



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사학위논문

그래프 컬러링 문제를 위한  
효과적인 유전 알고리즘

(An) Efficient genetic algorithm for graph coloring

2014 년 8 월

서울대학교 대학원

전기.컴퓨터 공학부

권 홍 준

# 그래프 컬러링 문제를 위한 효과적인 유전 알고리즘

(An) Efficient genetic algorithm for graph coloring

지도교수 문 병 로

이 논문을 공학석사 학위논문으로 제출함

2014 년 4 월

서울대학교 대학원

전기.컴퓨터 공학부

권 홍 준

권홍준의 공학석사 학위논문을 인준함

2014 년 6 월

위 원 장 : 신 현 식 (인)

부위원장 : 문 병 로 (인)

위 원 : 박 근 수 (인)

# 요약

그래프 컬러링 문제는 널리 알려진 유명한 문제의 하나로 다항시간안에 확정적으로 푸는 알고리즘이 존재하지 않는 NP완비문제로 알려져 있으며 그래프 컬러링 문제의 효율적인 해 공간 탐색을 위한 많은 알고리즘들이 제안되었다.

혼합형 유전 알고리즘은 유전알고리즘의 장점인 광범위한 공간 탐색과 지역 최적화를 결합하여 다항시간 안에 해결 불가능한 넓은 해 공간을 효율적으로 탐색하기 위한 메타 휴리스틱의 일종이다. 본 논문에서는 이전 연구를 활용하여 복잡도가 큰 그래프에서도 효율적으로 해 공간을 탐색 가능한 유전 알고리즘을 제안한다.

실험을 통해 본 논문에서 제시한 유전 알고리즘이 최적해가 알려지지 않은 그래프 컬러링 예시에 대해 이전에 제시된 알고리즘들 보다 더 좋은 해를 찾음을 보인다.

주요어 : 유전 알고리즘, 그래프 색칠 문제,

학 번 : 2012-20741

# 목차

요약	i
목차	ii
그림 목차	iii
표 목차	iv
제 1 장 서론	1
1.1 개요 . . . . .	1
1.2 논문구성. . . . .	2
제 2 장 배경지식	3
2.1 그래프 컬러링 문제 . . . . .	3
2.2 유전 알고리즘 . . . . .	6
2.3 혼합형 유전 알고리즘 . . . . .	9
제 3 장 선행연구	11
3.1 그래프 컬러링 문제를 유전 알고리즘으로 푼 연구들 . . . . .	11
제 4 장 유전 알고리즘 설계	16
제 5 장 실험 결과 및 분석	22
5.1 실험 환경 . . . . .	22

5.2 실험 결과 및 분석 . . . . .	25
제 6 장 결론	30
참고문헌	31
Abstract	33

# 그림 목차

그림 2.1 유전 알고리즘의 구조 . . . . .	6
그림 2.3 혼합형 유전 알고리즘의 구조 . . . . .	10
그림 3.1 GCP 문제의 유전자 표현 . . . . .	13
그림 4.1 지역최적화 알고리즘. . . . .	19
그림 5.2 $K=12$ 일 때 queen10_10.col에서의 세대별 해 도시 . . . .	27
그림 5.3 $K=12$ 일 때 queen10_10.col 그래프의 해중 하나 . . . .	28

# 표 목차

표 5.2 데이터 셋 결과 비교 . . . . .	23
표 5.3 100 회 수행 결과 . . . . .	24



# 제 1 장 서론

## 1.1 개요

GCP(Graph Coloring Problem)는 NP 하드 문제군에 속하는 잘 알려진 문제 중 하나로 무향 그래프의 모든 정점에 색깔을 칠하되, 서로 간선이 존재하는 이웃 정점과 다른 색깔을 칠하도록 하는 색깔 조합을 찾는 문제이다.

정점의 갯수와 간선의 개수가 작은 그래프들에 대해서는 간단하게 해를 찾을 수 있으나 수 백개의 정점과 수 천개의 간선이 있는 큰 규모의 그래프들에 대해서는 유전 알고리즘과 같은 강력한 공간 탐색 알고리즘을 기용하지 않고서는 최적해를 찾는 것은 불가능하다. 이에 따라 최적 색깔 개수에 가까운 근사 색칠 방법 구하기 위한 여러가지 휴리스틱 알고리즘이 제시되었다.

그 중 가장 강력한 방법 중 하나는 물론 유전 알고리즘 일 것이다. 1975 년 John Holland 에 의하여 최초로 제안된 유전 알고리즘은 결정 알고리즘으로 다항 시간 내에 해결 할 수 없는 문제들에 대해 좋은 품질의 근사해를 보장한다.[2] 유전 알고리즘은 생물 진화의

원리를 통해 문제를 해결하는 매력적인 방법이기도 하다. 문제의 해를 유전자로 보고

선택(selection), 교배(crossover), 변이(mutation)와 같은 자연의 진화과정을 흉내 내는 연산을 통해 인간이 계속해서 고민해온 난제들을 손쉽게 해결하는 것을 목도하면 새삼 자연의 경이로움에 감탄하게 된다.

본 연구에서는 GCP 문제 해결을 위해 제시된 여러 유전 알고리즘들을 살펴보고 강력한 변이 연산을 통해 복잡한 그래프에서 좋은 효과를 보이는 유전 알고리즘을 제시한다.

## 1.2 논문구성

이 논문은 다음과 같은 구성으로 되어 있다. 2 장에서는 그래프 컬러링 문제를 위한 몇 가지 용어와 유전 알고리즘에 대해 설명한다. 3 장에서는 최근 고안된 그래프 컬러링 문제를 위한 유전 알고리즘들을 살펴보고 4 장에서는 강력한 지역 탐색 방법을 채용한 혼합 유전 알고리즘을 제시하며, 5 장에서는 실험결과를 제시한다. 마지막으로 6 장에 결론과 함께 후속 연구를 위한 방향을 담는다.

