

정칙 이진 행렬을 이용한 메타 유전 알고리즘 설계 방안

박하얀^{O*} 김용혁* 윤유림^{**}

*광운대학교 컴퓨터소프트웨어학과, **서울대학교 컴퓨터공학부
klarfarbe@gmail.com, yhdfly@kw.ac.kr, yryoon@soar.snu.ac.kr

A Design Methodology of Meta Genetic Algorithms

Using Nonsingular Binary Matrices

Ha-yan Park^{O*} Yong-Hyuk Kim* Yourim Yoon^{**}

*Department of Computer Science and Engineering, Kwangwoon University

**School of Computer Science and Engineering, Seoul National University

요 약

본 논문에서는 정칙 이진 행렬을 이용하여 유전 알고리즘의 성능을 개선할 수 있는 메타 유전 알고리즘을 설계한다. 정칙 이진 행렬은 유전 알고리즘에서 사용되는 이진 인코딩에서의 기저 변환에 중요하게 쓰일 수 있다. 이 논문에서는 정칙 이진 행렬의 기저 변환을 위한 아이디어와 더불어 정칙 이진 행렬의 표현과 재조합 연산에 대한 아이디어를 제시했던 연구들을 소개하고, 메타 유전 알고리즘을 위한 변이와 초기 해집단 생성, 평가에 대한 방법론을 제시한다.

1. 서 론

행렬 A 가 $A \in M_{n \times n}(\mathbb{Z}_2)$ 를 만족하면, A 는 *이진 행렬* (binary matrix)이라고 한다. 이진 행렬은 \mathbb{Z}_2 벡터 공간 위의 그래프의 컷(cut) 부분 공간이나 사이클(cycle) 부분 공간을 다루는데 유용하다.

또한, \mathbb{Z}_2 벡터 공간 상에서 조합 문제의 기저 변경 표현에 사용할 수 있다. 이진 인코딩을 하는 기존의 접근 방식은 일반적으로 고유한 표준 기저를 사용한다. 다른 기저를 고려할 경우, 문제 공간의 기저 원소와 결합 구조의 견고함은 기존의 것과 완전히 달라질 수도 있다.

본 논문에서는 이진 인코딩의 기저 변환에 쓰이는 정칙 (nonsingular) 이진 행렬, $GL_n(\mathbb{Z}_2)$ 을 조사하는 아이디어를 제안한다.

논문은 다음과 같이 구성되어 있다. 2절에서는 이진 인코딩 시 기저 변환을 위한 아이디어와 유전 알고리즘을 위한 표현과 재조합을 $GL_n(\mathbb{Z}_2)$ 에서 설계한 연구들을 소개한다.

3절은 2절에서 소개된 논문에서 다루지 않은 아이디어들을 제안하고, 4절에서는 결론과 앞으로의 연구 방향을 제시한다.

2. 이진 연구

2.1 기저 변환 (change of basis)

Kim과 Yoon은 이진 인코딩(binary encoding)에서 기저 변환(change of basis)의 중요성을 연구하였다[1].

임의의 벡터가 기저 벡터의 선형 조합으로 유일하게

표현될 때, n 차원 벡터 공간의 기저는 n 개 벡터의 수열이다. 벡터 공간에서는 종종 하나 이상의 기저를 사용하는 것이 가능하기에, 하나의 기저에 대한 벡터의 좌표 표현을 가역 선형 변환을 이용하여 다른 기저에 대한 동등한 표현형으로 변환할 수 있음을 이해하는 것이 중요하다. 이러한 변환을 *기저 변환*이라고 한다. 아래 정리는 선형 대수학의 기본적인 이론으로부터 파생된 것이다.

정리 1. B_1 과 B_2 가 \mathbb{Z}_2^n 의 기저라고 하자. 기저 B 에 관한 v 의 표현형을 $(v)_B$ 라 할 때, 임의의 $v \in \mathbb{Z}_2^n$ 에 대하여, $T(v)_{B_1} = (v)_{B_2}$ 를 만족하는 정칙 행렬 $T \in M_{n \times n}(\mathbb{Z}_2)$ 가 존재한다.

