**유전 알고리즘과 지역 최적화를 이용한**

**그래프 컬러링 문제의 효율적 해결**

**-정올반 휴리스틱 프로젝트-**

**Efficient Solution of Graph Coloring Problem**

**using Genetic Algorithm and Local Optimize**

**2019년 2월 08일**

**선린 인터넷 고등학교**

**소프트웨어과**

**나정휘, 권노현**

**초 록**

그래프 컬러링 문제를 비롯한 NP-Complete클래스(NP-완비군)에 속하는 문제들은 다항 시간 안에 해결하는 방법이 알려지지 않았거나, 존재하지 않는다.

이러한 특징을 지닌 그래프 컬러링 문제를 효율적으로 해결하기 위해 많은 알고리즘들이 연구되어왔다.

이 프로젝트에서는 비교적 간단하면서도 대부분의 경우에 대해 최적해에 가까운 답을 구해주는 유전 알고리즘과 지역 최적화 기법 이용하여 그래프 컬러링 문제를 해결하고자 한다.

이 프로젝트 보고서에서는 지금까지 널리 알려진 방법들을 참고해 직접 구현한 알고리즘과 해당 알고리즘이 얼마나 효율적으로 문제를 해결하는지 소개한다.

**주요어:** 유전 알고리즘, 그래프 컬러링, NP-Complete, NP-완비군, 지역 최적화

**학번:** 10504 권노현, 10514 나정휘

**목 차**

**제 1 장 서 론 …………………………………………………………………………………04**

제 1 절 개 요 …………………………………………………………………………………………………..04

제 2 절 보고서의 구성 ………………………………………………………………………………………..04

**제 2 장 배경 지식 ………………………………………………………………………………05**

제 1 절 시간 복잡도 클래스 ……………………………………………………………………………….05

제 2 절 그래프 컬러링 ………………………………………………………………………………………..05

**제 3 장 모델 설계 ………………………………………………………………………………06**

**제 4 장 실험 결과 ………………………………………………………………………………07**

제 1 절 실험 환경 ………………………………………………………………………………………………07

제 2 절 실험 결과 ………………………………………………………………………………………………07

**제 5 장 최 적 화 ………………………………………………………………………………..08**

제 1 절 알고리즘 구현 ……………………………………………………………………………………….08

제 2 절 성 능 ………………………………………………………………………………………………….09

**참고 문헌 ………………………………………………………………………………………….10**

**Abstract …………………………………………………………………………………………..11**

**제 1 장 서 론**

**제 1 절 개 요**

그래프 컬러링 문제(GCP)는 무향 그래프에 있는 모든 정점들을 색칠하되 이웃한 정점끼리는 서로 다른 색깔로 칠하도록 하는 색깔의 조합을 찾는 문제로, 지금까지 다항 시간 안에 해결하는 방법이 알려지지 않은 NP-Complete클래스에 속하는 문제이다.

정점과 간선의 개수가 많아질수록 수행시간이 기하급수적으로 증가하는 이 문제를 빠르게 해결하기 위해 많은 방법이 연구되어왔고, 유전 알고리즘을 이용한 방법 또한 많이 연구되어왔다.

John Holland에 의해 제안된 유전 알고리즘은 생물의 진화 방법에서 아이디어를 얻은 알고리즘으로, 납득할 만한 근사해를 구해준다. [1]

이 프로젝트에서는 널리 알려진 유전 알고리즘을 이용한 방법들을 참고해 직접 구현을 해보고, 이 방법이 얼마나 효과적인지 이 보고서에서 실험을 통해 측정을 해보고자 한다.

**제 2 절 보고서의 구성**

2장 1절에서 NP-Complete를 비롯한 시간 복잡도 클래스에 대해 간략하게 알아본 뒤, 곧바로 그래프 컬러링에 대해 설명한다. 모델 설계에 대해 3장에서 설명하고, 해당 모델로 테스트한 결과를 4장에서 제시한다.

