

XÂY DỰNG PHÂN HỆ XẾP LỊCH THI VÀ TÍCH HỢP VÀO HỆ THỐNG MOODLE TẠI TRƯỜNG ĐẠI HỌC KINH TẾ, ĐẠI HỌC ĐÀ NẴNG

DEVELOPING THE EXAM TIME TABLE PLUGIN AND INTEGRATING IT INTO MOODLE AT THE UNIVERSITY OF DANANG, UNIVERSITY OF ECONOMICS

Vũ Hà Tuấn Anh

Trường Đại học Kinh tế, Đại học Đà Nẵng; anh.vht@due.edu.vn

Tóm tắt - Xếp lịch thi là một phần công việc quan trọng mà các cơ sở đào tạo luôn phải làm trước khi kết thúc học kỳ. Việc tạo ra một công cụ cho phép lập lịch thi học kỳ rất cần thiết cho các cơ sở đào tạo, hơn nữa lập lịch thi học kỳ được tích hợp vào hệ thống Moodle rất được quan tâm từ nhiều trường đại học, bởi Moodle là một hệ thống quản lý học tập nguồn mở đang được rất nhiều trường học áp dụng hiện nay. Bài báo này trình bày kết quả đã được xây dựng và phát triển một phân hệ xếp lịch thi được thực hiện trên hệ thống Moodle, ứng dụng cụ thể tại Trường Đại học Kinh tế, Đại học Đà Nẵng. Phân hệ xếp lịch thi này có sử dụng giải thuật backtracking với một số kỹ thuật kiểm tra trước để tự động xếp lịch thi. Kết quả thực hiện cho thấy khả năng ứng dụng của phân hệ xếp lịch thi là hữu ích và có khả năng phát triển các chức năng khác trên hệ thống quản lý học tập Moodle.

Từ khóa - xếp lịch thi; phân hệ xếp lịch thi; hệ thống quản lý học tập; module tích hợp; Moodle; giải thuật backtracking.

1. Đặt vấn đề

Ứng dụng công nghệ thông tin vào trường học được nhiều nhà quản lý giáo dục quan tâm, đặc biệt là hệ thống quản lý học tập Moodle ngày càng được áp dụng rộng rãi tại Việt Nam cũng như trên thế giới [8]. Có nhiều ứng dụng được triển khai tại trường học phải tốn kém nhiều chi phí cao, cần sự hỗ trợ phát triển bởi các nhà làm hệ thống thông tin, tuy nhiên khi áp dụng hệ thống quản lý giáo dục Moodle thì giúp các trường học giảm thiểu nhiều chi phí mà lại được hưởng nhiều chức năng sẵn có của hệ thống. Bởi hệ thống quản lý giáo dục Moodle là một hệ thống mã nguồn mở vì thế không những không tốn kém bản quyền phần mềm mà còn được sự hỗ trợ của nhiều nhà phát triển trên thế giới [9].

Hệ thống quản lý học tập Moodle hỗ trợ nhiều cho nhà quản trị hệ thống và được Alex Buchner [1] hướng dẫn cài đặt và cấu hình triển khai. William H. Rice [11] cũng đã hướng dẫn cho các nhà giáo xây dựng bài học, bài dạy dựa trên hệ thống Moodle. Đối với các nhà lập trình, Martin Dougiamas và cộng đồng lập trình Moodle [9] đã giúp rất nhiều bài viết để có thể giúp cách điều chỉnh hay viết thêm các chức năng. Đáng chú ý nhất là sách hướng dẫn phát triển thêm các phần mở rộng của Jonathan More [7] đã hướng dẫn rất chi tiết cách phát triển thêm các chức năng để gắn vào Moodle.