정리 1의 행렬 T 를 기저 B_1 으로부터 B_2 로의 *좌표 변환*(coordinate-change) 행렬이라고 한다.

$e_j = (0, \dots, 0, 1, 0, \dots, 0)$ 는 j 번째 위치만 1이고, 나머지는 모두 0인 \mathbb{Z}_2^n 의 원소라 할 때, \mathbb{Z}_2^n 의 표준 기저 B_S 는 $\{e_1, e_2, \dots, e_n\}$ 이다. 정칙 행렬 $T \in M_{n \times n}(\mathbb{Z}_2)$ 가 주어질 때, 행렬 T 는 표준 기저로부터 T 와 관계있는 또 다른 표준 기저 B_T 로의 좌표 변환 행렬로 여겨질 수 있다. 임의의 $v \in \mathbb{Z}_2^n$ 에 대하여, $T(v)_{B_S} = (v)_{B_T}$ 이 성립하고, 그러면 $B_T = \{Te_1, Te_2, \dots, Te_n\}$ 이다. 따라서, 새로운 기저를 찾는 문제는 적당한 정칙 이진 행렬(nonsingular binary matrix, NBM)을 찾는 문제와 같다.

유전자 재배치(gene reordering)는 기저 변환에서의 특별한 경우로 고려된다. 만약 T 가 치환 행렬

(permutation matrix)일 뿐이라면, 기저 변환의 의미는 인코딩 상에서의 유전자의 위치를 재배치하는 것을 의미한다. 기저를 변경하는 아이디어는 유전자의 재배치에 대한 것보다 일반적이나, 이미 연구된 내용은 이진 인코딩을 이용하는 문제에 적합하지 않으므로 몇 가지 제안이 필요하다[1].

논문[1]에서는 이진 변수의 n 차원 함수를 테스트에 사용하였다. 함수는 Kuffman의 NK 지형에 최적화 및 단순화 되어 있다. Kuffman은 n 비트의 함수에 대해 각 비트의 적합도는 자신과 이웃하는 k 개의 비트에 의존한다고 정의했다.

NK 지형은 종종 유전 알고리즘(genetic algorithms)의 성능을 테스트할 때 사용된다. 논문[1]에서 테스트에 사용한 3개의 함수는 각각 $k=1, 2, n-2$ 의 값을 가지고, 각 변수의 적합도는 다른 k 변수들에 의존한다.

모든 테스트 함수는 정칙 이진 행렬을 통해 각각의 테스트 함수에 대한 기저 변환 행렬을 선택할 수 있다[1]. 기저 변환 행렬을 T 라고 한다면, 이의 역행렬인 T^{-1} 는 가우스 소거법(Gaussian elimination)에 의해 효율적으로 계산 가능하다.

기저 변환 전과 후의 테스트 함수에 대해 실험한 결과, 3개의 테스트 함수 모두 기저 변환 후의 모양이 더 좋은 결과를 보여주었다[1]. 즉, 표준 기저보다 실험에 사용한 새로운 기저가 월등히 좋은 성능을 보여준 것이다. 이로써 기저 변환의 이점을 알 수 있었다.

논문[1]에서 제시한 이진 표현에서의 기저 변환은 단순하지만 강력한 아이디어다. 이진 인코딩에서 기저 변환의 중요성을 설립한 최초의 논문이기도 하다. 그러나 좋은 기저를 찾는 방법이 더욱 연구가 되어야 하고, 정규 선형 그룹과 같은 정칙 이진 행렬에 대한 조사가 시작되어야 한다. 몇몇의 문제들은 기저 변환으로 성능이 거의 변하지 않을 수도 있을 것이다.

본 논문에서는 최적의 정칙 이진 행렬을 찾는 방법론을 제시하고자 한다.

