지 적합도 평가 기준으로 평가하였다. 또한 여러 가지 적합도 평가 기준은 단일한 형태가 아닌 여러 가지 적합도를 포함한 적합도 함수의 가중합 형태이므로, <식6>을 사용하였다. 그리고 가중합을 적합도 스케일링하여 <식7>로 평가 하였다.

(1) 유사성 : 생성된 DNA 서열의 유사성을 평가하여 각 서열간의 고유성만을 측정하도록 <식2>를 사용하였다.

$$f_S = \sum_{s=0}^{s}\sum_{t=0}^{t} D_w(0,s,i:0,t,j) + H(s,\delta^j(t)) \quad <2>$$

$$D_w(0:s:i,0:t:j) = min \begin{matrix} dw(0:s:i,0:t:j) + m(s_i,t_j) \\ dw(0:s:i,0:t:j) + d(s_i,-) \\ dw(0:s:i,0:t:j) + ins(-,t_j) \end{matrix}$$

$D_w(0:s:i,0:t:j)$는 서열 s의 i번째 위치의 염기와 서열 t의 j번째 위치의 염기의 유사성을 평가한 값이고, $H(s,\delta^j(t))$는 서열 s와 t의 길이에 대한 해밍 거리(Hamming distance)의 값이다. 그리고 $m(s_i,t_j)$는 서열 s의 i번째와 서열 t의 j번째가 일치하는지를 의미하며, $d(s_i,-)$는 삭제를 의미하고 $ins(-,t_j)$는 삽입을 의미한다.

(2) H-측정 : 유사성과 같은 역할을 하지만 방향성에서 서로 상보적이다. 그러므로 <식3>은 <식2>로부터 상보적 특징을 고려한 표현이다.

$$f_H = \sum_{s=0}^{s}\sum_{t=0}^{t} (D_w(0,s,i:j,t,0) + H(s,\delta^j(t))) \quad <3>$$

(3) 이차 구조 : 한 서열이 자체적으로 휘어져서 상보 결합하는 것을 막기 위해, 발생 가능한 모든 경우를 수치화하여 <식4>로 계산하였다.

$$f_{sec} = \sum_{s=0}^{s}\sum_{j=i+6}^{s} (D_w(0,s,i:j,s,0) + n - H(s_i,\delta^w(s))) \quad <4>$$

(4) 연속성 : 특정 염기가 연속적으로 발생되는 경우에는 DNA 구조가 불안정하기 때문에 원하지 않는 결합이 발생할 가능성이 매우 높다. 이를 제어하기 위해 [9]에서 사용된 식을 사용하였다.

(5) 용융 온도 (Tm): Tm을 계산하기 위해 [10]에서 사용된 Nearest Neighbor (NN 모델)로 설정하였다.

(6) GC 함유량 : 모든 DNA 염기 서열에서 G와 C의 비율을 말하며, <식5>로 계산하였다.

$$f_{GC} = \frac{\sum_{i=0}^{s} Seq(G,C)}{s} \quad <5>$$

(7) 적합도 평가 함수의 가중합 : 여섯 가지의 적합도 평가 기준을 <식6>으로 평가 하였다. 그리고 실험 조건의 만족도는 0~1 사이의 값을 가지도록 설정하였다.

$$f_{total} = \sum_{i=0}^{n} H(S_{cond},\ \gamma,\ \delta(f_S,\ f_H,\ f_{GC},\ f_{sec},\ f_{con},\ f_{NN})^2) - \overline{f_{(range)}} \quad <6>$$

(8) 적합도 스케일링: 적합도 평가 함수의 가중합은 0~1 사이의 값을 가지므로, 실험 조건의 만족도를 높이기 위해서 개선된 <식7>의 스케일링 윈도우를 이용하여 실험 조건의 만족도를 조정하였다.

$$f(k) = f(s(k)) = F(x(k)) - \gamma \quad <7>$$

γ는 $\forall k \in [0, \infty]$에 대해 $f(k) \geq 0$의 관계를 보장하도록 결정되는 적합도 스케일링 상수로, 탐색 효율을 조정할 수 있다.