Giải thuật để giải quyết bài toán xếp lịch thi đã có nhiều tác giả mô hình bài toán này thành dạng lập trình ràng buộc. Trong đó có công trình của Boizumault và các cộng sự [2] đã thực hiện xếp lịch thi cho trường đại học Catholique de l'Ouest ở Angres, Pháp. Công trình này sử dụng ngôn ngữ lập trình logic có ràng buộc trên miền trị hữu hạn CHIP. Bài toán xếp lịch thi vẫn đáp cho kỳ thi tuyển sinh vào trường Ecole des Mines de Nantes, Pháp, đã được David P. [5] giải quyết bằng giải thuật giải hệ ràng buộc với kỹ thuật tu chỉnh lặp. Buke, Newall và Weare [4] đã sử dụng giải thuật di

Abstract - Scheduling exam time tables is an important part that training institutions always have to do before the end of a semester. Developing a tool that allows universities to schedule exam time tables is essential. Furthermore, a plugin with exam time tables integrated into Moodle is the great concern of many universities. Moodle is a learning management system provided freely as a open source software used by a variety of institutions and universities. This paper presents the results of building and developing an exam time table module integrated into Moodle, specifically applied at the University of Economics, The University of Danang. This exam time table plugin uses backtracking algorithm with a number of technical inspections done before the exam time table being automatically scheduled. The result shows that the application of the exam time table plugin is helpful and has the ability to develop other functions on Moodle.

Key words - schedule exam time table; exam time table module; learning management system; plugin; Moodle; backtracking algorithm.

truyền cho bài toán xếp lịch thi học kỳ và họ gọi giải thuật đó là memetic. Boufflet và Negre [3] đã giải bài toán xếp lịch thi cho trường Đại học Kỹ thuật Compiègne ở Pháp bằng giải thuật tabu search. White và Zhang [10] đã sử dụng giải thuật tabu search cho trường Đại học Ottawa ở Canada.

Trong các plugins [9] của hệ thống Moodle hiện nay chưa thấy có chức năng xếp lịch thi học kỳ hay các chức năng tương tự được tích hợp vào hệ thống quản lý học tập Moodle. Tại trường Đại học Kinh tế, Đại học Đà Nẵng, Phòng Đào tạo cũng đang sử dụng công cụ sẵn có của Excel để xếp lịch thi. Cách làm này có thể đáp ứng ngay nhu cầu xếp lịch thi của nhà trường nhưng khó làm được các chức năng hữu ích trong quá trình xếp lịch thi như là kiểm tra việc trùng lịch của một sinh viên học chế tín chỉ.

Bài báo này trình bày một cách thực tế việc khảo sát, phân tích, thiết kế, xây dựng một chức năng xếp lịch thi học kỳ tích hợp vào hệ thống Moodle với các yêu cầu dựa theo trường hợp cụ thể là Trường Đại học Kinh tế, Đại học Đà Nẵng. Trong đó, bài báo có sử dụng giải thuật backtracking với một số kỹ thuật kiểm tra trước dành riêng cho việc xếp lịch thi để tự động xây dựng một lịch thi.

Đóng góp của công trình là thực hiện phân hệ xếp lịch và tích hợp vào hệ thống Moodle trường hợp cụ thể của Trường Đại học Kinh tế, Đại học Đà Nẵng. Công trình tự xây dựng giải thuật backtracking theo mô hình cơ bản Dechter và Frost [6], trong đó bổ sung các kỹ thuật kiểm tra trước giúp giải thuật backtracking thực hiện thuận lợi hơn.

2. Phương pháp thực hiện

2.1. Khảo sát thực tế

Khảo sát thực tế và quan sát dữ liệu của Trường Đại học Kinh tế, Đại học Đà Nẵng làm dữ liệu điển hình cho bài toán xếp lịch thi học kỳ, qua đó cho thấy rõ công việc lập lịch thi học kỳ.

	A	B	C	D	E	F	G	H
1	Lớp	Mã sinh viên	Họ lót	Tên	Mã học phần	Tên học phần	Số tín chỉ	Lớp tín chỉ
2	34K01.2	081121601245	Nguyễn Thị Thanh	Thảo	4270212	Anh văn chuyên ngành 1 (NT-TM)	2	AVNT1_01
3	34K14	081121514004	Văn Công	Chinh	4270212	Anh văn chuyên ngành 1 (NT-TM)	2	AVNT1_01
4	35K01.1	091121601102	Nguyễn Yến Quỳnh	Đan	4270212	Anh văn chuyên ngành 1 (NT-TM)	2	AVNT1_01
5	35K01.1	091121601106	Nguyễn Thị Phương	Dung	4270212	Anh văn chuyên ngành 1 (NT-TM)	2	AVNT1_01
6	35K01.1	091121601112	Dương Bích	Hạnh	4270212	Anh văn chuyên ngành 1 (NT-TM)	2	AVNT1_01

