|  |
| --- |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  NGUYỄN THỊ HUYỀN TRANG  KHÓA 14  HỆ ĐẠI HỌC  **KHÓA LUẬN**  **TỐT NGHIỆP ĐẠI HỌC**  **CHUYÊN NGÀNH CÔNG NGHỆ GAME & MÔ PHỎNG**  **XÂY DỰNG CHƯƠNG TRÌNH XẾP LỊCH DẠY CHO GIẢNG VIÊN KHOA CÔNG NGHỆ THÔNG TIN**  **HÀ NỘI, NĂM 2019** |

**LỜI CẢM ƠN**

Để hoàn thành bài báo cáo thực tập này, trước hết, em xin cảm ơn các thầy giáo, cô giáo Khoa Công nghệ thông tin – trường Học viện Kỹ Thuật Quân Sự, những người đã dạy dỗ, tận tình giảng dạy, truyền đạt những kiến thức, kinh nghiệm quý báu trong suốt những năm học vừa qua, giúp chúng em hiểu rõ hơn các lĩnh vực đã nghiên cứu để hoàn thành đề tài được giao.

Em xin bày tỏ lòng biết ơn xâu sắc nhất tới thầy giáo Tiến sĩ Nguyễn Trung Tín, người đã hướng dẫn, chỉ bảo tận tình, định hướng cho em để em hoàn thành tốt khóa luận của mình.

Xin gửi lời cảm ơn đến gia đình, bạn bè và các bạn sinh viên lớp đã luôn động viên, giúp đỡ em trong suốt quá trình học cũng như làm tốt nghiệp, giúp em hoàn thành đề tài đúng thời hạn.

Em xin chân thành cảm ơn!

*Hà Nội, tháng 11 năm 2019*

Sinh viên

**Nguyễn Thị Huyền Trang**

MỤC LỤC

[**DANH MỤC HÌNH ẢNH** 5](#_Toc25961196)

[**DANH MỤC BẢNG BIỂU** 6](#_Toc25961197)

[**DANH MỤC CHỮ VIẾT TẮT** 7](#_Toc25961198)

[**LỜI MỞ ĐẦU** 8](#_Toc25961199)

[**CHƯƠNG I: TỔNG QUAN VỀ BÀI TOÁN XẾP LỊCH DẠY VÀ CÁC PHƯƠNG PHÁP TIẾP CÂN** 10](#_Toc25961200)

[1.1 Tổng quan về đề tài 10](#_Toc25961201)

[1.2 Bài toán xếp lịch dạy cho các trường Đại học – Cao đẳng 11](#_Toc25961202)

[1.3 Các phương pháp tiếp cận hiện nay 14](#_Toc25961203)

[1.4 Cơ sở lý thuyết 15](#_Toc25961204)

[1.4.1 Ngôn ngữ lập trình Javascript 15](#_Toc25961205)

[1.4.1.1 Giới thiệu về NodeJS 15](#_Toc25961206)

[1.4.1.2 Ứng dụng 16](#_Toc25961207)

[1.4.1.3 Ưu điểm 16](#_Toc25961208)

[1.4.2 Visual Studio Code 18](#_Toc25961209)

[1.4.3 Hệ quản trị cơ sở dữ liệu 18](#_Toc25961210)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 19](#_Toc25961211)

[2.1 Giải thuật di truyền 19](#_Toc25961212)

[**CHƯƠNG 3: BÀI TOÁN XẾP LỊCH DẠY VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 25](#_Toc25961213)

[3.1 Phân tích thiết kế hệ thống 25](#_Toc25961214)

[3.1.1 Nhiệm vụ cơ bản 25](#_Toc25961215)

[3.1.2 Quy trình xếp lịch dạy theo đào tạo tín chỉ và quy tắc quản lý 25](#_Toc25961216)

[3.1.3 Mẫu biểu 26](#_Toc25961217)

[3.1.4 Mô hình tiến trình nghiệp vụ 30](#_Toc25961218)

[3.1.5 Biểu đồ hoạt động 32](#_Toc25961219)