## 제 2 장 배경지식

### 2.1 그래프 컬러링 문제

무향 그래프  $G=(V,E)$ 에 대해  $G$ 의  $k$ -색칠( $k$ -coloring)은 정점 집합  $V$ 를 인접 정점이 같은 부분 집합에 속하지 않게  $k$ 개의 부분집합  $S_i : i=1,2,\dots,k$ 로 분할하는 것이다. GCP의 목적은 주어진 조건을 만족하는 가장 작은 자연수  $k$ 를 찾는 것이다. 주어진 그래프  $G$ 에서  $k$ -색칠을 만족하는 가장 작은 자연수  $k$ 를 Chromatic Number( $\chi(G)$ )라고 부르며, 주어진 인스턴스의 전역 최적해가 된다. 이론적인 Chromatic Number의 상한은 다음과 같이 알려져 있다. [4]

$$\chi(G) \leq \delta(G) + 1$$

위의 수식에서  $\delta(G)$ 는 그래프  $G$ 의 maximum degree를 의미한다. 이 수식은 직관적으로 maximum degree를 갖는 정점  $v$ 의 이웃들을 모두 서로 다른 색깔로 칠한 최악의 경우에는 색을 하나 더 추가할 수 밖에 없음을 의미한다. 뒤에서 다룰 알고리즘 중에는 위와

같은 이론적 상한값을 문제의 특성으로 삼아 좋은 성능을 내는 경우도 있다. 더 좋은 알고리즘을 만들기 위해 이와 같은 문제특정적(problem specific)인 정보들이 더 많이 활용되어야 할 것이다.[3]

주어진 정점 집합  $V$  에 속한 두 개의 정점  $a$  와  $b$  가 같은 색으로 칠해져 있고, 간선 집합  $E$  에  $a$  와  $b$  를 잇는 간선  $\{a,b\}$ 가 존재할 시, 이 간선을 충돌(conflicting)간선 혹은 나쁜 간선(bad edge)이라고 부른다. 이러한 간선을 포함하지 않는 해는 적합(legal)한 해라고 부른다.

GCP 를 풀기 위한 접근 방법을 크게 세 가지로 나눌 수 있다.

첫 번째 방법은 무조건 적합한  $k$ -색칠을 생성 하는 것이다. 이 때  $k$  가 작으면 분명 더 좋은 해이지만, 적합한  $k$ -색칠을 생성하는 것을 기준으로 문제를 풀어 나간다. 이러한 접근 방법의 경우  $k$  를 chromatic number 까지 최소화 하는 것이 목적이 된다.

두 번째 방법은 첫 번째 방법과 같이 무조건 적합한  $k$ -색칠을 하는 동시에  $k$  를 고정하여 생성하는 것이다. 물론 이러한 완벽한 해를 생성하는 것은 GCP 를 푸는 것과 동치이다. 따라서 어떠한 정점들은 어떠한 색도 가지지 못한 상태인 채로 생성되며, 이러한 해들을 수선하여 색깔이 칠해지지 않은 정점의 개수를 최소화 하는 것이 두 번째 방법의 목표가 된다.

세 번째 방법은  $k$  를 고정하고 가능한 나쁜 간선이 적은  $k$ -색칠을 생성하는 것이다. 이러한 접근에서는  $k$ -색칠의 나쁜 간선을 적절히 수선하거나 혹은 변형하여 나쁜 간선의 개수가 0 인  $k$ -색칠을 구하는 것을 목적으로 한다. 나쁜 간선의 개수가 0 인  $k$ -색칠을 구했으면  $k-1$  에 대해 다시 색칠을 수행한다. 이렇게 충돌 간선의 개수가 0 인 색칠이 존재하지 않을 때까지  $k$  값을 감소 시켜나가면서 알고리즘을 여러 번 반복 수행한다.

일반적으로 세번째 방법의 경우 알고리즘을 여러 번 수행해서 하므로 자원의 낭비가 심한 편이다. 그러나 강력한 지역최적화가 결합된 혼합형 유전 알고리즘에선 오히려 세 번째 방법이 더 효과적이다. 혼합형 유전 알고리즘의 경우 더 빨리 지역최적해(local optima)에 도달하며, 전역최적해(global optima) 근처에서의 미세한 조정에서는 시간이 많이 걸리나 chromatic number 보다 큰  $k$ -색칠에 대해서는 매우 빨리 풀어낸다. 따라서 본 논문에서는 세 번째 접근 방식을 선택한다.[6]

## 2.2 유전 알고리즘

유전 알고리즘은 생물의 진화 원리를 본따 문제의 해를 구하는 접근 방법이다. 유전 알고리즘은 메타 휴리스틱의 한 종류로 이와 같이 메타 휴리스틱에 속하는 잘 알려진 방법 중에는 Tabu Search나 SA(Simulated Annealing)이 있다. 유전 알고리즘은 비결정형 알고리즘으로, 다항 시간안에 풀 수 없을 정도로 문제 공간이 방대한 문제에 적합한 접근 방법이다. 유전 알고리즘은 구현이 쉬우며 폭넓은 문제에 활용 할 수 있으나, 강력한 공간 탐색 능력에 비해 미세한 조정에서는 부족한 모습을 보이기 때문에 아주 좋은 품질의 해를 얻기 위해 최적해 근처에서의 조정이 중요하다.[1]

```
Generate a populations of size n;  
Repeat{  
    for i = 1 to k {  
        choose parent1 and parent2 from the populations;  
        offspringi <- crossover(parent1,parent2);  
        offspringi <- mutation(offspringi);  
    }  
    replace(populations,[offspring1,...,offspringk]);  
}until(stopping criterion)  
Return the best individual of populations;
```

그림 2.1 유전 알고리즘의 구조

그림 2.1 는 유전자 알고리즘의 구조를 보여준다. 매 세대마다 n개의 해를 만들어서 k개의 기존 해와 교환한다. 이런 과정은 정지 조건을 만족할 때까지 수행되며, 해 집단 중 가장 좋은 해가 최종해가 된다.