4) Cross_Mut_Op Module

(1) 선택 연산 : 선택은 적응도가 가장 높은 개체에서 도태될 가능성이 있는 확률적 선택 방법의 단점을 보완한 엘리트 보존 전략 (elitist preserving strategy)을 사용하여, 생물학 실험에 효율적으로 반응할 수 있도록 <식8>로 수정하였다.

$$P(g+1) = select\ ' s_{(\mu,\lambda)}(P^n(f_{total}))\ or\ s_{(\mu+\lambda)}(P(f_{total}) \cup P^n(f_{total}))' \quad <8>$$

$s_{(\mu,\lambda)}$와 $s_{(\mu+\lambda)}$는 각각 정점 서열의 위치이고, (μ, λ)와 $(\mu+\lambda)$의 선택을 의미한다.

(2) 교차 연산 : 우수한 정점 서열을 서로 교차시켜 새로운 서열로 생성하는 연산이다. 교차 연산을 위한 교차점은 랜덤한 위치에서 교차시켜 선택된 정점 서열의 고유성을 확보하였다.

$$\alpha'_{\mu} = \gamma'(P(f_{total})) \quad <9>$$

$$\alpha'_{\mu} = \gamma'(P(f_{total})\ and\ P^n(f_{total})) \quad <10>$$

$$\alpha'_{\mu} = \chi \times \delta'_{ij} + (1 - (P(f_{total}/\gamma)) \quad <11>$$

그리고 <식9>는 일점 교차, <식10>은 복수점 교차를 위한 식이고, γ'은 교차의 위치를 의미한다. <식11>은 균일 교차 연산을 위한 식으로, χ는 0과 1사이에서 발생되는 랜덤 값이다. 그리고 δ'_{ij}는 서열의 최대 길이를 의미하며, γ는 적합도 스케일링 상수이다.

(3) 돌연변이 연산 : 교차로 생성된 정점 서열에 대하여 개체의 다양성을 높이기 위해 독립적으로 랜덤하게 발생하도록 하였다. 돌연변이가 일어날 확률은 0에서 1사이의 값으로 <식12>와 같은 확률적 돌연변이가 일어나도록 하였다.

$$\xi = \frac{\{\ exp(\gamma^2, N_i(0,1)) + seq\ \}}{2} \quad <12>$$

여기서 <식12>는 생화학 실험 조건을 만족하는 seq의 값과 랜덤한 위치에서 돌연변이가 일어날 확률 $N_i(0,1)$을 이용하여 돌연변이가 발생된다. 이와 같은 전처리 과정을 통해 가변 길이의 정점이 생성되며, 간선은 정점을 연결하는 연결 정보를 포함해야 하므로 3.3절에서 자세히 다루도록 한다.

3.2 후처리 과정

1) Synthesis Module

해를 찾기 위해 실험에 사용할 모든 정점과 간선의 서열을 넣고 섞는 과정이다. 이때 정점과 간선은 단일 가닥 DNA의 형태이며, 이 과정에서 모든 서열은 서로 결합을 하지 않아야 한다.

2) Hybri_Liga Module

합성된 서열을 연결 정보인 간선을 통해 이중 가닥의 긴 DNA 서열로 만들어 주는 과정이다. 이 과정은 생물학 연산자인 결합과 결찰을 사용한다.

3) PCR Module

생성된 이중 가닥의 서열을 PCR 연산을 사용하여 해가 될 가능성이 있는 서열을 증폭해서 해를 찾을 수 있는 확률을 높이는 과정이다.

4) Gel_Clon_Seq Module

증폭된 서열을 분자량에 따라 이동 거리가 다르다는 성질을 이용한 겔 전기 영동으로 특정 길이의 DNA 서열을 확인하고 추출한다. 추출된 결과를 좋은 해인지 판단하기 위해 클로닝을 거쳐 시퀀싱으로 판독한다.

3.3 TSP의 서열 디자인

본 논문에서는 DNA 서열 생성 알고리즘을 7개의 정점(city)과 5개의 가중치(cost), 25개의 간선(road)으로 설계된 [그림 4]의 TSP 그래프를 이용하였다.