2.2 정칙 이진 행렬의 재조합과 표현

Kim과 Yoon은 유전 알고리즘을 위한 표현과 재조합 연산을 $GL_n(\mathbb{Z}_2)$ 에서 설계하는 방법을 제안하였다[2]. 이진 행렬의 정의와 유용성은 2.1절에서 충분히 언급하였다.

정칙 이진 행렬에 관한 가장 직관적이고 자연스러운 표현은 2차원의 $n \times n$ 이진 인코딩이다. 또는 더욱 간단

하게 n^2 길이의 이진 선형 문자열을 사용할 수 있다.

Cohoon과 Paris[3], Anderson 등[4] 여러 사람들이 2차원 교차에 대한 다양한 방법을 주장했다. 또한 2차원 인코딩에서 새로운 스키마(schema) 창조의 힘이 선형 인코딩 교차에서 낮은 점을 보완하기 위해 지역적 교차의 방법이 제안되면서 2차원 표현과 지역적 교차의 결합으로 정보 손실의 감소와 새로운 스키마 창조의 힘 모두를 기대할 수 있었다.

그렇지만 이와 같은 방법의 피할 수 없는 단점은 타당성(feasibility) 검사와 구조의 수정이 필요하다는 것이다. 이에 대한 실례로, 각각 가역적인 두 부모해로부터 비가역적인 자식해가 생성되는 것을 볼 수 있었다[2]. 구조의 수정을 피할 수 있는 방법 중 하나는 최대의 독립적인 열(또는 행)을 찾는 방법이지만, 이는 가우스-조르당 소거법(Gauss-Jordan elimination)에 의해 처리되어야 하므로 $O(n^3)$ 의 시간 복잡도를 가져 오랜 시간이 걸린다.

따라서 일반적 접근에서 더 나아가 모든 가역행렬은 기본 행렬 곱으로 표현될 수 있다는 점에서 착안하여 기본 행렬을 이용한 방법을 제안하였다.

$M_{n \times n}(\mathbb{Z}_2)$ 의 기본 행렬을 소개하기 전에, 기본 행 연산을 정의하면 다음과 같다.

정의 1. 행렬 A 가 $A \in M_{n \times n}(\mathbb{Z}_2)$ 를 만족한다고 하자. A 의 어떠한 행이 다음 두 연산을 따르면, *기본행 연산*(*elementary row operation*)이라고 정의한다.

- (i) A 의 어떠한 두 행을 교환하고,
- (ii) A 의 한 행을 다른 행에 더한다.

기본 행렬은 기본 행 연산 수행이 (i) 또는 (ii)를 따르는지에 따라 타입 1이나 타입 2로 불리며, 이는 연산에 의존적이다. 이 연산으로 각각의 기본 행렬 E 에 대해 $E^2 = I$ 임을 쉽게 보일 수 있다.

논문[2]에서 i 번째 행과 j 번째 행을 서로 교환한 타입 1의 기본 행렬을 S_n^{ij} 로 정의하고, i 번째 행을 j 번째 행에 더한 타입 2의 기본 행렬을 A_n^{ij} 로 정의한다[2].

모든 가역 행렬은 기본 행렬 곱으로 표현된다는 것은 알고 있는 사실이다[5]. 그러므로 $GL_n(\mathbb{Z}_2)$ 상에서 기본 행렬 곱으로 해를 표현할 수 있다면, 타당성 유지에 큰 이득을 얻을 수 있다.

기본 행렬을 사용할 때, 기본 행렬의 각 원소를 가변 길이 선형 문자열로 표현할 수 있고[2], 이로 인해 새로운 인코딩이 주어진다. 가변 길이 문자열에 대한 어떤 재조합도 이 인코딩에 관한 재조합 연산자로 사용할 수

있다. 재조합 연산자 중 순차 정렬로 인한 기하학적 교차는 효율적인 수행이 기대된다. 순차 정렬로 인한 기하학적 교차는 자식해가 부모해의 고유한 특성을 유지하는데 도움이 된다.