[3.1.6 Mô hình hóa chức năng nghiệp vụ chi tiết 34](#_Toc25961220)

[3.1.7 Vẽ sơ đồ phân rã chức năng 37](#_Toc25961221)

[3.1.7 Danh sách hồ sơ dữ liệu sử dụng 38](#_Toc25961222)

[3.1.8 Ma trận thực thể chức năng 38](#_Toc25961223)

[3.1.9 Sơ đồ luồng dữ liệu 40](#_Toc25961224)

[3.2 Áp dụng giải thuật di truyền và bài toán 47](#_Toc25961225)

[3.2.1 Các yêu cầu cơ bản của thời khóa biểu theo đào tạo tín chỉ 47](#_Toc25961226)

[3.2.2 Mô tả bài toán 49](#_Toc25961227)

[3.2.3 Mô hình toán học của bài toán xếp lịch 51](#_Toc25961228)

[3.2.4.1 Thiết kế giải thuật 53](#_Toc25961229)

[3.4.2.2. Biểu diễn quần thể 53](#_Toc25961230)

[3.4.2.3 Đánh giá và lựa chọn cá thể 55](#_Toc25961231)

[3.4.2.4 Lai ghép 55](#_Toc25961232)

[**CHƯƠNG IV. KẾT QUẢ THỰC NGHIỆM** 56](#_Toc25961233)

[4.1. Chuẩn bị dữ liệu 56](#_Toc25961234)

[4.2 Một số chức năng vào giao diện của ứng dụng 56](#_Toc25961235)

[4.2.1 Chức năng nhập dữ liệu 56](#_Toc25961236)

[4.2.2 Chức năng hiển thị thời khóa biểu 58](#_Toc25961237)

[**KẾT LUẬN** 60](#_Toc25961238)

# **DANH MỤC HÌNH ẢNH**

[Hình 2. 1 Sơ đồ khối giải thuật di truyền đơn giản (GA). [1] 20](#_Toc25960364)

[Hình 2. 2 Minh họa quá trình lai ghép trong thuật toán di truyền 22](#_Toc25960365)

[Hình 2. 3 Minh họa quá trình đột biến trong thuật toán di truyền [2] 22](#_Toc25960366)

[Hình 3. 1 Mô hình tiến trình nghiệp vụ 1 29](#_Toc25960370)

[Hình 3. 2 Biểu đồ hoạt động 30](#_Toc25960371)

[Hình 3. 3 Biểu đồ phân rã chức năng 34](#_Toc25960372)

[Hình 3. 4 DFD mức 0 36](#_Toc25960373)

[Hình 3. 5 Sơ đồ luông dữ liệu mức 37](#_Toc25960374)

[Hình 3. 6 Sơ đồ tiến trình xếp TKB 37](#_Toc25960375)

[Hình 3. 7 Sơ đồ tiến trình xem TKB 38](#_Toc25960376)

[Hình 4. 1Trang nhập môn học dự kiến 51](#_Toc25960377)

[Hình 4. 2 Trang nhập giảng viên dự kiến 51](#_Toc25960378)

[Hình 4. 3 Trang nhập giảng đường dự kiến 52](#_Toc25960379)

[Hình 4. 4 Trang xem TKB các môn học 53](#_Toc25960380)

# **DANH MỤC BẢNG BIỂU**

[Bảng 3. 1Thời khóa biểu dân dự khóa 14,15, 16,17 25](#_Toc25960451)

[Bảng 3. 2 Bảng phân công dạy 25](#_Toc25960452)

[Bảng 3. 3 Ma trận thực thể 1 35](#_Toc25960453)

[Bảng 3. 4 Các thực thể, thuộc tính và khóa 38](#_Toc25960454)

[Bảng 3. 5 Mô hình ER 39](#_Toc25960455)

[Bảng 3. 6 LOP\_MONHOC 40](#_Toc25960456)

[Bảng 3. 7 MONHOC 40](#_Toc25960457)

[Bảng 3. 8 GIANGDUONG 41](#_Toc25960458)

[Bảng 3. 9 GIANGVIEN 41](#_Toc25960459)

[Bảng 3. 10 DAY 41](#_Toc25960460)