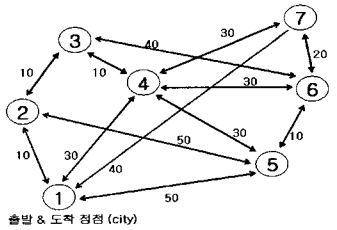

그림 4. TSP 그래프
Fig. 4. TSP graph

실험에 사용된 서열은 잘못된 결합, 이차 구조 등의 생물학적 실험 오류를 최대한 줄이기 위해, 개선된 적합도 평가를 거쳐 생성된 [표 1]을 사용하였다. 가중치는 편의상 5가지의 cost1~cost5 (10, 20, 30, 40, 50)로 나누었다.

표 1. TSP의 정점과 가중치 서열
Table 1. Vertexes and weights sequence of TSP

		서열 (5′ → 3′)	Tm (℃)	GC (%)
정점 (City)	1	ATGACGTGGGCATGAAAGTCTC	51.79	50
	2	ATGAAGTTCGTGAACGTCGC	49.71	50
	3	ATGTTCCAGGTTCGCATTCGTC	52.2	50
	4	ATGACACCACGGCTCCATTTGTAG	50.45	50
	5	ATGTTCTGCTTTTGACTCTCACCCCG	50.43	50
	6	ATGCTTTGCTCTGTACGCCAAG	51.87	50
	7	ATGGCTTGTGCTTCTGTACCTCCACT	51.38	50
가중치 (Cost)	1(10)	ATGCGAATGGTCTCATT	47.41	41
	2(20)	ATGTCGTTTAGGGAGACCTAGGTAA	52.92	44
	3(30)	ATGGCGATATCCGATCGAA	54.82	47
	4(40)	ATGGAGGCAGGATCCGATAGAA	57.41	50
	5(50)	ATGGGCCGACGATATCGAA	58.84	53

각 정점과 가중치 서열의 연결 정보를 갖고 있는 간선의 생성은 적합도 평가를 거친 7개의 정점과 5개의 가중치 서열을 바탕으로 생성된다. 간선 생성은 먼저 V_i 정점에 대해 시작 코돈인 ATG 코드 앞에서 끊어 간선을 표현하지 않고 AT*(ATT, ATC, ATA)의 3종류를 지정한다. 그리고 V_{i+1} 정점에 대해 정지 코돈인 TGA, TAA, TAG를 지정한다. 마지막으로 연결하려는 두 정점의 간선 표현은 V_i 정점에서 처음 나타나는 AT*와 V_{i+1} 정점에서 처음 나타나는 정지 코돈을 가중치를 포함하여 상보 결합한다. 만약 V_{i+1} 정점에서 정지 코돈이 없을 경우에는 정점의 염기 서열 1/2bp를 간선 생성에 사용한다.

하지만 짧은 서열의 경우, 위의 방법을 사용하면 처음에 설계하고자 했던 특성을 반영한 서열을 생성하기 힘들다. 따라서 50bp 이하의 짧은 길이로 설계된 서열로부터의 간선 표현은 위의 표현과 같지만 출발 정점의 서열에서 절반의 뒷부분과 도착 도착의 서열에서 절반의 앞부분을 이용하는 부분만 다르게 처리하여 [그림 5]와 같이 생성하였다.

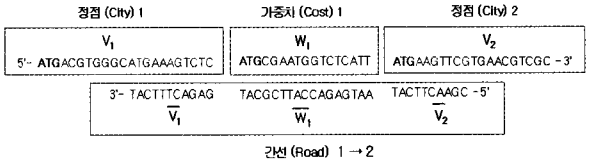

그림 5. 짧은 길이의 정점을 이용한 간선의 코드 표현
Fig. 5. Expression of edges code using vertexes with short length

4. 실험 및 분석

4.1 컴퓨터 시뮬레이션 결과 및 분석

본 실험에서는 [그림 4]의 TSP 그래프를 사용하여, 제안한 알고리즘과 단순 유전자 알고리즘을 사용한 DNA 컴퓨팅을 비교 평가 하였다. 실험에 사용된 파라미터는 [표 2]와 같이 설정하였다.

