Hamiltonian Path Problem을 위한 DNA 컴퓨팅의 코드 최적화

김은경[O] 이상용
공주대학교 컴퓨터 공학과[O], 공주대학교 정보통신공학부
{rotnrwk[O], sylee}@kongju.ac.kr

Code optimization of DNA computing for Hamiltonian path problem

Eun-kyoung Kim[0] Sang-yong Lee
Dept. of Computer Science & Eng, Kongju National University[0],
Division of Information & Communication Engineering

요 약

DNA 컴퓨팅은 생체 분자들이 갖는 막대한 병렬성을 정보 처리 기술에 적용한 기술이다. Adleman의 DNA 컴퓨팅은 랜덤한 고정길이의 형태로 문제를 표현하기 때문에 해를 찾지 못하거나 시간이 많이 걸리는 단점을 갖고 있다.

본 논문은 DNA 컴퓨팅에 DNA 코딩 방법을 적용하여 DNA 서열을 효율적으로 표현하고 반응횟수 만큼 합성과 분리 과정을 거쳐 최적의 코드를 생성하는 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. ACO를 NP-complete 문제 중 Hamiltonian path problem에 적용하여 실험한 결과, Adleman의 DNA 컴퓨팅 보다 초기 문제 표현에서 높은 적합도 값을 갖는 서열을 생성했으며, 경로의 변화에도 능동적으로 대처하여 최적의 결과를 빠르게 탐색할 수 있었다.

1. 서 론

Adleman은 DNA가 갖는 막대한 병렬성과 상보적인 특징을 이용하여 Hamiltonian path problem을 해결함으로써, 분자 수준의 컴퓨팅이 가능하다는 것을 증명하였다[1].

현재 DNA 컴퓨팅은 실제 생체 분자인 DNA를 이용하여 빠른 계산 및 거대한 저장 매체로 활용할 수 있어 많은 연구가 진행중이다[2]. 하지만 최적해를 찾는 연구에서 두 가지의 문제점이 제기되었다[3][4]. 첫째, 단순한 합성과 분리 과정을 사용하므로 해를 찾는데 많은 시간과 노력이 요구된다. 둘째, 문제를 염기 서열로 변환하는 효율적인 표현 방법이 없기 때문에 실제 화학적 특성에 대한 오류의 가능성을 포함하고 있다. 이 두 가지 문제점을 해결하기 위해 많은 노력과 연구가 진행되어 첫 번째 문제는 유전자 알고리즘을 이용한 반복 과정으로 해결되었다[3]. 그러나 두 번째 문제는 아직 확실한 방법이 제시되고 있지 않다.

따라서 본 논문에서는 DNA 컴퓨팅의 문제점에 대한 해결 방법으로 DNA 코딩 방법을 적용하여 초기 서열을 분리 하고 반응횟수 만큼 합성과 분리 과정을 거쳐 최적의 코드를 형성하는 ACO(Algorithm for Code Optimization)를 제안한다. ACO는 염기 서열에 대한 자유로운 표현이 가능하며, 화학적 오류를 미리 제거할 수 있기 때문에 최적의 해를 찾을 수 있다.

2. 관련 연구

2.1 DNA 컴퓨팅

DNA 컴퓨팅은 실제 생체 분자인 DNA나 RNA와 같은 살아있는 세포를 이용한 기술이다. DNA는 1㎤에 1조개의 CD보다 많은 정보를 가질 수 있다. 인간 유전체를 포함하는 2중 나선형 가닥은 A(Adenine), C(Cytosine), G(Guanine), T(Thymine)라는 4개의 염기에 거대한 메모리로 작동할 수 있는 데이터를 저장할 수 있다. 이들 염기는 정해진 상호 보완적인 방식의 Watson-Crick 결합을 한다[5]. 또한 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.

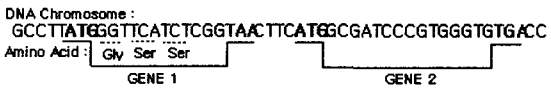
이러한 DNA 컴퓨팅의 특징을 살펴보면 매우 낮은 에너지로 작동되기 때문에 많은 에너지가 필요 없다는 것이다. 그리고 나노 수준의 막대한 병렬성을 이용하여 NP-complete에 효과적인 접근이 가능하게 되었다. 또한 계산 속도와 정보의 저장 및 처리 효율에서도 우수함을 보이고 있다[1][6].

Adleman이 DNA를 이용하여 Hamiltonian path problem을 해결한 이후, 많은 학자들이 *Turing machine* 구현, 암호 해독, DNA를 이용한 유전자 알고리즘 구현 등에 대하여 연구하고 있다.