가변 길이 문자열을 위한 거리로서, 편집 거리(edit distance)가 일반적이다. 이는 단일 기본 행렬의 삽입, 삭제, 대체의 최소 숫자로 하나의 문자열을 다른 것으로 변환한다. 이 거리와 연상되는 기하학적 교차를 상동 기하 교차(homologous geometric crossover)라고 한다.

두 개의 문자열은 재조합 전에 최적의 상태로 정렬된다. 여기서 정렬은 부모해 문자열이 서로 잘 일치하도록 쪽 나열하는 것을 의미한다. 다시 말해, 부모해 문자열의 길이가 서로 같지 않은 경우, 상대적으로 길이가 짧은 부모해에 형식적으로 ‘-’를 문자열의 임의의 자리에 끼워 넣어 길이를 확장시켜 해밍(Hamming) 거리가 최소가 되게 하고, 이 문자열들을 이용해 다수의 선형 문자열을 생성하는 것이다[2].

기본 행렬에 대한 표현을 기본 행렬의 순서로 고려할 때, 행렬에 대한 표현은 유일하지 않을 수도 있다[2]. 게다가 정칙 행렬에 대해 얼마나 많은 동등한 표현이 존재하는지를 아는 것은 쉽지 않다. 논문[2]에서 몇몇의 동치 관계를 간단한 아이디어를 통해 제안했다.

$C_n(i, j)$ 는 $\{S_n^{ij}, A_n^{ij}, A_n^{ji}\}$ 이라고 하자. $E_1 \in C_n(i_1, j_1)$, $E_2 \in C_n(i_2, j_2)$ 이고, $\{i_1, j_1\} \cap \{i_2, j_2\} = \emptyset$ 이면, $E_1 E_2 = E_2 E_1$ 이다. n 의 크기가 작으면 교환 가능한 경우가 드문 것처럼 보이지만[2], 큰 n 에 관해서는 그렇지 않다. 임의로 선택하는 기본 행렬 E_1 과 E_2 에 대하여, $E_1 E_2 = E_2 E_1$ 일 확률은 다음과 같다.

$$P(E_1 E_2 = E_2 E_1) = \begin{cases} \frac{(n-2)(n-3)}{n(n-1)}, & n > 3 \\ 0, & n \leq 3. \end{cases}$$

그림 1은 행렬 크기 n 에 따른 $E_1 E_2 = E_2 E_1$ 의 확률을 보여준다. 여기서 n 이 충분히 크면 $P(E_1 E_2 = E_2 E_1)$ 이 1에 가까워지는 것을 알 수 있다. 이는 $E_1 E_2 = E_2 E_1$ 의 경우가 n 이 커지면 매우 빈번하게 나타남을 의미한다.

이 동치 관계를 기반으로, 논문[2]에서 방향성 비순환 그래프(directed acyclic graph, DAG)의 노드를 기본 행렬로 사용하여 가환 행렬을 표현했다. 각 에지(edge)는 양 클래스 간의 비호환성을 의미한다.

$A = E_1 E_2 \cdots E_m \in GL_n(\mathbb{Z}_2)$ 일 때, 다음을 따르는 A 의 방향성 비순환 그래프를 가정해보자.