[Bảng 3. 11TKB 41](#_Toc25960461)

# DANH MỤC CHỮ VIẾT TẮT

TKB- Thời khóa biểu

GV- Giảng viên

DS- Danh sách

SV- Môn học

# **LỜI MỞ ĐẦU**

Trong cuộc sống ta thường gặp các bài toán liên quan đến xếp lịch như xếp lịch vận hành máy móc, xếp lịch biểu cho việc thực hiện một dự án, xếp lịch làm việc, xếp lịch thi đấu thể thao,… Đối với loại bài toán này cần phải tìm ra một phương án xếp lịch thỏa mãn tất cả các ràng buộc cũng như khai thác hiệu quả các nguồn tài nguyên hiện có, giảm thời gian và chi phí thực hiện. Bài toán xếp thời khóa biểu trong trường học nói chung và trong trường Đại học nói riêng là một trong những bài toán như vậy. Có rất nhiều các ràng buộc được đặt ra trong bài toán này như ràng buộc về đối tượng tham gia (giảng viên, lớp học, sinh viên), ràng buộc về thời gian (số tiết học, số lần học, số tiết mỗi lần), ràng buộc về chuyên môn và rất nhiều các ràng buộc khác tùy thuộc vào từng trường. Vấn đề đặt ra là cần xây dựng một thời khóa biểu thỏa mãn tất cả các ràng buộc đồng thời khai thác hiệu quả các nguồn tài nguyên phục vụ giảng dạy. Bài toán xếp thời khóa biểu thuộc lớp các bài toán NP-đầy đủ vì vậy có thể không tìm ra được lời giải tối ưu. Đây là một bài toán không mới và đã có nhiều giải thuật được đưa ra để giải quyết như giải thuật nhánh cận, giải thuật leo đồi, giải thuật luyện thép, giải thuật tô màu đồ thị, giải thuật xấp xỉ,… Tuy nhiên các giải thuật này thường không có tính tổng quát và chỉ áp dụng hiệu quả đối với các trường học có quy mô nhỏ, ít ràng buộc về mặt dữ liệu.

Trong những năm gần đây, phương pháp tiếp cận di truyền đã thu hút rất nhiều sự chú ý trong các lĩnh vực nghiên cứu khác nhau trong đó có khoa học máy tính. Phương pháp này có nhiều đặc điểm nổi trội như tránh tối ưu cục bộ, thực hiện tốt với các bài toán có không gian lời giải lớn và có thể áp dụng cho nhiều loại bài toán tối ưu khác nhau. Ví dụ, sử dụng giải thuật di truyền để xếp thời khóa biểu học cho sinh viên , dùng giải thuật di truyền để hỗ trợ lập lịch điều hành công tác bệnh viện. Trong khuôn khổ bài toán này, chúng tôi đã sắp xếp lịch giảng dạy cho giảng viên tại khoa Công Nghệ Thông Tin\_ Học viện Kỹ Thuật Quân Sự. Mục tiêu chính của bài toán này là xếp lịch giảng dạy cho các giảng viên sao cho thỏa mãn tối ưu các ràng buộc.

Cấu trúc của khóa luận gồm các chương sau:

Chương 1: Tổng quan về bài toán xếp lịch dạy và các phương pháp tiếp cận

Chương 2: Cơ sở lý thuyết

Chương 3: Phân tích và thiết kế hệ thống và áp dụng giải thuật

Chương 4: Xây dựng phần mềm xếp lịch dạy cho giảng viên

Và cuối cùng là phần kết luận.

# **CHƯƠNG I: TỔNG QUAN VỀ BÀI TOÁN XẾP LỊCH DẠY VÀ CÁC PHƯƠNG PHÁP TIẾP CÂN**

## Tổng quan về đề tài

Bài toán lập lịch thời khóa biểu trường học là một trong những bài toán thú vị nhất trong lớp các bài toán tối ưu vì tính chất đa dạng về mô hình thời khóa biểu, có nhiều ràng buộc phức tạp và tính chất thực tiễn của nó.