**제 2 장 배경 지식**

**제 1 절 시간 복잡도 클래스**

계산 복잡도 이론에서 다항 시간 알고리즘이 존재하는 문제들의 집합을 P문제라고 한다. P문제와 더불어 많이 언급되는 것은 NP문제가 있다. 어떤 문제에 대해 답이 주어졌을 때, 이것이 정답인지를 다항 시간 내에 판별할 수 있는 문제들의 집합을 NP문제, 혹은 NP클래스라고 한다. 모든 P문제는 NP문제에 속한다.

다항 시간 내에 해결하는 방법이 알려지지 않은 문제들의 집합을 NP-Hard 클래스, NP-난해 문제 등으로 부른다. NP-Hard에 속하면서 동시에 NP에 속하는, 즉 다항 시간 내에 해결할 수는 없지만 다항 시간 내에 답을 판별할 수 있는 문제들의 집합을 NP-Complete 클래스, NP-완비 문제 등으로 부른다.

**제 2 절 그래프 컬러링**

그래프 컬러링 문제(GCP)는 무향 그래프의 모든 정점을 색칠하되, 인접한 정점의 색깔을 다르게 색칠하는 조합을 찾는 문제이다. 즉, 그래프에서 독립 집합을 찾는 문제와 동치라고 볼 수 있다.

이외 또 다른 목적은 그래프를 k개의 색깔로 색칠하되, 인접한 정점의 색깔을 다르게 색칠할 수 있는 최소의 자연수 k를 구하는 것이다. 무향 그래프 G에 대한 이론적인 k값의 상한은 다음과 같이 알려져 있다. [2]

χ(G)는 k값의 상한, δ(G)는 그래프 G의 최대 차수(max degree)를 의미한다.

두 정점 u, v를 잇는 간선 {u, v}가 존재하고, u와 v에 칠해진 색깔이 같다면 해당 간선을 나쁜 간선 (bad edge)라고 부른다. 나쁜 간선이 존재하지 않는 해를 적합해라 하고, 이것을 구하는 것이 이 문제의 목적이다.

**제 3 장 모델 설계**

이 프로젝트에서는 GCP를 해결하기 위해 k값을 고정하고 색칠하는, K-Coloring 방식을 사용한다. k값을 MaxDegeree+1 부터 시작해 적합해를 발견할 때마다 k를 1씩 감소시켜 가능한 가장 작은 k값을 구하는 방식으로 진행이 된다. 500000 세대가 지나도 적합해가 나오지 않는 경우, 불가능한 것으로 판단하고 종료한다.

**해의 표현**

그래프의 정점의 개수를 V라고 하면, 각각의 유전자는 0부터 k-1까지의 정수로 이루어진 길이 V의 순열로 나타낸다.

**적합도 함수**

적합도는 유전자에서의 나쁜 간선의 개수로 판별한다. 모든 간선 {u, v}에 대해 정점 u와 v에 칠해진 색깔이 같을 때마다 적합도를 1 증가시킨다. 적합도 함수의 값을 0으로 만드는 것이 목적이다.

**선택 연산**

두 개의 유전자를 골라 그 중 더 우월한 유전자, 즉 적합도 함수의 값이 더 작은 유전자를 부모 유전자로 선택하는 *토너먼트 선택 기법*을 사용한다.

**교차 연산**

교차 연산으로는 유전자에서 각각의 점에 대해 두 부모 유전자 중 랜덤하게 하나를 골라 채용하는 *균등 교차 기법*을 사용한다.

**변이 연산**

유전자의 매 점마다 0.15%의 확률로 변이 연산을 수행해 0부터 k-1까지의 정수 중 하나로 바꾼다.

**대치 연산**

매 세대마다 토너먼트 선택에서 탈락된 유전자들을 교차 연산에서 생성된 유전자로 교체한다.