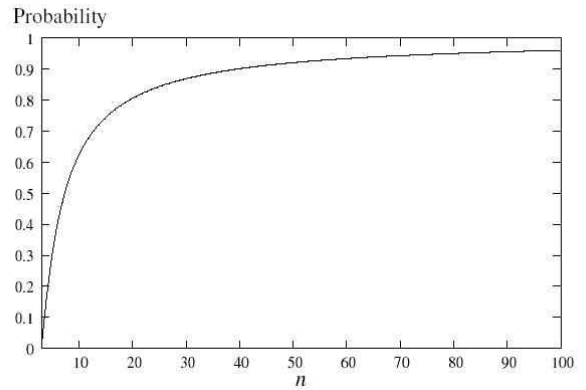
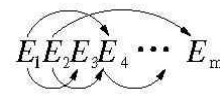
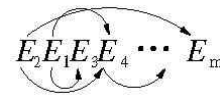


그림 1. 행렬 크기 n 에 따른 확률 $P(E_1 E_2 = E_2 E_1)$ [2]



그러면, $E_1 E_2 \cdots E_m$ 은 방향성 비순환 그래프 상의 위상 순서가 된다. 다음의 내용 역시 방향성 비순환 그래프 상의 또 다른 위상 순서이다.



정칙 이진 행렬에 방향성 비순환 그래프 표현을 사용할 때, 방향성 비순환 그래프 재조합과 같이 알려진 것을 적용하거나 동질성을 기반으로 기하학적인 연산자를 설계할 수 있다.

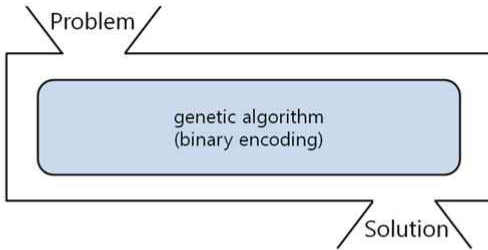
또한 방향성 비순환 그래프의 표현과 단순한 재조합을 이용하여, 부모해의 가변 길이 문자열을 기반으로 생성된 자식해보다 더 부모해의 고유한 성질을 잘 유지함을 알 수 있었다. 나아가 좀 더 정교한 동치 관계들을 발견할 수 있었다[2].

논문[2]에서 다른 정칙 이진 행렬 $GL_n(\mathbb{Z}_2)$ 은 표준 이진 인코딩의 기저 변환에 이용되어 중요하다. 논문[2]는 타당성을 유지하는 이점을 갖는 기본 행렬을 이용한 몇 가지 아이디어를 제공하였다. 가변 길이 선형 문자열과 방향성 비순환 그래프는 기본 행렬 곱으로 이해할 수 있는 각 정칙 이진 행렬 표현으로 제안되었다. 자명한 동치 관계만으로는 하나의 행렬이 서로 다른 기본 행렬 곱의 표현으로 종종 나타날 수 있어서, 자명하지 않은 동치 관계를 더 발견하여 보다 많은 정보를 얻고자 시도하였다.

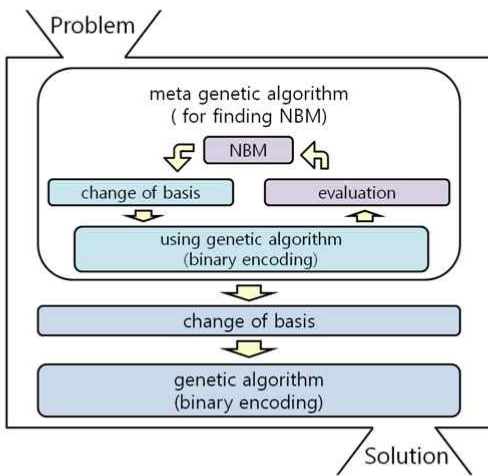
본 논문은 논문[2]의 결과를 토대로 유전 알고리즘을 이용하여 이진 인코딩의 좋은 기저를 찾는 아이디어들을 제공한다.

3. 메타 유전 알고리즘 (meta genetic algorithm)

어떤 문제에 대한 해를 찾기 위해 유전 알고리즘을 사용할 때, 유전 알고리즘의 성능을 개선하기 위해 사용하는 상위의 유전 알고리즘을 *메타 유전 알고리즘(meta*



(a) 기존의 문제 해결 과정



(b) 제안하는 문제 해결 과정

그림 2. 문제 해결 과정

genetic algorithm, meta GA)이라고 한다.

