제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

Travelling Salesman Problem을 위한 DNA 컴퓨팅의 코드 최적화

김은경*, 이상용**
*공주대학교 컴퓨터 공학과, **공주대학교 정보통신공학부
e-mail:{rotnrwk*, sylee**}@kongju.ac.kr

Code Optimization of DNA Computing for Travelling Salesman Problem

Eun-kyoung Kim*, Sang-yong Lee**
*Dept of Computer Engineering, Kongju National University,
**Division of information & Communication Engineering,
Kongju National University

요 약

DNA 컴퓨팅은 생체 분자들이 갖는 막대한 병렬성을 이용하여 조합 최적화 문제에 적용하는 연구가 많이 시도되고 있다. 특히 TSP(Travelling Salesman Problem)는 간선에 대한 가중치 정보가 추가되어 있기 때문에 가중치를 DNA 염기 배열로 표현하기 위한 효율적인 방법들이 제시되지 않았다.

따라서 본 논문에서는 DNA 컴퓨팅에 DNA 코딩 방법을 적용하여 정점과 간선을 효율적으로 생성하고 표현된 DNA 염기 배열의 간선에 실제값을 적용하여 가중치 정보를 계산하는 ACO(Algorithm for Code Optimization)를 제안한다. DNA 코딩 방법은 변형된 유전자 알고리즘으로 DNA 기능을 유지하며, 서열의 길이를 줄일 수 있으므로 최적의 서열을 생성할 수 있는 특징을 갖는다. 실험에서 ACO를 TSP에 적용하여 Adleman의 DNA 컴퓨팅 알고리즘과 비교하였다. 그 결과 초기 문제 표현에서 우수한 적합도 값을 생성했으며, 경로의 변화에도 능동적으로 대처하여 최적의 결과를 빠르게 탐색할 수 있었다.

1. 서론

DNA 컴퓨팅의 분야는 Aldeman이 Hamiltonian path problem을 풀기 위해 생체 분자들을 사용함으로써, 분자 수준의 컴퓨팅이 가능하다는 것을 증명하였다[1]. 그리고 Lipton은 satisfiability(SAT) 문제를 풀기 위해 DNA 컴퓨팅을 사용하였다[2].

이들은 막대한 병렬성과 빠른 계산 및 거대한 저장매체로 생체분자인 DNA를 사용할 수 있는 가능성을 제시하였고, 최근에는 Narayanan과 Zorbalas이 TSP(Traveling Salesman Problem)를 풀기 위하여 DNA 코드들에 가중치를 주어 표현하는 방법을 제안하였다. TSP는 Hamiltonian path problem의 변형으로 간선들에 의해 연결되고, 주어진 정점들을 모두 한번만 방문하여 스타트 정점으로 돌아오는 가장 짧은 거리를 찾는 문제이다. 이들은 각 간선을 DNA 서열에 가중치를 주어 표현하는 방법을 제안하였다[3]. 하지만 이 방법은 실제 값들을 표현하는데 적합하지 못한 단점을 가지고 있다.

따라서 본 논문에서는 DNA 컴퓨팅 알고리즘에 DNA 코딩 방법을 적용하여 정점과 간선을 효율적으로 생성하고, 표현된 DNA 염기 배열의 간선에 실제 값을 적용하여 가중치 정보를 계산하는 ACO (Algorithm for Code Optimization)를 제안한다. 이 방법은 염기 서열에 대한 자유로운 표현이 가능하며, 화학적 오류를 미리 제거할 수 있다. 또한 서열에 맞는 가중치를 주어 표현함으로써 짧은 거리를 쉽게 찾아낼 수 있으며, 계산속도를 최소화하였다.

2. 관련연구

1) DNA 컴퓨팅

DNA 컴퓨팅은 실제 생체 분자인 DNA나 RNA와 같은 살아 있는 세포를 응용한 기술이다. DNA는 4개의 염기인 A, T, G, C가 2중 나선 구조로 구성되어 있다. 이들 염기에 대용량 데이터를 저장할 수 있는 메모리 기능을 가지고 있으며, 정해진 규칙에 위해 상호 보완적인 방식의 Watson-Crick 결합을 하고 있다[4]. 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.

이러한 DNA 컴퓨팅의 특징을 살펴보면 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 우수

함을 보이고 있다[1][5].

