論文
2-10-12

DNA 컴퓨팅의 새로운 PCR 연산
(New PCR of DNA Computing)

김 정 숙*
(Jung-Sook Kim)

요 약

외판원 문제(Traveling Salesman Problem)는 주어진 n개의 도시들과 그 도시들 간의 거리비용이 주어졌을 때, 모든 도시들을 정확히 한번씩만 방문하면서 걸린 비용이 최소가 드는 경로를 찾는 문제이다. 따라서 최적해(optimal)를 구하는 것은 전형적인 NP-완전 문제 중의 하나로, 외판원 문제를 해결하려는 다양한 알고리즘들이 개발되고 있다. 특히 실제 생체 분자(bio-molecule)를 계산의 도구로 사용하는 새로운 계산 방법인 DNA 컴퓨팅은 DNA 분자가 잠재적으로 가지고 있는 막대한 병렬성을 이용해서 NP-완전 문제들을 해결하고자 하는 연구들이 많이 진행되고 있다. 그러나 아직 실제 생체 분자의 특성을 잘 반영하는 계산 모델이나 분자 생물학에서 사용하는 연산들이 많이 개발되지 않아 계산 효율이 비교적 좋지 않다. 따라서 본 논문에서는 외판원 문제를 해결하기 위한 DAN컴퓨팅의 새로운 중합 효소 연쇄 반응(Polymerase Chain Reaction, PCR) 연산을 개발하였다.

ABSTRACT

In the Traveling Salesman Problem(TSP), a set of N cities is given and the problem is to find the shortest route connecting them all, with no city visited twice and return to the city at which it started. Since TSP is a well-known combinatorial optimization problem and belongs to the class of NP-complete problems, various techniques are required for finding optimum or near optimum solution to the TSP. Especially DNA computing, which uses real bio-molecules to perform computations supported by molecular biology, has been studied by many researchers to solve NP-complete problem using massive parallelism of DNA computing. Though very promising, DNA computing technology of today is inefficiency because the effective computing models and operations reflected the characteristics of bio-molecules have not been developed yet. In this paper, I design new Polymerase Chain Reaction(PCR) operations of DNA computing to solve TSP.

* 정회원 : 김포대학 컴퓨터계열 소프트웨어 개발 전공 전임강사

논문접수 : 2001. 10. 13.
심사완료 : 2001. 10. 23.

1. 서론

외판원 문제는 주어진 n개의 도시들과 그 도시들간의 거리 비용이 주어졌을 때, 처음출발도시에서부터 정확히 한 도시는 한 번씩만 방문하여 다시 출발 도시로 돌아오면서 방문한 도시들을 연결하는 최소의 비용이 드는 경로를 찾는 문제로 최적해(optimal)를 구하는 것은 전형적인 NP-완전 문제중의 하나이다[3]. 이렇게 복잡한 외판원 문제가 다른 최적화 문제들에 응용이 될 뿐만 아니라, NP-완전 문제들을 현실적으로 해결하려는 노력의 일환이 되기 때문에 외판원 문제를 해결하기 위한 다양한 연구들이 시도되고 있다.

분기 한정법(Branch-and_Bound) 알고리즘이나 동적 프로그래밍(Dynamic Programming) 알고리즘 방법과 같이 최적해를 구하는 연구들이 있고, 확률적 탐색 휴리스틱(probability search heuristic)에 근거해서 근사해(near optimal)를 구하는 유전 알고리즘, 담금질 방법, 최근접 휴리스틱 알고리즘 등 다양한 방법들이 개발되었다.

특히 최근 들어 분자 생물학의 비약적인 발전으로 인해서 생체 분자를 이용하여 계산을 수행하고자 하는 DNA 컴퓨팅 기술을 이용한 연구가 활발히 진행되고 있다. DNA 컴퓨팅이 주목을 받고 있는 이유는 DNA 분자가 가지고 있는 막대한 병렬성으로 현재 사용되고 있는 병렬 컴퓨터로써는 상상도 할 수 없다. 이러한 병렬성을 이용하면 NP-완전 문제들과 같은 여러 가지 문제들을 효율적으로 해결할 수 있을 것으로 생각된다. 그런데, DNA 컴퓨팅은 생체 분자를 이용하기 때문에 일반 컴퓨터의 연산자를 사용할 수 없고, 분자 생물학에서 사용하는 여러 가지 실험 방법들을 연산의 도구로 사용하여야 한다. 그러나 지금은 개발되어 있는 연산이 많지 않다. 따라서 본 논문에서는 DNA 컴퓨팅의 연산 중 중합 효소 연쇄 반응 연산을 효율적으로 수행할 수 있도록 개발하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 DNA 컴퓨팅의 기본 개념과 주로 사용되는 연산자들과 DNA 컴퓨팅의 특징들에 대해 검토하였으며, 3장에서 외판원 문제를 DNA 컴퓨팅으로 해결하는 방법을 기술하고, 4장에서는 본 논문에서 제시한 중합효소 연쇄반응에 대해 설명하며, 5장에서 실험한 내용을 검토하고 마지막으로 결론을 맺고 향후 연구 과제에 대해 논의한다.