$k/n$ 을 세대차(Generation gap)이라고 하며,  $k/n$ 이 1에 가까워 매 세대마다 해 집단을 크게 갱신하는 유전 알고리즘을 세대형 유전 알고리즘(Generational Genetic Algorithm)이라고 한다. 이에 반해, 세대차가  $1/n$ 에 가까운 유전 알고리즘을 안정 상태 유전 알고리즘(Steady-State Genetic Algorithm)이라고 한다.

유전 알고리즘에서는 해를 염색체(Chromosome)으로 표현한다. 염색체의 표현 방법은 문제에 적합해야 하는데, 가장 기본적인 이진 문자열 표현부터 2차원 배열, 정수 문자열 표현 등 많은 선택지가 있다. 염색체의 표현을 어떻게 하느냐에 따라 염색체를 다루는 유전 알고리즘의 연산자 또한 변하는데 이에 따라 문제 공간에도 크게 영향을 끼치게 된다.

유전 알고리즘은 복수개의 염색체를 집합하여 해 집단을 유지한다. 해 집단은 세대 초기에는 보통 매우 다양하나 세대를 지나 대치가 진행됨에 따라 다양성은 감소하여 수렴이 진행된다.

적합도는 염색체가 해당 문제를 잘 풀 수 있는 지를 나타내는 값이며, 염색체 표현을 적합도 값으로 변환해주는 함수를 적합도 함수라고 한다. 적합도가 높은 염색체일수록 더 좋은 해임을 나타내며 적절한 적합도 함수 설정은 문제 공간의 탐색에 있어 큰 힘을 발휘한다.

유전 알고리즘 과정을 실제로 수행하는 세 개의 연산자는 각각 선택(Selection), 교차(Crossover), 변이(Mutation), 대치(Replace)가

있다.

선택 연산자는 교차를 위해 해 집단에서 부모 해를 선택하는 역할을 한다. 적합도가 좋은 부모 일수록 자식 또한 좋은 적합도를 가질 확률이 많기 때문에 좋은 선택 연산자의 선택은 해의 품질에 크게 기여한다.

교차 연산자는 유전 알고리즘의 성능을 결정하는 가장 큰 요소 중 하나이다. 교차 연산은 부모 유전자로부터 일정 부분을 조합하여 새로운 자식 해를 생성한다. 교차 연산자의 종류는 일점 교차, 다점교차, 사이클교차, 내추럴 교차 등 다양한 방법이 제안되었으며, 교차연산이 문제의 특성을 더 잘 반영하여 좋은 스키마를 넘겨줄 확률이 클수록 유전 알고리즘의 성능은 크게 향상 될 것이다. 빌딩 블록의 관점으로 보면 교차 연산을 통해 부모가 갖고 있는 작은 블록들을 결합하여 더 큰 블록을 쌓아가는 과정이라 볼 수 있다.

변이 연산자는 유전자의 일부를 변형시켜 단순히 부모해들로부터 유전 받은 특성 뿐 만 아니라 부모해들이 갖지 못한 좋은 특성들을 발현시키려는 의도에서 적용된다. 교차 연산자에 비해 해의 작은 부분을 적은 확률로 변형 시키는 경우가 일반적이기 때문에 변이 연산자를 통해 아주 뛰어난 해로 탈바꿈하긴 힘들다. 하지만 해집단이 수렴할수록 변이연산자를 통해 적합도의 향상이 운 좋게 일어난다면 전체 해집단의 품질 향상에 크게 기여할 수 있다. 그렇지 않은 경우 대



개 대치 연산자를 통해 좋지 못한 변이는 해 집단에서 사라지게 된다.

대치연산자는 만들어진 자식해를 해 집단에 포함시킨다. 원래 해 집단 중 어떤 해를 해 집단에서 제외시킬지를 결정하는 것은 전체 해 집단의 품질에 크게 관여한다. 가장 일반적인 선택은 가장 적합도가 낮은 해를 제외시키는 것이지만 해 집단의 다양성이 주어진 문제 특성에서 더 좋은 역할을 할 경우 선부르게 수렴하여 품질을 저하시킬 수 있다.

## 2.3 혼합형 유전 알고리즘

앞서 설명한 유전 알고리즘은 전역최적해를 향해 해의 품질을 계속 향상시킬 수는 있으나 매우 느리게 진행된다. 이런 단점을 보완하기 위해 해를 근처의 지역최적해로 빠르게 접근하도록 하는 지역최적화 알고리즘을 포함한 혼합형 유전 알고리즘이 고안되었다. 혼합형 유전 알고리즘은 변이와 교차에만 의존하던 순수 유전 알고리즘과는 달리 문제 공간에 맞는 방법으로 해의 품질을 지역적으로 향상시키기 때문에 더 빨리 수렴하게 되며, 좋은 품질의 해를 기대할 수 있다.

