# 배낭 문제를 해결하기 위해 DNA 코딩 방법을 적용한 DNA 컴퓨팅

# DNA Computing adopting DNA Coding Method to solve Knapsack Problem

김은경, 이상용
공주대학교 컴퓨터공학과, 공주대학교 정보통신공학부
Eun-Gyeong Kim, Sang-Yong Lee
Dept. of Computer Engineering, Division of Information & Communication Engineering, Kongju National University
E-mail : rotnrwk@kongju.ac.kr

## 요 약

배낭 문제는 단순한 것 같지만 조합형 특성을 가진 NP-hard 문제이다.이 문제를 해결하기 위해 기존에는 GA(Genetic Algorithms)를 이용하였으나 지역해에 빠질 수 있어 잘못된 해를 찾거나 찾지 못하는 문제점을 갖고 있다.

본 논문에서는 이러한 문제점들을 해결하기 위해 막대한 병렬성과 저장능력을 가진 DNA 컴퓨팅 기법에 DNA에 기반한 변형된 GA인 DNA 코딩 방법을 적용한 ACO(Algorithm for Code Optmization)를 제안한다. ACO는 배낭 문제 중 (0,1)-배낭 문제에 적용하였고, 그 결과 기존의 GA를 이용한 것 보다 초기 문제 표현에서 우수한 적합도를 생성했으며, 빠른 시간내에 우수한 해를 찾을 수 있었다.

## 1. 서론

배낭 문제는 조합 최적화 문제로서, 다항 시간(polynomial time)에 풀리지 않는 NP-hard 문제이다. 이 문제를 해결하기 위해 자연의 생물학적 진화 원리에 기반하여 확률적인 검색과 최적화를 수행하는 GA를 사용하고 있다. 그리고 탐색에 있어서 GA는 전역탐색(exploration)과 지역탐색(exploitation)을 사용하고 있다. 하지만 이 탐색 방법들은 다음과 같은 문제점을 갖고 있다. 전역탐색만 수행할 경우 무작위 검색과 같은 결과를 나타내며, 지역탐색만을 수행할 경우 국부 최적해를 찾는 결과를 초래하게 된다. 또한 개체 집단의 다양성과 다음 세대의 선택 강도가 각각 전역탐색과 지역탐색에 의해 결정되므로, 선택 강도의 증가와 감소는 개체 집단의 다양성을 감소 또는 증가 시킨다[1]. 따라서 검색 초기에는 개체 집단의 다양성을 강조하고, 검색이 진행될수록 점차로 선택 강도를 높여 지역탐색을 강조해야 한다. 하지만 검색 조절의 적합한 시점을 판단하는 것은 쉽지 않다.

따라서 본 논문에서는 이러한 문제점들을 해결하기 위해 ACO를 제안한다. ACO는 DNA 컴퓨팅 알고리즘에 DNA 코딩 방법을 적용한 것이다. DNA 코딩 방법은 GA의 문제점들을 잘 해결할 수 있었으며, 막대한 병렬성과 저장능력을 갖는 DNA 컴퓨팅은 빠른 탐색과 우수한 해를 효율적으로 찾을 수 있었다. 그리고 ACO의 성능을 평가하기 위해 (0,1) 배낭 문제에 적용하여 실험하였다.

## 2. 관련연구

### 2.1 배낭 문제

배낭 문제(knapsack problem)는 조합 최적화 문제로서 무게와 이익이 사전에 알려진 품목(물체)들의 집합이 주어질 때, 무게의 한계를 초과하지 않으면서 전체 이익이 최대가 되도록 배낭을 채우는 문제를 말한다[2][3]. 즉, n개의 품목과 하나의 배낭이 있을 때, 식(1)과 같이 이득 F(x)를 최대로 하는 품목을 선택하는 문제이다.

$$F(x) = \sum_{i=1}^{n} p_i x_i \qquad \text{식(1)}$$

배낭의 용량은 W이며, 식(2)의 조건을 만족해야 한다.

$$\sum_{i=1}^{n} w_i x_i \leq W \qquad (2)$$

이때 $x=(x_1,\cdots,x_n)$이며, $x_i$는 0또는 1의 값을 갖는다. 또한 $p_i$는 품목 i의 이득이며, $w_i$는 품목 i의 무게를 나타낸다.