2. DNA 컴퓨팅

2.1. DAN 컴퓨팅의 기본 개념

DNA 컴퓨팅 기법은 실제 생체 분자인 DNA을 계산의 도구 및 저장 도구로 사용한다. 따라서 가상의 DNA 컴퓨터는 A(Adenine), C(Cytosine), G(Guanine), T(Thymine), 4가지 염기의 정보를 표현한다. A는 T와 2개의 수소 결합을 형성하면서 결합하고 C는 G와 3개의 수소 결합을 형성하면서 결합하는데, 이것을 Watson-Crick 상보 결합이라고 한다[1][5]. 자료구조는 DNA 구조인 이중가닥(double strand)이나 단일가닥(single strand)을 사용하며, 정보를 저장하고 추출하는 기본 방법은 Watson-Crick 상보 결합이다. 일반적인 연산자로는 생물학 실험 방법들인 결찰법(ligation), 서냉복원법(annealing), 하이브리드형성(hybridization), 중합효소 연쇄반응, 겔 전기 영동법(gel electrophoresis), 항체 친화력 반응(antibody affinity) 등과 여러 가지 효소(enzyme) 들을 사용한다.

2.2. DNA 컴퓨팅에서 사용되는 연산자

1) 하이브리드 형성(Hybridization) : 각 염기들이 상보적인 염기들과 서로 결합하는 것을 의미한다. 이를 통해서 2개의 단일 가닥 사슬이 결합하여 1개의 이중 가닥 사슬을 형성할 수 있다.
2) 결찰법(Ligation) : 점착말단(sticky end)이 서로 결합하여 하나의 긴 사슬을 형성하는 것을 의미한다.
3) 서냉복원법(Annealing) : DNA 컴퓨팅에서는 결찰법 또는 하이브리드 형성법과 거의 동일한 의미로 사용된다.
4) 중합효소 연쇄반응 : 특정 염기배열(sequence)을 가지고 있는 DNA의 이중 가닥을 단일 가닥으로 분해(melting)한 후, 증폭시키고자 하는 염기배열의 상보 배열을 추가하여 이중 가닥으

로 다시 만드는 것이다. 이 연산 결과 하나의 염기 배열이 2개의 염기 배열로 증폭된다.

5) 겔 전기영동법(Gel Electrophoresis) : 특정 길이의 DNA를 추출하는 연산이다.
6) 항체 친화력 반응(Antibody Affinity) : 전체 염기배열 중에서 특정 염기 배열을 가지고 있는 DNA를 선택하는 방법이다.
7) 효소(Enzyme) 반응 : 결찰법에서 사용되는 ligase, 특정한 부분을 인식하여 이중 가닥 사슬을 쪼개서(cleave) 단일 가닥 사슬로 만든 후, 다른 과정에 적용할 수 있도록 하는 제한 효소(restriction enzyme)가 사용된다.

2.3. DNA 컴퓨팅의 특징과 문제점

DNA 컴퓨팅의 특징은 첫째로, 막대한 병렬성을 이용해서 주어진 문제의 탐색 공간을 전부 탐색하는 방법을 사용한다. 일반적인 알고리즘은 하나의 해를 유지하면서 그 해를 주어진 연산자를 이용해 개선해 나가면서 최종해를 찾는 접근 방법을 취하고 있는데 비해, DNA 컴퓨팅 알고리즘은 많은 수의 해를 유지하면서 연산자를 적용한 후 생성된 많은 수의 해 중에 최종 조건을 만족하는 해를 발견하는 방법을 취하고 있다. 이러한 방식은 진화 연산과 유사하지만, 개체군의 크기에서 진화 연산 개체군의 수와 비교할 수 없을 만큼 크다. 그리고 기본적인 DNA 컴퓨팅 알고리즘에서는 해를 진화시킨다는 개념이 없이 한 번의 과정을 통해 모든 가설 공간을 탐색하기 때문에 여러 가지 면에서 다르다.