Bài toán thời khóa biểu đưa ra một chuỗi sự kiện(các môn học, bài giảng hoặc môn thi) và bao gồm các giảng viên trong một khaorng thời gian định trước, và một tập các ràng buộc phải thỏa mãn của từng loại thời khóa biểu khác nhau. Tập ràng buộc bao gồm khả năng chứa của phòng học và các yêu cầu của giảng dạy theo phân công.

Phát biểu bài toán

* Mỗi trường có một danh sách các lớp học.
* Mỗi lớp có một danh sách xác định các giờ học trong một tuần, bao gồm tên môn học, tên giáo viên và số tiết.
* Các lớp học được phân bố trong các phòng học đã biết.

Tìm một phương án phân bổ giờ học, môn học và giảng viên thỏa mãn một số ràng buộc (ràng buộc cứng) và một số có thể có hoặc không có các ràng buộc không bắt buộc thỏa mãn triệt để (ràng buộc mềm).

Có thể nêu ra một số ràng buộc phổ biến như sau:

Ràng buộc cứng:

* Một giảng viên trong một tiết dạy không quá một lớp
* Một lớp trong một tiết học có không quá một giáo viên.
* Một lớp trong một tiết học có không quá một môn.
* Một số giảng viên không dạy quá k tiết trong một ngày.
* Trong mỗi buổi học của mỗi lớp các tiết học liên tục (không có tiết nghỉ ở giữa)
* Trong mỗi buổi học, các tiết học của cùng một môn học liên tục (không được tách rời).
* Một số môn phải phân vào các giờ xác định. Ví dụ: tiết sinh hoạt là tiết đầu của buổi đầu tuần.

Ràng buộc mềm:

* Các môn học có nhiều tiết trong tuần phải phân bố tương đối tập trung cho mỗi lớp.
* Một số giáo viên muốn dạy hoặc không dạy vào một số tiết hoặc một số buổi nhất định. Số buổi dạy của mỗi giáo viên là không quá nhiều (gom ngày dạy). Trường hợp một giáo viên dạy cả hai buổi thì nếu buổi sáng có tiết dạy thì buổi chiều ngày đó không phân lịch dạy, hoặc buổi sáng không phân lịch tiết cuối và buổi chiều không phân lịch tiết đầu…

## Bài toán xếp lịch dạy cho các trường Đại học – Cao đẳng

Đây là loại thời khóa biểu phức tạp vì tính biển động vì tính biến động và tính chất đa dạng của loại hình đào tạo (học theo niên chế, học theo tín chỉ…).

Bài toán lập thời khóa biểu cho trường Đại học là bài toán lập lịch cho các bài giảng vào từng khóa học với một số lượng phòng học và tiết học cho trước. Khóa học là điểm khác biệt của thời khóa biểu trường Đại học với trường Trung Học Phổ Thông. Các sinh viên tham dự khóa học, còn các lớp học ở trường phổ thông được tạo bởi tập học sinh.

Ở trường Đại học,Cao đẳng , hai khóa học có thể có trùng một số sinh viên tham dự và điều này tạo ra xung đột không thể lập lịch được trong một tiết học. Hơn nữa, các giảng viên thường chỉ dạy một khóa học hay một môn học trong một học kỳ. Cuối cùng, sức chứa của các phòng học là một yếu tố quan trọng trong việc lập lịch. Hiện nay, các trường Đại học ở Việt Nam thường đào tạo theo 2 mô hình: Mô hình lớp học niên chế: Sinh viên vào nhập học và các năm học được phân cố định vào các lớp học. Mô hình lớp học tín chỉ: Sinh viên được tự do đăng ký vào các lớp môn học đã được chuẩn bị trước của thời khóa biểu. Các lớp môn học này thực chất là các môn học được thiết kế thời khóa biểu giảng dạy chi tiết. Thông thường, sau khi thời khóa biểu của các lớp học này đã được lên kế hoạch thì sinh viên mới căn cứ vào thời khóa biểu cụ thể để đăng ký học.