표 2. TSP에서 사용한 파라미터
Table 2. Parameters of TSP

변 수		값
집단 크기		1000
세대수		200
교차 연산 비율		30% (0.3)
돌연변이 연산 비율		1% (0.01)
코드 길이	단순 유전자 방법	20mer
	DNA 서열 생성 방법	10mer ~ 30mer
반복 횟수		1000

[표 3]은 DNA 서열 생성 알고리즘의 전처리 과정으로부터 생성된 TSP의 정점 서열과 단순 유전자 알고리즘을 이용하여 생성된 정점 서열의 적합도 평가를 비교하였다. 그 결과, 연속성은 서로 비슷한 결과 값을 보이지만 유사성, H-측정, 이차 구조, Tm은 제안한 알고리즘으로부터 생성된 정점 서열이 전반적으로 좋은 결과를 보인다. 특히 서열의 안정성을 나타내는 Tm 값은 49~52℃로 변위가 작은 반면 단순 유전자 알고리즘의 Tm 값은 24~47℃로 변위가 매우 큰 것을 확인할 수 있다.

표 3. 정점 서열의 적합도 평가 결과
Table. 3. Fitness estimation result of vertex sequence

정점 서열 (5' → 3')	유사성	H-측정	이차 구조	연속성	Tm	GC 함유량
단순 유전자 알고리즘으로 생성한 정점						
CCTAGTCCTATCTGTAACCC	83	238	4	3	24.99	50
GTCCTAATTGAGTCCCGCAT	82	118	3	3	41.89	50
GATACTAGCCTGTGTAACCC	88	237	1	3	33.67	50
TCCTAATTGTCCCCTGGTAC	83	304	2	4	36.85	50
CGCGAATCCAGCATACTGTT	93	266	1	2	47.87	50
AATCCTATCGCCTTGAACCG	90	223	4	2	45.3	50
CGGCTTTACCTTGTGATCTC	95	290	4	3	42.89	50
DNA 서열 생성 알고리즘으로 생성한 TSP의 정점						
ATGACGTGGGCATGAAAGTCTC	81	176	1	3	51.79	50
ATGAAGTTCGTGAACGTCGC	75	176	0	2	49.71	50
ATGTTCCAGGTTCGCATTCGTC	80	178	1	2	52.2	50
ATGACACCACGGCTCCATTTGTAG	79	194	0	3	50.45	50
ATGTTCTGCTTTTGACTCTCACCCCG	70	202	1	4	50.43	50
ATGCTTTGCTCTGTACGCCAAG	78	200	0	3	51.87	50
ATGGCTTGTGCTTCTGTACCTCCACT	73	214	0	2	51.38	50

[그림 6]은 DNA 서열 생성 알고리즘으로 생성한 TSP 정점 서열과 단순 유전자 알고리즘으로 생성한 정점 서열을 세대수에 따라 평균 적합도 결과를 나타낸 그래프이다. 그 결과, 단순 유전자 알고리즘으로 생성된 서열은 모든 세대에서 불규칙한 평균 적합도를 생성하였으며, DNA 서열 생성 알고리즘으로 생성된 서열은 50세대 이후에 평균 적합도의 변위가 $9 \le f(x) < 14$로, 우수한 자식 개체를 지속적으로 유지하고 있음을 확인할 수 있다. 따라서 제안한 알고리즘은 50세대 이후에 생성된 자식 개체 중에서 우수한 해를 찾을 수 있다.