본 논문은 좋은 기저를 찾기 위해 정칙 이진 행렬을 조사하는 메타 유전 알고리즘과 이를 이용하여 유전 알고리즘의 성능을 개선할 수 있는 기저 변환에 대한 아이디어를 제안한다. 그림 2(a)는 기존의 문제 해결 과정을, 그림 2(b)는 제안하는 문제 해결 과정을 간단하게 도식화한 것이다.

2.1절에서는 기저 변환이 유전 알고리즘의 성능을 개선할 수 있다는 연구 결과를, 2.2절에서는 정칙 이진 행렬을 찾는 메타 유전 알고리즘의 아이디어 중 교차 연산과 인코딩에 관한 연구 결과를 얻었다. 이 절에서는 정칙 이진 행렬을 조사하는 메타 유전 알고리즘의 연산 중 자명하지 않으면서 논문[2]에서 언급되지 않은 변이 (mutation) 연산과 초기 해 집단 생성(population initialization), 평가(evaluation)에 대한 내용을 소개한다.

3.1 변이 (mutation)

논문[2]에서의 연구 내용을 통해 교차 연산이 일어나면 자식해는 두 부모해의 고유한 특성을 물려받는 것을 알 수 있다. 그러면 부모해의 유전자들을 좋은 방향으로 결합하는 것은 가능하지만 부모해에 없는 유전자를 도입하기는 힘들다. *변이 연산*은 부모해에 없는 속성을 자식해에 도입하는 역할을 한다[6].

두 부모해로부터 얻어진 자식해를 행렬 $B \in GL_n(\mathbb{Z}_2)$ 라고 하고, B 를 구성하는 기본 행렬들을 $E_i (i = 1, 2, \dots, 3n(n-1)/2)$ 라고 하자. 그러면 기본 행렬의 각 원소를 가변 길이 선형 문자열로 표현할 수 있다[2]. 즉, $B = E_1 E_2 \dots E_k$ 의 꼴로 나타낼 수 있다.

교차 연산으로 인해 부모해의 특성을 가지고 있는 B 에 대한 변이 연산은 삭제와 변경, 추가의 3가지 타입으로 분류 가능하고, 이는 다음과 같다.

타입 1. $B = E_1 E_2 \dots E_k$ 일 때, 각각의 E_i 는 p_1 의 확률로 삭제할 수 있다(최대 k 개).

타입 2. 각 E_i 에 대하여, p_2 의 확률로 E_i 를 $3n(n-1)/2$ 개의 기본 행렬 중 임의로 선택한 E_l 로 바꿀 수 있다(최대 k 개).

타입 3. 곱해진 각 기본 행렬의 앞과 뒤의 위치에 인덱스를 주어 이를 e_0, e_1, \dots, e_k 라고 하자. 각 a_i 의 위치에는 p_3 의 확률로 $3n(n-1)/2$ 개의 기본 행렬 중 임의로 선택한 E_l 이 삽입될 수 있다(최대 $k+1$ 개).

그림 3은 $B \in GL_n(\mathbb{Z}_2)$ 에 대한 $3n(n-1)/2$ 개의 기본 행렬 중 임의의 기본 행렬 E_l 을 선택하는 방법을 의사코드로 표현한 것이다.

그림 4(a)는 주어진 행렬 $E_1 E_2 \dots E_{i-1} E_i E_{i+1} \dots E_k$ 에 대하여, p_1 의 확률로 E_i 가 삭제되는 타입 1, 그림 4(b)는 p_2 의 확률로 E_i 가 E_l 로 바뀌는 타입 2, 그리고 그림 4(c)는 e_i 번째 위치에 E_l 이 추가되는 타입 3에 대한 예를 보여준다.

