

그래프 착색 방법에 기초한 대학교 시간표 작성 자동화

김분홍, 이광호
목포대학교 공과대학 컴퓨터공학교육과

Automation of a University Timetable Scheduling based on Graph Coloring

BoonHong Kim and KwangHo Lee
Dept. of Computer Eng. Education, Col. of Engineering
Mokpo National University

요 약

이전 연구의 무작위 또는 순차 시간배정으로 인해 발생했던 중복 배정을 없애고, 보다 효율적인 프로그램의 구현을 위해 시간 배정에 앞서 같은 시간에 올 수 있는 과목들을 그룹화 한다. 본 논문은 그래프 착색 알고리즘과 시간표 작성 알고리즘의 관련성을 보이고, 같은 색으로 착색 가능한 시간들을 묶어 그룹화 함으로써 같은 시간대에 여러 과목을 배정하는 방법을 제안한다.

1. 서 론

매 학기가 시작될 때마다 시간표작성이 전산화되지 않아 수작업으로 해결하는 경우가 많다. 시간표 작성은 학생, 교수, 강의실을 서로 겹치지 않도록 배치하여 보다 효율적인 수업 장소와 시간을 할당하기 위한 것이다. 그러나 제한 조건이 많아 모든 조건을 만족하는 시간표를 얻는다는 것은 쉬운 일이 아니다. 제약조건을 모두 맞추려다 보면 모두 회피하는 시간대가 나올 수 있다. 예를 들어, 중식시간을 보장하거나 교수의 연구 시간을 보장하려는 경우이다.

다양한 요구를 맞추기 위해서는 나머지 시간에 최대한 모든 강의실과 교수 및 학생이 겹치지 않도록 과목을 배치하는 것이 필요하다. 이 문제에 있어서 본 연구에서는 같은 시간대에 개강할 수 있는 과목을 그룹화 하여 시간표 상에 같은 시간에 배치함으로써 사용할 시간대를 줄여 요구시간에 대한 만족도를 높이는 데 주안점을 둔다.

자동화 프로그램을 이용한 기존의 연구로는 경희대학교와 계명대학교의 유전자 알고리즘을 이용한 강의실 배치 [1], 서울시립대의 시간표 작성 프로그램 [2], 본 저자의 무제약 탐색 방법 [3] 등이 있었다. 본 연구에서는 그래프 이론 분야에서 연구된 그래프 착색 문제를 시간표 작성 문제에 적용하였다. 실험에서는 목포대학교의 컴퓨터 관련 학과들의 99년 가을학기의 실제 데이터를 이용한 그 구체적인 실행 결과를 보임으로써 타당성을 보인다.

2. 본 론

2.1 그래프 착색

비방향 그래프(undirected graph)의 착색 문제는 그래프의 각각의 정점(node)에 색을 부여하는 문제이다. 단, 인접한 정점들끼리는 서로 다른 색을 가져야 한다. 그래프의 최소 착색은 가능한 한 가장 적은 개수의 색을 사용하여 착색한 것을 일컫는다. 일반적으로 그래프 착색에서는 최소의 색으로 착색하는 해를 찾고자 노력하게 된다. 본 절에서는 구현된 세 가지의 착색 알고리즘에 대해 간단히 기술한다.

(a) Welch-Powell 알고리즘 [4] 에 의한 착색법

1) 그래프의 정점의 차수(degree)가 내림차순이 되게 배열한다 (이 배열은 차수가 같은 정점이 여러 개 있을 수 있으므로 몇 가지 다른 순서가 존재할 수 있다).

2) 배열의 첫 번째 정점은 첫 번째 색으로 착색하고 계속해서 배열의 순서대로 이미 착색된 정점과 인접하지 않은 정점을 모두 같은 색으로 착색한다.

3) 배열에서 먼저 나타나는 착색되지 않은 정점을 두 번째 색으로 착색하고, 계속해서 배열의 순서대로 지금 착색하고 있는 색으로 이미 착색된 정점과 인접하지 않은 정점을 모두 착색한다.

4) 계속해서 위의 과정을 그래프의 모든 정점이 착색될 때까지 반복한다.

(b) Trick 알고리즘 [5] 에 의한 착색법

1) 최대 크기의 클릭크(clique)를 다음과 같은 과정으로 찾는다. (클릭크는 정점들이 완전 연결된 그래프의 부분집합)

1.1) 정점의 차수에 따라 내림차순으로 정점들을 정렬한다.