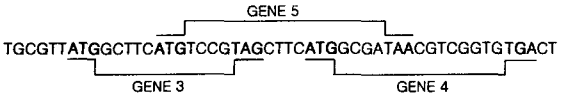
2.2 DNA 코딩 방법

DNA 코딩 방법은 1995년 Yoshikawa가 제시한 변형된 형태의 유전자 알고리즘으로 DNA를 이용한 선택, 재생, 교배, 돌연변이 연산자를 사용한다[7][8]. DNA는 아미노산 번역 표에 따라 20개의 아미노산으로 해석된다. 하나의 아미노산으로 해석되기 위해서는 3개의 염기 서열이 필요하며, 이것을 생물학적인 용어로 코돈(codon)이라고 한다.

[그림 1]에서 보는 것과 같이 염기서열은 Start 코돈인 ATG에서 시작하여 Stop 코돈인 TGA(TAA, TAG)에서 끝나며, 코돈을 아미노산으로 해석함으로써 짧은 DNA 코드에서도 많은 정보를 얻을 수 있다.



[그림 1] DNA 염색체의 번역 예



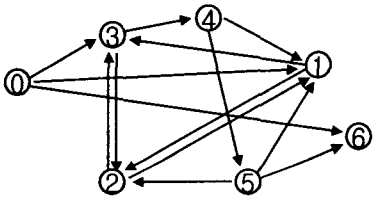
[그림 2] 유전자의 중복의 예

DNA 코딩 방법의 특징을 살펴보면, 첫 번째로 [그림 2]에서 보는 것처럼 염색체의 중복을 효율적으로 표현할 수 있다는 것이다. 두 번째로 하나의 아미노산을 만드는 코돈이 여러 개이므로 지식 표현이 쉽다는 것이다. 세 번째로 교차점이 임의로 주어지기 때문에 염색체의 길이가 가변적이라는 것이다.

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.

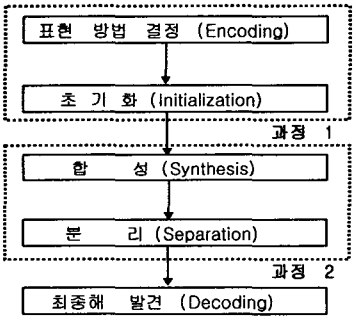
2.3 Hamiltonian path problem

Hamiltonian path problem은 입력정점 V_{in}에서 출력정점 V_{out}에 이르는 모든 정점을 반드시 한번만 포함한다는 전제 조건을 갖고 있는 NP-complete문제이다. 이 문제에 대하여 실제로 다항식 시간에 계산할 수 있는 효율적인 알고리즘은 존재하지 않는다.



[그림 3] Hamiltonian path problem

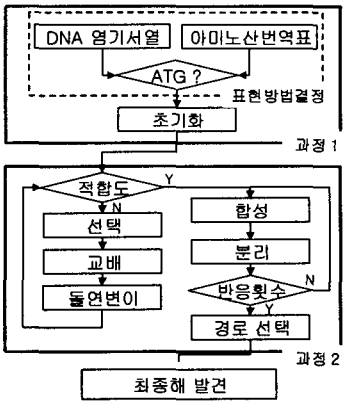
[그림 3]은 Adleman이 Hamiltonian path problem를 표현한 유향 그래프로, 정점 0에서 출발하여 모든 경로를 거쳐 정점 6에 도달하는 문제이다. Adleman은 이 문제를 풀기 위해 생체 분자와 생화학적인 방법을 이용하여 [그림 4]와 같이 DNA 컴퓨팅 알고리즘을 제시하였다.



[그림 4] Adleman의 DNA 컴퓨팅 알고리즘

3. ACO (Algorithm for Code Optimization)

본 연구에서는 Adleman의 DNA 컴퓨팅 알고리즘의 문제 표현을 개선하기 위하여 DNA 코딩 방법을 적용하여 최적의 서열을 생성하고 합성과 분리 과정을 유전자 알고리즘으로 반복, 처리하면서 최종해를 찾는 ACO를 제안한다.



[그림 5] ACO 흐름도

[그림 5]는 ACO의 전체 흐름도를 나타낸 것이다. 각 단계별로 살펴보면, 과정1은 DNA 염기 서열을 이용하여 정점(V)으로 표현하고 간선을 생성하는 단계이다. [그림 6]과 같이 DNA 염기 서열로 변환하기 위해서는 세 가지 처리 단계를 거친다.

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

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

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

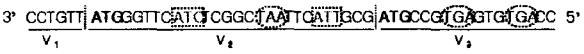
첫째, Start 코돈인 ATG 코드 앞에서 끊어 간선을 표현하지 않고 AT*(ATT, ATC, ATG, ATA)의 4종류를 지정한다.

둘째, Stop 코돈인 TGA, TAA, TAG를 지정한다.

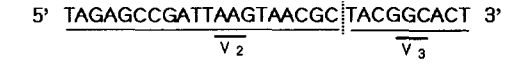
셋째, 연결하려는 두 정점의 간선 $\overline{V_2 \rightarrow V_3}$의 표현은 V_2에서 처음 나타나는 AT*과 V_3에서 처음 나타나는 Stop 코돈의 상보결합을 간선으로 사용한다.