$x_i$의 값에 0과 1사이의 소수를 허용하면 분할가능 배낭 문제(fractional knapsack problem)라고 하며, 0과 1만 허용하면 (0,1)-배낭 문제라고 부른다. 분할가능 배낭 문제에서는 품목들을 쪼개서 배낭에 넣을 수 있으나, (0,1)-배낭 문제에서는 품목을 통채로 배낭에 넣든지 아니면 버리든지를 결정해야 한다. 분할가능 배낭 문제는 욕심쟁이법으로 간단히 해결할 수 있다. 그러나 (0,1)-배낭 문제는 NP-hard에 속하는 문제로서 욕심쟁이법과 동적계획법 등으로 다항 시간에 풀리지 않는 어려운 문제이며, 이를 풀기 위해서는 $\Omega(2^n)$의 시간이 요구된다.

### 2.2 DNA 컴퓨팅

DNA 컴퓨팅은 합성 DNA를 정보소자로 사용하며, 풀고자 하는 문제에 대한 해법을 DNA 코드로 기술한다. 그리고 주어진 DNA 조각들로부터 화학적으로 합성, 검출함으로써 답을 찾아 내는 기술을 말한다. DNA 컴퓨터는 기존 컴퓨터의 이진법을 사용하지 않으며, DNA를 구성하는 4가지 염기인 A(Adenine), C(Cytosine), G(Guanine), T(Thymine)를 사용한다. 이들 염기는 상온에서 A와 T, G와 C가 결합하는 상보성을 갖으며[4], 대용량 데이터를 저장할 수 있는 메모리 기능 또한 가지고 있다. 그리고 복잡한 염기 조합의 패턴은 하나의 유전 정보를 담고 있으며, 인체 내에서 자연 발생하는 효소에 의해 읽혀지고 있다. 이러한 효소는 생물학 실험 방법들과 함께 DNA 컴퓨팅의 연산자로 사용되고 있다.

이러한 DNA 컴퓨팅은 매우 낮은 에너지로 작동되기 때문에 많은 에너지가 필요없다. 그리고 나노 수준의 막대한 병렬성을 이용하여 NP-complete에 효과적인 접근이 가능하며, 계산 속도와 정보의 저장 및 처리 효율에서도 우수함을 보이고 있다[5,6].

### 2.3 DNA 코딩 방법

DNA 코딩 방법은 1995년 Yoshikawa가 제시한 변형된 형태의 유전자 알고리즘이다[7]. 일반적인 유전자 알고리즘은 0, 1을 사용하지만, DNA 코딩 방법은 A(Adenine), G(Guanine), T(Thymine), C(Cytosine)를 사용한다. 그리고 A, T, G, C 중 3개의 염기(코돈 ; codon)가 하나의 의미 단위인 아미노산을 지정하며, 그 수는 중복을 제외한 20가지가 있다.

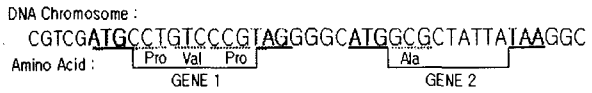


그림 1 DNA 염색체의 변역 예

그림 1에서 보는 것처럼 염기서열은 시작 코돈인 ATG에서 부터 정지 코돈인 TGA (TAA, TAG)까지 아미노산 번역을 한다. 이러한 아미노산 번역은 짧은 DNA 코드에서도 많은 정보를 얻을 수 있다.

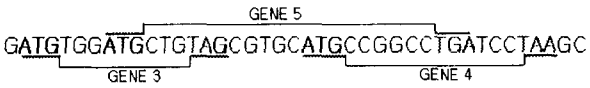


그림 2 유전자 중복의 예

DNA 코딩 방법의 특징은 그림 2에서 보는 것처럼 염색체의 중복을 효율적으로 표현할 수 있으며, 하나의 아미노산을 만드는 코돈이 여러 개이므로 지식 표현이 쉽다. 또한 교배점이 임의로 주어지기 때문에 염색체의 길이가 가변적이다. 이러한 가변길이 표현방법은 염색체의 길이가 길수록, 고정길이 표현 방법 보다 성능이 훨씬 우수하며 다양한 개체군을 생성한다. 이러한 특징들은 염색체의 기능과 동작을 더욱 생물학적으로 가깝게 모델링할 수 있다.

## 3. ACO

ACO는 배낭 문제를 GA로 해결할 때, 발생했던 문제점들을 해결하기 위해 DNA 컴퓨팅에 DNA 코딩 방법을 적용하여 해결하였다.