Bảng 1.1 So sánh giữa mô hình niên chế và tín chỉ

|  |  |  |
| --- | --- | --- |
| Đặc thù | Lớp niên chế | Lớp tín chỉ |
| Tạo lớp học | Bắc buộc phải phân lớp cho mỗi khóa học đầu năm học | Không cần phân lớp cụ thể, sinh viên tự đăng ký |
| Phân bố môn học | Phân bố môn học và các bài giảng cho các lớp học dễ dàng | Việc phân bố, tạo lớp tín chỉ hàng năm tương đối phức tạp |
| Lập TKB | Lập thời khóa biểu rất phức tạp vì phải chú ý đến việc trùng giờ, trùng tiết trên lớp, giáo viên và phòng học, chưa kể các phát sinh do ghép lớp, tách lớp | Lập thời khóa biểu tương đối dễ dàng vì chỉ phải quan tâm đến giáo viên và phòng học |
| Quản lý giảng dạy | Quản lý lớp học và sinh viên dễ dàng | Quản lý việc lên lớp rất phức tạp |
| Lớp ghép, lớp tách | Rất phức tạp khi tổ chức ghép và tách các lớp niên chế | Không cần ghép hay tách các lớp tín chỉ |
| Phòng học | Yêu cầu chung về phòng học là lớn và phức tạp | Yêu cầu phòng học đơn giản |

Ta nhận thấy, đối với lớp tín chỉ, , việc tổ chức thời khóa biểu đơn giản hơn, nhưng rất phức tạp cho việc quản lý chuyên môn, đào tạo, còn đối với lớp niên chế, đơn giản về mặt tổ chức, quản lý chuyên môn, nhưng rất phức tạp trong việc lập thời khóa biểu. Trong trường hợp phải ghép hoặc tách lớp thì công việc lập thời khóa biểu lại càng phức tạp hơn.

Vì nội dung đồ án đề cập về mô hình tín chỉ, nên phần này chỉ để cập đến hệ đào tạo theo tín chỉ.

Đối với các trường Đại học có hình thức đào tạo theo tín chỉ, bài toán thời khóa biểu được phát biểu như sau:

* Có N môn học được các sinh viên đăng ký tham dự cần lập lịch vào một tuần gồm K tiết học tương ứng.
* Các môn học được tổ chức tại các phòng học đáp ứng đủ các điều kiện học tập của môn học đó.

Một lời giải hay một thời khóa biểu chấp nhận được là tất cả các môn học đều được chia vào các tiết học và các phòng học tương ứng, đồng thời thỏa mãn các ràng buộc sau:

Ràng buộc cứng:

* Không có sinh viên nào tham dự hơn một môn học trong cùng một thời gian.
* Phòng học có sức chứa và điều kiện để tổ chức dạy môn học đó. Chỉ có một môn học được tổ chức tại một phòng học trong một khoảng thời gian cho trước.
* Các môn học thường được học từ 2 đến 4 tiết mỗi ngày.

Ràng buộc mềm:

* Hạn chế số sinh viên phải tham dự nhiều môn học liên tiếp nhau trong cùng một ngày.
* Hạn chế số sinh viên chỉ học đúng một môn học trong một ngày…
* Số buổi dạy của mỗi giáo viên là không quá nhiều (gom ngày dạy).

## Các phương pháp tiếp cận hiện nay

Trước hết, chúng ta cùng điểm qua các giải thuật truyền thống:

* Giải thuật vét cạn (tìm kiếm theo chiều rộng hoặc chiều sâu) về mặt tài nguyên luôn tìm được nghiệm nếu bài toán có nghiệm. Nhưng trên thực tế, các bài toán thời khóa biểu không nên áp dụng phương pháp này, vì ta phải phát triển không gian cực lớn trước khi đi đến trạng thái đích. Do các hạn chế về thời gian, dung lượng bộ nhớ, không cho phép ta thực hiện được.
* Giải thuật leo đồi (Hill Climbing) sử dụng kỹ thuật nâng cấp lặp, áp dụng cho một số điểm đơn (điểm hiện hành) trong không gian tìm kiếm. Mỗi lần nâng cấp, một điểm trong lân cận của điểm hiện hành được chọn làm điểm kế tiếp, nếu nó cho kết quả tốt hơn của hàm mục tiêu. Việc tìm kiếm kết thúc khi không thể nâng cấp được nữa. Rõ ràng, giải thuật leo đồi chỉ cho kết quả tối ưu cục bộ, kết quả này phụ thuộc vào sự chọn lựa điểm xuất phát, mặt khác ta không có được thông tin về sai số giữa tối ưu cục bộ tìm được và tối ưu toàn cục. Mặc dù đã cải tiến bằng cách tăng số lượng điểm xuất phát (chọn ngẫu nhiên hoặc chọn theo kết quả của lần chạy trước), nhưng khi có nhiều cực trị địa phương thì khả năng tìm được kết quả tối ưu toàn cục của giải thuật leo đồi còn rất thấp.

