

BÀI TOÁN TÔ MÀU ĐỒ THỊ VÀ ỨNG DỤNG XÂY DỰNG PHẦN MỀM XẾP LỊCH THI CHO HỌC CHẾ TÍN CHỈ

THE PROBLEM OF GRAPH COLORING AND ITS APPLICATION TO THE
DEVELOPMENT OF AN EXAMINATION SCHEDULE SOFTWARE FOR
CREDIT-BASED ACADEMIC COURSES

Trần Quốc Chiến

Trường Đại học Sư phạm, ĐH Đà Nẵng

Phan Thị Ngà

Trường Đại học Thể dục Thể thao Đà Nẵng

TÓM TẮT

Với mô hình đào tạo mới theo học chế tín chỉ, bài toán xếp lịch thi cũng có nhiều yêu cầu mới khác với bài toán lập lịch cổ truyền. Ứng dụng thuật toán tô màu đồ thị vào bài toán lập lịch được coi là một giải thuật tối ưu cổ điển, thì với yêu cầu lập lịch thi cho học chế tín chỉ cần phải cải tiến lại giải thuật cho phù hợp với các yêu cầu ràng buộc mới.

Đề tài tập trung nghiên cứu về lý thuyết đồ thị và bài toán tô màu, tìm hiểu về học chế tín chỉ. Ứng dụng giải thuật tô màu đồ thị để đề ra giải pháp, thuật toán cho bài toán xếp lịch thi cho học chế tín chỉ. Xây dựng, thiết kế phần mềm xếp lịch thi cho học chế tín chỉ.

ABSTRACT

With new credit-based academic programmes, the math problem of an examination schedule has a number of new requisites that differ from those of a traditional examination schedule. While the application of algorithms to colored graphs in the math problem of examination schedule is considered to be a classic optimum, the schedule for credit-based programme examinations needs improved algorithms in accordance with new constraints.

This topic focuses on the graph theory, the problem of coloring, the credit-based programme investigation, the application of algorithms to colored graphs in the math problem solving, the algorithms for the problem of examination schedule for credit-based academic programmes and the development and design of a software for credit-based examination schedule.

1. Đặt vấn đề

Hiện nay sự phát triển của các thuật toán trên đồ thị là một trong các mối quan tâm chính của ngành khoa học máy tính. Đề án đổi mới giáo dục đại học Việt Nam đang được thực thi, một trong những nội dung quan trọng là xây dựng mô hình đào tạo ở bậc đại học theo học chế tín chỉ. Quản lý và xếp lịch thi cho học chế tín chỉ vẫn luôn là sự quan tâm của nhiều trường đại học. Các phần mềm về quản lý đào tạo theo học chế tín chỉ đã có đều là phần mềm đã đóng gói. Vấn đề nghiên cứu về quản lý học chế tín chỉ đang được phát triển. Hướng nghiên cứu và kết quả của đề tài nhằm đóng góp một phần vào việc đưa ra giải pháp và thuật toán để xây dựng phần mềm xếp lịch thi cho học chế tín chỉ.

2. Bài toán xếp lịch thi cho học chế tín chỉ.

Các trường đại học tổ chức học theo tín chỉ. Nếu sinh viên tích lũy đủ số chứng

chỉ cho một số môn quy định của một ngành là có quyền nhận bằng tốt nghiệp của ngành đó. Đối với các đại học như thế, việc học và thi không tổ chức theo lớp mà theo các môn học. Hàng năm nhà trường thông báo các môn sẽ học để sinh viên tự đăng ký học các môn học theo ngành mình chọn. Cuối kỳ hoặc cuối năm nhà trường tổ chức thi cho các môn đã giảng trong năm. Mỗi môn thi trong một ngày nhưng trong một ngày có thể tổ chức thi nhiều môn. Do một sinh viên có thể đăng ký thi nhiều môn nên lịch thi cần phải bố trí để nếu có một sinh viên đăng ký thi nhiều môn nào đó thì các môn đó không được thi cùng ngày.