1.2) 최대 차수를 갖는 정점을 포함하는 클릭크를 찾는다. (이후에 생성되는 다른 클릭크들은 여기에서 구한 클릭크의 크기와 비교하여, 크기가 더 클 때에만 최대 클릭크가 되기 위한 후보자가 된다)

1.3) 1.2에서 구한 클릭크에 포함되지 않은 정점들 각각에 대하여, 정렬된 정점들 중에서 그 정점을 마지막으로 포함하는 최대 클릭크를 찾는다 (단계1의 recursion). 각 정점들에 대해 이 과정을 수행하는 중에, 그때까지 발견된 클릭크의 최대 크기보다 큰 클릭크가 발견될 때마다 그 값을 가장 큰 클릭크로 수정한다.

2) 단계1에서 찾은 최대 클릭크의 크기가 n 이면, 클릭크를 구성하고 있는 각 정점에 색을 1부터 n 까지 착색한다.

3) 단계2에서 착색되지 않은 나머지 정점들에 대하여 다음과 같은 과정으로 착색한다.

3.1) 각 정점의 이웃 정점들에 착색된 색의 가짓수 (degree of colors) 에 따라 내림차순으로 정점들을 정렬한다.

3.2) 정렬된 정점의 순서대로 각 정점에 대해, 착색 제약조건 (인접한 정점들끼리는 같은 색을 가질 수 없다)을 만족하면서 칠할 수 있는 사용된 경험이 있는 색들 각각에 대하여 착색하고, 나머지 정점들에 대해서 착색 과정을 수행한다 (단계3의 recursion). 만일 기존의 색 집합 속에 착색 제약조건을 만족하는 색이 없다면 새로운 색을 생성하여 착색한 후, 나머지 정점들에 대해서 착색 과정을 수행한다 (단계3의 recursion). 이 과정을 수행하는 중에, 그

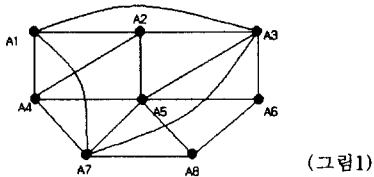
때까지 발견된 가장 적은 수의 색 개수보다 적은 수의 색 개수가 발견될 때마다 그 값을 가장 적은 착색수로 수정한다.

(c) W.-P. 알고리즘의 변형에 의한 착색법

W.-P. 알고리즘을 변형시킨 형태로 그래프의 정점의 차수를 비교해서 내림차순으로 정점을 하나씩 순차적으로 착색하는 방법으로, 이는 원래의 알고리즘이 병렬적인데 반해 순차적인 착색 방법이다.

- 1) 현재 정점을 무조건 첫 번째 색으로 착색한다.
- 2) 현재 색칠된 정점과 이웃하는 정점의 색과 비교한다.
- 3) 같은 경우에는 색의 순서에 따라 같은 색이 이웃하지 않도록 다시 착색한다.
- 4) 같지 않은 경우에는 색칠할 다음 정점으로 이동하고 위의 과정을 반복한다.

그림1의 그래프에 대한 세 가지 착색 결과는 다음과 같다.



· Welch-Powell 알고리즘으로 착색한 경우

정점을 내림차순으로 정리하면 A5, A3, A7, A1, A2, A4, A6, A8 이 된다. A5와 A1이 먼저 첫 번째 색으로 착색되고, 다음 A3, A1, A8이 두 번째 색으로 착색되며, 마지막으로 A7, A2, A6이 세 번째 색으로 착색된다. 따라서 3색 착색이 가능하다.

· Trick 알고리즘으로 착색한 경우

최대 크기의 정점들의 집합(클리크)으로 {A3, A5, A7} 이 구해지고, 이 들 정점들에 대해 첫 번째 색, 두 번째 색, 세 번째 색으로 착색한다. 나머지 정점들에 대해서는, 정점 A1, A2, A6, A4, A8 의 순서로 착색이 되며 각각 두 번째 색, 세 번째 색, 세 번째 색, 첫 번째 색, 첫 번째 색이 할당된다. 따라서 3색으로 착색이 가능하다.

· W.-P. 변형 알고리즘의 경우

정점을 내림차순으로 정리하면 A5, A3, A7, A1, A2, A4, A6, A8 이 되므로 A5를 먼저 첫 번째 색으로 칠한다. 다음, A3을 첫 번째 색으로 칠하고 이웃하는 정점에 첫 번째 색이 있으므로 두 번째 색을 칠한다. A7은 A5와 A3과 이웃하므로 세 번째 색을, A1은 이웃하는 정점에 첫 번째 색이 없으므로 그대로 첫 번째 색을 칠한다. A2는 A1, A3와 연결되어 있으므로 세 번째 색을, A4는 A1과 연결되어 있으므로 두 번째 색을, A6은 A5와 A3에 연결되어 있으므로 세 번째 색을, 그리고 A8은 A5와 연결되어 있으므로 두 번째 색을 칠한다.