Đã có nhiều giải thuật được đề xuất để giải các bài toán thời khóa biểu. Các giải thuật này tìm được lời giải gần tối ưu và là một trong các xu thế phát triển hiện nay đối với các bài toán chưa thể tìm được lời giải tối ưu thực sự. Các giải thuật này đều mô phỏng theo tự nhiên như giải thuật luyện kim, giải thuật di truyền, phương pháp tính toán tiến hóa, giải thuật hệ kiến… Hiện nay giải thuật di truyền và giải thuật tối ưu đàn kiến là hai phương pháp được sử dụng nhiều nhất để giải quyết bài toán lập thời khóa biểu. Xét về thời gian thực hiện, chi phí thực hiện thì giải thuật tối ưu đàn kiến tốt hơn nhưng cũng phức tạp hơn giải thuật di truyền. Trên thực tế việc lập thời khóa biểu chỉ diễn ra khoảng hai đến ba lần trong một năm phụ thuộc vào chương trình đào tạo của từng trường cụ thể, nên thời gian và chi phí cũng không ảnh hưởng nhiều tới việc lập thời khóa biểu, vì vậy trong đồ án này để đơn giản em sử dụng giải thuật di truyền để giải quyết bài toán lập thời khóa biểu cho đào tạo tín chỉ.

## 1.4 Cơ sở lý thuyết

## 1.4.1 Ngôn ngữ lập trình Javascript

### 1.4.1.1 Giới thiệu về NodeJS

* Node.js là một mã nguồn mở, một môi trường cho các máy chủ và ứng dụng mạng
* Node.js dữ dụng Google V8 JavaScript engine để thực thi mã và một tỷ lệ lớn các modun cơ bản được viết bằng JavaScript. Các ứng dụng node.js thì được viết bằng JavaScript.
* Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Websever mà không cần phần mềm như Ngĩn, Apche HTTP Sever hoặc IIS.
* Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao
* Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing)

### 1.4.1.2 Ứng dụng

* Xây dựng websocket server (Chat server)
* Hệ thống Notification (Giống như facebook hayTwitter)
* Ứng dụng upload file trên client
* Các máy chủ quảng cáo
* Các ứng dụng dữ liệu thời gian thực khác.

### 1.4.1.3 Ưu điểm

* Đặc điểm nổi bật của Node.js là nó nhận và xử lý nhiều kết nối chỉ với một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP. Ngoài ra, tận dụng ưu điểm non-blocking I/O của Javascript mà Node.js tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP
* JSON APIs Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.
* Ứng dụng trên 1 trang( Single page Application) Nếu bạn định viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng bạn định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của bạn.
* Shelling tools unix NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.
* Streamming Data (Luồng dữ liệu) Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả xử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
* Ứng dụng Web thời gian thực Với sự ra đời của các ứng dụng di động & HTML 5 nên Node.js rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter,…

Trong bài chúng tôi đã dùng các thư viện:

* Thư viện 1 là Knex để kết nối đến CSDL và thực hiện việc đọc ghi dữ liệu vào CSDL
* Thư viện 2 là Express làm framework cấu trúc lên ứng dụng web

Express.js là một trong những framework quan trọng nhất của Node.js. Đây là một framework tối giản để xây dựng một loạt các ứng dụng web và di động cũng như các giao diện lập trình ứng dụng (API). Rất nhiều ứng dụng phổ biến và các trang web như MySpace, [Geekli.st](http://geekli.st/), Klout, [Segment.io](http://segment.io/) và Yummly được xây dựng bởi Express.js. Express.js cung cấp các tính năng khác nhau, như đơn giản hóa nhiều định tuyến, tích hợp cơ sở dữ liệu và ...