혼합형 유전 알고리즘의 경우 지역최적화 알고리즘을 수행하는데 대부분의 시간을 소비하며, 지역최적화 알고리즘의 성능에 따라 해의 품질도 크게 차이가 난다. 그림 2.3은 혼합형 유전 알고리즘의 구조를 보여준다.

```
Generate a populations of size n;  
Repeat{  
    for i = 1 to k {  
        choose parent1 and parent2 from the populations;  
        offspringi <- crossover(parent1,parent2);  
        offspringi <- mutation(offspringi);  
        offspringi <- local_optimization(offspringi);  
    }  
    replace(populations,[offspring1,...,offspringk]);  
}until(stopping criterion)  
Return the best individual of populations;
```

그림 2.3 혼합형 유전 알고리즘의 구조

## 제 3 장 선행 연구

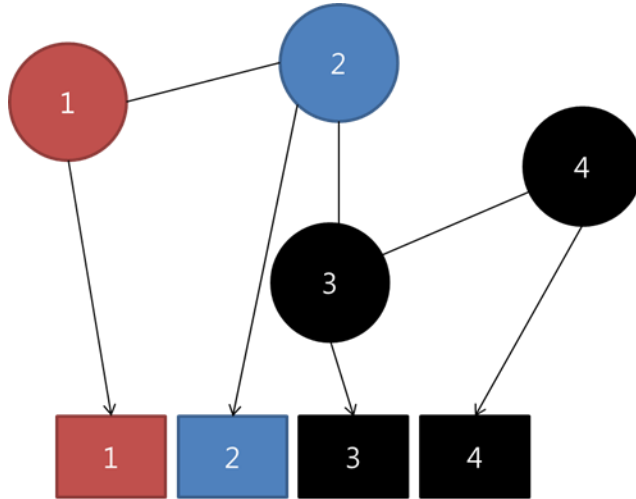
### 3.1 그래프 컬러링 문제를 유전 알고리즘으로 풀

#### 연구들

M. M. Hindi가 2012년 발표한 논문에서 *Wisdoms of Crowds* 개념을 활용한 유전 알고리즘을 제안했다.[5] 이 알고리즘은 세대형 알고리즘으로 50개의 고정된 개수의 해집합을 유지하나 매 세대마다 품질이 좋지 못한 해집합의 반에 해당하는 개수의 유전자를 새로 만든 유전자들로 대체하는 점이 특징적이다. 유전자는 일반적인 정수열 표현법을 사용한다. 이 알고리즘은 나쁜 간선 수를 0으로 만드는  $k$ -색칠을 구하는 접근법을 사용한다.

M. M. Hindi가 제안한 알고리즘은 두 가지 군의 연산자를 상황에 맞게 바꿔 도입하여 해의 품질을 높이는 미세조정을 시도하였다. 두 가지 군의 연산자를 사용한 정책 두 가지는 각각 두 가지의 선택 연산과 변이 연산을 선택하여 채용 한다. 해의 품질이 충분히 수렴하지 않은 상황 (현재 최적해의 나쁜 간선수가 4개 초과)에서는 랜덤으로 두 개의 해를 골라 더 좋은 해를 부모해로 삼아 자식 해를 생성하여, 모든 정점을 순회하며 나쁜 간선을 수리하는 변이 정책을 사용한다.

해가 충분히 수렴한 상황에서는 무조건 해집합에서 가장 좋은 해 두 개를 골라 부모해로 삼고 모든 정점을 순회하며 나쁜 간선을 적합한 색깔이 아닌 랜덤한 색깔로 칠하도록 한다. 이 과정에서 해의 품질이 퇴보하면 다시 첫 번째 정책으로 돌아가 해집합의 품질을 향상시키는 과정을 반복적으로 수행하게 된다. 즉 전역 최적해에 붙들린 상황에서는 해 집합을 크게 변이 시켜 전역 최적해를 빠져 나갈 수 있을만한 스키마를 확보할 수 있도록 하고, 해 집합의 다양성이 충분히 확보된 상황에서는 강한 수리 기능을 포함한 변이 연산자를 통해 해 집합을 수렴시킨다. M. M. Hindi가 고안한 유전 알고리즘은 20000세대를 수행한 후 종료되는데, 20000세대까지 수행해도 나쁜 간선수가 0인 최적해를 찾지 못하면 Wisdoms of Crowds 개념을 이용해 가장 좋은 해를 보완하였다. Wisdoms of Crowds 개념은 이미 TSP에 적용되어 성공을 거둔바 있다. [10]



a) 정수열 표현법

정점/ 색깔	1	2	3	4
빨강	1	0	0	0
파랑	0	1	0	0
검정	0	0	1	1

b) 이차원 이진표현법

그림 3.1 GCP문제의 정수열 유전자 표현과 이차원 이진표현법

Hasin은 2013년에 발표한 논문에서 이차원 이진 표현법을 포함한 혼합 유전 알고리즘을 발표하였다.[3] 그림 3-1은 정수열 표현 방법과 이차원 이진 표현법의 예이다. Hasin은 제안한 논문에서 한 개의 유전자를 사용하는 색깔의 개수를 행 수로, 정점의 개수를 열로 두는

이차원 이진 표현 방법을 사용한다. Hasin은 새로 고안한 유전자 표현 방법과 함께 새로운 유전자의 수리,교차,지역탐색 방법을 제시하였으며, 현존하는 알고리즘 중 대상 데이터 셋에 대해 일관되게 좋은 성능을 보여준다.