Hình 1. Lịch học tập của sinh viên Trường Đại học Kinh tế

10	STT	Mã học phần	Tên học phần	Số tín chỉ	Thi				
11					Tuần	Giờ	Giờ	Ngày	Phòng
12	1	2150062	Hệ thống thông tin quản lý	3	40	X1	07h00	12/5/2014	D101,D102,D103,D104,D105,D106,D201,D202,D203
13	2	2140243	Hệ thống thông tin quản trị trong ngân hàng	2	40	X1	07h00	12/5/2014	D203
14	3	2110183	Kinh tế vùng	2	40	X1	07h00	12/5/2014	D204,D205
15	4	2111562	Luật tài chính	2	40	X1	07h00	12/5/2014	D206,D301,D302
16	5	2100162	Phân tích tài chính doanh nghiệp	3	40	X1	07h00	12/5/2014	D303,D304,D305,D306,D401

Hình 2. Lịch thi của sinh viên Trường Đại học Kinh tế

Chuẩn bị cho học kỳ mới, Phòng Đào tạo sẽ tiến hành cho sinh viên đăng ký tín chỉ học tập trực tuyến trên trang web daotao.due.edu.vn. Lấy mẫu về lịch học tập của sinh viên cụ thể tại Hình 1, trong đó có các thông tin: Lớp sinh hoạt; Mã sinh viên; Họ tên sinh viên; Mã học phần; Tên học phần; Số tín chỉ; Lớp tín chỉ; Học kỳ.

Vào gần cuối của mỗi học kỳ, thầy cô phòng Đào tạo tiến hành sắp xếp lịch thi và công bố trên trang web của nhà trường. Thông tin của mỗi lịch thi được trình bày trong Excel như trong Hình 2 và trong đó cụ thể gồm: Mã học phần; Tên học phần; Số tín chỉ; Tuần; Giờ; Ngày; Phòng thi.

2.2. Phân tích dữ liệu lịch thi

Phân tích các số liệu và tổng kết đánh giá thông tin về bảng đăng ký học tập của sinh viên như sau:

Bảng 1. Bảng tổng kết về đăng ký học tập tín chỉ

Tên bảng	Thuộc tính	Tổng số mẫu tin
Đăng ký học phần	Mã sinh viên, mã học phần, lớp tín chỉ, học kỳ	51191
Sinh viên	Mã sinh viên, họ lót, tên, lớp	7055
Học phần	Mã học phần, tên học phần, số tín chỉ	200
Phòng	Tên phòng, sức chứa	75 (5 khu nhà)

Đăng ký học phần gồm có các thông tin mã sinh viên, mã học phần, lớp tín chỉ. Dữ liệu về sinh viên gồm có mã sinh viên, tên, họ lót, lớp sinh hoạt. Dữ liệu về học phần gồm có mã học phần, tên học phần, số tín chỉ.

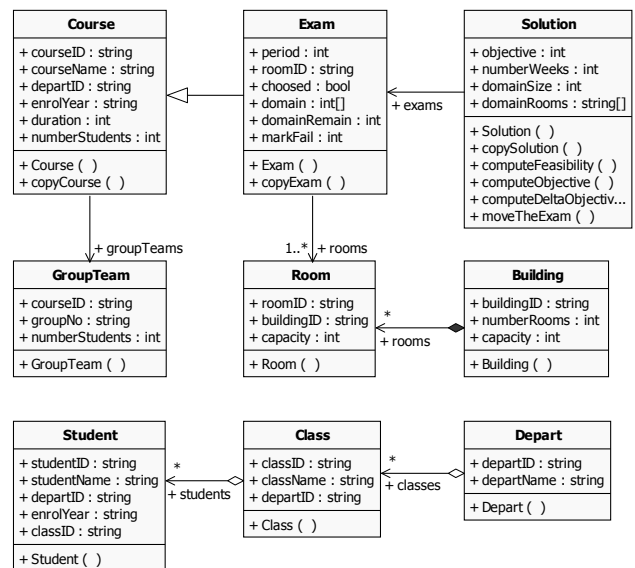
2.3. Thiết kế cấu trúc

Để xây dựng thành công chương trình xếp lịch thi học kỳ, cần phải xây dựng các dữ liệu cơ bản của bài toán, với mục tiêu dễ dàng hiện thực được ý đồ giải thuật vào chương trình. Chương trình cũng cần có khả năng nhập liệu cho tất cả các thông tin cần thiết làm nền tảng cho việc xếp lịch thi.