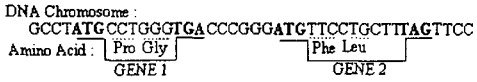
Aldeman이 DNA 컴퓨팅을 이용하여 Hamiltonian path problem을 해결한 이후, Lipton과 Narayanan, Zorbalas을 선두로 하여 많은 학자들이 Turing machine 구현, 암호 해독, DNA를 이용한 유전자 알고리즘 구현 등에 대하여 연구하고 있다.

2) DNA 코딩 방법

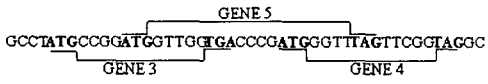
DNA 코딩 방법은 Yoshikawa가 1995년에 처음으로 제시한 변형된 형태의 유전자 알고리즘으로 DNA를 이용한 선택, 재생, 교배, 돌연변이 연산자를 사용한다[6][7].

DNA가 하나의 아미노산으로 해석되기 위해서는 3개의 염기 서열이 필요하며, 이것을 생물학적인 용어로 코돈(codon)이라고 한다. 이것을 아미노산 번역표에 따라 총 64가지의 아미노산으로 해석된다. 하지만 중복된 것을 제외한 20개의 아미노산만으로 해석된다.

[그림 1]에서 보는 것과 같이 염기 서열은 Start 코돈인 ATG에서 시작하여 Stop 코돈인 TGA (TAA, TAG)에서 아미노산 번역이 끝나며, 염기 서열을 아미노산으로 번역하기 때문에 짧은 DNA 코드로도 많은 정보를 얻을 수 있다.



[그림 1] DNA 염색체의 번역 예



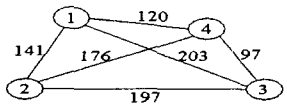
[그림 2] 유전자 중복의 예

DNA 코딩 방법의 특징을 살펴보면 첫번째로, [그림 2]에서 보는 것처럼 염색체의 중복을 효율적으로 표현할 수 있다. 두번째로, 하나의 아미노산을 만드는 코돈이 여러개이므로 지식 표현이 쉽다. 세번째로, 교차점이 임의로 주어지기 때문에 염색체의 길이가 가변적이다.

이러한 특징들로 인해 긴 길이의 염기 서열이 아닌 적은 수의 아미노산 서열을 사용할 수 있고, 0과 1을 사용하는 유전자 알고리즘에 비하여 DNA 코딩 방법은 4가지 염기를 사용하여 코딩하기 때문에 해의 표현이 다양하다.

3) TSP(Traveling Salesman Problem)

TSP는 n개의 도시와 도시사이의 거리가 주어질 때, 어떤 도시에서 시작하여 모든 도시를 단 한번만 방문하고, 원래의 출발지로 되돌아오는 최단 길이의 여행을 찾는 것이다. 다시 말하면 도시들의 가능한 방문의 모든 순열이 주어질 때, 각 도시와 다음 도시와의 유클리드 거리의 합이 최소가 되는 여행을 선택하는 것이다. 따라서 TSP의 탐색 공간은 가능한 모든 여행의 집합($T_1, T_2, \ldots, T_{n!}$)이 되고, 그 크기는 n!이며, 이 중에서 여행거리가 가장 짧은 것이 해가 된다[8]. [그림 3]은 간단한 4개 도시 TSP를 나타낸 것이다.



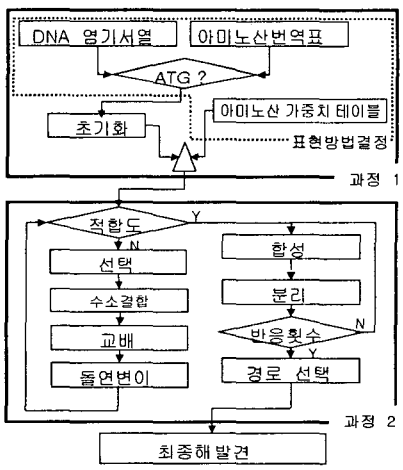
[그림 3] TSP의 예제 그래프

이 문제에서 가능한 경로 $n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow n_4$가 최단 여행경로이고, 이때의 여행거리는 D=555이다.

