

비유사 중복 인코딩을 사용하는 유전 알고리즘을 위한 정규화 연산

(Non-Synonymously Redundant Encodings and Normalization in Genetic Algorithms)

최 성 순 [†] 문 병 로 ^{**}

(Sung-Soon Choi) (Byung-Ro Moon)

요 약 정규화는 교차 연산 전에 두 부모해 사이의 일관성을 유지하기 위하여 한 부모해를 다른 부모해에 맞추어 변환하는 연산이다. 본 논문은 비유사 중복 인코딩이 유전 알고리즘의 성능을 떨어뜨리는 이유와 정규화 연산이 비유사 중복 인코딩에 의해 유발되는 문제점들을 어떻게 완화하는지 설명한다. 이를 위해 완전 비유사 중복 인코딩을 정의하고, 완전 비유사 중복 인코딩에 의해 적합도와 거리의 상관성이 없는 탐색 공간이 만들어짐을 증명한다. 또한, 완전 비유사 중복 인코딩을 사용하는 다수의 문제들에 대한 실험을 바탕으로, 정규화를 통해 상관성이 없는 탐색 공간이 상관성이 있는 탐색 공간으로 변화되어 유전 알고리즘의 성능이 향상됨을 보인다.

키워드 : 유전 알고리즘, 혼합형 유전 알고리즘, 정규화, 탐색 공간 분석

Abstract Normalization transforms one parent genotype to be consistent with the other before crossover. In this paper, we explain how normalization alleviates the difficulties caused by non-synonymously redundant encodings in genetic algorithms. We define the encodings with maximally non-synonymous property and prove that the encodings induce uncorrelated search spaces. Extensive experiments for a number of problems show that normalization transforms the uncorrelated search spaces to correlated ones and leads to significant improvement in performance.

Key words : genetic algorithms, hybrid genetic algorithms, normalization, search space analysis

1. 서 론

유전 알고리즘에서, 주어진 해는 특정 인코딩 방식에 따라 스트링으로 표현되어 유전 연산자들에 의해 처리된다. 생물학의 용어를 빌어 스트링과 해는 각각 *유전자형(genotype)*과 *표현형(phenotype)*으로 불린다. 인코딩 방식은 유전자형 공간(genotype space)과 표현형 공간(phenotype space) 사이의 관계를 결정한다. 결정 방식에 따라 인코딩 방식은 세 부류로 나누어진다. 첫 번째 부류에 속하는 인코딩은 하나의 표현형을 하나의 유전자형으로 표현한다. 이 경우 유전자형 공간은 정보의 손실 없이 표현형 공간을 표현하는 가장 작은 공간들 중 하나이다. 주어진 문제에 대한 사전 지식이 없다면 이러

한 표현 방식이 좋은 선택일 수 있다[1,2].

두 번째 부류의 인코딩 방식은 다수의 표현형을 하나의 유전자형으로 표현한다. 이러한 인코딩은 문제와 관련된 중요한 파라미터만을 고려함으로써 탐색 공간의 크기를 줄인다. 이는 중요한 파라미터들을 유전 연산 과정에서 처리하고 덜 중요한 파라미터들은 지역 최적화 방법으로 미세 조정하는 혼합형 진화 알고리즘에 적합하다. 함수 근사 문제에서 회귀 엔진을 지역 최적화 휴리스틱으로 결합한 유전 프로그래밍을 예로 들 수 있다 [3,4]. 그러나, 실제 문제에서는 파라미터들 사이의 복잡한 의존 관계 때문에 상대적으로 중요한 파라미터를 선정하는 일조차 어려운 경우가 많다[5].

마지막 부류의 인코딩 방식은 하나의 표현형을 다수의 유전자형으로 표현한다. 이러한 인코딩을 *중복 인코딩(redundant encoding)*이라 부른다. 전형적인 인코딩 방식을 사용할 경우 문제 자체의 특성으로 인해 하나의 표현형을 하나의 유전자형만으로 표현하기 어려운 문제들이 많다. 분할 문제들과 구조 최적화 문제들을 예로

[†] 정 회 원 : 서울대학교 컴퓨터공학부
irranum@soar.snu.ac.kr

^{**} 정 회 원 : 서울대학교 컴퓨터공학부 교수
moon@soar.snu.ac.kr

논문접수 : 2005년 10월 13일

심사완료 : 2007년 4월 10일

들 수 있다. 분할 문제들은 주어진 아이템 집합을 서로 겹치지 않는 부분 집합들로 분할하는 것과 관련이 있다 [2]. 그래프 분할 문제[6-8], 그래프 색칠 문제, 상자 채우기 문제, 작업장 배치 문제[2] 등이 여기에 속한다. 구조 최적화 문제들은 주어진 조건을 만족하는 최적의 구조를 찾는 문제들이다. 그 문제들에서는 흔히 한 해와 구조적으로 동일한 다른 해들이 존재한다. 정렬 네트워크 최적화 문제[9], 신경망 최적화 문제[10], RNA 구조 추정 문제[11] 등을 예로 들 수 있다.

최근 들어, 진화 알고리즘의 중복 인코딩이 가지는 장단점에 대한 많은 연구들이 있었다. 특정 문제에 대하여 유전 알고리즘의 탐색을 더 효과적으로 만든 중복 인코딩의 사례가 몇몇 문헌에 제시되어 있다[12,13]. Ebner, Shackleton, Shipman은 진화의 중립 이론(neutral theory of evolution)[14]에서 영감을 얻어 중복 인코딩이 중립망을 야기한다는 사실에 주목하였다. 여기서, 중립망이란 같은 표현형으로 발현되며 일점 변이에 의하여 서로 연결되는 유전자형들의 집합을 의미한다[15-17]. 그들은 랜덤 부울 네트워크 매핑(random Boolean network mapping)과 같은 다소 복잡한 형태의 중복 인코딩을 설계하고 이러한 중복 인코딩이 적합도 값이 갑자기 변하는 동적 환경에서 유용함을 보였다. 그들은 중립망이 표현형들 사이의 연결성을 향상시켜 해집단의 다양성을 유지하는데 도움을 준다고 주장하였다.

Knowles와 Watson은 나름의 실험을 통해 랜덤 부울 네트워크 매핑이 몇몇의 최적화 문제에 대하여 변이 기반 진화 알고리즘의 성능을 저해한다는 결과를 얻었다[18]. 이러한 결과를 바탕으로 Knowles와 Watson은 인코딩의 중복성이 항상 효율적인 최적화 과정을 이끄는 것은 아니라고 주장하였다. Van Hoyweghen 등은 유전자형 공간에 어떤 대칭성이 있는 중복 인코딩은 동기화 문제(synchronization problem)를 야기한다는 실험 결과를 제시하였다[19-21]. 동기화 문제란, 서로 다른 최적해들을 구성하는 빌딩 블럭들이 조화롭게 결합하지 못하여 알고리즘이 지역 최적해에 교차되는 문제를 말한다. Weicker와 Weicker는 유전 알고리즘에서 중복 인코딩을 사용함에 따른 이해 득실을 기술하였다[22]. 이와 함께 그들이 고려한 문제에 대해서는 중복성이 성능 향상을 이끌지 못한다는 실험 결과를 제시하였다.

Rothlauf와 Goldberg는 중복 인코딩을 유사 중복 인코딩(synonymously redundant encoding)과 비유사 중복 인코딩(non-synonymously redundant encoding)으로 구분하였다[23]. 주어진 인코딩하에서 동일한 표현형으로 발현되는 유전자형들이 서로 유사한 경우 그 인코딩을 유사 중복 인코딩이라 부르고 그렇지 않은 경우 비유사 중복 인코딩이라 부른다. 그들은 단순화된 이론

적 모델을 바탕으로 유사 중복 인코딩은 유전 알고리즘의 성능을 해치지 않지만 비유사 중복 인코딩은 유전 알고리즘의 성능을 저해함을 입증하였다. 중복 인코딩이 교차 연산의 탐색 능력을 저해함을 시사하는 다수의 실험 결과가 있다. 이 문제를 완화하기 위한 접근법으로, 표현형의 관점에서 부모해들을 재결합하는 적응형 교차 연산자들이 제안되었다[6,24,25]. 이와는 달리, 부모해들 사이의 일관성을 유지하도록 하나의 부모해를 다른 부모해에 맞추어 변환한 뒤 전통적인 교차 연산자들을 사용하여 두 부모해를 재결합하는 정규화(normalization) 기법이 있다[7-9,26].

본 논문은 비유사 중복 인코딩이 일반적으로 사용되는 문제들에서 정규화 연산의 효용성에 대하여 살펴 본다. 구체적으로, 실험을 통해 관찰된 다음 두 가지 현상, 전통적인 교차 연산자들이 비유사 인코딩을 사용하는 유전 알고리즘의 성능을 떨어뜨리는 현상과 정규화 연산이 그러한 유전 알고리즘의 성능을 크게 개선하는 현상에 대하여 설명하고자 한다. 적합도 거리 분석(fitness distance analysis)은 정규화에 의한 탐색 공간의 변화를 설명하는 주요한 도구로 사용된다. 본 논문은 인코딩이 가지는 유사성 정도를 정량화하여 완전 비유사 성질을 가지는 중복 인코딩을 정의하고 그러한 인코딩에 의해 만들어진 탐색 공간은 적합도와 거리의 상관성을 갖지 않음을 증명한다. 또한, 다수의 문제들에 대한 실험을 바탕으로 비유사 인코딩이 만드는 탐색 공간이 정규화를 통해 상관성이 있는 탐색 공간으로 변화됨을 보인다. 이는 정규화에 의해 교차 연산자가 잘 동작하는 탐색 공간이 만들어짐을 의미한다.

본 논문의 구성은 다음과 같다. 2절에서 비유사 중복 인코딩이 유전 알고리즘의 탐색 능력, 특히, 전통적인 교차 연산자들의 탐색 능력을 저해함을 살펴본다. 비유사 중복 인코딩의 문제점을 완화하기 위한 기법의 하나로 정규화를 설명하고 정규화가 주어진 탐색 공간을 더 작은 크기의 공간으로 줄이는 역할을 함을 보인다. 3절에서 탐색 공간 지형 분석을 위한 통계적 측도들을 간략히 설명한다. 완전 비유사 인코딩을 정의하고 그들에 의해 만들어지는 탐색 공간 지형의 특징들을 고찰한다. 4절에서는 여러 문제들을 중심으로 실제 사용되는 중복 인코딩과 정규화 기법을 설명하고 정규화가 탐색 공간에 미치는 영향을 실험적으로 규명한다. 마지막으로, 5절에서 결론을 맺는다.

2. 중복 인코딩에서의 정규화

동일한 표현형을 공유하는 유전자형들이 서로 유사하지 않은 경우 관련 인코딩을 비유사 인코딩이라 불렀다. 이 절에서는 전통적인 교차 연산자들에 대하여 비유사

인코딩이 야기하는 문제점들을 논한다. 그 문제점들을 완화하는 기법의 하나로 정규화를 소개한다.

2.1 비유사 인코딩의 문제점

교차 연산이 가져야 할 두 가지 주요 특징으로 보존성(respectfulness)과 결합성(combination)을 들 수 있다. 보존성이란 부모해가 공통으로 가지는 형질이 자손에게 유전되는 특성을 의미한다. “평균 이상의 품질을 가지는 해들 사이에서 공통으로 존재하는 스키마는 평균 이상의 품질을 가진다” 라는 공통성 가설(commo-nality hypothesis)[27]은 보존성의 개념을 지지한다. 다수의 논문에서 교차 연산자 설계에서의 보존성의 중요성이 언급된 바 있다[1,27-29]. 결합성이란 부모해가 가지는 낮은 차수의 고품질 스키마들을 재결합하여 높은 차수의 고품질 스키마를 만드는 특성을 의미한다. 결합성의 개념은 빌딩 블록 가설(building block hypothesis)[30]에 기반하며 여러 문헌에서 반복해서 강조되고 있다 [1,28,31,32].