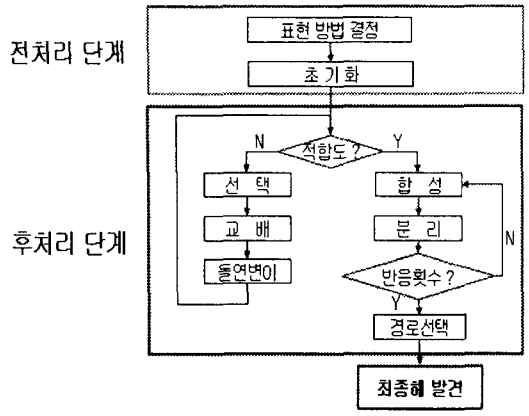


그림 3 ACO의 흐름도

ACO는 그림 3과 같이 전처리 단계와 후처리 단계로 구성된다.

먼저 ACO의 전처리 단계를 살펴보면, 표현 방법 결정 과정과 초기화 과정으로 나뉜다. 표현 방법 결정 과정에서는 주어진 DNA 코드를 DNA 코딩 방법으로 품목 i의 무게 $w_i$와 이득 $p_i$를 생성한다. $w_i$와 $p_i$는 바로 DNA 코드로 표현할 수 없으므로 다음의 순서에 따라 생성한다. 먼저 시작 코돈(ATG)의 위치를 파악하고, i번째 시작 코돈에서 (i+1)번째 시작 코돈 전까지의 DNA 코드를 $w_i$로, (i+1)번째 시작 코돈에서 (i+2)번째 시작 코돈 전까지의 DNA 코드를 $p_i$로 표현한다. 그 다음 순서는 $w_i$의 실제 무게를 표현하기 위해 A/T쌍과 G/C쌍의 수소 결합 수를 각각 낮은 무게와 높은 무게에 포함시켜 $w_i$를 생성한다. 그리고 $p_i$는 방사능 라벨을 부착한다. 그림 4는 품목 i의 표현 예이다.

3' ATGCCAG | ATGCAATG | ATGAACGGTAACTG | ATGGGCAT 5'
$w_1$ $p_1$ $w_2$ $p_2$

그림 4 품목 i의 표현 예

이렇게 표현된 품목 i인 $w_ip_i$와 품목 (i+1)인 $w_{i+1}p_{i+1}$의 연결은 상보결합인 $\overline{p_iw_{i+1}}$ 로 표현한다. 그림 5는 품목 i와 품목 (i+1)에 대한 연결을 나타낸다.

TACGTTAC | TACTTGCCATTGAC
$\overline{p_1}$ $\overline{w_2}$

그림 5 품목 i와 품목 i+1의 연결 예

위와 같은 방법으로 표현 방법 결정 과정이 끝나면, DNA 코드를 아미노산 번역표에 따라 아미노산 코드로 번역하는 초기화 과정을 수행한다.

ACO의 후처리 단계를 살펴보면 적합도를 만족하지 않을 경우, DNA 코딩 방법의 연산자인 선택, 교배, 돌연변이를 이용하여 적합도를 재평가한다. 만족하는 경우, 반응횟수만큼의 합성과 분리과정을 거친 후 우수한 해를 찾는다.

표 1 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 |

적합도 평가는 <표 1>의 아미노산에 부여된 코드를 적용하여 비례 선택법(roulette wheel)으로 평가한다. 그리고 잘못된 결합이나 결합 위치 이동과 같은 생물학 실험에서 발생할 수 있는 오류의 조건을 미리 제거한다. 교배는 2점 교배를 하고 국소 해에 빠질 위험성을 벗어나기 위해 랜덤하게 교배점을 선택한다. 돌연변이는 모든 코드에서 수행하며, 세대수 만큼 반복한다.

이렇게 생성된 코드 중, 적합도를 만족하는 코드를 선택하여 주어진 반응횟수 만큼의 합성과 분리 과정을 거친다. 이 분리 과정에서 해가 될 가능성이 없는 것 즉, 배낭의 용량을 초과한 값은 항체 친화력 반응, 겔 전기 영동법 등과 같은 생물학적 연산자를 이용하여 제거한다. 그리고 겔 전기 영동법으로 배낭의 용량이 W인 것을 선택하고, BAS를 이용하여 최고의 이득을 갖는 DNA 코드만을 추출하여 최종해로 한다.

## 4. 실험 및 분석

ACO의 성능을 확인하기 위해 <표 2>의 배낭 문제를 GA와 ACO로 해결하여 비교 평가하였다. 단, 배낭의 용량(C)는 1105의 조건을 만족해야 한다.