1) $1 \leq i \leq n, 1 \leq j \leq n$ 이고 $i < j$ 인 두 i, j 로 생성할 수 있는 S_n^{ij} 는 i 와 j 의 순서가 바뀌어도 같은 기본 행렬의 꼴을 갖는다. 따라서 ${}_n C_2 = n(n-1)/2$ 개까지 생성될 수 있다. A_n^{ij} 는 i 와 j 의 순서가 바뀌면 서로 다른 기본 행렬이 된다. 즉, $n(n-1)$ 개만큼 생성할 수 있다. 따라서, 임의의 정칙 이진 행렬 $B \in GL_n(\mathbb{Z}_2)$ 의 생성 가능한 기본 행렬의 개수는 $n(n-1)/2 + n(n-1) = 3n(n-1)/2$ 가 된다.

3.2 초기 해집단 생성 (population initialization)

자식해를 얻기 위해서는 먼저 부모해가 존재해야 한다. 부모해 역시 가변 길이 선형 문자열로 나타낼 수 있

```
Select_Random_Elementary_Matrix(n) {
  do {
    i ← rand()%n+1;
    j ← rand()%n+1;
  } until (i ≠ j);
  if i > j, then swap(i, j);
  switch (rand()%3) {
    case 0 :
      return Snij;
    case 1 :
      return Anij;
    case 2 :
      return Anji;
  }
}
```

그림 3. 기본 행렬 E_i를 임의로 선택하는 의사 코드

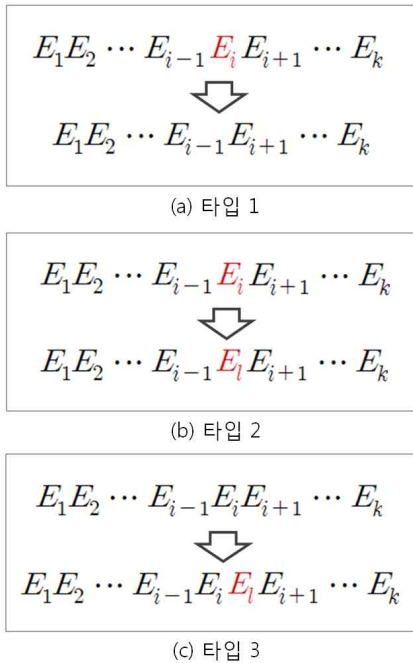


그림 4. 변이 연산의 각 타입에 대한 예

어야 하므로, 초기해를 생성하는 방법은 길이를 고정하는 타입 1과 고정시키지 않는 타입 2로 분류할 수 있다.

타입 1. 행렬 $E \in GL_n(\mathbb{Z}_2)$ 에는 $3n(n-1)/2$ 개의 기본 행렬 $E_1, E_2, \dots, E_{3n(n-1)/2}$ 이 존재하고, 부모해는 k 개의

기본 행렬의 곱으로 그 길이가 정해져 있다고 하자. k 개의 기본 행렬은 각각 $1 \leq p_i \leq 3n(n-1)/2, i = 1, 2, \dots, k$ 을 만족하는 확률 상수 p_i 를 갖는다고 할 때, m 번째로 생성되는 부모해 $P_m = E_{p_1}E_{p_2} \dots E_{p_k}$ 와 같이 정의할 수 있다. 이 때, $i \neq j$ 인 임의의 p_i 와 p_j 에 대해 $p_i = p_j$ ($1 \leq i \leq k, 1 \leq j \leq k$)가 성립할 수도 있다. 즉, 어떤 기본 행렬 E_i 가 중복되어 나타날 수 있다.

타입 2. 부모해 P_m 의 선형 문자열 길이가 정해져 있지 않다고 하자. k 는 $1 \leq k \leq 3n(n-1)/2$ 으로 범위가 정해져 있다고 하면, 그 중 하나의 값을 임의로 선택하여 k 로 정한다.

그러면, P_m 의 길이는 k 로 정해지므로, 타입 1의 방법으로 부모해를 생성하도록 한다.