비유사 중복 인코딩은 하나의 최적 표현형을 유전자형 공간 속에서 서로 멀리 떨어진 다수의 유전자형들로 표현한다. 그로 인해 유전자형 공간은 다수의 봉우리를 가지는 다봉형(multimodal) 공간이 된다. 유전 알고리즘에 대한 다봉형 문제의 어려움은 여러 문헌에서 지적된 바 있다[30,33,34]. 마찬가지로, 비유사 인코딩하에서 고품질의 표현형 스키마¹⁾는 서로 다른 다수의 유전자형 스키마들로 표현된다. 한편, 다점 교차나 균등 교차와 같은 전통적인 교차 연산자들은 표현형과는 무관하게 유전자형 공간에서 작용한다. 비유사 중복 인코딩은 보존성과 결합성 두 측면 모두에서 이러한 교차 연산자들에게 해롭다.

- 보존성: 다점 교차나 균등 교차와 같은 전통적인 교차 연산자들은 유전자형 공간에서 보존성을 가진다. 그러나, 그들은 표현형 공간에서 보존성을 갖지 않는다. 만일 하나의 표현형 스키마가 두 부모해에서 서로 다른 유전자형 스키마로 표현되어 있다면 표현형 스키마는 보존되지 않을 수 있다. 표현형 스키마에 대응되는 유전자형 스키마가 많을수록 표현형 스키마가 자손에게 유전될 확률은 작아진다.
- 결합성: 상위(epistasis)의 정도가 높은 문제에서, 유전자형 내의 낮은 차수의 스키마는 다른 유전자들이 가지는 형질에 따라 즉 유전 문맥에 따라 유전자형 전체 품질에 긍정적 혹은 부정적 기여를 할 수 있다 [35]. 전통적인 교차 연산들은 유전 문맥과는 무관하게 부모해 내의 스키마들을 재결합한다. 교차 연산 과정에서 나타나는 문맥상의 차이는 고품질 스키마의

수명을 단축시킨다. 이러한 이유로, 병렬 배치(juxtaposition)를 통해 높은 차수의 고품질 스키마를 만드는 것은 어려워진다. 표현형 스키마에 대응되는 유전자형 스키마가 많을수록 그 스키마가 만들어질 확률은 작아진다.

따라서, 규모가 크고 상위 정도가 높은 문제들에 대해 전통적인 교차 연산을 사용하는 유전 알고리즘이 비유사 인코딩에 의해 확장된 유전자형 공간을 효율적으로 탐색하기는 어렵다. 정규화는 두 부모해의 유전 문맥이 교차 과정 중에 가능한 일치하도록 부모해 중 하나를 다른 하나에 맞추어 변환하는 것이다. 정규화는 전통적인 교차 연산자들이 보존성 및 결합성과 관련된 문제들을 극복하는데 도움을 준다. 다음 절에서 이에 대해 살펴보고자 한다.

2.2 정규화와 탐색 공간 변환

중복 인코딩은 유전자형 공간 X 와 표현형 공간 S 사이에 정의된 전사 함수 $\varphi: X \rightarrow S$ 이다. 여기서 $|X| \geq |S|$ 가 성립한다. 표현형 $s \in S$ 에 대하여 s 를 공유하는 유전자형들의 집합 $H(\varphi, s) \subseteq X$ 는 다음과 같다.

$$H(\varphi, s) = \{x \in X \mid \varphi(x) = s\}$$

이때 $H(\varphi, s)$ 를 s 에 대응되는 잉여 집합 (혹은 중립 집합)이라 부른다. 잉여 집합의 정의는 유전자형들 사이에 동치 관계를 주며 X 는 잉여 집합들에 의해 분할된다. 잉여 집합들의 모임을 \bar{X} 로 나타내면 \bar{X} 와 S 는 일대일 대응 관계에 있다.

앞서 언급했듯이, 비유사 인코딩의 문제들을 완화하기 위해서는 표현형 수준에서 부모해들을 재결합하는 것이 필요하다. 그러면, 부모해 $x, y \in X$ 가 주어지면, 대응되는 표현형 $\varphi(x)$ 와 $\varphi(y)$ 를 어떻게 재결합할 것인가? 정규화의 본질은 잉여 집합내의 유전자형들을 하나의 표현형으로 보는 것이다. 그러한 관점에 따라 정규화 연산자는 y 를 x 와 가장 밀접한 잉여 집합내의 원소로 대체한다. 즉 고품질 스키마들을 보존하고 결합하는데 가장 적합한 $y' \in H(\varphi, \varphi(y))$ 을 선택하여 x 와 y' 을 교차 연산자에 넘겨준다.

좀 더 엄밀하게, 유전자형 쌍에 정의된 일관성 측도 $m: X \times X \rightarrow \mathbb{R}$ 이 있다고 가정하자. (\mathbb{R} 은 실수 집합을 나타낸다.) 일관성 측도는 흔히 유전자형 공간상에 정의된 거리 측도에 반비례하게 정의된다. 4절의 예에서 보듯이 문제의 인코딩에 따라 거리 측도와 무관하게 정의될 수도 있다. 부모해 $x, y \in X$ 가 주어지면, 정규화 연산자는 y 를 $m(x, y')$ 값을 최대화하는 $y' \in H(\varphi, \varphi(y))$ 으로 변환한다. 그 뒤, 교차 연산자는 두 유전자형 x 와 y' 을 재결합한다. 유전자형들 사이의 거리는 정규화에 의해 재정의되고 그에 따라 탐색 공간 지형은 변화된다. \bar{X} 와

1) 암묵적으로, 표현형 스키마는 표현형을 이루는 부분 특성으로 정의한다.

5는 일대일 대응 관계에 있으므로, 교차 연산의 관점에서 유전자형 공간은 정규화에 의해 더 작은 크기의 표현형 공간으로 변환된다.

정규화에 의한 탐색 공간 변환은 탐색 공간 지형의 구조를 바꾼다. 이는 유전 알고리즘에 대한 문제의 난이도를 변화시켜 유전 알고리즘의 성능에 영향을 미친다. 난이도의 변화는 일관성 측도 m 과 밀접한 관련이 있으며 적합한 측도 m 을 선택하는 것은 정규화 연산자 설계에서 중요한 일이다. 정규화는 부모해들 사이의 유전자형 거리(예를 들면, 해밍 거리)를 줄이는 것 이상의 의미를 가진다.

3. 적합도 거리 분석

이 절에서 적합도 거리 분석에 사용되는 몇 가지 통계적 측도들을 간략히 설명한다. 비유사 중복 인코딩을 정의한 뒤 이러한 인코딩에 의해 만들어지는 지형에서 적합도와 거리는 선형적으로 상관성을 가지지 않음을 보인다.

3.1 상관성 측도들

주어진 알고리즘의 성능을 예측함에 있어 탐색 공간 지형의 구조를 이해하는 것은 중요하다. 굴곡의 정도는 탐색 공간 지형의 구조를 나타내는 가장 중요한 특징 가운데 하나이다. 지형의 굴곡이 심할수록 진화 알고리즘의 성능은 나빠진다는 것이 정설이다. 탐색 공간 지형의 굴곡의 정도는 적합도와 거리의 상관성과 큰 관련이 있다. 완만한 지형에서 적합도와 거리의 상관성은 높은 반면 굴곡이 심한 지형에서는 상관성이 낮다. 이러한 이유로, 굴곡의 정도를 추정하는 많은 측도들이 적합도와 거리의 상관성에 기반한다. 자기 상관(autocorrelation) 계수[36], 상관 길이(correlation length)[37], 적합도 거리 상관(fitness distance correlation, FDC) 계수[38], Boese 등에 의해 제안된 측도[39] 등을 예로 들 수 있다.

Jones와 Forrest에 의해 제안된 FDC 계수[38]는 진화 연산 분야에서 문제의 난이도를 예측하는 데 널리 사용되었다[40,41]. FDC 계수는 다음과 같이 정의된다.

$$\rho_{FDC} = Corr[f_{opt}, d_{opt}] = \frac{Cov[f_{opt}, d_{opt}]}{\sigma[f_{opt}] \cdot \sigma[d_{opt}]}$$

여기서, f_{opt} 와 d_{opt} 는 각각 랜덤하게 추출된 유전자형과 그로부터 가장 가까운 최적해 사이의 적합도 차이와 거리를 나타내는 확률 변수이다. 최소화 문제의 경우, $\rho_{FDC} = 1.0$ 값은 적합도와 거리가 완전한 상관성을 가져 일반적으로 진화적 탐색이 실패를 거들 것임을 의미한다. 영에 가까운 ρ_{FDC} 값은 적합도와 거리의 선형적인 상관성이 없음을 의미하며 진화적 탐색이 실패할 수 있음을 나타낸다. Boese 등은 또다른 상관성 기반 측도,

ρ_{avg} 를 제안하였다[39]. 그 정의는 다음과 같다.

$$\rho_{avg} = Corr[f_{opt}, d_{avg}] = \frac{Cov[f_{opt}, d_{avg}]}{\sigma[f_{opt}] \cdot \sigma[d_{avg}]}$$

여기서 f_{opt} 는 랜덤하게 추출된 유전자형과 최적해 사이의 적합도 차이를 나타내는 확률 변수이고 d_{avg} 는 추출된 유전자형과 다른 유전자형들 사이의 평균 거리를 나타내는 확률 변수이다. 그들은 이 측도를 몇 가지 조합 최적화 문제의 지역 최적해 공간에 적용하여 그 문제들의 지형이 전역적 볼록 구조(globally convex structure, 큰 계곡 구조(big valley)로도 알려져 있다)를 가짐을 보였다.

유전자형 x 에 대하여 다음과 같은 상관성 측도 $\rho(x)$ 를 고려하자.

$$\rho(x) = Corr[f_x, d_x] = \frac{Cov[f_x, d_x]}{\sigma[f_x] \cdot \sigma[d_x]}$$

여기서 f_x 와 d_x 는 각각 랜덤하게 추출된 유전자형과 x 사이의 적합도 차이와 거리를 나타내는 확률 변수이다. 측도 $\rho(x)$ 는 x 를 중심으로 적합도와 거리의 상관성 구조를 나타낸다. 만일 x 가 유전자형 공간내의 유일한 최적해라면 ρ_{FDC} 와 $\rho(x)$ 는 동일한 측도를 나타낸다. 그러나, 다봉형 탐색 공간의 경우, 가장 가까운 최적해들은 유전자형마다 다를 수 있다. 이로 인해 최적해 x 에 대한 $\rho(x)$ 값과 ρ_{FDC} 값은 보통 달라진다.

전형적인 진화적 탐색은 유전자형들의 해집단이 진화하여 하나의 최적(혹은 최적에 가까운) 유전자형으로 수렴하는 과정으로 볼 수 있다. 한편, 다봉형 탐색 공간의 경우 하나 이상의 최적 유전자형들이 ρ_{FDC} 값에 반영된다. 비유사 인코딩에서 하나의 최적 표현형은 서로 멀리 떨어진 다수의 유전자형으로 표현되므로 ρ_{FDC} 측도는 진화 알고리즘의 탐색 능력을 잘못 추정할 가능성이 있다. 이러한 이유로, 본 논문에서는 주어진 최적해 x 에 대한 측도 $\rho(x)$ 를 사용하여 비유사 인코딩에 의해 만들어지는 탐색 공간 지형을 분석한다.