2.2 그래프 착색과 시간표 작성 문제와의 관련성

시간표의 주요 요소인 과목명과 교수명, 대상학년의 3-tuple 을 하나의 정점으로 하여, 두 정점간의 교수명이 같거나 대상학년이 같은 경우에 두 정점을 edge로 연결하면 그래프가 형성된다. 이 그래프를 그래프 착색 알고리즘에 적용시키면 교수명과 대상학년이 모두 다른 정점들끼리 같은 색을 갖게

된다. 같은 색으로 착색된 정점은 inconsistent하지 않으므로 같은 시간대에 올 수 있다는 것을 의미하며, 이를 그룹화해서 같은 시간대에 배치하게 된다. 한 시간대에 배치 가능한 최대 과목 수는 그 학과의 교수의 수와 대상학년 수(보통 4 학년제이므로 학년 수는 4 임) 중 최소 값이 되기 마련이다.

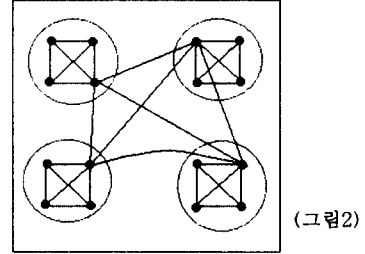


그림2에서 정점을 묶어둔 4개의 타원은 각각 학년을 의미하고, 실선으로 연결된 정점은 모두 한 명의 교수가 가르치는 과목을 연결한 것이다. 한 타원 안에는 같은 학년만 묶여 있기 때문에 서로 완전히 연결된 상태가 되고, 또 같은 교수의 과목은 모두 완전히 연결된 상태가 된다. 한 교수가 한 시간대에 두 번 있을 수 없으므로 만약 교수가 5명이면 최대 5개이지만 한 학년이 한번만 배치되므로 최대 4개가 된다. 반대로 학년을 한번씩 배치해서 4개지만, 교수의 수가 3명이라면 같은 시간대에 한 교수가 두 번 들어가지 못하므로 최대 3개 과목을 배치할 수 있는 것이다.

3. 구현 및 실험결과

3.1 구현 방법

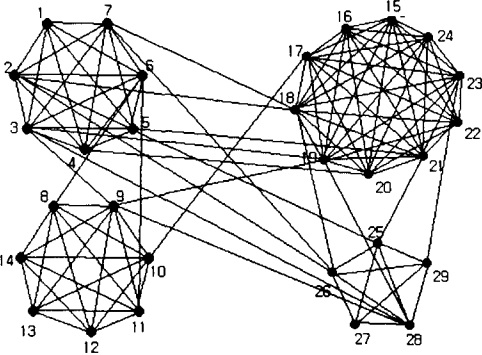
실험은 99학년도 2학기의 목포대학교 컴퓨터공학교육과의 교과과정을 데이터로 이용한 실험-1, 그리고 컴퓨터공학과와 합친 데이터를 이용한 실험-2 (이는 실제로 두 학과의 과목 운영이 마치 하나의 학과처럼 운영되는 특수성에 기인하며, 제약조건의 증가로 인하여 타 학과에 비해 조교가 수작업 시간표 작성 시 더 어려움을 겪는 이유가 되고 있다) 의 결과를 보인다. 한 개의 과목은 한 개의 정점에 해당하며, (과목, 교수명, 해당학년)의 레코드로 구성된다. 입력 확인은 정점의 수, 연결선의 수, 각 edge를 구성하는 정점 쌍들을 내용으로 한다. 이미 설명한 바와 같이, 그래프는 교수명이 같거나 학년이 같은 정점을 edge로 연결하여 준비되었다.

3.2 실험 결과

(a) 실험-1

| 과목명 | 교수명 | 학년 | 과목명 | 교수명 | 학년 |
|-----|------|----|-----|-----|----|
| 1 | 시간강사 | 1 | 15 | 김도윤 | 3 |
| 2 | 김동혁 | 1 | 17 | 안동순 | 3 |
| 3 | 이광호 | 1 | 18 | 김동혁 | 3 |
| 4 | 김종화 | 1 | 19 | 이광호 | 3 |
| 5 | 조성의 | 1 | 20 | 김종화 | 3 |
| 6 | 박효진 | 1 | 21 | 조성의 | 3 |
| 7 | 김동혁 | 1 | 22 | 서재현 | 3 |
| 8 | 이광호 | 2 | 23 | 박경우 | 3 |
| 9 | 박효진 | 2 | 24 | 김규형 | 3 |
| 10 | 안동순 | 2 | 25 | 조성의 | 4 |
| 11 | 박효진 | 2 | 26 | 김동혁 | 4 |
| 12 | 오수열 | 2 | 27 | 강성준 | 4 |
| 13 | 김민철 | 2 | 28 | 이광호 | 4 |
| 14 | 김종문 | 2 | 29 | 박경우 | 4 |
| 15 | 김원희 | 3 | | | |