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 6]에서 사각형의 Start 코돈과 원의 Stop 코돈을 제한조건에 의해 [그림 7]과 같이 표현할 수 있다.



[그림 6] 정점 표현 방법 예



[그림 7] 간선 표현 방법 예

이러한 방법으로 정점과 간선을 표현하면 주어진 정점의 개수보다 적거나 많을 수 있다. 이 문제를 해결하기 위해 과정2에서 유전자 알고리즘을 통하여 정점의 개수가 같은 적합한 서열을 생성한다. 적합도 평가는 [표 1]의 아미노산 코드를 적용하여 비례 선택법(roulette wheel)으로 계산하고, 잘못된 결합이나 결합위치 이동과 같은 화학적 오류가 일어날 수 있는 조건

을 미리 제거한다. 교배는 2점 교배를 하고 국소 해에 빠질 위험성을 벗어나기 위해 랜덤하게 교배 점을 선택한다. 돌연변이는 모든 코드에서 수행하며, 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 특정 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 7개와 간선 14개를 대상으로 한 Hamiltonian path problem에 적용하여 ACO와 Adleman의 DNA 컴퓨팅 알고리즘을 비교하였다. 또한 ACO의 과정1과 과정2를 각각 Adleman의 DNA 컴퓨팅 알고리즘에 적용하여 효율성을 같이 비교 평가하였다. 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

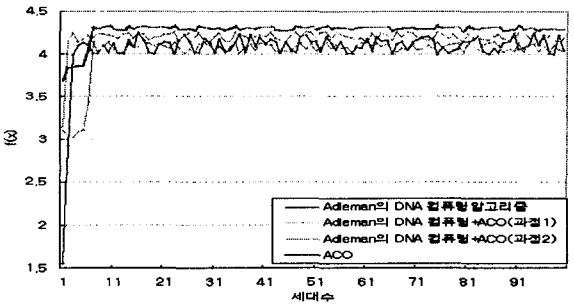
변 수	ACO	Adleman의 DNA 컴퓨팅 알고리즘
집단 크기	100	100
세대수	100	100
교배 연산 비율	0.5	0.5
돌연변이 연산 비율	0.5	0.5
반복횟수	10	1
반응횟수	10	100
화학적 오류률	0.01	0.01

[표 3]은 실험에서 나온 각각의 평균 적합도 값을 나타낸 것이다. ACO는 평균 적합도에서 Adleman의 DNA 컴퓨팅 알고리즘보다 더 높은 최적해를 생산하여 보다 나은 결과를 얻었으며, Adleman의 DNA 컴퓨팅 알고리즘에 과정1과 과정2를 각각 적용한 방법보다 나은 결과를 얻었다.

[표 3] ACO의 비교 평가값

구분	ACO	Adleman의 DNA 컴퓨팅 알고리즘	ACO의 과정1 적용	ACO의 과정2 적용
평균 적합도	4.2814	4.0656	4.0153	4.2459
최적해	4.3284	4.2467	4.1320	4.2754

[그림 8]에서 알 수 있듯이 ACO는 11세대 이후 고른 적합도 값을 유지하고 있으며, Adleman의 DNA 컴퓨팅 알고리즘과 ACO의 과정1과 과정2를 적용한 방법은 불규칙한 적합도 값을 나타내고 있다. 또한 ACO는 높은 적합도 값을 유지하면서 최적의 해를 가장 많이 생산하였다. 그리고 DNA 염기 서열의 길이에 대한 화학적 반응 오류율은 ACO가 평균 0.9%, Adleman의 DNA 컴퓨팅 알고리즘은 평균 23%, ACO의 과정1과 과정2를 적용한 방법에서는 각각 24%와 16%로 반응하였다.



[그림 8] 세대별 적합도

5. 결론

본 연구에서는 Hamiltonian path problem을 통하여 ACO가 Adleman의 DNA 컴퓨팅 알고리즘보다 높은 코드 최적화를 보였다. ACO는 DNA 염기 서열에 따라 능동적으로 대처하였으며, 높은 적합도를 유지하면서 고른 코드 값을 생성하였다. 또한 과정2의 합성과 분리 과정에서는 유전자 알고리즘으로 불필요한 화학적 오류들을 제거하여 Adleman의 DNA 컴퓨팅 알고리즘보다 많은 수의 최적해를 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] 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.

[4] Rose, J. A., Deaton, R. J., Franceschetti, D. R., Garzon. M., and Stevens, S. E. Jr., "A Statistical Mechanical Treatment of Error in the Annealing Biostep of DNA Computation" In [GECCO99], pp. 1829-1834.

[5] Watson, J. D., Gliman, M., Wikowski, J., Zoller, M., Recombinant DNA, 2nd Ed., Scientific American Books, New York, 1992.

[6] 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.

[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] 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.