3.2 완전 비유사 인코딩의 무상관성

중복 인코딩 $\varphi: X \rightarrow S$ 와 $X \times X$ 상에서 정의된 거리 측도 d 가 주어져 있다고 가정하자. 본 논문에서는 관계에 따라 d 를 해밍 거리 측도라고 가정한다. 유전자형들 사이의 거리함은 다음과 같이 분해될 수 있다.

$$\sum_{x,y \in X} d(x,y) = D(\varphi) + D(\varphi).$$

여기서, $D(\varphi) = \sum_{s \in S} \sum_{y \in H(\varphi,s)} d(x,y)$ 이고 $D(\varphi) = \sum_{s \neq t} \sum_{x \in S} \sum_{y \in H(\varphi,t)} d(x,y)$ 이다. $D(\varphi)$ 는 서로 같은 잉여 집합에 속하는 유전자형들 사이의 거리함, $D(\varphi)$

는 서로 다른 잉여 집합에 속하는 유전자형들 사이의 거리합을 나타낸다. $D(\varphi)$ 는 인코딩 φ 하에서 같은 잉여 집합내의 유전자형들이 유전자형 공간에서 얼마나 가깝게 위치해 있는지를 나타내므로, $D(\varphi)$ 값이 상대적으로 작으면 (고정된 G 에 대하여 $D(\varphi) + D'(\varphi)$ 값은 상수이므로 $D'(\varphi)$ 값이 상대적으로 크면) φ 를 유사 인코딩으로, 그렇지 않으면 φ 를 비유사 인코딩으로 간주할 수 있을 것이다. $D(\varphi)$ 와 본질적으로 동일한 측도가 참고 문헌 [23]에 언급된 바 있다.

우선 어떠한 인코딩이 완전 비유사 특성을 가지는지 고찰한다. 다음에서, $x[i]$ 는 유전자형 x 의 i 번째 유전자형질을 나타낸다.

명제 1. 알파벳 Σ 에 대하여 표현형 공간 S 가 유전자형 공간 Σ^n 에 의해 표현되며 표현형들의 잉여 집합의 크기가 미리 정해져 있다고 가정하자. 인코딩 φ 가 임의의 $s \in S$, $\{i \in \{0, 1, \dots, n-1\}\}$, $a, b \in \Sigma$ 에 대하여

$$|\{x \in H(\varphi, s) | x[i] = a\}| = |\{x \in H(\varphi, s) | x[i] = b\}| \quad (1)$$

를 만족하면, φ 는 L 값을 최대화한다.

증명. 우선, 다음이 성립함을 알 수 있다.

$$\begin{aligned} D(\varphi) &= \sum_{s \in S} \sum_{x, y \in H(\varphi, s)} d(x, y) \\ &= \sum_{s \in S} \sum_{x, y \in H(\varphi, s)} \sum_{i=0}^{n-1} 1(x[i] \neq y[i]) \\ &= \sum_{s \in S} \sum_{i=0}^{n-1} \sum_{x, y \in H(\varphi, s)} 1(x[i] \neq y[i]). \end{aligned}$$

여기서, $1(\cdot)$ 은 지표 함수를 의미한다. 이제, 고정된 $s \in S$ 와 i 에 대하여 $\sum_{x, y \in H(\varphi, s)} 1(x[i] \neq y[i])$ 의 값을 고려하자. 임의의 $a \in \Sigma$ 에 대하여 $t(\varphi, s, i, a) = |\{x \in H(\varphi, s) | x[i] = a\}|$ 라 하면,

$$\begin{aligned} &\sum_{x, y \in H(\varphi, s)} 1(x[i] \neq y[i]) \\ &= \sum_{x, y \in H(\varphi, s)} \sum_{a \neq b \in \Sigma} 1(x[i] = a \text{ and } y[i] = b) \\ &= \sum_{x, y \in H(\varphi, s)} \sum_{a \neq b \in \Sigma} 1(x[i] = a) \cdot 1(y[i] = b) \\ &= \sum_{a \in \Sigma} \sum_{x \in H(\varphi, s)} 1(x[i] = a) \sum_{b \in \Sigma, b \neq a} \sum_{y \in H(\varphi, s)} 1(y[i] = b) \\ &= \sum_{a \neq b \in \Sigma} t(\varphi, s, i, a) \cdot t(\varphi, s, i, b) \\ &= \sum_{a, b \in \Sigma} t(\varphi, s, i, a) \cdot t(\varphi, s, i, b) - \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 \\ &= \left(\sum_{a \in \Sigma} t(\varphi, s, i, a) \right)^2 - \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 \\ &= |H(\varphi, s)|^2 - \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 \end{aligned}$$

이 성립한다. 잉여 집합의 크기는 미리 정해져 있으므로

$|H(\varphi, s)|$ 값은 상수이고, 따라서 $\sum_{x, y \in H(\varphi, s)} 1(x[i] \neq y[i])$ 값이 최대가 될 필요 충분 조건은 $\sum_{a \in \Sigma} t(\varphi, s, i, a)^2$ 값이 최소가 되는 것이다. 코시-슈바르츠 부등식에 의해, $\sum_{a \in \Sigma} t(\varphi, s, i, a)^2$ 값은 모든 $a, b \in \Sigma$ 에 대하여 $t(\varphi, s, i, a) =$

$$t(\varphi, s, i, b) \text{ 일 때 최소값 } \frac{\left(\sum_{a \in \Sigma} t(\varphi, s, i, a) \right)^2}{|\Sigma|} = \frac{|H(\varphi, s)|^2}{|\Sigma|} \text{ 을}$$

가진다. 따라서, 명제가 증명된다. \square

명제 2. 알파벳 Σ 에 대하여 표현형 공간 S 가 유전자형 공간 Σ^n 으로 표현되고 표현형들의 잉여 집합의 크기가 $|\Sigma|$ 의 배수라고 가정하자. 인코딩 φ 가 L 값을 최대화하면, φ 는 조건 (1)을 만족한다.

증명. 우선 조건 (1)을 만족하는 임의의 인코딩은 조건을 만족하지 않는 어떠한 인코딩보다 더 큰 L 값을 가짐을 보인다. 그리고, 조건 (1)을 만족하는 인코딩이 존재함을 보임으로써 L 값을 최대화하는 임의의 인코딩이 조건 (1)을 만족함을 증명한다.

조건 (1)을 만족하는 임의의 인코딩 φ 와 조건을 만족하지 않는 임의의 인코딩 φ' 이 주어져 있다고 가정하자. 다음이 성립함을 알 수 있다.

$$\begin{aligned} D(\varphi) - D(\varphi') &= \sum_{s \in S} \sum_{i=0}^{n-1} \left(\sum_{x, y \in H(\varphi, s)} |H(\varphi, s)|^2 - \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 \right) \\ &\quad - \sum_{s \in S} \sum_{i=0}^{n-1} \left(\sum_{x, y \in H(\varphi', s)} |H(\varphi', s)|^2 - \sum_{a \in \Sigma} t(\varphi', s, i, a)^2 \right) \\ &= \sum_{s \in S} \sum_{i=0}^{n-1} \left(\sum_{a \in \Sigma} t(\varphi', s, i, a)^2 - \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 \right). \end{aligned}$$

모든 $s \in S$, $\{i \in \{0, 1, \dots, n-1\}\}$, $a, b \in \Sigma$ 에 대하여

$$t(\varphi, s, i, a) = t(\varphi, s, i, b) \text{ 이므로 } \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 = \frac{|H(\varphi, s)|^2}{|\Sigma|}$$

$$= \frac{|H(\varphi', s)|^2}{|\Sigma|} \text{ 이다. 코시-슈바르츠 부등식에 의해, 임의의}$$

$s \in S$, $\{i \in \{0, 1, \dots, n-1\}\}$ 에 대하여, $\sum_{a \in \Sigma} t(\varphi', s, i, a)^2 \geq$

$$\frac{|H(\varphi', s)|^2}{|\Sigma|} = \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 \text{ 이 성립한다. 인코딩 } \varphi' \text{은 조}$$

건 (1)을 만족하지 않으므로 $t(\varphi', s, i, a) \neq t(\varphi, s, i, b)$ 인 어떤 $s \in S$, $\{i \in \{0, 1, \dots, n-1\}\}$, $a, b \in \Sigma$ 가 존재한다. 따라서,

$$\text{코시-슈바르츠 부등식에 의해, } \sum_{a \in \Sigma} t(\varphi', s, i, a)^2 > \frac{|H(\varphi', s)|^2}{|\Sigma|}$$

$$= \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 \text{ 인 } s \in S, i \in \{0, 1, \dots, n-1\} \text{가 존재함을 알}$$

수 있다. 임의의 $s \in S$, $\{i \in \{0, 1, \dots, n-1\}\}$ 에 대하여,

$$\sum_{a \in \Sigma} t(\varphi', s, i, a)^2 \geq \sum_{a \in \Sigma} t(\varphi, s, i, a)^2 \text{ 이고 } \sum_{a \in \Sigma} t(\varphi', s, i, a)^2 >$$

$\sum_{a \in \Sigma} t(\varphi, s, i, a)^2$ 인 $s \in S$, $\{i \in \{0, 1, \dots, n-1\}\}$ 가 존재하므로, $D(\varphi) - D(\varphi') > 0$ 가 성립한다.

이제, 조건 (1)을 만족하는 인코딩이 존재함을 보인다. π 를 모든 $1 \leq i \leq |\Sigma| - 1$ 와 $a \in \Sigma$ 에 대하여 $\pi^i(a) \neq a$ 를 만족하는 Σ 상의 순열이라 하자. 유전자형 공간 Σ^n 을 순열 π 에 의해 생성되는 궤도(orbit)들로 분할하자. 각각의 $s \in S$ 에 대하여 $|H(\varphi, s)|$ 값은 $|\Sigma|$ 의 배수이고 궤도 각각의 크기는 $|\Sigma|$ 이므로, 각 궤도내의 유전자형들이 같은 잉여 집합에 속하는 인코딩 φ 가 존재한다. 인코딩 φ 는 조건 (1)을 만족하므로 증명은 완결된다. \square

위의 두 명제로부터, 조건 (1)을 만족하는 인코딩을 **완전 비유사 인코딩**이라 한다. 완전 비유사 인코딩에서 고정된 잉여 집합으로부터 하나의 유전자형을 균일한 분포로 뽑으면 각 유전자가 가지는 형질 역시 알파벳 상에서 균일한 분포를 따른다. 예를 들면, 하나의 유전자형과 그것의 비트별 반전이 같은 표현형을 공유하는 인코딩 $\varphi: \{0, 1\}^n \rightarrow S$ 는 완전 비유사 인코딩이다. 완전 비유사 인코딩은 많은 진화 알고리즘에서 사용되어 왔다. 4절에서 몇 가지 예를 살펴볼 것이다.

이제 완전 비유사 인코딩에 의해 만들어지는 탐색 공간 지형의 성질들을 고찰해 보자. 유전자형 공간의 두 부분 집합 A 와 B 에 대하여 A 와 B 로부터 균일한 분포로 뽑은 유전자형들 사이의 평균 거리를 $\bar{d}(A, B)$ 로 표시하자.