3.3 평가 (evaluation)

3.2절에서 제안한 아이디어로 생성된 부모해에 2.2절과 3.1절에서 제안한 연산을 통해 얻어진 자식해를 행렬 B 라고 하고, 그 모양은 $B = E_1E_2 \dots E_k$ 와 같다고 가정한다. 그러면 B 는 정칙 이진 행렬이므로, 기저 변환을 하여 새로운 이진 인코딩으로 유전 알고리즘을 수행한다. 행렬 B 가 좋은 기저라고 판단되면 2.1절에서 제시되었던 기저 변환을 거친 후, 새로운 이진 인코딩으로 유전 알고리즘을 수행하여 기존의 문제 해결 방법보다 성능이 좋아졌는지 평가한다. 이를 각 단계로 나타내면 다음과 같다.

단계 1. 3.2절의 방법으로 초기 해집단을 생성한다.

단계 2. 두 부모해 P_1, P_2 를 선택한다.

단계 3. 2.2절의 교차연산으로 자식해 B 를 얻고, 3.1절의 변이 연산을 거친다.

단계 4. 정칙 이진 행렬인 B 로부터 기저 변환 B_T 를 구한다.

단계 5. B_T 에 의한 이진 인코딩으로 유전 알고리즘을 수행하여 기존의 기저보다 더 좋을 수 있는지 평가한다.

단계 6. 행렬 B 가 좋은 기저로 평가되었다면, 2.1절에서 제안하였던 아이디어로 기저 변환한다.

단계 7. 이진 인코딩으로 기존의 문제 해결 방법보다 성능이 좋아졌는지 평가한다.

4. 결 론

본 논문은 정칙 이진 행렬을 이용하여 유전 알고리즘의 성능을 개선할 수 있는 새로운 메타 유전 알고리즘을 설계했다.

정칙 이진 행렬을 통해 기저 변환한 이진 인코딩으로 유전 알고리즘의 성능을 검사하고, 유전 알고리즘의 성능이 개선될 수도 있다는 사실은 이전 연구 내용을 통해 보여졌다[1]. 논문은 기저 변환에 쓰이는 임의의 정칙 이진 행렬을 조사하는 방법을 중점적으로 연구했는데, 유전 알고리즘을 이용한 아이디어로 제시하였다.

기본 행렬을 이용한 해의 표현과 그에 따른 교차 연산은 이미 연구되었으나[2], 초기 해집단의 생성과 변이 연산, 그리고 평가 방법에 대한 아이디어는 본 논문에서 최초로 제안했다.

유전 알고리즘의 성능을 개선하는데 강력하게 사용될 정칙 이진 행렬을 찾는 메타 유전 알고리즘은 매우 중요하다. 향후 연구 과제는 본 논문에서 제시한 아이디어들을 토대로 실험하고 평가하여 그 결과를 분석하는 것이다.

5. 참고 문헌

- [1] Y.-H. Kim and Y. Yoon. Effect of Changing the Basis in Genetic Algorithms Using Binary Encoding, *KSII Transactions on Internet and Information Systems*, Vol.2, No.4, pp.184-193, August 2008.
- [2] Y.-H. Kim and Y. Yoon. Representation and Recombination over Nonsingular Binary Matrices, In *proceedings of the World Summit on Genetic and Evolutionary Computation*, pp.855-858, 2009.
- [3] J. P. Cohoon and W. Paris. Genetic placement. In *proceedings of the IEEE International Conference on Computer-Aided Design*, pp.422-425, 1986.
- [4] C. A. Anderson, K. F. Jones, and J. Ryan. A two-dimensional genetic algorithm for the Ising problem. *Complex Systems*, Vol.5, pp.327-333, 1991.
- [5] S. H. Friedberg, A. J. Insel, and L. E. Spence. *Linear Algebra*. thirde edition, Prentice Hall, 1997.
- [6] 문병로. *쉽게 배우는 유전 알고리즘 - 진화적 접근법*, 한빛미디어, 2008.