둘째로, DNA 컴퓨터는 계산 속도나 계산 효율에서도 우수하다. 계산 속도 측면에서 살펴보면 일반 PC는 초당 10^6번의 계산을 할 수 있고 슈퍼컴퓨터라 할지라도 초당 10^{12}번의 연산만이 가능하다. 그러나 DNA는 대략 초당 10^{14}번의 계산이 가능하다.

셋째로, DNA의 막대한 병렬성을 계산에 이용하는 것뿐만이 아니라 정보를 저장하는 용도로도 사용할 수 있다. 일반적인 비디오 테이프인 경우 1bit를 저장하기 위해서 $10^{12}nm^3$이 필요하지만, DNA를 사용하면 $1nm^3$만이 필요하다. 이처럼 현재 사용되고 있는 컴퓨터에 비해 여러 가지 장점을 가지고 있으나 다음처럼 몇 가지 해결되지 못한 문제점들을 살펴볼 수 있다.

먼저 기본적으로 생물학 실험을 연산자로 사용하기 때문에 계산 시간이 많이 걸리며, 생물학 실험들은 확률적이고 통계적으로 정확하게 반응한다는 가정이 큰 지장이 없지만, 실험 방법들을 연산자로 사용하는 DNA 컴퓨팅에서는 치명적인 결과를 가져올 수 있어서 연산이 정확하지 않다. 이러한 예로는 최종해를 발견했는데도 해가 없다고 하거나(false negatives), 적절한 해가 아닌데도 최종해라고 판단하는(false positives) 등의 오류가 발생할 수 있다. 따라서 이런 오류들의 발생 가능성을 최소화시킨 후에 계산을 하거나, 오류가 발생하더라도 그 오류를 극복할 수 있는 방법이 있어야만 한다.

3. 외판원 문제를 위한 DNA 컴퓨팅 알고리즘

코드라고 하는 것은 주어진 문제를 표현하고 해결하기 위한 정보를 가지고 있는 염기배열(DNA sequence)을 의미한다. 문제에 따라 다양한 표현 방법이 존재하지만, 여기서는 Adleman의 그래프 표현 방법을 사용한다. 외판원 문제는 간선 염기배열은 연결을 하는 역할뿐만 아니라 가중치를 표현하기 위한 역할도 해 주어야 한다. 다음 <표 1>은 DNA 컴퓨팅 알고리즘을 간략화 한 것이다.

<표 1> DNA 컴퓨팅 알고리즘

<Table 1> DNA computing algorithm

1. 코드 표현 방법 결정(Encoding) : 분자 구조를 이용하여 주어진 문제에 대한 표현 방법 결정
2. 초기화(Initialization) : 결정한 표현 방법을 반영한 분자들을 생성
3. 합성(Synthesis) : 결찰법 등의 연산자(실험과정)를 이용하여 가능한 모든 초기 해집합을 생성함
4. 분리(Separation) : 생성된 초기해 집합에서 주어진 조건을 만족하지 않는 해들을 분리해 냄
5. 최종해 발견 : 분리 과정 이후 존재하는 해들을 최종해라고 판단함

염기 배열의 길이(l)를 20bp(base pair, 염기쌍)로 고정한다면 그래프 정점 i를 표현하기 위해 각각의 길이가 10bp(l/2)인 2개의 "위치 염기배열(position sequence)", P_{i1}, P_{i2}를 사용한다. 3' - 5'는 DNA 염

기배열의 방향성을 나타내는 것이다. 간선 *i* -> *j*를 표현하기 위해서는 간선 *i* -> *j*가 연결하는 두 정점 *i*, *j*의 위치 염기배열을 이용하는데, 정점 *i*의 3'-end 10 bp의 상보 염기배열(complementary sequence)인 Pi2와 정점 j의 5'-end 10bp의 상보 염기배열인 P_{j1}를 사용한다. 다음 [그림 1]은 정점 염기 배열의 예를 보였고, [그림 2]는 간선 염기 배열의 예를 나타내었다.

3' ATCGCATTAA | TGCCACCGGC 5'
P_{i1} P_{i2}
Vertex i
3' CTCACCGTAA | GCACCATTCA 5'
P_{j1} P_{j2}
Vertex j

[그림 1] 정점 염기 배열
[Fig. 1] vertex sequences

5' ACGGTGGCCG | GAGTGGCATT 3'
P_{i2} P_{j1}
Edge i -> j
[그림 2] 간선 염기 배열
[Fig. 2] edge sequences

