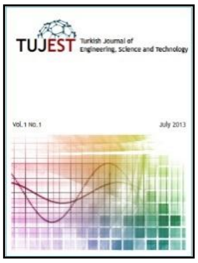
Turkish Journal of Engineering, Science and Technology 02 (2013) p32-35

Turkish Journal of Engineering, Science and Technology

journal homepage: www.tujest.com



A Graph-Based Solution to University Timetabling

Mohammad Hadi Alaeiyan[a,*]

[a,] *Assistant Department of Computer Engineering, Iran University of Science and Technology, Narmak, Tehran, 16844, Iran*

Article Info

Article history:
Received 21 August 2013
Accepted 26 August 2013
Available online 01 September 2013

Keywords:
Best Fit Decreasing (BFD) method,
Overlap graph,
K-coloring

Abstract

Time-tabling can be considered to be a certain type of scheduling problem. There is a wide variety of time-tabling problems one of which is known as university time-tabling. In this kind of problem, there are a lot of university courses requiring scheduling while avoiding overlap in any particular term. There are several approaches to solving such problems. This article focuses on a graph-based solution. By modeling a graph-based solution, we are able to gain a coloring problem which can then be used to obtain a complete solution to the problem through SFSG: SDO proper coloring algorithm, followed by BFD method.

1. Introduction

To clarify the issue, we start with fundamental concepts of graph theory. A graph $G = (V_{(G)}, E_{(G)})$ is a finite nonempty set $V_{(G)}$ of objects called vertices (also called points or nodes) and a (possibly empty) set $E_{(G)}$ of 2-element subsets of $V_{(G)}$ called edges (or lines). The set $V_{(G)}$ is called the vertex set of G and $E_{(G)}$ its edge set.

Let G be a graph and $\{u, v\}$ an edge of G. Since $\{u, v\}$ is a 2-element set, we may write $\{v, u\}$ instead of $\{u, v\}$. It is often more convenient to represent this edge by uv (or vu).

If $e = uv$ is an edge of a graph G, then we say that u and v are adjacent in G, and that e joins u and v. (We may also say that each of u and v is adjacent to or with the other.)

For example, a graph G is defined by the sets

$$V_{(G)} = \{u, v, w, x, y, z\}$$

and

$$E_{(G)} = \{uv, uw, wx, xy, xz\}$$

Every graph has a diagram associated with it. These diagrams are useful for understanding problems involving such a graph. We represent the vertices by means of points (actually small circles) and join two points by a line (segment or curve) whenever the corresponding pair of vertices forms an edge. Hence, Figure 1 is a diagram of the graph in the above stated example. It is often convenient to refer to a diagram of a graph as the graph itself [4].

A k-coloring of a graph G is a labeling $f: V_{(G)} \rightarrow S$, where $|S| = k$ (often we use $S = [k]$). The labels are colors; the vertices of one color form a color class. A k-coloring is proper if adjacent vertices have different labels. A graph is k-colorable if it has a proper k-coloring. The chromatic number $X_{(G)}$ is the least k such that G is

*Corresponding Author:
M. H. Alaeiyan, E-mail: hadi_alaeiyan@comp.i

k-colorable [7].

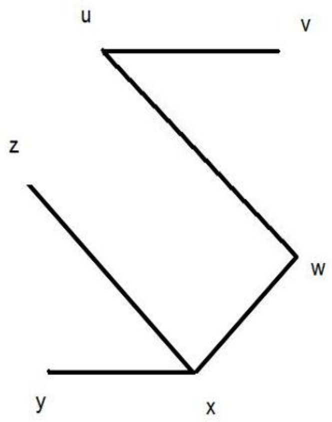


Figure 1. The diagram of graph G

Suppose we want to present n different courses in a certain term, where $n \epsilon \mathbb{N}$. We show these courses with n vertices of a graph, such as $G = (V, E)$. Therefore, the vertex of v_i from G represents the i-th course. So, we will have $\mathrm{V}_{(\mathrm{G})} = \{v1, v2, \dots, vn\}$, which $\mathrm{V}_{(\mathrm{G})}$ is vertex set of G.