현재 TSP는 많은 연구자들이 몇 십개에서 많게는 수백만개의 도시를 가진 TSP를 다양한 방법으로 근사 최적해를 연구하고 있다. 또한 유전자 알고리즘을 이용한 새로운 인코딩 변환 기법이나 교차기법, 국소 최적화 알고리즘의 스피드업, 문제 공간 분석 등이 주요한 기술로 등장하고 있다.

3. ACO(Algorithm for Code Optimization)

본 연구에서는 조합 최적화 문제인 TSP를 풀기 위해 Aldeman의 DNA 컴퓨팅 알고리즘을 개선하여 정점과 가중치를 표현한 간선을 생성하고, 합성과 분리 과정을 유전자 알고리즘으로 반복, 처리하면서 최종해를 찾는 ACO를 제안한다. [그림 4]는 ACO의 전체 흐름도를 나타낸 것이다.



[그림 4] ACO의 흐름도

각 단계 별로 살펴보면, 과정1은 DNA 염기 서열을 DNA 코딩 방법에 적용하여 각각의 아미노산 코드로 변환한 후 정점과 가중치를 표현한 간선을 생성하는 단계이다. [그림 5]는 최적화 문제를 ACO로 표현하기 위한 초기 방법으로 다음과 같은 과정을 거친다.

[그림 5] ACO를 적용한 초기 문제 표현의 예

정점 생성단계는 아래와 같이 크게 3단계로 구분짓는다.

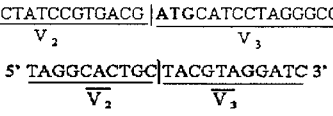
첫째, DNA 염기 서열에 DNA 코딩 방법을 적용하여 각각의 아미노산 코드로 변환하고 Start 코돈인 ATG 코드 앞에서 잘라 정점(V)을 표현한다.

둘째, 처음 부분에 Strat 코돈이 나타나지 않을 경우 Strat 코돈의 앞부분을 하나의 정점으로 표현한다.

셋째, DNA 염기 서열을 정점으로 표현한 후 간선 $\overline{V_2 \rightarrow V_3}$를 표현하기 위해 다음과 같은 4가지 제한조건이 필요하다.

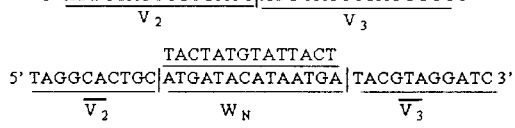
첫째, Start 코돈인 ATG 코드 앞에서 끊어 간선을 표현하지 않고 AT*(ATT, ATC, ATG, ATA)의 4종류를 지정한다.
둘째, Stop 코돈인 TGA, TAA, TAG를 지정한다.
셋째, 연결하려는 두 정점의 간선 $\overline{V_2 \rightarrow V_3}$의 표현은 V_2에서 처음 나타나는 AT*과 V_3에서 처음 나타나는 Stop 코돈의 상보결합을 간선으로 사용한다.
넷째, Stop 코돈이 없을 경우에는 정점의 염기 배열 1/2bp를 간선으로 하여 상보 염기 배열을 생성한다.

간선 $\overline{V_2 \rightarrow V_3}$의 생성은 [그림 5]에서 사각형의 Start 코돈과 원의 Stop 코돈을 제한 조건에 의해 [그림 6]과 같이 표현할 수 있다.



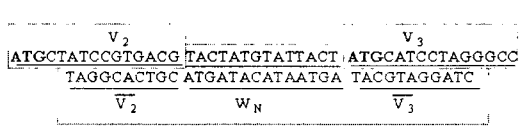
[그림 6] 정점과 간선의 염기 배열의 예

[그림 6]에서 생성된 간선 염기 배열은 가중치를 표현하기 위해 간선에 가중치 염기 배열을 포함한 후 가중치에 대해서만 상보결합을 해준다.



[그림 7] 정점과 간선의 가중치 염기 배열의 예

[그림 7]과 같이 가중치에 대해 상보결합을 해주게 되면 정점의 표현 방법에는 변화가 없으며 간선의 가중치 염기 배열의 길이를 가변적으로 변화시킬 수 있는 장점을 가지고 있다. [그림 8]은 정점과 간선의 가중치를 통합한 2중 가닥의 DNA를 이용한 경로 생성의 예이다.



[그림 8] 경로 생성 예

이러한 방법으로 정점과 간선의 가중치를 표현하면 주어진 정점의 개수보다 적거나 많을 수 있다. 이 문제를 해결하기 위해 과정2에서 유전자 알고리즘을 통하여 정점의 개수가 같은 적합한 서열을 생성한다.