R.Abbasian은 그의 논문[6]에서 계층형 병렬 알고리즘의 일종으로 GCP를 병렬 프로그래밍 방법론으로 푸는 방법을 제시하였다. 또한 GCP와 같은 제약된 최적화 문제(Constraint optimization problem)을 풀기 위한 Genetic Modification을 제안하였으며, 데이터셋에서 좋은 성능을 보여주었다. [8]

B.Ray는 그의 논문에서 새로운 변이 연산인 guided mutation연산을 고안하여 순수 GA의 성능을 크게 끌어올렸다. 새로 제시된 guided mutation 연산자는 염색체의 여러 지점을 골라 색깔을 삭제한 후 수선 연산을 통해 염색체의 적합도를 향상시킨다.[7]

최근에 제안된 유전 알고리즘들을 살펴보면 모두 강력한 공간 검색을 위한 지역최적화 방법이 빠져있음을 알 수 있다. 가장 최근 발표된 M. M. Hindi가 제안한 방법은 새로운 염색체 표현법과 지역 최적화를 결합하여 해의 품질을 비약적으로 향상시켰다. 이 알고리즘에서는 지역최적화에 버금가는 강력한 변이 연산을 채용하였으나 세대형 해집합을 사용하고 지역 최적해 근처에서 미세 조정을 포기함으로써 강력한 수렴능력을 잃었다. R.Abbasian가 제안한 병렬형 유전 알고리

증에서는 지역최적화를 채용하지 않았으며, Bray가 제안한 guided mutation이 기존 변이 연산보다 적합도에 더 기여하는 것은 사실이나 지역최적화의 강력한 채굴 능력에 비할 바는 아니다.

## 제 4 장 유전 알고리즘 설계

이 논문에서 제안하는 새로운 혼합형 유전 알고리즘은 기본적인 혼합형 유전 알고리즘의 구조를 따른다. 앞서 제안된 알고리즘들의 경우 지역최적화 알고리즘을 채용하지 않거나 대신 변이 알고리즘을 사용한 경우가 많기 때문에 실험 결과의 대비를 위해 항상 모든 정점들을 순회하며 나쁜 간선을 고치는 지역 최적화 알고리즘을 사용하였다. 이는 논문[4]의 변이정책의 일부분으로도 사용된 바 있다. 강한 지역최적화 알고리즘을 사용하므로 설익은 수렴을 할 위험 또한 크기 때문에 0.2 이라는 큰 변이확률을 도입하여 수렴 상태에서도 돌연변이를 일으켜 지역 최적화 알고리즘과 시너지를 내도록 조정하였다.

- 그래프 컬러링 접근 방법

$k$  값을 고정한  $k$ -색칠 접근을 사용한다. 현재 프로그램에 인자로 주어진  $k$  에 대해 나쁜 간선을 하나도 갖지 않은 유전자를 발견할 때까지 수행한다. 프로그램에 주어지는



$k$  값은 보통 주어진 데이터셋의 Chromatic number 혹은 좋은 근사 값을 사용하였다.

- 해의 표현

기본적인 정수열 표현법을 사용한다. 주어진 그래프  $G=(V,E)$ 의 정점 집합의 크기  $n=|V|$ 일 때, 한 개의 유전자는 0 부터  $k-1$  까지의 숫자로 이루어진 순열로 표현된다. 유전자의 첫 번째로 나오는 숫자는 각 정점에 1 부터  $n$  까지 번호를 매길 때 1 번에 해당하는 정점에 칠한 색깔을 의미한다.

- 적합도 함수

적합도는 유전자가 지닌 나쁜 간선의 개수의 합이 된다. 그래프  $G(V,E)$ 의 정점 집합  $V$ 에 속한 정점  $a$ 와  $b$ 에 대해  $a$ 와  $b$ 를 잇는 간선  $\{a, b\}$ 가  $E$ 에 속하고 해당 유전자에서  $a$ 와  $b$ 가 같은 색깔로 칠해졌을 시 적합도에 1을 더하며 유전자의 전체 적합도를 계산하기 위해 정점 집합  $V$ 에 속한 모든 정점들에 대해 이 과정을 반복한다.

- 해집합 초기화

해집합의 크기는 50 개이며, 유전자의 각 정점 항목에 0 부터  $k-1$  까지의 난수를 생성하여 초기화한다.

- 선택 연산자

랜덤 하게 두 개의 유전자를 골라 적합도가 더 좋은 쪽을 부모 유전자로 사용하는 간단한 토너먼트 선택 방법을 사용하였다.

- 교차 연산자와 변이 연산자

교차 연산자는 유전 알고리즘의 성능에 큰 영향을 끼치는 핵심적인 연산자이다. GCP 문제에 적합한 교차에는 기본적인 일점 교차, 다점교차, 균등교차부터 시작해 싸이클 교차, 순서교차 등과 같은 여러 선택지가 존재한다. 이 중에서 가장 좋은 성능을 보여준 균등 교차를 사용한다. 균등 교차는 유전자의 매 한 점 마다 부모 유전자 둘 중 랜덤 하게 고른 한 부모의 유전자를 복사하는 방법이다. 가장 간단한 방법 중의 하나이나 싸이클 교차, 순서 교차에 비해 일부 테스트 셋에 대해 좋은 성능을 보여주었다.

변이 연산자는 20%의 확률로 수행되며, 염색체 상의 임의의 항목 하나를 임의의 색깔로 변형한다.

```

//Q; an empty queue

Localsearch (chromosome)
{
    Choose a vertex V randomly;
    Q.enqueue(V);
    while Q is not empty {
        V ← Q.dequeue;
        If V has conflict with adjacent vertex) {
            adjacentColors ← all adjacent color
            validColors ← allColors−adjacentColors;
            newColor ← random color from
validColors;
            Assign newColor to a vertex;
        }
        for each V' adjacent to V
            Q.enqueue(V');
    }
}

```

}

그림 4.1 지역 최적화 알고리즘

- 지역 최적화 알고리즘

지역 최적화 알고리즘은 혼합 알고리즘에서 가장 핵심적인 역할을 담당한다. 한 개의 정점을 골라 나쁜 간선이 존재할 경우 정점의 색깔을 바꾸며, 이후 인접 정점들을 너비 우선 탐색으로 순회한다. 한번 수행 될 때 마다 정점의 개수의 제곱에 비례하는 수행시간을 가지나, 전처리 과정에서 각 정점의 인접 정점에 대한 리스트를 모두 수집한 후 이 정보를 활용하여 각 정점의 인접 정점의 색깔을 확인 할 때 우선 순위 큐를 통해 구현함으로써 수행 시간을 줄이도록 하였다.