Also we define the edge set as:

$v_i \leftrightarrow v_j$ <===> course i and course j cannot present simultaneously.

This definition specifies $\mathrm{E}_{(\mathrm{G})}$ – edge set – completely. We call G overlap graph. It can be easily shown that solution of the real problem is equal to finding a proper coloring for G. Burke and Werra introduced an inventive method for solving of coloring problem in their papers [1,6], which is suitable for timetabling. Also Carter and Laporte in their papers [3] presented creative ways for solution of coloring problem. Examples of common graph coloring /time-tabling heuristics, which have been used in [2], are:

- Largest degree first: prioritizes the events that have a large number of conflicts with other events
- Largest weighted degree: takes into account the conflicts of events with other events weighted by the number students involved.
- Saturation degree: schedules firstly those events that have the smallest number of valid periods available for scheduling in the time-table constructed so far.
- Coloring degree: orders the events by the number of conflicts that an event has with the events that have already been scheduled.

Although each proper coloring of G will be a solution, we are seeking a minimum proper coloring. It helps us in optimizing uses of facilities, such as rooms and laboratories.

There are several algorithms for coloring a graph, the most being Greedy algorithms. Some of well-known Greedy algorithms are:

- **Simple-Search-Greedy (SSG):** this method chooses the color classes from a normal-sized class that is defined in advance.
- **Largest-First-Search-Greedy (LFSG):** by which color-classes are ordered according to their size in a decreasing order. This method tends to produce classes with the largest size.
- **Smallest-First-Search-Greedy (SFSG):** this method orders color-classes according to increasing order. Therefore, it has a tendency to make a flat distribution of vertices in color-classes, meaning that a color-class is defined as a subset of $\mathrm{V}_{(\mathrm{G})}$ including all of vertices with a certain color when our coloring is a proper coloring.

Each of these three Greedy methods, combining vertex ordered techniques, makes a whole Greedy algorithm for graph coloring. Some of the most important vertex ordered techniques are:

- Decreasing Degree Ordering: DDO
- Increasing Degree Ordering: IDO
- Saturation Degree Ordering: SDO

Saturation Degree of a vertex is the number of spent colors in adjacent vertices. This method chooses the vertex with maximum colors in its adjacent vertices to color.

As mentioned before, our overlap graph is produced by a real problem, university timetabling. Because of specific characteristics of a real problem, we will also have some limitations. Some of these restrictions are critical, denoted as hard constraints, while other limitations which are not so essential are dented as feasible solutions.

Common hard constraints include:

- No person can be allocated to more than one place at any one time.
- The total resources required in each time period must not be greater than the resources that are available.

Soft constraints are those that it is desirable to satisfy, but they are not essential. In real-world university time-tabling problems, it is usually impossible to satisfy all of the soft constraints. The quality of a feasible time-table can be assessed based on how well the soft constraints are satisfied. However, some problems are so complex that it is difficult to find even a feasible solution.

Some of feasible solutions are:

- Desire to distribute courses uniformly in hours of weekdays.
- It will be preferable for each professor to attend their classes as near as possible to their office.

Comprehensive overviews of different constraints that are imposed by universities are given in [1,3].

2. BFD Method

The Below, presents an overlap graph for a real problem.

Consider we want to present these four courses in a certain term: Physics I, Chemistry II, Chemistry-lab II and Physics II. We have these restrictions:

- The professors of Physics I and Physics II are the same.
- Chemistry II and Chemistry-lab II must be taken simultaneously.
- Physics II and Chemistry II must be presented in room 102.

We consider one vertex for each course and build overlap graph considering the above restrictions. Each restriction produces one edge. Figure 2 show overlap graph considering the above restrictions.