Dữ liệu cơ bản của bài toán xếp lịch thi học kỳ thiết kế theo UML trong Hình 3 và được thiết kế cụ thể như sau:

Học phần (Course): Là dữ liệu cơ bản của công tác giáo vụ, bao gồm các thông tin cơ bản như mã số (courseID),

tên học phần (courseName), mã số khoa (departID), năm học sinh viên (enrolYear), số tiết (duration), số sinh viên (numberStudents).



Hình 3. Thiết kế cấu trúc theo mô hình UML

Nhóm sinh viên (GroupTeam): Là dữ liệu mô tả việc chia nhóm của sinh viên theo một học phần nào đó, bao gồm các thông tin như mã số học phần (courseID), số của nhóm sinh viên (groupNo), tổng số sinh viên của nhóm (numberStudents).

Môn thi (Exam): Là dữ liệu cơ bản của việc sắp xếp lịch thi. Môn thi kế thừa thông tin của học phần, và khi môn thi được xác định trong lời giải của lịch thi nó còn thêm các thông tin như tiết thứ (period), phòng thi (roomID).

Lời giải (Solution): Lời giải được cấu tạo bao gồm tất cả các môn thi, trong đó từng môn thi đã được gán tiết và gán phòng thi.

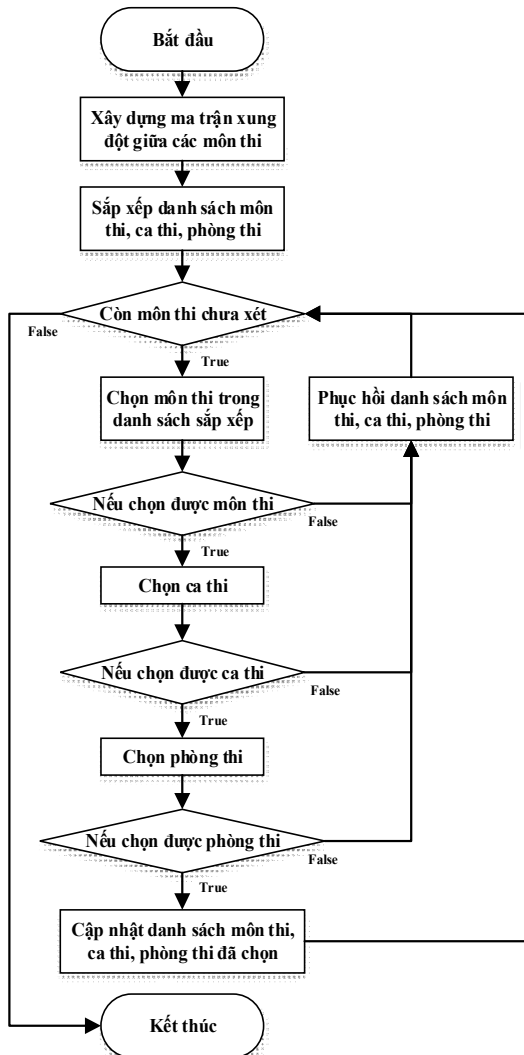
Phòng thi (Room): Là dữ liệu cơ bản của công tác giáo vụ. Phòng thi là một trong các tài nguyên thường được dùng cho việc sắp xếp lịch học tập hoặc lịch thi học kỳ.

Từng thời điểm phòng thi sẽ được sử dụng cho các học phần và môn thi khác nhau. Phòng thi có các dữ liệu như mã số phòng thi (roomID), mã số tòa nhà (buildingID), sức chứa (capacity).

Sinh viên (Student): Là thông tin của các sinh viên tham gia học trong trường. Sinh viên là đối tượng rất cơ bản của công tác giáo vụ. Thông thường sinh viên được chia theo học từng lớp, cũng tùy vào từng trường đại học mà cách tổ chức lớp học là chặt hoặc lỏng lẻo, nghĩa là sinh viên có thể gán cứng theo lớp hoặc không nhất thiết phải quá cứng nhắc theo một lớp nào đó. Thông tin của một đối tượng sinh viên có các nội dung cơ bản như mã số sinh viên (studentID), tên sinh viên (studentName), mã lớp (classID), mã khoa (departID), sinh viên đang theo học năm thứ mấy (enrolYear).