Để sinh viên có thời gian ôn tập, lịch thi sẽ được xếp theo sao cho: nếu sinh viên thi nhiều môn thì điều kiện là hai môn thi kế tiếp phải cách nhau tối thiểu là $nDay$ ngày.

Mỗi đợt thi chỉ giới hạn trong một khoảng thời gian nhất định. Trong một ngày, một phòng có thể tổ chức nhiều ca thi.

3. Xây dựng giải thuật cho bài toán

Đầu vào:

- + *dsInpMHoc*: là danh sách các môn học
- + *dsInpDK*: là danh sách đặc tả mối quan hệ sinh viên đăng ký dự thi môn học nào
- + *dsInpPHoc*: là danh sách các phòng học, tương ứng với thông tin về số hiệu phòng, sức chứa của phòng
- + *dtiBegin*: là ngày bắt đầu tổ chức thi
- + *nDay*: là số lượng ngày tối thiểu tương ứng với khoảng cách hai lần thi của một thí sinh tương ứng
- + *numOfCathi*: là số lượng ca thi tổ chức trong ngày, mặc định nhận giá trị là 1 hoặc 2 ca thi/1 ngày
- + *numOfRoom*: là số lượng thí sinh tối thiểu. Trong trường hợp số lượng thí sinh không vượt quá *numOfRoom* là không tổ chức thi.

Đầu ra:

- + *numOfSubject*: là số lượng môn học được tổ chức thi, nghĩa là có thể có một số môn học trong danh sách các môn học không đủ thí sinh để tổ chức thi, nên sẽ không được phép tổ chức thi
- + *numOfDate*: là số lượng ngày cần thiết tối ưu để tổ chức đợt thi tương ứng với danh sách các môn học đó
- + *dtiBegin* và *dtiEndDate*: tương ứng là ngày bắt đầu tổ chức thi và ngày kết thúc đợt thi. Và *lstDate*: là danh sách các ngày thi tương ứng sẽ tổ chức thi cho các môn học với số lượng ca thi trong một ngày cho trước.
- + *dsLichThi*: là danh sách lịch thi tương ứng với thông tin ngày thi với từng ca thi sẽ tổ chức thi ở các phòng thi tương ứng với môn thi và số lượng thí sinh dự thi.

CÁC BƯỚC GIẢI THUẬT

Bước 1: Định nghĩa đồ thị với tập đỉnh là các môn học đủ, mối quan hệ giữa các đỉnh của đồ thị dựa vào thông tin đăng ký dự thi của thí sinh đầu vào.

- $n = dsInpMHoc.Tables[0].Rows.Count$ là số lượng đỉnh của đồ thị trong giải thuật lập lịch.
- $arrSubjectColor[i]$: đặc tả thông tin đỉnh thứ i của đồ thị tương ứng với môn học i sẽ được tô với màu là $arrSubjectColor[i]$. Nếu $arrSubjectColor[i]$ nhận giá trị là 0 tức là môn học thứ i chưa được tô màu.
- $graph[i][j]$: thể hiện mối quan hệ giữa môn học thứ i và môn học thứ j , trong đó: i và j thể hiện môn học có sinh viên thi cả hai môn.

Bước 2: Áp dụng bài toán tô màu trong lập lịch

- Bắt đầu với màu $colorCurrent$ được khởi tạo bằng 0.
- Lặp lại cho đến khi giải thuật tô màu kết thúc

while (true)

1: Tìm đỉnh có bậc lớn nhất chưa được tô màu

int iHasMaxLevelCurrent = findMaxLevelCurrent();

2: Nếu không tìm được đỉnh nào chưa tô có bậc nhỏ hơn 0 thì thuật toán dừng

if (iHasMaxLevelCurrent < 0)

break;

3: Ngược lại thì tô màu đỉnh này

$colorCurrent++$;

$arrSubjectColor[iHasMaxLevelCurrent] = colorCurrent$;

4: Tô màu các đỉnh không kề với đỉnh iHasMaxLevelCurrent với cùng màu $colorCurrent$

//Lặp lại cho đến khi không tìm thấy đỉnh nào không kề với đỉnh iHasMaxLevelCurrent nữa thì giải thuật quay lại 1:

while (true)