명제 3. 알파벳 Σ 에 대하여 중복 인코딩 $\varphi: \Sigma^n \rightarrow S$ 가 완전 비유사 인코딩이라 가정하자. 그러면, 모든 $x \in \Sigma^n$ 와 $s \in S$ 에 대하여 $\bar{d}(x, H(\varphi, s)) = \left(\frac{|\Sigma| - 1}{|\Sigma|}\right)n$ 이 성립한다.

증명. 다음이 성립함을 알 수 있다.

$$\begin{aligned} \bar{d}(x, H(\varphi, s)) &= \frac{1}{|H(\varphi, s)|} \sum_{y \in H(\varphi, s)} d(x, y) \\ &= \frac{1}{|H(\varphi, s)|} \sum_{y \in H(\varphi, s)} \sum_{i=0}^{n-1} 1(x[i] \neq y[i]) \\ &= \frac{1}{|H(\varphi, s)|} \sum_{i=0}^{n-1} \sum_{y \in H(\varphi, s)} 1(x[i] \neq y[i]). \end{aligned}$$

φ 는 완전 비유사 인코딩이므로, 조건 (1)로부터

$$\sum_{y \in H(\varphi, s)} 1(x[i] \neq y[i]) = \left(\frac{|\Sigma| - 1}{|\Sigma|}\right)|H(\varphi, s)|$$

이고 따라서

$$\bar{d}(x, H(\varphi, s)) = \frac{1}{|H(\varphi, s)|} \sum_{i=0}^{n-1} \left(\frac{|\Sigma| - 1}{|\Sigma|}\right)|H(\varphi, s)| = \left(\frac{|\Sigma| - 1}{|\Sigma|}\right)n$$

이 성립한다. \square

명제 3은 임의의 유전자형으로부터 임의의 잉여 집합내의 유전자형들까지의 평균 거리가 일정함을 의미한다.

유사한 방식으로, 다음 결과를 얻는다.

명제 4. 알파벳 Σ 에 대하여 중복 인코딩 $\varphi: \Sigma^n \rightarrow S$ 가 완전 비유사 인코딩이라 가정하자. 그러면, 모든 $s, t \in S$ 에 대하여 $\bar{d}(H(\varphi, s), H(\varphi, t)) = \left(\frac{|\Sigma| - 1}{|\Sigma|}\right)n$ 이 성립한다.

완전 비유사 인코딩은 임의의 두 잉여 집합 사이의 평균 거리를 균일하게 만든다. 이로 인해 탐색 공간 지형은 적합도와는 무관한 대칭 구조를 가지게 된다. 알파벳의 크기가 클수록 임의의 두 잉여 집합으로부터 균일한 분포로 추출된 유전자형 쌍들은 유전 형질을 거의 공유하지 않게 된다. 이는 보존성이 적절히 유지될 수 없음을 의미한다.

각각의 잉여 집합내의 유전자형들이 같은 확률로 추출되는 경우, 명제 3은 완전 비유사 인코딩에 의해 만들어진 탐색 공간 지형에서 적합도와 거리가 선형적으로 상관성을 가지지 않음을 함축한다.

명제 5. 알파벳 Σ 에 대하여 중복 인코딩 $\varphi: \Sigma^n \rightarrow S$ 가 완전 비유사 인코딩이라 가정하자. 잉여 집합내의 유전자형들이 같은 확률로 추출된다면, $\rho_{avg} = 0$ 이고 임의의 유전자형 x 에 대하여 $\rho(x) = 0$ 이다.

증명. 임의의 유전자형 x 에 대하여 $\rho(x) = 0$ 임을 증명하기 위해 $Cov[f_x, d_x] = E[f_x d_x] - E[f_x]E[d_x] = 0$ 이 성립함을 보이면 된다. 임의의 $y \in \Sigma^n$ 에 대하여 x 와 y 사이의 적합도 차이를 $f(x, y)$ 로 표시하자. 그러면,

$$\begin{aligned} E[f_x d_x] &= \sum_{s \in S} \Pr[y \in H(\varphi, s)] E[f(x, y) d(x, y) | y \in H(\varphi, s)] \\ &= \sum_{s \in S} \Pr[y \in H(\varphi, s)] f(x, y) E[d(x, y) | y \in H(\varphi, s)] \end{aligned}$$

이다. 명제 3으로부터, 모든 $s \in S$ 에 대하여 $E[d(x, y) | y \in H(\varphi, s)] = \bar{d}(x, H(\varphi, s))$ 값은 불변이므로, 모든 $s \in S$ 에 대하여

$$\begin{aligned} E[d_x] &= \sum_{s \in S} \Pr[y \in H(\varphi, s)] E[d(x, y) | y \in H(\varphi, s)] \\ &= E[d(x, y) | y \in H(\varphi, s)] \end{aligned}$$

이 성립한다. 따라서,

$$E[f_x d_x] = \sum_{s \in S} \Pr[y \in H(\varphi, s)] f(x, y) E[d_x] = E[f_x] E[d_x]$$

이다. 유사한 방식으로 $\rho_{avg} = 0$ 임을 보일 수 있다. 이에 대한 자세한 증명은 생략한다. \square

명제 5의 증명은 표현형에 대한 확률 분포와 무관함에 주목하자. 따라서, 전체 유전자형 공간 뿐만 아니라 지역 최적해 공간에도 이 명제를 적용할 수 있다.

완전 비유사 인코딩에 의해 만들어지는 탐색 공간 지형은 전역적 블록 구조를 가지지 않는다. 또한 어떠한 최적해에 대해서도 적합도와 거리의 선형적 상관성은 존재하지 않는다. 따라서, 완전 비유사 인코딩을 사용하

는 진화 알고리즘이 우수한 성능을 낼 것이라 기대하기 어렵다. 4절에서 정규화가 탐색 공간 지형의 구조를 변화시켜 최적해 x 에 대한 $\rho(x)$ 의 값을 크게 증가시킴을 확인할 것이다. 정규화에 의한 성능 향상 역시 실험적으로 검증될 것이다.

4. 정규화의 예들

이 절에서는 몇 가지 문제들에서 사용되는 정규화 기법들을 살펴본다. 또한, 정규화가 탐색 공간에 미치는 영향들을 살펴보고 정규화 전후의 탐색 공간 지형들을 분석한다. 유전 알고리즘 성능에 대한 실험 결과 역시 살펴본다. 이 절에서 살펴볼 문제들은 그래프 분할 문제, 계층적 동치 문제, 클러스터 동기화 문제, 순회 판매원 문제, 정렬 네트워크 문제이다.

4.1 그래프 분할 문제

그래프 $G=(V,E)$ 가 주어질 때(V 는 정점들의 집합을, E 는 간선들의 집합을 나타낸다), k -분할이란 정점 집합 V 를 k 개의 서로 겹치지 않는 부분 집합들로 나누는 것을 말한다. k -분할에서 두 개의 끝점이 서로 다른 부분 집합에 속하는 간선의 총 개수를 *컷사이즈(cut size)*라 부른다. 가장 큰 부분 집합과 가장 작은 부분 집합의 크기의 차가 1을 넘지 않을 때 k -분할은 *균형 잡혀 있다고* 말한다. k -분할 문제는 최소의 컷사이즈를 갖는 균형 잡힌 k -분할을 찾는 것이다. 이 문제는 NP-hard임이 알려져 있다[42].

k -분할 문제에서는 일반적으로 k 개의 부분 집합을 0부터 $k-1$ 까지의 정수로 표현하는 k -진수 인코딩이 사용되어 왔다[6-8,24,43]. 각 부분 집합에 어떤 정수를 할당하는지에 따라 하나의 표현형(k -분할)은 k 개의 서로 다른 유전자형들로 표현된다. 임의의 i 에 대하여 i 번째 유전자가 특정 형질을 가지는 유전자형의 개수는 $(k-1)!$ 이다. 따라서, 명제 1로부터 주어진 k -진수 인코딩은 완전 비유사 인코딩이다.

참고문헌 [8]에서 k -분할 문제를 위한 정규화 기법이

사용되었다. 이는 참고문헌 [6]과 [24]에서 제안된 적응형 교차 연산에 기반하고 있다. 이 기법은 우선 부모해에서 가장 많은 정점들을 공통으로 가지는 부분 집합을 하나씩 선택하고 그 부분 집합들에 공통의 숫자를 할당한다. 이 과정을 남아 있는 부분 집합들과 숫자들에 대하여 반복한다. 이 기법은 $O(|V|+k^2 \log k)$ 의 계산 복잡도를 가지도록 구현될 수 있다. k 를 상수라 하면, 이는 정점의 개수에 대하여 선형이다. 그림 1은 정규화의 한 예를 보여 준다. 부모해 2는 부모해 1과 가능한 한 일치되도록 변환되었다. 그림 2는 정규화를 사용한 경우와 그렇지 않은 경우의 교차 연산 결과를 보여 준다. 여기서, 그림 1의 부모해가 사용되었다. 정규화를 사용하지 않은 경우 교란의 정도가 크다는 사실을 확인할 수 있다. 부모해의 좋은 형질들이 자손에게 거의 유전되지 못했다.

표 1은 32-분할 문제에 대하여 정규화를 사용하지 않은 경우(Ordinary)와 사용한 경우(Normalization)의 탐색 공간 비교 실험 결과를 보여 준다. 실험을 위해 Johnson의 벤치마크 인스턴트들을 사용하였다[7]. 표에서 $E[d]$ 는 랜덤하게 생성된 10,000개의 지역 최적해들 사이의 평균 거리를 나타낸다. $\hat{\rho}(x)$ 는 생성된 지역 최적해들 가운데 가장 품질이 좋은 해 x 에 대한 $\rho(x)$ 값의 추정치를 나타낸다. 지역 최적해를 얻기 위하여 참고문헌 [43]에서 제안된 휴리스틱이 사용되었다.