2.4. Giải thuật xếp lịch thi

Tham khảo mã giả về giải thuật backtracking của giáo sư Rina Dechter trường đại học California [6], tác giả đã xây dựng sơ đồ khối về giải thuật backtracking với một số kỹ thuật giúp kiểm tra trước để xây dựng nên một lịch thi đề nghị cho nhà quản lý lịch thi.



Hình 4. Giải thuật backtracking xếp lịch thi

Giải thuật bắt đầu bằng việc xây dựng các danh sách môn thi (học phần thi), thời gian thi (tiết thi), phòng thi.

Các miền trị để chạy giải thuật đó là tập hợp thời gian thi cho trước, tập hợp phòng thi theo thời gian thi được xây dựng phục vụ cho giải thuật.

Giải thuật backtracking xếp lịch thi tiến hành vòng lặp xem xét các môn thi chưa được xét, trong các môn thi đó chọn các môn thi nào đó để xem xét, kế tiếp sẽ chọn thời gian thi để gán cho môn thi đó, sau đó là chọn phòng thi cho môn thi đó.

Nếu việc chọn lựa môn thi, thời gian thi, phòng thi là thành công thì giải thuật cập nhật các số liệu cần thiết và di chuyển đến môn thi kế tiếp. Nếu giải thuật không chọn được môn thi, thời gian thi hay phòng thi thích hợp thì giải thuật sẽ qua trở lại khôi phục các trị đã gán của môn thi trước đó, đánh dấu môn thi không thành công để xét sau cùng. Giải thuật chạy tiếp bằng môn thi kế tiếp cho tới khi hết môn thi cần xem xét.

2.4.1. Kỹ thuật xây dựng ma trận xung đột giữa các học phần thi

Hai học phần thi xung đột nhau là hai học phần có ít nhất một sinh viên tham gia thi cả hai học phần đó. Như vậy, hai học phần xung đột không được phép tổ chức thi cùng một thời điểm. Để có thể trình bày việc xung đột giữa các học phần, ta có thể dùng ma trận xung đột.

Giả sử có n học phần, ta có thể lập ma trận xung đột với n hàng, n cột và các giá trị trong ma trận là số sinh viên học tham gia thi cả hai học phần.

CM là ma trận xung đột $n \times n$, trong đó n là số học phần.

$CM(i, j)$ = tổng số sinh viên cùng tham gia học phần i và học phần j

$CM(i, j) = 0$ khi không có sinh viên nào cùng tham gia học phần i và học phần j .

Theo ví dụ dưới đây ở Hình 5 thì học phần 0 đụng độ với học phần 1 là 45, học phần 0 đụng độ với học phần 2 là 34, học phần 0 không đụng độ với các học phần khác.

	0	1	2	...	n-2	n-1
0	0	45	34	0	0	0
1	45	0	0	0	89	56
2	34	0	0	0	0	0
...	0	0	0	0	0	0
n-2	0	89	0	0	0	32
n-1	0	56	0	0	32	0

Hình 5. Ma trận xung đột giữa các học phần

Để hiện thực ma trận xung đột này có thể sử dụng các cấu trúc dữ liệu mảng hai chiều, mỗi chiều có kích thước bằng tổng số môn thi. Hoặc có thể hiện thực ma trận xung đột bằng mảng hai chiều nhưng kích thước thay đổi tùy vào số học phần xung đột.

2.4.2. Kỹ thuật mô hình hóa miền trị của thời gian thi

Thời gian thi trong lịch thi được quy định như sau: số ca thi trong một ngày là 4, số ngày trong một tuần là 5. Tổng số thời gian thi trong một tuần là 20. Giả sử chọn 6 tuần để xếp lịch thi, ta sẽ có 4 ca x 5 ngày x 6 tuần = 120 ca thi, và xây dựng miền trị của môn thi từ 0 đến 119.

Tổng quát hóa kỹ thuật xây dựng miền trị này: cho quy định số ca thi trong một ngày là p , số ngày trong một tuần là d , và số tuần thi là w , vậy tổng số ca thi sẽ là $n = p \times d \times w$, và miền trị sẽ từ 0 đến $n-1$.