· 그래프



· 그룹화 결과

W.-P.의 변형 Trick W.-P.

| 분류 | 해당과목 | 분류 | 해당과목 | 분류 | 해당과목 |
|----|---------------|----|------------|----|---------------|
| 1 | 18, 3, 8, 25 | 1 | 4, 15 | 1 | 18, 9, 5, 27 |
| 2 | 19, 2, 10, 27 | 2 | 1, 14, 16 | 2 | 19, 7, 10, 25 |
| 3 | 21, 9, 6, 26 | 3 | 5, 13, 17 | 3 | 21, 3, 11, 26 |
| 4 | 17, 5, 11, 28 | 4 | 6, 12, 18 | 4 | 20, 2, 8, 28 |
| 5 | 20, 7, 12, 29 | 5 | 11, 19, 26 | 5 | 17, 6, 12, 29 |
| 6 | 15, 4, 13 | 6 | 7, 10, 20 | 6 | 15, 4, 13 |
| 7 | 16, 1, 14 | 7 | 2, 21, 29 | 7 | 16, 1, 14 |
| 8 | 22 | 8 | 8, 22, 28 | 8 | 22 |
| 9 | 23 | 9 | 9, 23, 27 | 9 | 23 |
| 10 | 24 | 10 | 3, 24, 25 | 10 | 24 |

(b) 실험-2

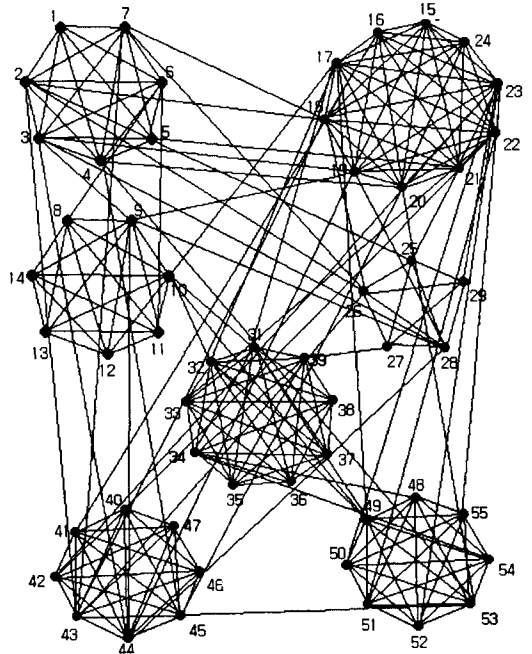
| 과목명 | 교수명 | 학년 | 과목명 | 교수명 | 학년 |
|-----|------|----|-----|-----|----|
| 1 | 시간강사 | 1 | 16 | 김도윤 | 3 |
| 2 | 김동혁 | 1 | 17 | 안동순 | 3 |
| 3 | 이광호 | 1 | 18 | 김동혁 | 3 |
| 4 | 김종화 | 1 | 19 | 이광호 | 3 |
| 5 | 조성익 | 1 | 20 | 김종화 | 3 |
| 6 | 박호진 | 1 | 21 | 조성익 | 3 |
| 7 | 김동혁 | 1 | 22 | 서재현 | 3 |
| 8 | 이광호 | 2 | 23 | 박경우 | 3 |
| 9 | 박호진 | 2 | 24 | 김규형 | 3 |
| 10 | 안동순 | 2 | 25 | 조성익 | 4 |
| 11 | 박호진 | 2 | 26 | 김동혁 | 4 |
| 12 | 오수열 | 2 | 27 | 강성준 | 4 |
| 13 | 김민철 | 2 | 28 | 이광호 | 4 |
| 14 | 김정준 | 2 | 29 | 박경우 | 4 |
| 15 | 김양희 | 3 | | | |