지역 최적해 사이의 평균 거리는 정규화를 통해 크게 감소되었다. k -진수 인코딩은 완전 비유사 인코딩이므로, 명제 4는 정규화 전의 평균 거리가 $\left(\frac{k-1}{k}\right)^n$ 값으로 추정될 수 있음을 말해 준다. 표를 통해 이를 확인할 수 있다. 32-분할 문제에서 정규화 전의 평균 거리는 $\frac{31}{32}^n$ 으로 염색체 길이에 가깝다. 이는 지역 최적해들을 표현하는 유전자형들 사이에 공통 스키마가 드물 것을 말해 준다. 명제 4는 또한 정규화를 사용하지 않은 경우 지역 최적해들 사이의 평균 거리가 랜덤하게 추출된 유

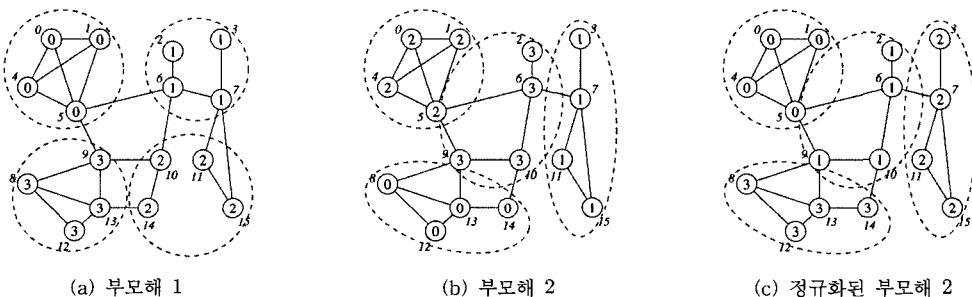


그림 1 4-분할 문제에서의 정규화의 예

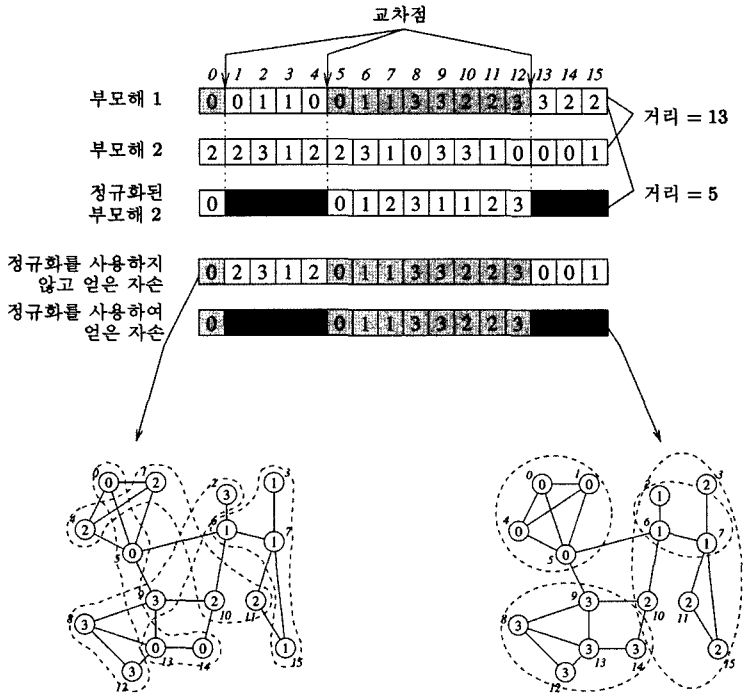


그림 2 4-분할 문제에서의 정규화의 예

표 1 32-분할 문제에서의 탐색 공간 비교 실험

Graph	Ordinary		Normalization	
	$E[d]$	$\hat{\rho}(x)$	$E[d]$	$\hat{\rho}(x)$
G500.005	483.59	0.019	300.61	0.621
G500.04	484.36	0.013	358.82	0.412
G1000.0025	968.15	0.009	723.75	0.472
G1000.02	968.78	0.030	813.99	0.303
U500.05	484.00	0.005	170.76	0.602
U500.40	484.04	0.016	173.56	0.338
U1000.05	968.47	0.021	525.48	0.610
U1000.40	969.04	0.006	225.09	0.672

전자형들 사이의 평균 거리와 같음을 말해 준다. 그러나, 정규화를 사용한 경우 지역 최적해들 사이의 평균 거리는 랜덤하게 추출된 유전자형들 사이의 평균 거리보다 훨씬 작다.²⁾ 이는 정규화를 통해 지역 최적해들이 서로 더 가까워짐을 의미한다. 명제 5는 정규화를 사용하지 않은 경우 $\hat{\rho}(x)$ 값이 0에 가까울 것임을 암시하는데 실제 표에서 확인할 수 있다. 또한, 정규화를 통해 $\hat{\rho}(x)$ 값이 크게 증가함을 알 수 있다. 이는 정규화가 탐색 공간 지형을 유전 알고리즘의 탐색에 유리하도록 변

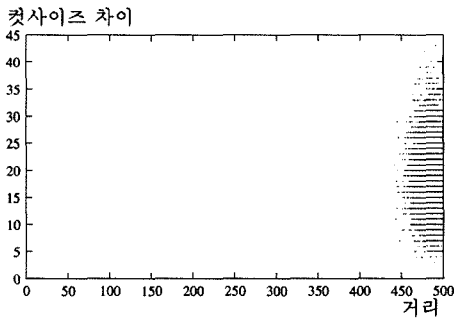
화시킴을 의미한다.

표 2는 32-분할 문제에 대하여 정규화를 사용하지 않은 경우(Ordinary)와 사용한 경우(Normalization)의 성능 비교 실험 결과를 보여 준다. 표에서 Best, Avg., σ/\sqrt{n} , CPU는 참고문헌 [43]의 지역 최적화 휴리스틱과 결합된 혼합형 유전 알고리즘의 100회 수행에 대한 최소 컷사이즈, 평균 컷사이즈, 표본 표준 편차, 평균 수행 시간을 나타낸다. 이 유전 알고리즘은 비례 선택과 오점 교차(5-point crossover)를 사용한다. 해집단의 크기는 50이고 2,500 세대 후에 정지한다. 표를 통해 알 수 있듯이, 정규화는 큰 성능 향상을 이끌었다. 정규화를 사용한 유전 알고리즘이 그렇지 않은 유전 알고리즘에 비해 수행 시간이 짧다는 것은 주목할 만한 사실이

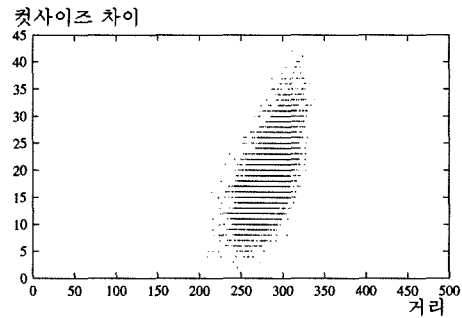
2) 32-분할 문제에서 100,000개의 랜덤 유전자형들 사이의 평균 거리는 500개의 정점을 가지는 인스턴스들의 경우 432.50, 1,000개의 정점을 가지는 인스턴스들의 경우 899.54였다.

표 2 32-분할 문제에서의 성능 비교 실험 (CPU: seconds on Intel Pentium IV 1.5 GHz)

Graph	Ordinary				Normalization			
	Best	Avg.	σ/\sqrt{n}	CPU	Best	Avg.	σ/\sqrt{n}	CPU
G500.005	183	185.37	0.11	168	178	181.56	0.09	160
G500.04	4052	4055.76	0.19	1093	4045	4050.58	0.19	1056
G1000.0025	333	340.40	0.23	573	322	327.95	0.16	497
G1000.02	7847	7861.16	0.43	2994	7836	7850.04	0.43	3015
U500.05	114	120.45	0.16	313	114	116.99	0.11	256
U500.40	5356	5384.85	0.70	2459	5357	5376.94	0.68	1858
U1000.05	149	158.03	0.39	1055	128	135.86	0.26	623
U1000.40	7395	7420.83	0.95	5301	7399	7417.87	0.76	4491



(a) 정규화를 사용하지 않은 경우(Ordinary)



(b) 정규화를 사용한 경우(Normalization)

그림 3 32-분할 문제에서의 정규화 효과 (G500.005)

다. 이에 대해서는 4.4절에서 다시 언급하겠다.

그림 3은 32-분할 문제의 한 인스턴스 G500.005의 탐색 공간 지형의 변화를 보여준다. 수평축은 10,000개의 지역 최적해 중 가장 좋은 해로부터의 거리를 나타낸다. 정규화를 사용하지 않은 경우, 해의 품질과 거리 사이에 상관성은 보이지 않는다. 정규화를 사용한 경우, 강한 상관성이 나타남을 알 수 있다.

4.2 계층적 동치 문제

계층적 동치 문제(Hierarchical-if-and-only-if problem, H-IFF 문제)는 빌딩 블럭들 사이의 상호 의존성을 모델링할 목적으로 Watson 등에 의하여 소개되었다[44]. 차수 $n = 2^k$ (k 는 양의 정수)에 대하여 H-IFF 함수 $f_n: \{0,1\}^n \rightarrow \mathbb{R}$ 은 재귀적으로 정의된다.

$$f_n(x_1, \dots, x_n) = \begin{cases} 1 & n=1 \text{인 경우} \\ n + f_{\frac{n}{2}}(x_1, \dots, x_{\frac{n}{2}}) + f_{\frac{n}{2}}(x_{\frac{n}{2}+1}, \dots, x_n) & n > 1 \text{이고 } x_i = x_j \forall i, j \text{인 경우} \\ f_{\frac{n}{2}}(x_1, \dots, x_{\frac{n}{2}}) + f_{\frac{n}{2}}(x_{\frac{n}{2}+1}, \dots, x_n) & \text{그밖의 경우} \end{cases}$$

함수 f_n 은 계층 구조를 이루는 빌딩 블럭들로 구성되며 더 큰 함수 값을 얻기 위해서는 서로 다른 빌딩 블럭들이 결합되어야 한다. H-IFF 문제는 f_n 값을 최대로 하는 $x \in \{0,1\}^n$ 를 찾는 것이다.

H-IFF 함수 f_n 은 두 개의 최적해 $x^* = (0, \dots, 0)$ 와 $\bar{x}^* = (1, \dots, 1)$ 를 가지며 이 때의 함수값은 $(\lg n + 1)n$ 이다. 유전자형 x 의 비트별 반전을 \bar{x} 라 하면, 모든 $x \in \{0,1\}^n$ 에 대해서 $f_n(x) = f_n(\bar{x})$ 가 성립한다. 이러한 성질을 스핀 반전 대칭성(spin-flip symmetry)이라 부른다[21]. 해집단내의 다양성이 유지되는 경우, 교차와 변이를 사용하는 유전 알고리즘은 H-IFF 문제를 푸는데 적합하다[44]. 그렇지 않은 경우, 스핀 대칭 다양성은 유전 알고리즘의 동기화 문제를 야기한다고 알려져 있다[21]. 진화 알고리즘 분야에서 H-IFF 문제에 대한 많은 연구들이 있었다[17,18,32,45].

H-IFF 문제에 대하여 이진 인코딩은 자연스럽다. 비트별 반전 관계에 있는 한 쌍의 유전자형이 하나의 표현형에 대응된다고 가정하면 이진 인코딩은 완전 비유사 인코딩이다. 명제 4로부터 임의의 두 잉여 집합 사이의 평균 거리는 $\frac{n}{2}$ 이다. 명제 5는 $\rho(x^*) = \rho(\bar{x}^*) = 0$ 임을 말해준다. 다음과 같은 간단한 정규화 기법을 고려하자. 부모해 x 와 y 가 주어지면, 둘 사이의 해밍 거리가 $\frac{n}{2}$ 보다 큰 경우에 한해서 y 를 비트별 반전시킨다 (이것은 결국 $k=2$ 인 k -분할 문제를 위한 정규화 기법과 같다). 그 후, x 와 정규화된 y 를 교차 과정에서 재결합한다.

정규화의 효과를 살펴보기 위하여 우선 랜덤하게 추출된 10,000개의 유전자형들로부터 $\rho(x^*)$ 값을 계산하였다. 정규화를 사용하지 않은 경우, $n=16, 32, 64$ 일 때의 값은 각각 $-0.001, 0.000, -0.001$ 이었다. 정규화를 사용한 경우, 그 값은 각각 $0.370, 0.270, 0.187$ 이었으며 이들은 정규화를 사용하지 않은 경우의 값들에 비해 상당히 크다(정규화를 사용하는 경우에도 인코딩의 스핀 반전 대칭성에 의해 $\rho(x^*)$ 와 $\rho(\overline{x^*})$ 의 값이 같음을 주목하자). 그림 4는 $n=32$ 인 H-IFF 함수 f_n 에 대하여 x^* 를 기준으로 한 탐색 공간 지형의 변화를 보여준다. 정규화를 사용한 경우, 적합도와 거리는 분명한 상관성을 가진다.