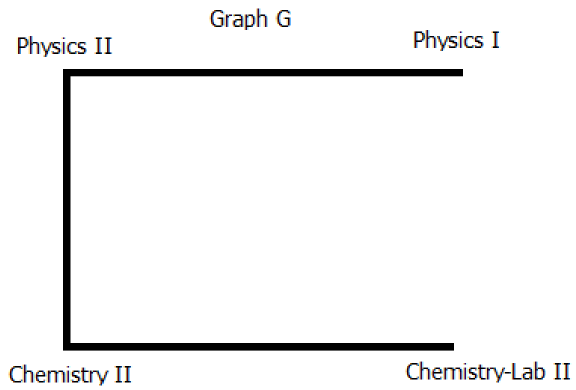


Figure 2. Overlap graph considering the above restrictions.

The solution of the real problem is equal to finding a proper coloring for G, and because we want to use minimum facilities, using minimum colors will be the best approach. Here we introduce a SFSG: SDO proper coloring for a certain graph, such as above overlap graph G.

Repeat until:

All of graph vertices are colored.

Choose the next vertex V to color, such as V which has maximum various colors on its adjacent vertices. If there is more than one proper vertex, choose the one with the maximum number of its adjacent vertices uncolored. If there is more than one proper vertex again, choose one of them randomly. For coloring, choose one of the colors that you have used before but with minimum frequency. If none of them is proper, color the vertex with a new color.

End Repeat:

After coloring the overlap graph, we have to designate courses with a same color to available rooms. We do this by using Best Fit Decreasing (BFD) method.

For a certain time period, like T we name the number of courses N and the number of rooms M. When $N \leq M$ we define:

$$S_{(ci)} = \text{the number of students in course } i.$$

$$S_{(rj)} = \text{the capacity of room } j.$$

We sort $S_{(ci)}$s and $S_{(rj)}$s in decreasing order:

$$S_{(c1)} \geq S_{(c2)} \geq \cdots \geq S_{(cn)}$$

$$S_{(r1)} \geq S_{(r2)} \geq \cdots \geq S_{(rm)}$$

We then assume:

$$S_{(r1)} \geq S_{(c1)} \;\&\; S_{(rm)} \geq S_{(cn)}$$

Now, we implement BFD method, starting with the largest course, we consider course c_i to room r_k such as $S_{(rk)} - S_{(ci)} \geq 0$ and $S_{(rk)} - S_{(ci)}$ is minimum.

These conditions always don't result in the best arrangement. The following list illustrates the matter more clearly:

i	$S_{(ci)}$		$S_{(rk)}$
	.		.
	.		.
j	120		150
		X	
$j+1$	100		125
	.		95

Better allocation would be to allocate the room with the capacity of 150 students to the course attended by 120 students, and the room which can hold of 125 students to the course with 100 students. When the lines cross, such as this example, it is enough to change $S_{(cj)}$ and $S_{(cj+1)}$

By repeating BFD method for each color-class finalizes our term schedule.

3. Conclusions

There is a wide variety of time-tabling problems, one of which is known as university time-tabling. In this kind of problem, there are a lot of university courses requiring scheduling while avoiding overlap in any particular term. There are several approaches to solving such problems. We focuses on a graph-based solution. We are able to gain a coloring problem which can then be used to obtain a complete solution to the problem through SFSG: SDO proper coloring algorithm, followed by BFD method.

References

[1] Burke E.K, et al (2007). A graph-based hyper-heuristic for educational timetabling problems. European Journal of Operational Research. 176: 192-177.

[2] Burke E.K, Petrovice S (2002). Recent research directions in automated timetabling. European Journal of Operational Research. 140: 280-266.

[3] Carter M W, Laporte G (1998). Recent developments in practical course timetabling. Second International Conference on Practice and Theory of Automated Timetabling (PATAT 1997), LNCS,Springer, Toronto, Vol. 1408, p. 19-3.

[4] Chartrand G, Oellermann O.R (1993). Applied and algorithmic graph theory. McGraw-Hill, New York, USA.

[5] Michael W. Carter, Gilbert Laporte (1996). Recent developments in practical examination timetabling. The Practice and Theory of Automated Timetabling. First International Conference Edinburgh, U.K., August 29–September 1. p. 21-1.

[6] Werra D (1985). An introduction to timetabling. European Journal of Operational Research. 19(2): 162-151.

[7] West D.B (2002). Introduction to graph theory, 2th ed. Indian Branch, Delhi, Indian.