Ngược lại, với một chỉ số i bất kỳ của miền trị, ta cũng dễ dàng xác định được tuần, thứ và ca thi như sau:

$$w_i = i / (p \times d)$$

$$d_i = (i \bmod (p \times d)) / p$$

$$p_i = (i \bmod (p \times d)) \bmod p$$

Kỹ thuật mô hình hóa miền trị này so với cách xem miền trị theo từng tuần, từng ngày, từng ca đã tỏ ra hữu hiệu hơn nhiều khi phải thao tác lập trình, giúp đơn giản hóa quá trình tính toán và chọn tiết, đồng thời có thể chuyển đổi dễ dàng khi cần.

2.4.3. Kỹ thuật sắp thứ tự học phần thi

Nhận xét thấy rằng những học phần nào có các ràng buộc càng chặt thì càng khó xếp vào thời khóa biểu, nếu những học phần đó phải xếp sau cùng thì độ khó càng cao và có thể không xếp được, vì vậy những học phần khó xếp nhất thì nên được xếp trước tiên. Đây là cách tiếp cận theo hướng sắp thứ tự biến nhằm có thể loại bỏ các nhánh không đi đến kết quả.

Để có thể định nghĩa được độ khó xếp của học phần ta có thể định nghĩa hàm tính điểm ưu tiên như sau:

$$\text{markFail} = x \times \text{domainSize} / \text{markDomain} + y \times \text{markConflict} + z \times \text{markStudent};$$

Trong đó:

domainSize: kích thước tổng miền trị của học phần.

markDomain: kích thước miền trị còn lại của học phần đang xét.

markConflict: số học phần xung đột với học phần đang xét.

markStudent: số sinh viên tham gia học phần đang xét.

x, y, z : là các tham số có thể quy định từ chương trình.

Sau khi tính toán điểm ưu tiên của toàn bộ các học phần, ta sẽ xếp chúng vào danh sách theo thứ tự tăng dần, danh sách này còn gọi là danh sách học phần first-fail. Và những học phần nào có điểm cao nhất sẽ được chọn để xếp lịch thi trước.

Kỹ thuật này sẽ phải thực hiện ở đầu giải thuật backtracking, và các điểm ưu tiên của môn thi sẽ được cập nhật sau mỗi bước lặp của giải thuật để bảo đảm rằng giải thuật luôn chọn được học phần có độ khó cao nhất.

3. Kết quả thực hiện

3.1. Kết quả xây dựng phân hệ tích hợp vào Moodle

Để thực hiện tích hợp các chức năng thêm vào trong hệ thống Moodle, nghiên cứu này đã cài đặt XAMPP 1.8.2, và cài đặt Moodle 2.7 sử dụng Apache, PHP và MySQL đã được tích hợp trong XAMPP 1.8.2. Công trình đã thực hiện viết một plugin dạng block để tích hợp vào trong Moodle, theo đó người sử dụng Moodle có thể thêm một plugin vào Moodle dễ dàng.

Người sử dụng có thể khai thác menu bên trong block xếp lịch thi để quản lý các dữ liệu liên quan đến việc xếp lịch thi như quản lý học phần, quản lý phòng thi, quản lý sinh viên, quản lý thông tin sinh viên đăng học phần. Diễn hình là quản lý học phần được trình bày trong Hình 6.

Sau khi đã chuẩn bị xong các dữ liệu cần thiết, người sử dụng có thể khai thác các chức năng quản lý lịch thi như: xây dựng ma trận các học phần xung đột, tạo một lịch thi và trong từng lịch thi thực hiện xếp lịch thi cho các môn thi.

Quản lý học phần				Thêm học phần	
Mã	Tên học phần	Số tín chỉ	Số sinh viên		
3250041	Lý thuyết xác suất và thống kê toán	3	2225	Sửa	Xóa
600321	Thế dục 2	2	2039	Sửa	Xóa
600341	Thế dục 4	2	2027	Sửa	Xóa
2111221	Kinh tế vĩ mô	3	1862	Sửa	Xóa
4270021	Ngoại ngữ đại cương II	2	1734	Sửa	Xóa
2120251	Các nguyên lý cơ bản của chủ nghĩa Mac-Lênin 2	3	1718	Sửa	Xóa
2150011	Tin học đại cương	3	1563	Sửa	Xóa
2130032	Quản trị học	3	1539	Sửa	Xóa
2100012	Nguyên lý kế toán	3	1481	Sửa	Xóa