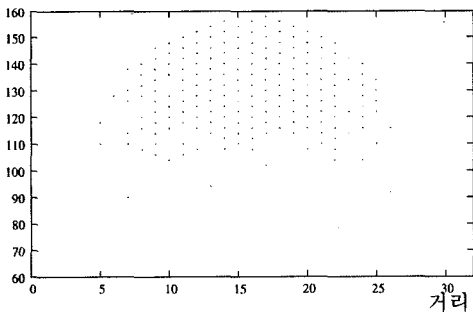
다음으로, 이진 토너먼트 선택, 일점 교차(1-point crossover), 확률 $\frac{1}{n}$ 의 비트별 변이 연산을 가지는 안정 상태 유전 알고리즘을 사용하여 실험을 수행하였다. 유전 알고리즘이 100,000 세대 안에 최적해를 찾으면 유전 알고리즘은 H-IFF 문제를 푸는 것이다. 정규화를 사용한 유전 알고리즘의 경우 각 세대마다 0.8의 확률로 교차 연산 전에 정규화를 수행하였다. 그림 5는 H-IFF 함수 f_{32} 와 f_{64} 에 대하여 100번의 독립 시행 중 성공 횟

수를 보여 준다. 유전 알고리즘의 성능은 해집단의 크기에 크게 의존함을 알 수 있다. 이는 해집단의 다양성이 유지되지 못할 때 인코딩의 스핀 반전 대칭성이 동기화 문제를 유발한다는 참고문헌 [21]의 주장을 뒷받침한다. 전체적으로, 정규화를 사용한 유전 알고리즘이 그렇지 않은 유전 알고리즘보다 크게 우수하였다. 다양성이 적절히 유지되면(해집단의 크기가 500인 경우), 정규화를 사용한 유전 알고리즘은 f_{32} 와 f_{64} 에 대해 100번의 수행 모두에서 문제를 풀었다.

4.3 클러스터 동기화 문제

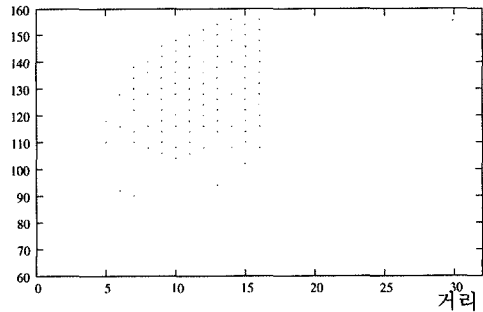
클러스터 동기화 문제(Connected clusters problem)는 빌딩 블록들 사이의 동기화를 보다 명시적으로 고찰할 목적으로 참고문헌 [21]에서 소개되었다. 문제의 인스턴스는 동기화 조건(synchronization constraint)이라 불리는 다수의 간선들로 연결되는 두 개의 같은 크기의 완전 그래프(클러스터)로 구성된다. 동기화 계수(synchronization tightness) τ 는 서로 다른 클러스터에 속하는 두 정점 사이에 동기화 조건(간선)이 존재할 확률을 의미한다. 이 값은 전체 동기화 조건의 개수를 결정한다. 색상 집합 $\{0, \dots, k-1\}$ 이 주어지면, 문제는 간선으로

적합도 차이



(a) 정규화를 사용하지 않은 경우 (Ordinary)

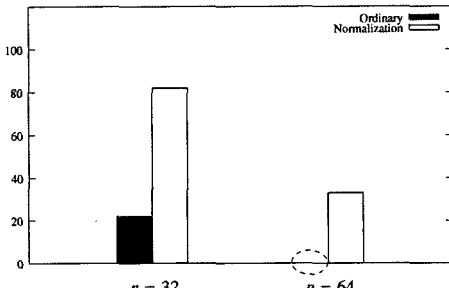
적합도 차이



(b) 정규화를 사용한 경우 (Normalization)

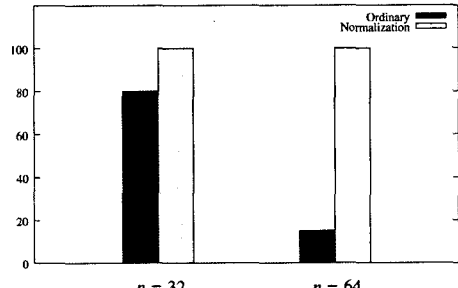
그림 4 H-IFF 문제에서의 정규화 효과 ($n=32$)

성공 횟수



(a) 해집단 크기 100

성공 횟수



(b) 해집단 크기 500

그림 5 H-IFF 문제에서의 비교 실험

연결된 정점들을 최대한 같은 색으로 칠하는 것이다. 주어진 색칠의 적합도를 구하기 위해 다른 색으로 칠해진 간선마다 동등하게 페널티를 준다.

일반적으로 색상의 개수 k 가 증가함에 따라 문제는 어려워진다. 문제의 난이도는 동기화 계수에도 의존한다. 동기화 계수가 0 또는 1이면, 문제는 쉽게 풀린다. 동기화 계수가 0에 가까우면 즉 소수의 동기화 조건만이 두 클러스터 사이에 존재하면, 문제는 풀기 어려운 것으로 알려져 있다[21].

색상의 순서를 임의로 바꾼다 하더라도 적합도는 변하지 않는다. 이러한 의미에서 문제는 대칭성을 가진다. 이 문제를 위해 흔히 k -진수 인코딩이 사용되는데, 이는 완전 비유사 인코딩이다. 잉여 집합의 크기는 표현형에 따라 다를 수 있음에 주목하자. 예를 들어, 정점들을 모두 같은 색으로 칠하는 표현형의 잉여 집합의 크기는 k 인 반면, k 개의 서로 다른 색을 사용하는 표현형의 잉여 집합의 크기는 n 이다. 이러한 인코딩을 비균등 중복 인코딩(non-uniformly redundant encoding)이라 부른다[46]. 명제 4로부터, 임의의 두 잉여 집합 사이의 평균 거리는 $\left(\frac{k-1}{k}\right)n$ 이다. 명제 5는 최적의 표현형에 대응되는 임의의 유전자형 x^* 에 대하여 $\rho(x^*)=0$ 임을 말해 준다. 주어진 문제에 대해 그래프 분할 문제에서와 같은 정규화 기법을 사용한다.

정규화가 미치는 영향을 살펴보기 위하여 64개의 정점과 8개의 색상을 가지는 (즉, $n=64$ 이고 $k=8$) 클러스터 동기화 문제를 사용하였다. 해집단의 크기가 500이고, 이전 토너먼트 선택, 일점 교차, 확률 $\frac{1}{n}$ 의 비트별 변이를 사용하는 안정 상태 유전 알고리즘으로 실험하였다. 유전 알고리즘은 최대 100,000 세대까지 진행되며, 0.1의 간격으로 0.1에서 0.9까지의 동기화 계수 τ 값이 실험에 사용되었다. 정규화를 사용한 유전 알고리즘의 경

성공 횟수

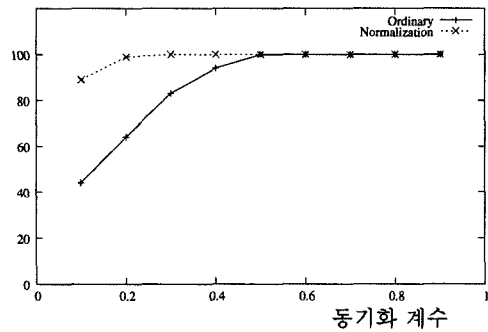


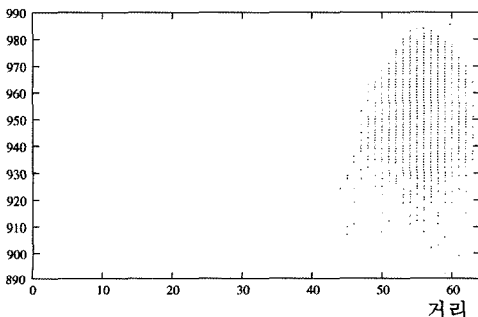
그림 6 클러스터 동기화 문제에서의 실험 결과

우, 각 세대마다 0.8의 확률로 교차 연산 전에 정규화를 수행하였다. 그림 6은 각각의 동기화 계수값에 대하여 100회의 독립 수행 중 성공 횟수를 보여준다. 전체적으로, 정규화를 사용한 유전 알고리즘이 그렇지 않은 유전 알고리즘보다 우수하였다. 특히, 문제가 가장 어려울 때($\tau=0.1$), 성능의 차이는 뚜렷하였다. $\tau=0.1$ 과 $x^*=(0, \dots, 0)$ 에 대하여, 랜덤 추출된 10,000개의 유전자형으로부터 추정된 $\rho(x^*)$ 값은 정규화를 통해 0.039에서 0.609로 증가하였다. (모든 정점을 같은 색으로 칠하는 표현형에 대응하는 임의의 유전자형 x 에 대하여 $\rho(x)$ 의 값은 $\rho(x^*)$ 값과 같다. 이는 인코딩의 대칭성 때문이다.) 그림 7은 $\tau=0.1$ 에 대하여 x^* 를 중심으로 한 탐색 공간 지형의 변화를 보여준다. 정규화를 사용한 경우, 적합도와 거리 사이에 강한 상관성이 나타남을 알 수 있다.

4.4 순회 판매원 문제

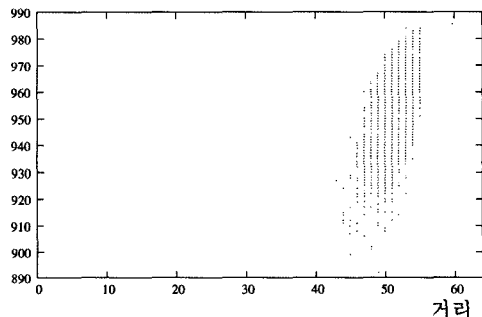
간선들에 가중치가 부여된 완전 그래프 $G=(V,E)$ 가 주어지면, 순회 판매원 문제(traveling salesman problem, TSP)는 최소 가중치를 가지는 해밀토니안 투어를 찾는 것이다. 이 문제는 NP-hard로 알려져 있다

적합도 차이



(a) 정규화를 사용하지 않은 경우 (Ordinary)

적합도 차이



(b) 정규화를 사용한 경우 (Normalization)

그림 7 클러스터 동기화 문제에서의 정규화 효과 ($\tau=0.1$)

표 3 순회 판매원 문제에서의 탐색 공간 비교 실험

Instance	Ordinary		Normalization	
	$E[d]$	$\hat{\rho}(x)$	$E[d]$	$\hat{\rho}(x)$
att532	530.37	0.024	457.11	0.552
rat783	781.25	0.034	737.03	0.370
dsj1000	998.60	0.035	933.27	0.351
pcb3038	3036.12	0.038	2954.20	0.251

표 4 순회 판매원 문제에서의 성능 비교 실험 (CPU: seconds on Intel Pentium IV 1.5 GHz)

Instance	Ordinary				Normalization			
	Best	Avg.	σ/\sqrt{n}	CPU	Best	Avg.	σ/\sqrt{n}	CPU
att532	27686	27691.38	0.72	53	27686	27691.36	0.74	31
rat783	8806	8806.32	0.06	45	8806	8806.21	0.05	25
dsj1000	18659688	18667213.60	506.54	1053	18659688	18663945.73	769.04	549
pcb3038	137906	138031.77	5.72	444	137773	137903.70	5.49	317

[42]. 폭넓은 응용성으로 인해 많은 연구들이 있었다. 진화 연산 분야에서의 최근 연구들로는 참고문헌 [47-50] 등이 있다.