본 논문에서는 [10]에서 제시한 코드 표현 방법을 사용한다. 정점 염기 배열은 위치 염기배열만을 포함하고, 간선 염기배열은 위치 염기배열과 가중치 염기배열을 모두 포함하도록 한다. 가중치 표현 방법은 낮은 가중치를 가지는 간선에 A/T쌍이 G/C쌍보다 많이 포함되도록 한다. 즉 A, C, G, T가 결합할 때 필요로 하는 수소결합의 수를 직접적으로 이용해서 가중치를 표현하는 것이다. 가중치 변환식은 식 (1)과 같다.

$$Fi = \begin{cases} \left| \frac{Kei}{Sh} - \frac{Wei}{Sw} \right| & \text{if } \left| \frac{Kei}{Sh} - \frac{Wei}{Sw} \right| \geq \theta, \\ 0 & otherwise. \end{cases} \quad (1)$$

식 (1)에서 K_{ei}는 간선 i의 수소 결합수이고, S_h는 전체 간선의 수소 결합 개수 총합이다. 그리고 W_{ei}는 간선 *i*의 실제 가중치이며, S_w는 전체 그래프에서 가중치 합계이다. θ값은 가중치를 표현하기 위해서 각 간선 가중치의 전체 가중치에 대한 비율을 구하고 이 값을 수소결합 변환 함수값의 비율과 비교하여 표현하는 것이다. 이 방법을 사용하면 정점 염기배열에 가중치 염기배열이 포함되지 않기 때문에 필요한 정점 염기배열이 단지 정점의 수만큼만 필요하게 되어, 필요한 염기배열의 수를 줄일 수 있다. 그리고 정점 표현 방법에는 변화가 없으나, 간선 염기배열에서 가중치 염기배열의 길이를 가변적으로 변화시키도록 한다(가변 길이 표현법, variable length representation). 수소 결합수를 직접적으로 반영한다는 성질을 이용하는 작은 가중치를 가지는 간선은 짧은 염기배열로 표시하고, 큰 가중치를 가지는 간선은 긴 염기배열로 표시한다는 것이다.

4. 새로운 중합효소 연쇄반응

앞에서 기술한 것처럼 중합효소 연쇄반응은 특정 염기 배열을 가지고 있는 DNA의 이중 가닥을 단일 가닥으로 분해 한 후, 증폭시키고자 하는 염기 배열의 상보 배열을 추가하여 이중 가닥으로 다시 만드는 것이다. 여기서는 증폭시키고자 하는 염기 배열을 단순하게 상보 결합을 추가하여 이중가닥으로 만들지 않고 좀 더 새로운 염기 배열을 얻을 수 있도록 하는 중합효소 연쇄 반응을 제안한다.

먼저 첫 번째 개발한 방법은 특정한 염기 배열 즉 시작 부분과 끝 부분이 특정 정점(시작 정점)인 염기배열을 선택한 후 임의의 두 위치를 선택하여 그 두 위치를 서로 바꾼 후, 염기 배열을 증폭시켜 상대적으로 다른 염기 배열들의 비율이 낮아지도록 하였다.

다음은 중합효소 연쇄 반응을 할 때 특정 염기 배열을 가지고 있는 DNA를 선택한 후, 그 선택된 염기 배열을 마치 환형 큐가 수행되는 것처럼 하여 한번의 연산이 일어나 1회전을 할 경우 주어진 염기 배열의 모든 염색체 위치가 하나씩 변형되고 2회전을 할 경우 1회전한 결과에다 다시 한번씩의 염색체 위치가 바뀌는 등으로 연산을 한 다음 증폭시킨다.

마지막으로 세 번째 방법은 중합효소 연쇄 반응 연산을 할 특정한 염기 배열을 선택하여, 그 염기 배열내의 가중치를 검사한 후 가장 가중치가 많은 두 곳을 선택해서 서로 위치를 교환한 후 중합 효소 연쇄 반응을 수행한다. 위에서 제안한 방법으로 연산을 수행하면서 현재 생성된 해보다 나쁜해를 가지는 경로를 생성하면 그 경로는 최종해가 될 가능성이 없으므로 해 집단에서 삭제하였다.

5. 실험 내용

DNA 컴퓨팅은 실제 생물학 실험 방법들을 연산자로 사용한다. 따라서 연산 비용과 시간이 일반 컴퓨터에 비해서 아주 많이 필요하게 된다. 또한 연산자인 실험 방법들이 오류의 가능성을 항상 포함하고 있기 때문에 실험 방법이 정확했더라도 원하는 결과가 나오지 않을 수 있다. 그래서 비용과 시간을 단축하면서 DNA 분자 실험의 효율을 극대화할 수 있는 시뮬레이션을 주로 이용한다. [9]는 VNA(Virtual Nucleic Acid) 라는 시뮬레이터를 개발하였고, [10]에서는 NACST(Nucleic Acid Computing Simulation Toolkit)이라는 시뮬레이터를 각각 개발하였다. 본 논문에서 실험은 진행중이며, 실험에서 사용한 파라미터는 [10]에 있는 파라미터를 동일하게 사용하였다. 다음 <표 2>는 외판원 문제를 해결하기 위해 사용한 파라미터들이다.