**정지 조건**

적합도 함수의 값이 0인 유전자를 발견하거나, 500000세대가 넘을 시 진화를 종료한다.

**제 4 장 실험 결과**

**제 1 절 실험 환경**

이 프로그램은 Intel® Core™ i7-7700HQ Processor, DDR4 8GB 환경에서 실행되었다. 성능 측정용 프로그램은 C++로 개발되었고, 시연용 visualization 프로그램은 JavaScript로 개발되어 Chrome 브라우저 상에서 실행된다. 해당 소스코드는 <https://github.com/justiceHui/GCP-by-GeneticAlgorithm/> 에서 확인 가능하다.

성능 측정용 그래프 데이터는 Center for Discrete Mathematics and Theoretical Computer Sciences(DIMACS) benchmarking graph collection에 있는 Queen7\_7.col 데이터셋을 사용했고, 시연용 그래프는 무작위로 생성된 그래프에서 진행된다. [8]

**제 2 절 실험 결과**

정점 49개, 간선 952개, 최대 차수 48로 이루어진 Queen7\_7데이터셋에서 측정한 결과, 알려진 최소 k값인 7에는 도달하지 못하였지만, k가 8까지 가는데 단 53초가 채 걸리지 않았다.

적합해에 수렴하는 속도를 보기 위해 정점 200개, 간선 400개로 이루어진 무작위 그래프에서 6개의 색으로 색칠한 결과, 적합도 함수의 변화는 다음과 같다.

예상보다 많이 느려서, 유전 알고리즘에 지역 최적화 기법을 적용시켜보았다.

**제 5 장 최 적 화**

**제 1 절 알고리즘 구현**

실행 시간을 단축시키기 위해 BFS를 이용한 지역 최적화 기법을 도입했다.

각각의 유전자(색 조합)을 그래프에 대입을 시킨 뒤, BFS로 그래프를 순회하며 인접한 정점의 색깔이 같다면 정점의 색깔을 바꿔주는 방법을 도입했다. 코드의 흐름은 아래와 같다.

|  |
| --- |
| localOptimization(gen){  V <- choose a vertex randomly  Q <- empty Queue  Q.push(V)    while Q is not empty then  V <- Q.pop  adjColor <- all adjacent color  canColor <- all of color - adjColor  V.color <- choose a color in canColor randomly  for each U adj to V  Q.push(U)  } |

**제 2 절 성 능**

무작위로 생성한 정점 200개, 간선 400개 로 구성된 그래프를 4개의 색깔로 칠하면서 시간을 측정한 결과 모두 1초 이내에 결과가 나왔다.

정점 49개, 간선 952개, 최대 차수 48로 이루어진 Queen7\_7데이터셋에서 알려진 K의 최솟값인 7개의 색깔로 칠할 때의 수렴 속도와 실행 시간은 아래와 같다.

**참고 문헌**

[1] Holland, J., Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, 1975.

[2] M. Kubale, “Graph colorings,” American Mathematical Society, 2004.

[3] M. Trick. (2013, March 1). Graph coloring instances. Michael Trick's Operations Research Page. [Online]. Available: <http://mat.gsia.cmu.edu/COLOR/instances.html>

**Abstract**

**Efficient Solution of Graph Coloring Problem**

**using Genetic Algorithm and Local Optimize**

나정휘(Jeounghui Nah)

권노현(Nohyun Kwon)

Sunrin Internet High School

Division of Software

Problems belong to the NP-Complete class, including GCP, are not known or exist method to be solved in polynomial time.

Many algorithms have been studied to efficiently solve GCP.

In this project, we try to solve GCP by using genetic algorithm and Local Optimize which is simple but finds near optimal solution.

In this project report, we introduce the algorithm that we have implemented by referring to various methods and how efficient this algorithm solves the problem.

**Keywords**: Genetic Algorithm, Graph Coloring Problem, NP-Complete, Local Optimize

**Student Number**: 10504 Nohyun Kwon, 10514 Jeounghui Nah