TSP를 위한 많은 인코딩 기법들이 제안되고 연구되었다. 다양한 인코딩 기법들에 대한 개괄적 연구로 참고문헌 [51]을 참고하기 바란다. 제안된 많은 기법들 중, 정점들을 0부터 $n-1$ 까지의 숫자로 표시하고 하나의 투어를 정점들의 방문 순서로 표현하는 순서 기반 인코딩을 고려한다. 그 이유는 이 인코딩이 충분한 중복성을 가지는 완전 비유사 인코딩이기 때문이다. 하나의 투어는 한쪽 방향으로 n 개의 방문 순서 표현이 가능하고 다른 방향으로 n 개의 방문 순서 표현이 가능하므로 표현형의 잉여 집합의 크기는 $2n$ 이다. 인코딩의 알파벳 크기는 정점의 개수 n 이므로 명제 4는 임의의 두 잉여 집합 사이의 평균 거리가 $\left(\frac{n-1}{n}\right)n = n-1$ 임을 말해준다. 또한, 명제 5는 최적 투어에 대응하는 임의의 유전자형 x^* 에 대하여 $\rho(x^*)=0$ 임을 말해준다. 두 부모해 사이의 해밍 거리를 최소화하도록 하나의 부모해를 다른 부모해에 맞추어 변환하는 정규화 기법을 사용한다. 부모해 내에서 각 도시의 위치를 미리 계산해 놓으면 정규화 과정은 계산 복잡도 $O(n)$ 을 가지도록 구현될 수 있다.

표 3과 4는 정규화를 사용한 경우와 그렇지 않은 경우의 실험 결과를 보여 준다. 실험은 TSPLIB의 인스턴스들을 사용하여 수행되었다.³⁾ 표 3에서 $E[d]$ 는 랜덤하게 생성된 10,000개의 지역 최적해들 사이의 평균 거리를 나타낸다. $\hat{\rho}(x)$ 는 지역 최적해들 중 가장 좋은 품질의 해 x 에 대하여 10,000개의 지역 최적해들로부터 추정된 $\rho(x)$ 값을 나타낸다. 지역 최적해들은 Lin-Kernighan(LK)

휴리스틱[52]을 사용하여 얻었다. 표 4에서 Best, Avg., σ/\sqrt{n} , CPU는 LK 휴리스틱과 결합된 혼합형 유전 알고리즘의 100회 수행에 대한 최소 투어 길이, 평균 투어 길이, 표본 표준 편차, 평균 수행 시간을 나타낸다. 유전 알고리즘의 해집단의 크기로 50을 사용하였다. 유전 알고리즘은 비례 선택, 오점 교차를 사용하고 2,500 세대 후에 정지하였다.

실험 결과는 앞선 절들의 결과들과 일관성을 보인다. 정규화는 지역 최적해들 사이의 평균 거리를 줄이고 상관성 측도 값을 증가시키며 유전 알고리즘의 성능을 향상시켰다. 정규화를 사용한 유전 알고리즘은 추가적인 정규화 알고리즘을 수행함에도 불구하고 그 수행 시간이 적게 걸린다. 이는 매 세대마다 교차 연산에 의해 만들어진 자손의 품질이 나쁘지 않아 지역 최적화 과정이 더 적은 반복 안에 종료되기 때문이다. 그림 8은 인스턴스 att532에 대하여 매 100세대마다 자손해들의 품질을 평균한 결과를 보여준다. 정규화를 사용한 경우의 자손들의 품질이 정규화를 사용하지 않은 경우에 비하여 상

투어 길이

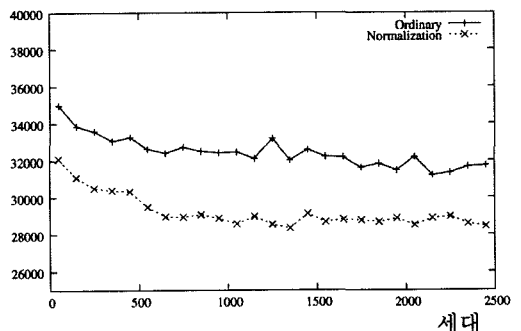
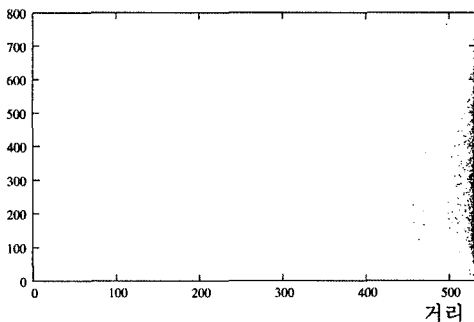


그림 8 매 100세대별 평균 자손 품질 (att532)

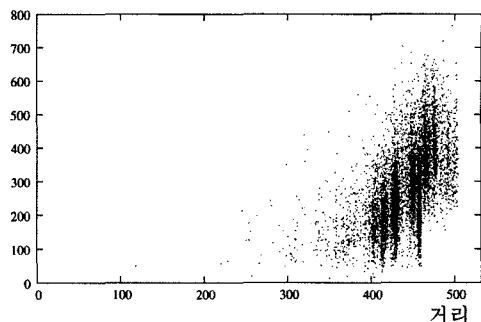
3) <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>

투어 길이 차이



(a) 정규화를 사용하지 않은 경우 (Ordinary)

투어 길이 차이



(b) 정규화를 사용한 경우 (Normalization)

그림 9 순회 판매원 문제에서의 정규화 효과 (att532)

대적으로 좋다는 사실을 알 수 있다. 따라서, 낮은 계산 복잡도를 가지는 정규화 연산은 혼합형 유전 알고리즘의 수행 시간을 줄이는 데 도움을 준다.⁴⁾ 그림 9는 인스턴스 att532에 대하여 가장 품질이 좋은 지역 최적해 x 를 중심으로 한 탐색 공간 지형의 변화를 보여준다.

4.5 정렬 네트워크 문제

정렬 네트워크(sorting network)은 정렬을 위한 하드웨어 논리 회로로서 정해진 순서로 데이터의 비교와 교환이 이루어진다. 정렬 네트워크는 회선(bus)들과 비교기(comparator)들로 구성된다. 비교기 $c(a,b)$ 는 a 번째 회선과 b 번째 회선의 값을 비교하여 정렬되어 있지 않으면 두 값을 교환한다. n 개의 입력을 가지는 정렬 네트워크를 n -회선 정렬 네트워크이라 부른다. 정렬 네트워크 문제는 주어진 입력 크기 n 에 대하여 가장 작은 수의 비교기를 가지는 n -회선 정렬 네트워크를 찾는 것이다.

정렬 네트워크 문제에서, 정렬 네트워크는 일반적으로 비교기들의 열로 인코딩된다[9,53-55]. Choi와 Moon[9]은 구조의 관점에서 정렬 네트워크 동형(sorting network isomorphism)을 정의하고 비교기들의 열로 정렬 네트워크를 표현하는 인코딩이 중복 인코딩임을 보였다. 이 인코딩에서 잉여 집합의 크기는 표현형마다 다를 수 있다. 그림 10은 동형 네트워크의 예를 보여준다. 두 네트워크의 회선들을 대응시키는 순열은 $\begin{pmatrix} 012345 \\ 543210 \end{pmatrix}$ 이다. 겉모습은 크게 다르지만 두 네트워크는 구조의 관점에서 같은 네트워크이며 같은 잉여 집합에 속한다.

정렬 네트워크에서 앞쪽(그림에서 왼쪽)의 비교기들은 이후의 비교기들에 큰 영향을 미친다는 사실이 알려져 있다[55]. 이는 앞쪽의 비교기들이 이후의 비교기들에 비해 네트워크의 품질에 더 큰 영향을 미친다는 것을 말

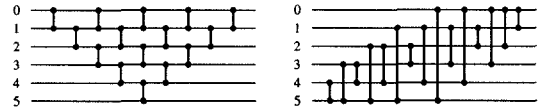


그림 10 두 동형 네트워크

해준다. 이러한 맥락에서 부모해의 앞쪽의 비교기들을 가능한 한 일관되게 맞추는 정규화 기법이 참고문헌 [9]에서 제안되었다. 예를 들어, 그림 11에서 부모해 1과 2를 고려해 보자. 두 네트워크의 처음 여덟개의 비교기들로 구성된 부분 네트워크는 겉모습은 다르지만 동형 관계에 있다. 그림 11(c)는 부모해 1에 대하여 정규화된 부모해 2를 보여준다. 실제 교차 연산은 부모해 1과 정규화된 부모해 2 사이에서 수행된다.

16-회선 정렬 네트워크 문제에 대하여 실험을 수행하였다. 실험을 위해 참고문헌 [9]에서 사용된 지역 최적화 휴리스틱과 혼합형 유전 알고리즘이 사용되었다. 그림 12는 지역 최적화 휴리스틱에 의해 랜덤하게 생성된 10,000개의 지역 최적해들과 그 중 가장 품질이 좋은 해 x 에 대하여 탐색 공간 지형의 변화를 보여준다. 이전의 문제들에서와 같이, 정규화에 의하여 $\rho(x)$ 의 추정치는 크게 증가되었고(0.079에서 0.847로) 상당한 성능 향상이 있었다. 알려진 최고 품질의 해 60-비교기 네트워크를 찾는 데 걸리는 시간이 Intel Pentium IV 1.5 GHz 상에서 758분에서 414분으로 줄었다.

5. 결론

중복 인코딩이 가지는 유사성 정도를 정량화함으로써 인코딩이 유전 알고리즘의 성능에 미치는 영향을 좀 더 엄밀하게 분석할 수 있을 것이다. 그러한 목적의 일환으로, 본 논문은 유사성에 대한 자연스러운 척도를 바탕으로 완전 비유사 인코딩을 정의하고 그에 대한 다수의

4) 그래프 분할 문제의 경우도 같은 맥락에서 설명될 수 있다.

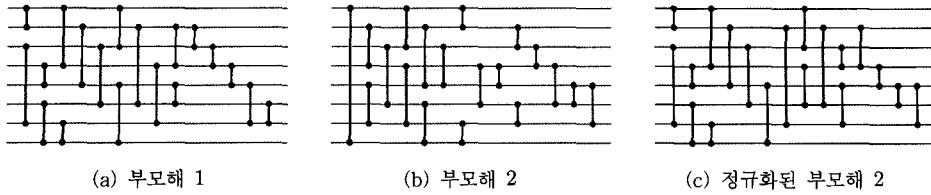
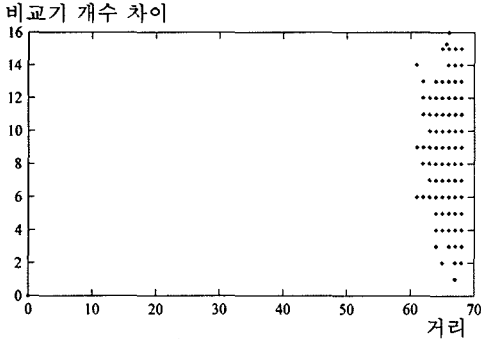
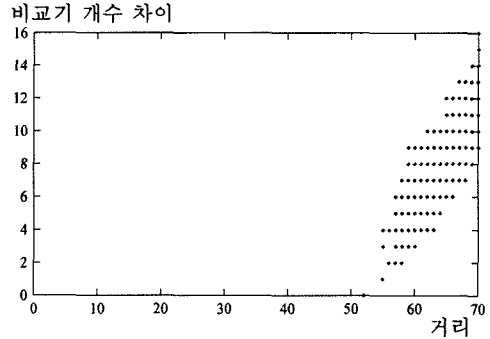


그림 11 8-회선 정렬 네트워크 문제에서의 정규화의 예



(a) 정규화를 사용하지 않은 경우 (Ordinary)



(b) 정규화를 사용한 경우 (Normalization)

그림 12 16-회선 정렬 네트워크 문제에서의 정규화 효과

엄밀한 결과들을 제시하였다. 특히, 완전 비유사 인코딩에 의해 적합도와 거리의 상관성이 없는 탐색 공간이 만들어짐을 증명하였다. 이는 완전 비유사 인코딩 혹은 그와 비슷한 성질을 가지는 인코딩을 사용하는 경우 유전 알고리즘의 성능이 저하되는 이유를 설명한다.