<표 2> DNA 컴퓨팅에 사용한 파라미터들

<Table 2> Parameters for DNA computing of TSP

변 수	값
위치 염기배열 길이	20
가중치 염기배열 최대 길이	40
가중치 염기배열 최소 길이	1
가중치 염기배열 증가크기	1
하이브리드 형성 오류확률	0.001

반응 오류는 없다고 가정하므로 항상 정확하게 반응한다고 생각하였다. 또한 매번 반응이 있을 때마다 해를 생성한다고 가정하였다. 실험은 도시수가 많지 않은 단순한 그래프이고, 생물학 실험 방법들에서 전기 영동법은 미리 정해진 길이 이상의 염기배열을 삭제하는데 사용하였으며 항체 친화력 반응을 이용하여 각 정점 방문 여부를 검사하는 실험이 진행중이다.

6. 결론 및 향후 연구과제

DNA 컴퓨팅 알고리즘이 가지고 있는 초 병렬성을 이용하여 NP-완전 문제들을 해결하는 새로운 방법들이 제안될 것으로 보이며, 또한 기존의 진화 연산 모델보다 생체 컴퓨터(bio-computer)의 구현 가능성이 높고, 다른 기술 분야로의 파급 효과도 클 것으로 기대한다. 그러나 DNA 컴퓨팅 알고리즘은 아직 연산 개발이나 코드 표현 방법들이 부족하다. 따라서 본 논문에서는 이런 문제점을 해결하고자 새로운 PCR 연산 기법을 3가지 설계하고 실험을 진행하고 있으며, 앞으로 연구과제는 문제의 크기가 아주 큰 문제를 해결할 수 있는 효율적인 방법들을 개발하는 일과 코드를 최적화하는 알고리즘을 추가하고 개발하는 일이다. 더불어 실험을 실제 실체처럼 수행할 수 있는 시뮬레이터를 확장하여 구현하는 일이다.

※ 참고문헌

[1] Leonard M. Adleman, "Molecular Computation Of Solutions To Combinatorial Problems", Science 266, pp.1021-1024, 1994.

[2] Thomas Back, Joosy N. Kok, Grzegorz Rozenberg, "Evolutionary Computation as a Paradigm for DNA-Based Computing", DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 2000.

[3] Soo-Yong Shin, Byoung-Tak Zhang, Sung-Soo Jun, "Solving Traveling Salesman problem Using Molecular Programming", Proc. ICEC, pp. 994-1000, 1999.

[4] Jan. J. Mulawka, Piotr Wasiewicz, Katarzyna Pietak, "Virus-enhanced Genetic Algorithms Inspired by DNA Computing", Lect. Not. Art. Int. - LNAI 1609, pp. 527-537, 1999.

[5] R. Deaton, R. C. Murphy, J. A. Rose, M. Garzon, D. R. Franceschetti, S. E. Stevens, Jr., "A DNA Based Implementation of an Evolutionary Search for Good Encodings for DNA Computation", Proc. ICEC, pp. 267-217, IEEE computer society, 1997.

[6] Erik Winfree, "Whiplash PCR for O(1) Computing", CIT, pp. 1-14, 1998.

[7] E. Stoschek, M. Sturm, T. Hinze, "On a DNA experiment for solving a certain NP-complete problem", 1999.

[8] Masami Hagiya, Masanori Arita, Daisuke Kiga, Kensaku Sakamoto, Shigeyuki Yokoyama, "Towards parallel evaluation and learning of boolean μ-formulas with molecules", Proc. DIMACS, 1997.

[9] Akio Nishikawa and Masami Hagiya, "Towards a System for Simulating DNA Computing with Whiplash PCR", Proc. ICEC, pp. 960-966, 1999.

[10] 신수용, "DNA 컴퓨팅 기법을 이용한 조합 최적화 문제의 해결", 서울대학교 석사학위 논문, 2000.

김 정 숙

1993년 2월 동국대학교 컴퓨터공학과 졸업(공학사)
1995년 2월 동국대학교 대학원 컴퓨터공학과 졸업 (공학석사)
1999년 8월 동국대학교 대학원 컴퓨터공학과 졸업 (공학박사)
2000년 3월 ~ 현재 김포대학 컴퓨터계열 소프트웨어 개발 전공 전임강사