표 2 배낭 문제의 상태값

| 번호(n) | 이득($p_i$) | 무게($w_i$) |
|---|---|---|
| 1 | 50 | 810 |
| 2 | 34 | 215 |
| 3 | 78 | 275 |
| 4 | 18 | 80 |

시뮬레이션은 PIV, 2GHz, RAM 512M의 PC에서 C언어를 사용하여 진행되었다. 실험에 사용한 파라미터들은 <표 3>와 같이 설정하였다.

표 3 파라미터들

| 변 수 | | ACO | GA |
|---|---|---|---|
| 집단 크기 | | 400 | 400 |
| 세대수 | | 100 | 100 |
| 교배 연산 비율 | | 0.4 | 0.4 |
| 돌연변이 연산 비율 | | 0.1 | 0.1 |
| 총 반응횟수 | 반복횟수 | 10 | |
| | 반응횟수 | 100 | |
| 생물학적 실험 오류율 | | 0.01 | 0.01 |

또한 집단의 크기를 100, 200, 300, 400의 상태에서 GA와 ACO의 초기 개체군의 크기를 실험하였다. 그리고 각 집단의 크기에서 적합도 범위 안에 만족하는 초기 개체수는 <표 4>와 같다.

표 4 집단의 크기에 대한 초기 개체수

| 집단의 크기 | ACO의 초기 개체수 | GA의 초기 개체수 |
|---|---|---|
| 100 | 6 | 8 |
| 200 | 7 | 9 |
| 300 | 7 | 10 |
| 400 | 8 | 10 |

적합도 평가는 집단의 크기가 100인 경우와 400인 경우를 가지고 적합도를 비교하였다. 그림 6에서와 같이 ACO가 GA의 해보다 적합도에서 우수하다는 것을 알 수 있었다. 또한 각 세대별 평균 적합도에서도 ACO는 우수한 결과를 얻었다. 이는 ACO가 초기의 적은 개체수를 가지고 배낭문제의 용량을 충분히 만족시킬 수 있었기 때문이다. 따라서 고정길이가 아닌 가변길이의 적은 서열로도 최대의 이득을 표현할 수 있었다.

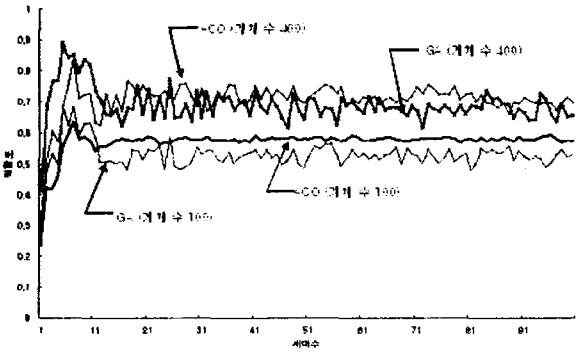

그림 6 적합도 평가

## 5. 결론

본 연구에서는 배낭 문제를 기존의 방법인 GA로 해결하였을때 발생되는 문제점을 분석하고, 이를 해결하기 위해 DNA 컴퓨팅에 DNA 코딩 방법을 적용한 ACO를 제안하였다. ACO는 GA보다 집단의 크기에 상관없이 적합도를 만족시키는 결과를 얻었다. 이는 어떠한 조건을 주어도 배낭문제의 조건인 용량을 정확히 만족하면서도, 이득을 최대로 발생시킬 수 있다는 것이다. 따라서 GA의 전역탐색이나 지역탐색의 한계점을 극복할 수 있으며, 빠른 시간내에 우수한 해(품목)을 찾을 수 있었다.

## 6. 참고문헌

[1] Z. Michalewicz, Genetic Algorithms + Data Structures=Evolution Programs, Springer-Verlag, 3rd, revised and extended edition, 1999.

[2] 진강규, "유전알고리즘과 그 응용", 교우사, pp. 282-289, 2000.

[3] S. Khuri, T. Bäck, J. Heitkötter, "The Zero/one Multiple Knapsack Problem and Genetic Algorithms", Proc. '94 ACM Symp. on Applied Computing, Phoenix, AZ, 1994.

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

[5] G. H. Gonnet, C. Korostensky, S. A. Benner, "Evaluation Measures of Multiple Sequence Alignments", Journal of Computational Biology 7(1-2): 261-276 , 2000.

[6] R. Deaton, S. A. Karl, "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.