- 대치 연산

매 세대마다 한 개의 자식 유전자를 생성하여 해집합에서 가장 나쁜 품질을 가진 해와 교체한다.

- 정지 조건

적합도가 0 인 유전자를 발견하거나 15000 세대를 넘을 시  
종료한다.

## 제 5 장 실험 결과 및 분석

### 5.1 실험 환경

본 논문에서 제안한 유전 알고리즘을 위한 데이터 셋은 모두 Center for Discrete Mathematics and Theoretical Computer Sciences(DIMACS) benchmarking graph collection으로부터 얻었다. [8] 모든 데이터 셋은 “p edge 457 2440”과 같은 헤더를 포함하며, 이는 해당 데이터가 457개의 정점과 2440개의 간선으로 이루어졌음을 의미한다.

실험은 주어진 표에 17개의 데이터 셋에 의해 수행되었으며, 이전 연구에서 제시된 최적해와 본 연구에서 찾은 해를 함께 표기하였다.

실험 수행은 각 데이터에 대해 100번씩 반복 되었으며, 100번의 수행에서 얻은 최적 해를 표 5.2에 표기하였다. 각 테스트 셋에 대해 최적 해를 얻은 횟수와 평균 성공 수행 시간을 표 5.3에 표기하였다.

모든 코드는 C++로 작성되었으며 gnu g++로 컴파일 하였다. 실험은 Xeon 2.4ghz 컴퓨터에서 수행하였다.

표 5.2 데이터 셋 결과 비교

Dataset	V	E	X(G)	[5]	[6]	[7]	[3]	Proposed GA
Myciel3.col	11	20	4	4	4	-	4	4
Myciel4.col	23	71	5	5	5	-	5	5
Myciel5.col	47	236	6	6	6	-	6	6
Myciel6.col	95	755	7	-	-	-	7	7
Myciel7.col	191	2360	8	-	8	-	8	8
Queen5_5.col	25	160	5	5	5	5	5	5
Queen6_6.col	36	290	7	7	8	8	7	7
Queen7_7.col	49	476	7	7	8	7	7	7
<b>Queen10_10.col</b>	<b>100</b>	<b>2940</b>	<b>?</b>	<b>-</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>
Miles250.col	128	387	8	8	8	-	8	8
Miles500.col	128	1170	20	-	-	-	20	20
Miles750.col	128	4226	31	-	-	31	31	31
Miles1000.col	128	3216	42	42	42	42	42	42
Miles1500.col	128	5198	73	-	73	73	73	73
Homer.col	561	1629	13	13	13	13	13	13
Mulsol.i.1.col	197	3925	49	-	49	49	49	49
Zeroin.i.1.col	211	4100	49	-	-	-	49	49

표 5.3 100회 수행 실험 결과

Dataset	V	E	X(G)	MG	SUCCESS RATIO(%)	AVG SUCCESS TIME(s)	Proposed GA
Myciel3.col	11	20	4	5	100	0.10	4
Myciel4.col	23	71	5	11	100	0.10	5
Myciel5.col	47	236	6	23	100	0.52	6
Myciel6.col	95	755	7	47	100	0.61	7
Myciel7.col	191	2360	8	95	100	6.41	8
Queen5_5.col	25	160	5	16	100	0.29	5
Queen6_6.col	36	290	7	19	52	1.84	7
Queen7_7.col	49	476	7	24	7	3.12	7
<b>Queen10_10.col</b>	<b>100</b>	<b>2940</b>	<b>?</b>	<b>35</b>	<b>68</b>	<b>13.37</b>	<b>12</b>
Miles250.col	128	387	8	16	100	0.29	8
Miles500.col	128	1170	20	38	100	3.23	20
Miles750.col	128	4226	31	64	100	13.27	31
Miles1000.col	128	3216	42	86	100	29.21	42
Miles1500.col	128	5198	73	106	100	4.23	73
Homer.col	561	1629	13	99	100	171.82	13
Mulsol.i.1.col	197	3925	49	121	100	11.43	49
Zeroin.i.1.col	211	4100	49	111	100	39.47	49



## 5.2 실험 결과 및 분석

표 5.2는 제안한 유전 알고리즘의 최적해를 다른 논문에서 제시된 결과와 비교한 것이다. 첫 번째 열부터 네 번째 열까지 데이터의 이름, 정점의 개수, 간선의 개수와 알려진 최적해인 chromatic number를 표시하였다. 그리고 다음 열부터 선행 연구 논문에 포함된 결과들을 각각 표기하였다. 마지막 열은 본 논문에서 제안한 알고리즘이 찾은 최적해를 표기하였다.

Queen10\_10.col의 경우 아직 알려진 최적 chromatic number가 존재하지 않아 chromatic number 열에 “?”와 같이 표시하였으며, “-“와 같이 표시된 셀은 해당 데이터 셋에 해당하는 논문의 결과값이 존재하지 않음을 의미한다.

표 5.2는 논문에서 제시된 알고리즘이 17개의 테스트셋 중 queen10\_10.col을 제외한 모든 데이터에서 전역최적해인 chromatic number를 만족하는 해를 찾아냈음을 보여준다. Chromatic number가 알려지지 않은 queen10\_10.col에서는 가장 좋은 근사해를 찾아내었다.

Myciel3.col과 같은 작은 규모의 그래프부터 homer.col과 같은 매우 큰 크기의 그래프까지 최적해를 전부 찾아 냈으며,

miles750.col,miles1000.col과 같은 밀집된 그래프에서도 유전 알고리즘의 정지조건인 15000세대 이하에서 전부 해를 찾아내는 뛰어난 성

능을 보였다.