Hình 6. Màn hình quản lý học phần

XẾP LỊCH THI	Xếp lịch thi > Thêm học phần thi	
Quản lý dữ liệu	Chọn học phần thi	
Quản lý học phần	3250041 Lý thuyết xác suất và thống kê toán 3 (2225)	
Quản lý phòng thi	Ngày thi	Giờ thi
Quản lý sinh viên	5 November 2014	Ca 1, 07:00
Quản lý đăng ký	Phòng thi	
Quản lý lịch thi	A203 (45)	
Học phần xung đột	A204 (19)	
Danh sách lịch thi	A205 (18)	
Tối ưu hóa lịch thi	A206 (10)	
	A207 (46)	
NAVIGATION	Save changes Reset Cancel	
Home		
My home		
Site pages		
My profile		
Courses		

Hình 7. Màn hình thêm học phần thi

Lịch thi của các môn thi cần được chọn thời gian và phòng thi sao cho không có sinh viên nào thi cùng một thời điểm, và cùng một thời điểm không có môn thi nào trùng phòng thi. Giao diện thêm một học phần vào lịch thi được thấy trong Hình 7.

3.2. Kết quả xây dựng giải thuật backtracking

Để thực hiện được giải thuật backtracking với các kỹ thuật kiểm tra trước, trong bài báo đã thực hiện chức năng xây dựng ma trận học phần xung đột như trong Hình 8.

Ma trận xung đột > Xây dựng ma trận	
100.0% (250/250)	
Tổng số học phần: 200, tổng số đăng ký: 51191	
Thời gian bắt đầu: 23:02:08	
1. Học phần 4270212 xung đột với 53 học phần: 2120251(12), 2160133(12), 2150022(94), 2130092(29), 2160132(46), 2160143(2), 2160091(5), 2130302(8), 2150123(13), 2130042(83), ...	
2. Học phần 4270232 xung đột với 54 học phần: 2130082(14), 2160143(8), 2150022(2), 2160153(3), 213061(2), 2130072(14), 2130052(6), 2130071(3), 2130092(4), 2130302(2), ...	
3. Học phần 4270232 xung đột với 49 học phần: 2160153(2), 2130072(94), 2160143(94), 2150122(12), 3250041(2), 2140112(2), 2120211(98), 2100032(58), 2160123(60), 2160143(48), ...	
4. Học phần 4270042 xung đột với 63 học phần: 2150022(186), 2130302(70), 2130162(146), 2130062(17 0), 060094(404), 2150122(69), 2130051(137), 2120211(166), 2160022(130), 2160152(49), ...	
5. Học phần 4270071 xung đột với 34 học phần: 2120211(55), 2160012(30), 2160143(13), 2160091(13), 2130092(48), 2150122(28), 060094(138), 2130052(192), 2150022(48), 2110091(48), ...	
6. Học phần 4270143 xung đột với 72 học phần: 2160123(1), 2160123(5), 2100071(48), 2160091(5), 2	

Hình 8. Chức năng xây dựng ma trận học phần xung đột

Kết quả thực hiện ma trận các học phần xung đột thể hiện trong Bảng 2 là khoảng 28 phút 11 giây:

Bảng 2. Kết quả thực hiện ma trận các học phần xung đột

Tổng số học phần	200
Tổng số sinh viên đăng ký các học phần	51.191
Thời gian bắt đầu	23:02:08
Thời gian kết thúc	23:30:19
Thời gian thực hiện	28 phút 11 giây

Giải thuật backtracking sử dụng ma trận các học phần xung đột được thực hiện khi người sử dụng chọn chức năng tự động xếp lịch thi như trong Hình 9.

The screenshot shows the Moodle 'Xếp lịch thi' (Exam Scheduling) interface. It displays a list of subjects (học phần) and their associated rooms (phòng) and times (giờ). The interface includes a sidebar with navigation links like 'Quản lý học phần', 'Quản lý phòng thi', 'Quản lý sinh viên', and 'Quản lý đăng ký'. The main content area shows a list of subjects with their respective rooms and times, and a 'Danh sách lịch thi' (Exam Schedule) section.