식(1)은 간선의 가중치 값을 구하는 식으로 간선의 i의 수소결합 변환 함수값(Ne_i)와 간선 i의 실제 가중치(W_{ei}), 전체 그래프에서 가중치의 합계(S_w), 전체 간선의 수소결합 총합(S_v), 실험을 통해 결정한 임계값(θ)을 가지고 계산한다. 이렇게 생성된 간선의 가중치를 표1의 아미노산 코드에 적용한 결과와 함께 비례선택법(roulette wheel)을 이용하여 적합도를 계산한다. 그리고 잘못된 결합이나 결합위치 이동과 같은 화학적 오류가 일어날 수 있는 조건을 미리 제거한다.

$$F_i = \begin{cases} \left| \sqrt{\frac{Ne_i}{Sv} - \frac{We_i}{Sw}} \right| & if \sqrt{\frac{Ne_i}{Sv} - \frac{We_i}{Sw}} \geq \theta \\ otherwise\ is\ 0 & \end{cases} \quad \text{식(1)}$$

적합도 계산 후 새로운 개체군을 생성하기 위해 A, T, G, C의 수소결합을 이용한다. 필요로 하는 수소결합의 수를 직접적으로 이용하기 위해 낮은 가중치를 가지는 간선에 A/T, 높은 가중치를 가지는 간선에 G/C를 포함한다. 실제 수소결합 수를 직접적으로 반영함은 물론 염기 배열의 길이까지 조절하여 가중치를 표현할 수 있다. 또한 가중치의 표현 범위가 훨씬 확장되어 짧은 염기배열을 가지고도 넓은 범위의 가중치를 표현할 수 있는 장점을 갖는다.

정점 염기 배열을 교배와 돌연변이를 통해 새로 생성한다. 교배는 간선의 염기 배열에서만 일어나게 하며 2점 교배를 하고 국소 해에 빠질 위험성을 벗어나기 위해 랜덤하게 교배 점을 선택한다. 돌연변이는 간선 염기 배열 중 임의의 염기쌍을 선택해 한 염기쌍을 변화시키는 방법을 사용한다. 위의 방법에서 간선은 정점을 이용해서 만들어주고 간선 염기배열을 이용해 간선의 가중치 염기배열을 생성하여, 반복은 세대수 만큼 반복한다.

[표 1] 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

높은 적합도를 갖는 최적의 코드를 선택하여 주어진 반응횟수만큼 합성과 분리 과정을 거친다. 이 분리 과정에서 해가 될 가능성이 없는 것은 항체 친화력 반응과 PCR 반응, 겔 전기 영동법으로 파악하여 미리 제거한다. 마지막으로 다시 한번 PCR을 이용하여 특성 부위의 서열을 증폭시킨다. 그리고 겔 전기 영동법으로 일정 길이의 염기 배열만 추출하고, 항체 친화력 반응을 통하여 그래프의 모든 정점에 대해서 최소한 한번 이상 방문한 경로만을 선택하여 최종해를 발견한다.

4. 실험 및 분석

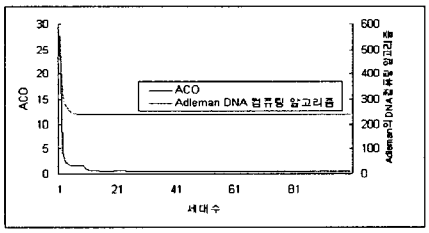
실험은 [그림 3]의 정점 4개와 간선 6개를 대상으

로 한 TSP에 적용하여 ACO와 Adleman의 DNA 컴퓨팅 알고리즘을 비교 평가하였고, 정점이 7개인 경우도 같이 실험하였다. 모의실험에서 사용된 파라미터들은 [표 2]와 같이 설정하였다. 그러나 Adleman의 DNA 컴퓨팅 알고리즘은 단순한 합성과 분리 과정이므로 반응횟수와 반복횟수를 곱한 총 반응횟수를 ACO와 동일하게 적용하였고, 정점과 간선 표현에서는 염기 배열이 최소 10bp, 최대 20bp 사이에서 실험하였다.