{

Tìm đỉnh v có số bậc lớn nhất chưa được tô

int v = findVOptimized(iHasMaxLevelCurrent, colorCurrent);

if (v < 0)

break;

Tô màu đỉnh v với màu hiện hành

$arrSubjectColor[v] = colorCurrent$;

}

5: Xoá mối quan hệ tương ứng các đỉnh đã tô màu ra khỏi đồ thị

for (**int** i = 0; i < n; i++)

for (**int** j = 0; j < n; j++)

$graph[i][j] = 0$;

Trong bước 2 này, đã thiết kế một số hàm đặc trưng cho một số chức năng chuyên biệt, xem chi tiết trong code chương trình minh họa, cụ thể là hàm:

+ Hàm *findMaxLevelCurrent()*: đảm nhận chức năng tìm đỉnh có số bậc lớn nhất trong đồ thị cho đến thời điểm hiện tại của vòng lặp, nhưng chưa được tô màu.

+ Hàm *findVOptimized(iHasMaxLevelCurrent, colorCurrent)*: đảm nhận chức năng tìm đỉnh v để tô màu thỏa mãn ràng buộc v đi đến *iHasMaxLevelCurrent* thông qua một đỉnh duy nhất và v có bậc lớn nhất trong số các đỉnh có cùng tính chất hoặc đỉnh v có số bậc lớn nhất không kề với đỉnh *iHasMaxLevelCurrent* trong số tập các đỉnh không kề với nó.

Bước 3: Lập lịch thi – dựa trên các đợt thi tương ứng với từng màu thu được.

- Gọi hàm kiểm tra tùy chọn số ca thi trên 1 ngày
- Sắp xếp các ca thi đã thực hiện ở bước trên tương ứng vào các ngày thi để tổ chức thi

1: Lấy về số lượng ngày cần thiết dùng để tổ chức thi

```
int numOfDateNeed = getNumOfRoomForSubject(indexCathi, numOfCathi);
```

```
CS.clsNgayThi[] lstNgayThi =
```

```
new vCheduler.CS.clsNgayThi[numOfDateNeed];
```

2: Sắp xếp các ca thi vào các ngày thi tương ứng

Bảo đảm điều kiện hai môn thi liên tiếp của một thí sinh cách nhau nDay ngày

```
for (int iDate = 0; iDate < numOfDateNeed; iDate++)
```

```
for ( i=0; i<m; i++)// duyệt danh sách sinh viên
```

```
for ( j=0; j<n-1 ; j++)// duyệt danh sách môn thi
```

```
for (k=j+1; k<n; k++)// duyệt danh sách 2 ngày thi liên tiếp
```

```
if (sv[i].DayOfxam[j] - sv[i].DayOfxam[k] < nDay)
```

```
{ DayOfxam = DayOfxam+nDay - abs(sv[i].DayOfxam[j] -
sv[j].DayOfxam[k])
```

```
Gọi thủ tục sắp xếp ca thi
```

```
}
```

Bước 4. Hiện thị kết quả xếp lịch thi

- Chuẩn hóa tham số đầu ra: *numOfSubject, numOfDate*

Gọi hàm lấy ngày thi

- Chuẩn hóa danh sách các ngày thi tương ứng với lịch thi thỏa mãn ràng buộc số lượng ngày tối thiểu giữa hai lần thi là nDay cho trước.

```
for (int i = 1; i < numOfDate; i++)
```

In lịch thi

- Chuẩn hóa dữ liệu thành lịch thi tương ứng

```
for (int i = 0; i < numOfDate; i++)
```

```
{
```

```

string ngaythi = lstDate[i];
for (int iCathi = 0; iCathi < numOfCaThi; iCathi++)
    CS.clsCaThi
    CS.clsPhongThi
}