표 5.3은 100회 수행실험의 결과를 자세히 표기하였다. 첫 번째 열부터 네 번째 열까지 각각 데이터셋의 이름, 정점의 개수, 간선의 개수와 알려진 chromatic number를 표기하였다. 다섯 째 열은 그래프의 복잡 정도를 알아보기 위해 주어진 그래프의 최다 간선 개수를 표기하였다. 6 번째 열은 100회 수행시 전역 최적해를 찾은 수행 횟수이다. 일곱 번째 열은 최적해를 찾는 데 성공 했을 시 평균 수행시간을 표기하였다. 마지막 열은 본 논문에서 제안한 알고리즘에서 찾아 내는데 성공한 최적해의 색깔 수이다.

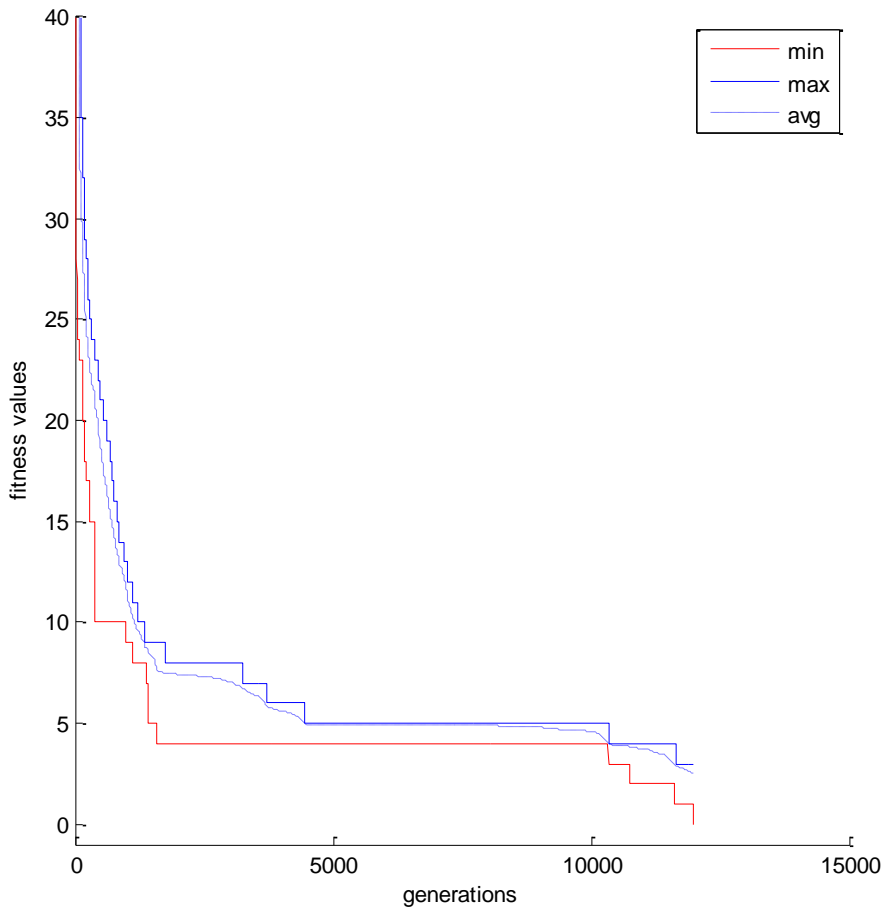


그림 5.2 k=12 일 때 queen10\_10.col에서의 세대별 해 도시

그림 5.2는 queen10\_10.col의 최적 근사해인 12를 찾았을 시 세대 진행에 따라 해 집단의 최적해, 가장 품질이 나쁜해, 평균 품질을 그래프로 도시하였다. 강력한 지역최적화 알고리즘의 영향으로 해가 매우 빠르게 수렴하는 모습을 볼 수 있으며, 강한 수렴 이후 5000세대 이후 많은

세대수 동안 최적해의 품질은 크게 향상되지 않았으나, 해집합의 평균이 점진적으로 최적해에 접근하여, 해집단이 완전히 수렴하기 직전에 성공적으로 지역최적해를 벗어나 전역최적해로 나아가는 모습을 볼 수 있다.

20%라는 높은 변이 확률을 채용한 변이 연산자와 지역최적화 알고리즘의 강한 탐색능력이 시너지를 일으켜 복잡한 그래프에서의 최적해를 얻는데 크게 기여했음을 알 수 있다.

```
11 3 7 2 5 0 6 8 1 9 0 8 10 11 9 3 4 7 5 2 3 5 9 6 8 10 2 1 11 0 6 2 11 0 1 4 8 10 3 5 10 4
5 7 6 2 0 11 9 8 1 11 2 8 10 7 5 3 6 4 2 6 0 3 4 8 1 9 10 7 9 10 4 1 7 11 3 2 0 6 5 0 3 10 2
1 7 6 4 11 7 9 1 4 11 5 10 0 2 3
```

그림 5.3 k=12 인 queen10\_10.col 그래프의 해 중 하나

그림 5.3는 k=12일 때 queen10\_10.col 그래프에서 찾아낸 최적해의 유전자 표현형이다.