Hình 9. Thực hiện xếp lịch thi tự động

Kết quả của giải thuật backtracking được thực hiện như trong Bảng 3 cho thấy thời gian để giải thuật thực hiện xong là 07 giây.

Bảng 3. Kết quả thực hiện xếp lịch thi tự động

Tổng số học phần	200
Tổng số sinh viên đăng ký các học phần	51.191
Thời gian bắt đầu	17:24:05
Thời gian kết thúc	17:24:12
Thời gian thực hiện	07 giây

Thời gian để chạy giải thuật backtracking như trên là rất khả quan, tuy nhiên thời gian xây dựng ma trận các học phần xung đột là khá lâu, nhưng người sử dụng vẫn có thể chờ đợi để có được một lịch thi xếp tự động.

4. Kết luận

Xếp lịch thi học kỳ là một công việc khó trong các công tác của nhà quản lý đào tạo, bài báo đã trình bày những công việc mà công trình đã thực hiện bao gồm khảo sát, phân tích, thiết kế và xây dựng một phân hệ xếp lịch thi học kỳ, đồng thời tích hợp phân hệ này vào trong hệ thống quản lý học tập Moodle.

Trong phân hệ xếp lịch thi học kỳ có sử dụng giải thuật backtracking với những kiểm tra cần thiết giúp xếp được

lịch thi học kỳ tự động, việc này có ý nghĩa rất lớn cho nhà quản lý đào tạo.

Phân hệ xếp lịch thi này đã được tích hợp vào hệ thống quản lý học tập Moodle làm tiền đề cho nhiều nhà phát triển có thể tiếp tục phát triển phân hệ này phù hợp với cơ sở đào tạo của mình và cũng có thể phát triển thêm các giải thuật khác áp dụng vào việc xếp lịch thi học kỳ.

Với hệ thống quản lý học tập Moodle, những nhà phát triển có thể tích hợp nhiều chức năng hơn nữa phục vụ cho các trường học và phù hợp với môi trường hiện tại.

TÀI LIỆU THAM KHẢO

- [1] Alex Buchner, *Moodle 2 Administration, An administrator's guide to configuring, securing, customizing, and extending Moodle*, PACKT Publishing, 2011.
- [2] Boizumault, P., Delon, Y. and Peridy, L., Constraint Logic Programming for Examination Timetabling, *Journal of Logic Programming*, 26, 217-233, 1996.
- [3] Boufflet, J. P. and Negre, S., Three Methods Used to Solve an Examination Timetabling Problem, *Lecture Notes in Computer Science 1153*, Springer-Verlag, 327-344, 1996.
- [4] Burke, E., Newall, J.P. & Weare, R. F. A Memetic Algorithm for University Exam Timebling, *Lecture Notes in Computer Science 1153*, Springer-Verlag, 241-250, 1996.
- [5] David, P., A Constraint-based Approach for Examination Timetabling using Local Repair Techniques, *Proc. of Practice and Theory of Automated Timetabling, PAPAT'97*, Toronto, Canada, August, 132-145, 1997.
- [6] Dechter, R., Frost, D., Backjump-based Backtracking for Constraint Satisfaction Problems, *Artificial Intelligence*, 136, 147-188, 2002.
- [7] Jonathan Moore, Michael Churchward, *Moodle 1.9 Extension Development, Customize and extend Moodle by using its robust plugin systems*, PACKT Publishing, 2010.
- [8] Martin Dougiamas and et al, About Moodle, from https://docs.moodle.org/27/en/About_Moodle, retrieved 15/8/2014.
- [9] Martin Dougiamas and et al, Developer documentations are for developers who work with Moodle code, plugins and themes, from https://docs.moodle.org/dev/Main_Page, retrieved 15/8/2014.
- [10] White G.M., Zhang J., Generating Complete University Timetables by Combining Tabu Search with Constraint Logic, In: *E. Burke, M. Carter (Eds.): PATAT'97, LNCS 1408*, pp. 187-198, 1998, Springer-Verlag Berlin Heidelberg 1998.
- [11] William H. Rice IV, *Moodle E-Learning Course Development, A complete guide to successful learning using Moodle*, PACKT Publisng, 2006.

(BBT nhận bài: 09/11/2014, phản biện xong: 12/01/2015)