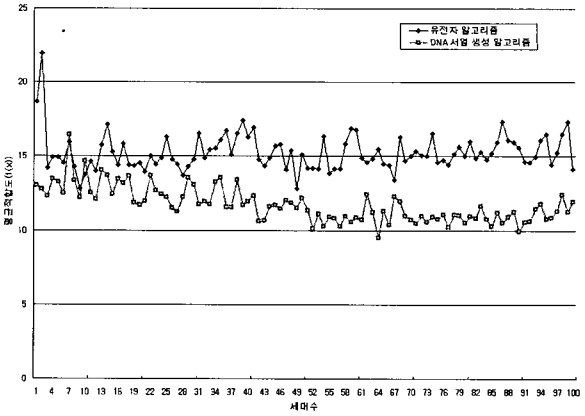

그림 6. 세대수에 따른 평균 적합도
Fig. 6. Average fitness by generations

4.2 생물학 시뮬레이션 결과 및 분석

DNA 컴퓨팅에서 DNA 올리고뉴클레오티드의 설계는 잘못된 결합, 이차 구조 등을 최소화 할 수 있기 때문에 중요하다. 그러므로 설계된 [표 1]의 서열이 예기치 않은 결합이 일어나는지 확인하기 위해, 정점의 서열을 랜덤하게 결합하여 그 결과를 10% 폴리아크릴아미드 겔 전기 영동으로 확인하였다. 그 결과 [그림 7]을 얻을 수 있었다. lane1~lane7은 각각 정점1~정점7을 확인한 결과이며, lane8~lane13은 2개의 정점을 랜덤하게 결합한 결과이다. 그리고 lane14~lane16은 3개의 정점을 결합한 결과이다(lane8: 정점1과 2, lane9: 정점2와 3, lane10: 정점6과 7, lane11: 정점3과 4, lane12: 정점4와 5, lane13: 정점5와 6, lane14: 정점1과 2, 3, lane15: 정점3과 4, 5, lane16: 정점5와 6, 7). 그 결과는 예상했던 대로 각 정점들 간에 결합 없이 각각의 밴드를 확인할 수 있었다. 따라서 DNA 서열은 잘 설계되었고, 실험에 의한 오류를 최소화 할 수 있다.

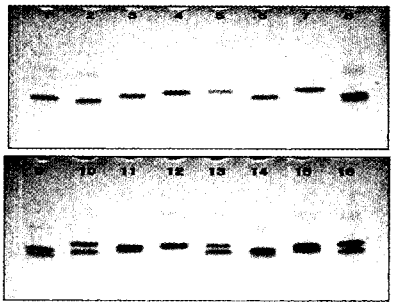

그림 7. 각 정점들의 hybridization 결과
Fig. 7. Hybridization result for vertexes

오류가 없는 정점과 가중치 서열은 긴 경로를 생성하기 위해 반응시간과 시약을 제안하여, 결합과 결찰 과정을 수행하였다. 그리고 대략 300~350bp 정도에서 elution하여 PCR하였다. 그리고 T-easy vector에 클로닝하여 시퀀싱으로 결과를 확인하였다. [그림 8]은 TSP의 전체 경로 중 일부분의 정점과 간선의 서열을 확인한 결과물이다. 그 결과 319bp의 최소의 길이를 갖는 정점1→가중치1→정점2→가중치1→정점3→가중치1→정점4→가중치3→정점5→가중치1→정점6→가중치2→정점7→가중치4→정점1 (총 가중치 : 130)의 경로를 확인할 수 있었다.

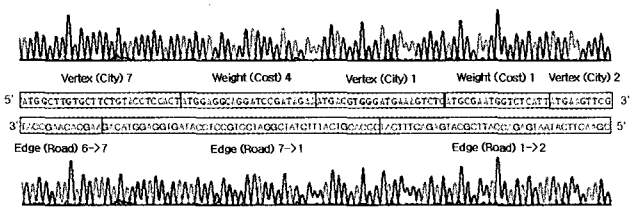

그림 8. 정점1 주변의 시퀀스
Fig. 8. Sequences around vertex 1

5. 결 론

본 연구에서는 DNA 컴퓨팅으로 TSP를 해결할 때 발생되는 문제점들을 분석하고, 이를 해결하기 위해 DNA 서열 생성 알고리즘을 제안하였다. DNA 서열 생성 알고리즘은 생물학적으로 더욱 가깝게 모델링 할 수 있는 진화 모델의 하나인 DNA 코딩 방법을 도입한 것으로, 생물학적 실험 오류의 발생 가능성이 낮은 DNA 서열을 생성하고자 하였다.