Queen6\_6.col, queen7\_7.col, queen10\_10.col을 제외한 모든 대상 데이터 셋은 항상 최적해를 찾는 데에 성공하였으나, 저 셋의 경우 초기해의 품질에 따라 성능이 크게 차이가 나는 한계점을 보였다. 다양한 교차 연산자를 시험해 보았을 때 나머지 테스트 셋에 대해서는 유의미한 차이를 보이지 않았으나 queen7\_7.col 의 경우 균등 교차를 사용했을시 chromatic number에 해당하는 최적해를 매우 낮은 확률로 찾을 수 있었다. 이는 queen7\_7.col의 경우 특정한 지역 최적해의 응집력이 매우 강

해 탈출하기 어렵다는 점을 의미하며, 본 논문에서 제안한 지역 최적화 방법의 경우 수렴 속도는 매우 빠르나 해의 다양성을 추구하는 교란 작용은 그리 크지 않기 때문에 최적해를 찾는 데 어려움을 겪었다고 분석할 수 있다. 다양한 교차 방법 중 해의 초기 품질은 나쁠지 몰라도 다양한 자식해를 만들 수 있는 균등 교차의 경우가 이러한 한계점을 상당 부분 보강했다고 볼 수 있다.

수행 시간 면에서 최적해를 찾는 경우 대부분 5000세대 이하에 수행이 종료되었으며, 그래프의 규모가 크더라도 상대적으로 chromatic number가 큰 경우에는 수행시간이 매우 빨랐음을 알 수 있다. 본 논문의 경우 색깔 수를 고정한 상태에서 나쁜 간선을 0개로 만드는 해를 찾을 때 수행이 종료되므로, 색깔 수를 고정하지 않은 상태에서 점점 줄여나가는 알고리즘 보다는 수행시간이 훨씬 빨랐다. 그러나 chromatic number가 알려지지 않거나 근사값이 주어지지 않은 그래프에 대해서는 여러 번 수행을 통해 최적 근사해를 찾아야 하므로 현재 테스트 셋에 표기된 시간보다 수행시간은 사실상 더 걸릴 것이다. 그러나 강한 수렴능력을 보여주기 때문에 사실 주어진 그래프의 chromatic number보다 더 큰 값을 목표로 했을 때는 훨씬 빠른 수행을 가지므로 큰 격차는 없을 것으로 예상된다.

## 제 6 장 결론

유전 알고리즘은 자연의 원리를 본따 넓은 해 공간을 탐색하는 알고리즘으로써 다항시간 내에 해결이 어려운 여러 문제들에 대해 좋은 성능을 보여준다.

본 논문에서는 그래프 색칠 문제를 해결하기 위해 혼합형 유전 알고리즘을 사용하였다. 그래프 색칠 문제는 다항시간 내에 해결이 불가능한 NP-완비군의 문제로서 매우 넓은 해 공간을 갖고 있다.

그래프 컬러링 문제들을 풀기 위한 선행 연구가 매우 많았으며, 본 논문에서는 이전 선행 연구들의 한계점을 고려하고 취합하여 강력한 지역 탐색 알고리즘을 채용하여 복잡한 테스트 셋에 대해서도 빠른 시간 안에 최적해를 풀어내는 혼합 알고리즘을 제시하였다.

실험의 결과 전역 최적해가 알려진 문제에 대해 모두 전역 최적해를 찾아 냈으며, 아직 전역 최적해가 알려지지 않은 테스트 셋에 대해서도 현재까지 가장 좋은 품질의 근사해를 찾아내는 데 성공하였다.

그러나 일부 테스트 셋에 대해서는 매 수행마다 항상 최적해를 보완하지 못했으며, 향후 연구에서 더 안정적으로 좋은 해를 얻을 수 있도록 보완되어야 할 것이다.

## 참고문헌

- [1] 문병로, “유전알고리즘”, 두양사, 2003.
- [2] Holland, J., *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.
- [3] Chans M., Hertz A. and Werra D, “Some experiments with simulated annealing for coloring graphs, *European Journal of Operational Research*, 32, 260–266.”, 1987.
- [4] M. Kubale, “Graph colorings,” American Mathematical Society, 2004.
- [5] M. M. Hindi and R. V. Yampolskiy, “Genetic algorithm applied to the graph coloring problem,” in *Proc. 23rd Midwest Artificial Intelligence and Cognitive Science Conf.*, April 2012, pp. 61–66.
- [6] R. Abbasian and M. Mouhoub, “An efficient hierarchical parallel genetic algorithm for graph coloring problem,” in *Proc. 13th annual conf. on Genetic and evolutionary computation*, 2011, pp. 521–528.
- [7] B. Ray, A. J. Pal, D. Bhattacharyya, and T. Kim, “An efficient GA with multipoint guided mutation for graph coloring problems,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 3, no. 2, pp. 51–58, 2010.
- [8] M. Trick. (2013, March 1). Graph coloring instances. Michael Trick's Operations Research Page. [Online]. Available:

<http://mat.gsia.cmu.edu/COLOR/instances.html>

[9] Hasin Al Rabat Chowdhury, Tasneem Farhat, and Mozammel H. A. Khan, "Memetic Algorithm to Solve Graph Coloring Problem", International Journal of Computer Theory and Engineering, Vol. 5, No. 6, December 2013.

[10] Yampolskiy, Roman V., and El-Barkouky, Ahmed, "Wisdom of Artificial Crowds Algorithm for Solving NP-Hard Problems", International Journal of Bio-Inspired Computation, 2011.



# Abstract

GCP(Graph coloring problem) is a well-known NP-hard problem. Due to its NP-hardness, There is no exact method to solve GCP for reasonable time scale. Many heuristic algorithms were proposed to solve GCP efficiently.

Genetic algorithm is most famous for its dominant ability for searching huge problem space. Combined with local search algorithm, Hybrid Genetic Algorithm has been known as most efficient tool for solving NP-hard problems. On this thesis, I propose new Hybrid Genetic Algorithms solving GCP instances of highly complex graphs efficiently.

The Proposed algorithm succeeded at solving the one of sample data set and outperformed one of a recent approach in terms of finding the minimum number of colors needed to color some of the graphs.

**keywords : Hybrid genetic algorithm,Graph Coloring**  
**student number : 2012-20741**