정규화는 비유사 인코딩이 만드는 탐색 공간을 유전 알고리즘이 잘 동작하는 탐색 공간으로 변환시키는 하나의 방법이다. 정규화의 타당성을 보이기 위해, 완전 비유사 인코딩이 일반적으로 사용되며 정규화 기법이 자연스럽게 정의되는 다수의 문제들을 살펴보았다. 그 문제들에 대한 실험을 통해, 본 논문은 정규화 기법이 탐색 공간이 가지는 굴곡의 정도를 완화시켜 유전 알고리즘의 성능을 크게 향상시킴을 보였다. 이는 주어진 문제에 따라 효과적으로 설계된 정규화 기법이 비유사 인코딩을 사용하는 유전 알고리즘의 문제점을 해결할 수 있음을 시사한다.

현재 정규화가 탐색 공간 변화에 미치는 영향에 대한 이론적인 분석을 수행중이다. 또한, 신경망 최적화 문제와 같은 다른 문제들을 위한 정규화 기법 설계에 대하여 연구중이며 유전 프로그래밍에 정규화의 개념을 적용하는 연구도 병행하고 있다.

참고 문헌

[1] N. J. Radcliffe. Forma analysis and random respectful recombination. In *Fourth International*

Conference on Genetic Algorithms, pages 222-230, 1991.

[2] E. Falkenauer. *Genetic Algorithms and Grouping Problems*. Wiley, 1998.

[3] N. Y. Nikolaev and H. Iba. Regularization approach to inductive genetic programming. *IEEE Trans. on Evolutionary Computation*, 5(4):359-375, 2001.

[4] Y. S. Yeun, W. S. Ruy, Y. S. Yang, and N. J. Kim. Implementing linear models in genetic programming. *IEEE Trans. on Evolutionary Computation*, 8(6):542-566, 2004.

[5] M. J. Martin-Bautista and M. A. Vila. A survey of genetic feature selection in mining issues. In *Congress on Evolutionary Computation*, pages 1314-1321, 1999.

[6] G. Laszewski. Intelligent structural operators for the k -way graph partitioning problem. In *Fourth International Conference on Genetic Algorithms*, pages 45-52, 1991.

[7] T. N. Bui and B. R. Moon. Genetic algorithm and graph partitioning. *IEEE Trans. on Computers*, 45(7):841-855, 1996.

[8] S. J. Kang and B. R. Moon. A hybrid genetic algorithm for multiway graph partitioning. In *Genetic and Evolutionary Computation Conference*, pages 159-166, 2000.

[9] S. S. Choi and B. R. Moon. Isomorphism, normalization, and a genetic algorithm for sorting network optimization. In *Genetic and Evolutionary Computation Conference*, pages 327-334, 2002.

- [10] C. Igel and P. Stagge. Effects of phenotypic redundancy in structure optimization. *IEEE Trans. on Evolutionary Computation*, 6(1):74-85, 2002.
- [11] P. Schuster. Molecular insights into evolution of phenotypes. In J. P. Crutchfield and P. Schuster, editors, *Evolutionary Dynamics - Exploring the Interplay of Accident, Selection, Neutrality and Function*. Oxford Univ. Press, 2002.
- [12] M. Gerrits and P. Hogeweg. Redundant coding of an NP-complete problem allows effective genetic algorithm search. In *Parallel Problem Solving from Nature*, pages 70-74, 1990.
- [13] B. A. Julstrom. Redundant genetic encodings may not be harmful. In *Genetic and Evolutionary Computation Conference*, page 791, 1999.
- [14] M. Kimura. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, 1983.
- [15] R. Shipman. Genetic redundancy: Desirable or problematic for evolutionary adaptation. In *Fourth International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 337-344. Springer-Verlag, 1999.
- [16] M. A. Shackleton, R. Shipman, and M. Ebner. An investigation of redundant genotype-phenotype mappings and their role in evolutionary search. In *Congress on Evolutionary Computation*, pages 493-500, 2000.
- [17] M. Ebner, M. Shackleton, and R. Shipman. How neutral networks influence evolvability. *Complexity*, 7(2):19-33, 2002.
- [18] J. D. Knowles and R. A. Watson. On the utility of redundant encodings in mutation-based evolutionary search. In *Parallel Problem Solving from Nature*, pages 88-98, 2002.
- [19] C. Van Hoyweghen and B. Naudts. Symmetry in the search space. In *Congress on Evolutionary Computation*, pages 1072-1079, 2000.
- [20] C. Van Hoyweghen, D. E. Goldberg, and B. Naudts. Building block superiority, multimodality and synchronization problems. In *Genetic and Evolutionary Computation Conference*, pages 694-701, 2001.
- [21] C. Van Hoyweghen, B. Naudts, and D. E. Goldberg. Spin-flip symmetry and synchronization. *Evolutionary Computation*, 10(4):317-344, 2002.
- [22] K. Weicker and N. Weicker. Burden and benefits of redundancy. In *Foundations of Genetic Algorithms*, volume 6, pages 313-333. Morgan Kaufmann, 2001.
- [23] F. Rothlauf and D. E. Goldberg. Redundant representations in evolutionary computation. *Evolutionary Computation*, 11(4):381-415, 2003.
- [24] H. Muhlenbein. Parallel genetic algorithms in combinatorial optimization. In *Computer Science and Operations Research: New Developments in Their Interfaces*, pages 441-453, 1992.
- [25] R. Dorne and J. K. Hao. A new genetic local search algorithm for graph coloring. In *Parallel Problem Solving from Nature*, pages 745-754. Springer-Verlag, 1998.
- [26] J. H. Kim, S. S. Choi, and B. R. Moon. Neural network normalization for genetic search. In *Genetic and Evolutionary Computation Conference*, pages 398-399, 2004.
- [27] S. Chen. *Is the Common Good ? A New Perspective Developed in Genetic Algorithms*. PhD thesis, Robotics Institute, Carnegie Mellon University, 1999.
- [28] G. Syswerda. Uniform crossover in genetic algorithms. In *Third International Conference on Genetic Algorithms*, pages 2-9, 1989.
- [29] S. Chen and S. Smith. Commonality and genetic algorithms. Technical Report CMU-RI-TR-96-27, Robotics Institute, Carnegie Mellon University, 1996.
- [30] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.
- [31] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.
- [32] R. A. Watson and J. B. Pollack. Recombination without respect: Schema combination and disruption in genetic algorithm crossover. In *Genetic and Evolutionary Computation Conference*, 2000.
- [33] S. Forrest and M. Mitchell. Relative building-block fitness and the building-block hypothesis. In *Foundations of Genetic Algorithms*, volume 2, pages 109-126. Morgan Kaufmann, 1993.
- [34] J. Horn and D. E. Goldberg. Genetic algorithm difficulty and the modality of fitness landscapes. In *Foundations of Genetic Algorithms*, volume 3, pages 243-270. Morgan Kaufmann, 1995.
- [35] S. A. Kauffman. Adaptation on rugged fitness landscapes. In D. Stein, editor, *Lectures in the Sciences of Complexity*, pages 527-618. Addison Wesley, 1989.
- [36] E. D. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325-336, 1990.
- [37] P. F. Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20:1-45, 1996.
- [38] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Sixth International Conference on Genetic Algorithms*, pages 184-192. Morgan Kaufmann, 1995.
- [39] K. D. Boese, A. B. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16:101-113, 1994.
- [40] P. Merz and B. Freisleben. Fitness landscapes,

- memetic algorithms and greedy operators for graph bi-partitioning. *Evolutionary Computation*, 8(1):61-91, 2000.
- [41] P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Trans. on Evolutionary Computation*, 4(4):337-352, 2000.
- [42] M. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
- [43] J. P. Kim and B. R. Moon. A hybrid genetic search for multi-way graph partitioning based on direct partitioning. In *Genetic and Evolutionary Computation Conference*, pages 408-415, 2001.
- [44] R. A. Watson, G. S. Hornby, and J. B. Pollack. Modeling building block interdependency. In *Parallel Problem Solving from Nature*, pages 97-106. Springer-Verlag, 1998.
- [45] M. Dietzfelbinger, B. Naudts, C. Van Hoyweghen, and I. Wegener. The analysis of a recombinative hill-climber on H-IFF. *IEEE Trans. on Evolutionary Computation*, 7(5):417-423, 2003.
- [46] F. Rothlauf. *Representations for Genetic and Evolutionary Algorithms*. Second Edition. Springer, 2006.
- [47] P. Merz and B. Freisleben. Genetic local search for the TSP: New results. *IEEE Conference on Evolutionary Computation*, pages 159-164, 1997.
- [48] Y. Nagata and S. Kobayashi. Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. In *Seventh International Conference on Genetic Algorithms*, pages 450-457. Morgan Kaufmann, 1997.
- [49] S. Jung and B. R. Moon. Toward minimal restriction of genetic encoding and crossovers for the 2D-Euclidean TSP. *IEEE Trans. on Evolutionary Computation*, 6(6):557-565, 2002.
- [50] D. I. Seo and B. R. Moon. Voronoi quantized crossover for traveling salesman problem. In *Genetic and Evolutionary Computation Conference*, pages 544-552, 2002.
- [51] N. J. Radcliffe and P. D. Surry. Fitness variance of formae and performance prediction. In *Foundations of Genetic Algorithms*, volume 3, pages 51-72. Morgan Kaufmann, 1995.
- [52] E. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, 1997.
- [53] W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In C. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*. Addison Wesley, 1992.
- [54] G. L. Drescher. Evolution of 16-number sorting networks revisited. Unpublished manuscript, 1994.
- [55] S. S. Choi and B. R. Moon. A graph-based

Lamarckian-Baldwinian hybrid for the sorting network problem. *IEEE Trans. on Evolutionary Computation*, 9(1):105-114, 2005.



최 성 순

2000년 서울대학교 전산학과 학사. 2006년 서울대학교 전기컴퓨터공학부 박사. 현재 서울대학교 전기컴퓨터공학부 두뇌한국21 박사후연구원 및 컴퓨터연구소 연구원. 관심분야는 알고리즘 설계 및 분석, 최적화, 진화 알고리즘, 랜덤 알고리즘, 인공지능

증, 인공지능



문 병 로

1985년 서울대학교 계산통계학과 학사
1987년 KAIST 전산학과 석사. 1994년 펜실바니아 주립대 박사. 1987년~1991년 (주)LG전자 중앙연구소 선임연구원
1994년~1995년 UCLA VLSI CAD Lab 박사후연구원. 1996년~1997년 8월 (주)LG

반도체 DT연구소 책임연구원. 1997년 9월~현재 서울대학교 컴퓨터공학부 부교수. 관심분야는 최적화, 유전 알고리즘, 알고리즘 설계 및 분석, 스케줄링 등