실험에서는 제안한 DNA 서열 생성 알고리즘과 단순 유전자 알고리즘을 TSP에 적용하여 유사성, H-측정, 이차 구조, 연속성, Tm의 적합도 평가하였다. 그 결과, 단순 유전자 알고리즘으로 생성된 서열보다 적은 변위차로 생물학적 실험 오류가 줄어들었으며, 세대수에 따른 평균 적합도 평가를 통해 생성된 서열은 우수하다는 것을 확인하였다. 또한 추가적으로 생물학적 실험으로 확인한 결과 최적의 경로를 찾을 수 있었다.

위와 같이 본 논문에서 제안한 DNA 서열 생성 알고리즘은 TSP를 위해 생성한 서열이 모든 적합도 평가에서 우수한 결과를 나타냈으며, 적은 오류율을 가진 많은 서열들을 이용하여 빠른 결과를 얻을 수 있었다.

참 고 문 헌

[1] C. H. Papadmitriou, *Computational Complexity*, 1994.
[2] J. A. Rose, R. J. Deaton, D. R. Franceschetti, M. Garzon.S. E. Jr. Stevens, "A Statistical Mechanical Treatment of Error in theAnnealing Biostep of DNA Computation", *In [GECCO99]*, pp. 1829-1834, 1999.
[3] A. J. Hartemink, D. K. Gifford, and J. Khodor, "Automated constraint based nucleotide sequence selection for DNA computation", *in Proc. 4th DIMACS Workshop DNA Based Comput.*, pp. 227-235, 1998.
[4] R. Penchovsky and J. Ackermann, "DNA library design for molecular computation", *J. Comput. Bio.*, Vol. 10, No. 2, pp. 215-229, 2003.
[5] U. Feldkamp, S. Saghafi, W. Banzhaf, and H. Rauhe, "DNA sequence generator-A program for the construction of DNA sequences", *in Proc. 7th Int. Workshop DNA Based Comput.*, pp. 179-188, 2001.
[6] L. M. Adleman, "Molecular computation of solutions to combinatorial problems", *Science*, Vol. 266, pp. 1021-1024, 1994.
[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 CFSA/IFIS/SOFT'95 on Fuzzy Theory and Applications.*
[8] A. Narayanan, S. Zorbalas, "DNA algorithms for computing shortest paths", *Genetic Programming 1998, Koza, J. R. et al. (eds.), Morgan Kaufmann,* pp. 718-723, 1998.
[9] D. Faulhammer, A. R. Cukras, R. J. Lipton, and L. F. Landweber, "Molecular computation: RNA solutions to chess problems", *in Proc. Natl. Acad. Sci. U.S.A.*, Vol. 97, pp. 1385-1389, 2000.
[10] S. Kashiwamura, A. Kameda, M. Yamamoto, and A. Ohuchi, "Two-step search for DNA Sequence Design", *Proceedings of the 2003 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 03)*, pp. 1815-1818, 2003.

저 자 소 개

김은경(Eun-Gyeong Kim)
2001년 : 공주대학교 화학과 졸업(학사)
2003년 : 공주대학교 대학원 컴퓨터공학과 (공학석사)
2006년 : 공주대학교 대학원 컴퓨터공학과 (공학박사)
현재 : 공주대학교 컴퓨터공학부 시간강사

관심 분야: 바이오인포메틱스, 인공생명, DNA 컴퓨팅, 유전자알고리즘 등
E-mail : rotnrwk@kongju.ac.kr

이상용(Sang-Yong Lee)
1984년 : 중앙대학교 전자계산학과 (공학사)
1988년 : 일본동경대학대학원 총합이공학 연구과(공학석사)
1988년~1989년 : 일본 NEC 중앙연구소 연구원
1993년 : 중앙대학교 일반대학원 전자계산학과 (공학박사)
1993년~현재 : 공주대학교 정보통신공학부 교수
1996년~1997년 : University of Central Florida 방문교수

관심 분야: 인공지능, 에이전트, 컴퓨터게임, 바이오인포매틱스
E-mail : sylee@kongju.ac.kr