[표 2] DNA 컴퓨팅에 사용한 파라미터들

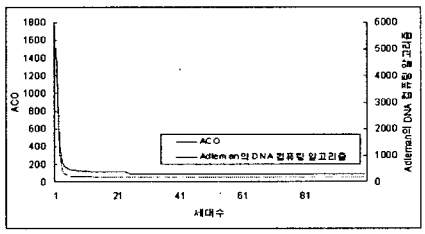
변 수	ACO	Adleman의 DNA 컴퓨팅 알고리즘
집단 크기	100	100
세대수	100	100
교배 연산 비율	0.5	0.5
돌연변이 연산 비율	0.5	0.5
임계값	0.1	0.1
반복횟수	10	1
반응횟수	10	100
화학적 오류률	0.01	0.01

4개의 정점과 7개의 정점을 갖는 TSP에 적용한 결과 다음과 같은 최적의 성능을 얻었다. [그림 9]는 정점이 4개인 경우로 ACO는 5세대 이후, Aldeman의 DNA 컴퓨팅 알고리즘은 13세대 이후에 각각 최적의 경로 값을 탐색하였다. 이들의 최적의 거리는 555로 동일한 값을 찾았다.



[그림 9] 정점 4개의 최적 성능 평가

또한 [그림 10]은 정점이 7개인 경우로 ACO는 11세대 이후, Aldeman의 DNA 컴퓨팅 알고리즘은 34세대 이후에 각각 최적의 경로 값을 탐색하였다. 이들의 최적의 거리는 2020으로 동일한 값을 찾았다.



[그림 10] 정점 7개의 최적 성능 평가

ACO는 평균 적합도에서 Adleman의 DNA 컴퓨팅 알고리즘보다 더 높은 최적해를 생산하여 보다 나은 결과를 얻었다. 따라서 실험 결과를 평균 적합도와 최적해를 가지고 정리하면 [표 3]과 같다.

[표 3] ACO의 비교 평가값

구분		ACO	Adleman의 DNA 컴퓨팅 알고리즘
평균 적합도	정점4개	3.02614	4.51964
	정점7개	3.65944	9.74181
최적해	정점4개	5세대이후 11개	13세대이후 11개
	정점7개	11세대이후 8개	34세대이후 8개

5. 결과

본 연구에서는 TSP을 통하여 ACO가 Adleman의 DNA 컴퓨팅 알고리즘 보다 빠른 최적화 값을 찾아냈다. ACO는 DNA 염기 서열의 길이를 가변적으로 표현하면서 수소결합 수를 직접 반영하였기 때문에 가중치의 표현 범위가 많이 확장되어 짧은 염기 배열을 가지고도 넓은 범위의 가중치를 표현할 수 있었다. 또한 과정2의 합성과 분리 과정에서는 유전자 알고리즘으로, 불필요한 화학적 오류들을 제거하여 Adleman의 DNA 컴퓨팅 알고리즘보다 평균 적합도에서 우수한 해를 얻을 수 있었다.

향후, 더 복잡한 문제에 적용하였을 경우, 우수한 결과를 얻을 수 있는지 계속적인 실험과 DNA의 특징을 보다 효과적으로 표현할 수 있는 알고리즘에 대한 연구가 필요할 것이다.

참고문헌

[1] Adleman, L. M., "Molecular computation of solutions to combinatorial problems", Science, 266:1021-1024, 1994.

[2] N. Jonoska & N. C. Seedman (Eds.), "Preliminary Proceedings of 7th International Meeting on DNA Based Computers", University of South Florida, Tampa, FL, June, 10-13, 2001.

[3] A. Narayanan and S. Zorbalas, "DNA algorithms for computing shortest paths", *Genetic Programming 1998*, Koza, J. R. *et al.* (eds.), Morgan Kaufmann, pp. 718-723, 1998.

[4] Deaton, R., Murphy, R. C., Garzon, M., Franceschetti, D. R., Stevens, S. E. Jr., "Reliability and efficiency of a DNA-based computation", Physical Review Letters, 82(2):417-420, 1998.

[5] Deaton, R. and Karl, S. A., " Introduction to DNA Computing", 1999 Genetic and Evolutionary Computation Conference Tutorial Program, pp. 75-93, Orlando, Florida, July 14, 1999.

[6] 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.

[7] 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.

[8] O. Martin, S. Otto, and E. Felten, "Large-step Markov Chains for the Traveling Salesman Problem.", Complex System, Vol. 5, No. 3, pp. 299-326, 1991.