```

4. Thiết kế xây dựng phần mềm xếp lịch thi cho học chế tín chỉ

Sử dụng hệ quản trị cơ sở dữ liệu SQL Server 2005, ngôn ngữ lập trình C# trong bộ Visual studio 2005. Demo phần mềm đã cài đặt chạy ổn định với giao diện dễ sử dụng. Có các modul chức năng như: Quản lý sinh viên học tín chỉ, quản lý phòng học, môn học, chức năng lập lịch, tìm kiếm môn học, tìm kiếm sinh viên học tín chỉ, tra cứu lịch thi. Trong đó chức năng lập lịch, kết quả lập lịch là chức năng trọng tâm nhất của phần mềm có giao diện chính như sau:

CHỨC NĂNG LẬP LỊCH THI

Tiền xử lý Quay về

1. Danh sách Môn học 2. Thông tin Đăng ký dự 3. Danh sách các Phòng học

Xác nhận Chọn tất cả Chọn các Môn học thi trong đợt này

Chọn	Mã Môn học	Tên Môn học	Số lượng tín chỉ	Mô tả thêm
<input checked="" type="checkbox"/>	CNPM01	Công nghệ Phần mềm	4	
<input checked="" type="checkbox"/>	DB004	Cơ sở dữ liệu Nâng cao	4	
<input type="checkbox"/>	GT001	Toán giải tích	3	
<input checked="" type="checkbox"/>	HCM001	Tư tưởng Hồ Chí Minh	3	
<input checked="" type="checkbox"/>	KTCT01	Kinh tế chính trị	3	
<input type="checkbox"/>	LSD001	Lịch sử Đảng	3	
<input checked="" type="checkbox"/>	MA0001	Mạng và kỹ thuật truyền số liệu	5	
<input checked="" type="checkbox"/>	TMNT01	Trí tuệ thông minh nhân tạo	5	
<input checked="" type="checkbox"/>	TRR001	Toán rời rạc	3	
<input type="checkbox"/>	TTNC01	Cấu trúc giải thuật nâng cao	4	

Thông tin Lập lịch thi

Thông tin Số Ca thi tổ chức trong Ngày: ☐ 1 Ca thi/ 1 Ngày ☒ 2 Ca thi/ 1 Ngày

Thông tin lập lịch thi: Số lượng thí sinh tối thiểu: / 1 phòng thi

Từ ngày: 05/05/2009

Lập lịch

KẾT QUẢ LẬP LỊCH THI

Quay về

Thông tin thống kê

Số lượng môn: 15

Số lượng Phòng: 10

Sức chứa: 54

Tổng kết lịch thi

Tổng số ngày tổ chức thi: 9

Số ca thi trong ngày: 2

Ngày bắt đầu: 05/05/2009 Ngày kết thúc: 13/05/2009

5. Kết luận

Nội dung đề tài khái quát được các kiến thức chung về lý thuyết đồ thị và ứng dụng. Nêu được chi tiết bài toán tô màu và thuật toán tô màu trên đồ thị. Phân tích các yêu cầu của bài toán lập lịch thi cho học chế tín chỉ, thiết kế cơ sở dữ liệu, áp dụng giải thuật ô màu cổ điển để xây dựng giải thuật cải tiến cho yêu cầu mới của bài toán đề ra.

Qua các bước xây dựng thuật toán, thiết kế cơ sở dữ liệu, code chương trình đã được cài đặt, xây dựng được demo phần mềm kiểm định thuật toán cho kết quả khả quan và mang tính ứng thực tế.

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Thiên Bằng, Phương Lan, *Khám phá SQL Server 2005*, Nhà Xuất bản Lao động Xã hội, 2006.
- [2] PGS.TSKH.Trần Quốc Chiến, *Giáo trình Lý thuyết đồ thị và ứng dụng*, giáo trình lưu hành nội bộ, Đà Nẵng 2007.
- [3] Nguyễn Ngọc Bình Phương – Thái Thanh Long, *Các giải pháp lập trình C#*, Nhà xuất bản Giao thông Vận tải, năm 2006.
- [4] Phạm Đức Tuấn, *Xây dựng hệ thống thông tin quản lý đào tạo theo học chế tín chỉ*, Luận văn Thạc sĩ Kỹ thuật, Đại học Đà Nẵng, 2006.
- [5] Các văn bản về mô hình đào tạo theo học chế tín chỉ của Bộ Giáo dục và Đào tạo.