| 과목명 | 교수명 | 학년 | 과목명 | 교수명 | 학년 |
|-----|-----|----|-----|-----|----|
| 31 | 안동순 | 2 | 44 | 이광호 | 3 |
| 32 | 안동순 | 2 | 45 | 김종화 | 3 |
| 33 | 서재현 | 2 | 46 | 박경우 | 3 |
| 34 | 오수열 | 2 | 47 | 박진영 | 3 |
| 35 | 임현 | 2 | 48 | 오수열 | 4 |
| 36 | 최성욱 | 2 | 49 | 안동순 | 4 |
| 37 | 김희봉 | 2 | 50 | 박경우 | 4 |
| 38 | 이진 | 2 | 51 | 서재현 | 4 |
| 39 | 강성준 | 2 | 52 | 박경우 | 4 |
| 40 | 서재현 | 3 | 53 | 김종화 | 4 |
| 41 | 김도윤 | 3 | 54 | 오수열 | 4 |
| 42 | 박경우 | 3 | 55 | 박경우 | 4 |
| 43 | 김동혁 | 3 | | | |

4. 결론

시간표 작성에 착색 알고리즘을 이용함으로써 보다 적은 시간대에 많은 과목들을 배치하고, 그룹화시킨 과목 중에서도 학생이나 교수의 요구에 따라 해당과목을 따로 떼어내어 남은 시간대에 배정하거나 원하지 않은 시간대를 피해 조정할 수 있는 방법을 제시하였으며, 교양이나 계열기초같이 중앙에서 미리 정해진 과목은 학과에서 다른 과목들과 같이 섞어

· 그래프



· 그룹화 결과

W.-P.의 변형 Trick W.-P.

| 분류 | 해당과목 | 분류 | 해당과목 | 분류 | 해당과목 |
|----|---------------------------|----|-----------------------|----|---------------------------|
| 1 | 23, 31, 43, 3, 51, 8, 25 | 1 | 4, 11, 15, 37, 46 | 1 | 23, 31, 43, 3, 51, 8, 25 |
| 2 | 17, 33, 42, 9, 2, 53, 27 | 2 | 3, 8, 16, 36, 54 | 2 | 18, 32, 42, 9, 4, 48, 27 |
| 3 | 18, 32, 44, 50, 4, 11 | 3 | 17, 27, 35, 47, 48 | 3 | 19, 33, 46, 10, 7, 53 |
| 4 | 19, 46, 10, 34, 7 | 4 | 18, 25, 39, 41, 53 | 4 | 17, 44, 34, 50, 2, 11 |
| 5 | 20, 40, 49, 29, 39, 5, 12 | 5 | 19, 26, 34, 45, 52 | 5 | 22, 45, 49, 39, 29, 6, 12 |
| 6 | 22, 45, 52, 6, 26, 35, 13 | 6 | 1, 14, 20, 28, 42, 49 | 6 | 20, 40, 52, 35, 5, 26, 13 |
| 7 | 21, 28, 36, 41, 48, 1, 14 | 7 | 6, 13, 21, 33, 44, 50 | 7 | 21, 36, 41, 28, 54, 1, 14 |
| 8 | 16, 37, 47, 54 | 8 | 5, 10, 22, 38, 43 | 8 | 16, 37, 47 |
| 9 | 15, 38 | 9 | 7, 9, 23, 32, 40 | 9 | 15, 38 |
| 10 | 24 | 10 | 2, 12, 24, 29, 31, 51 | 10 | 24 |

그룹화시킨 다음 미리 정해진 시간대에 배정함으로써 해결하였다. 추후 연구과제는 수용 가능한 강의실과의 관계를 고려하여, 수강인원이 많은 과목이 한 시간대에 한꺼번에 그룹화 되었을 때 모두 수용할 강의실이 없다면 다시 재조정해주는 문제, 그리고 그룹화된 과목 중 3시간 강의의 경우, 2시간 강의에 2시간 실험의 경우, 또는 2시간 강의만 있는 경우 등 수업 시수에 따른 배정 문제 등이다.

참고문헌

- [1] 정태춘, "종합강의시간표 자동작성 시스템 개발연구 과제 보고서," 경희대학교 전자계산공학과, 1996.
- [2] "서울시립대 1998년 2학기 종합강의시간표 자료제출안내서," 서울시립대학교 교무과 수업계, 1998.
- [3] 김분홍, 이광호, "목포대학교 시간표작성 프로그램 개발을 위한 기초실험," 1999년 4월, 한국정보과학회 '99봄 학술발표논문집(B), 목포대학교, 제26권 1호, 310-312쪽.
- [4] 황종선, 박두순, "이산수학개론," 1998, 정익사, 278-280쪽.
- [5] A. Mehrotra and M.A. Trick, "A Column Generation Approach For Graph Coloring," 1995, Technical Report, Carnegie Mellon Univ., April 11, pp.1-20.