* Thư viện 3 là Lodash để xử lý dữ liệu
* Thư viện 4 là body-parser là thư việc để lấy dữ liệu từ các request

## 1.4.2 Visual Studio Code

Visual Studio Code hay viết tắt là VS Code hay VSC là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên.VS Code rất nhanh và nhẹ, và nó cũng rất mạnh! Microsoft đã thiết kế VS Code như một trình soạn thảo mã nguồn đa nền tảng để viết các ứng dụng web và Cloud.

* **Visual Studio Code là trình soạn thảo Cross-Platform**:Visual Studio Code là trình soạn thảo đa nền tảng, mã nguồn mở và miễn phí, hoạt động trên Windows, Linux và macOS.
* VS Code hỗ trợ vô số ngôn ngữ lập trình: Nó hỗ trợ các ngôn ngữ lập trình như: Python, JavaScript, HTML, CSS, TypeScript, C ++, Java, PHP, Go, C, PHP, SQL, Ruby, Objective-C và thậm chí nhiều hơn thế nữa ...
* VS Code tích hợp tính năng Debug
* VS Code tích hợp sẵn Git
* Visual Studito Code cung cấp sẵn tính năng IntelliSense cho các ngôn ngữ lập trình JavaScript, CSS, HTML, TypeScript, JSON, Sass và Less.
* Visual Studio Code cũng cung cấp các tính năng để quản lý code như Go to Definition, Peek definition, Find all references and rename Symbol.

## 1.4.3 Hệ quản trị cơ sở dữ liệu

MariaDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở hoàn toàn miễn phí được xây dựng bởi một số tác giả sang lập MySQL và hiện tại đang được đông đảo cộng đòng các nhà phát triển sử dụng.

MariaDB có sẵn để cài đặt trên[Linux](https://blog.tinohost.com/he-dieu-hanh-linux-la-gi/) [CentOS](https://blog.tinohost.com/centos-la-gi-tim-hieu-ve-he-dieu-hanh-centos/), [Ubuntu](https://blog.tinohost.com/he-dieu-hanh-ubuntu/) và [Window](https://blog.tinohost.com/he-dieu-hanh-windows-la-gi-uu-va-khuyet-diem/). MariaDB được phát triển nhằm thay thế công nghệ cơ sở dữ liệu MySQL, vì thế nó tương thích và cho một hiệu suất cao hơn so với MySQL.

MariaDB có các phiên bản cho các hệ điều hành khác nhau: Windows, Linux,.. với các gói cài đặt tar, zip, MSI, rpm cho cả 32bit và 64bit. Hiện tại phiên bản mới nhất của MariaDB là 10.1.

Ngoài việc hỗ trợ các storage engines cơ bản như MyISAM, BLACKHOLE, CSV, MEMORY, ARCHIVE, and MERGE thì trên MariaDB còn bổ sung thêm các storage engines sau: Aria (được xem như một phiên bản cập nhập của MyISAM), XtraDB (thay thế cho InnoDB), FederatedX, OQGRAPH, SphinxSE, IBMDB2I, Spider, PBXT, ….

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## 2.1 Giải thuật di truyền

Hoạt động của GAs đơn giản là việc mô phỏng sự tiến hóa và chọn lọc tự nhiên bằng máy tính bắt đầu từ một quần thể ngẫu nhiên . Bên cạnh đó để tối ưu ta cần hàm lượng giá hoặc hàm thích nghi để chọn cá thể tốt và loại bỏ cá thể xấu .

Thuật toán di truyền ( GAS ) khác với kĩ thuật tối ưu khác ở chỗ:

- GAs làm việc với bộ mã của biển chứ không phải làm việc trực tiếp trên biến.

- Hầu hết các kĩ thuật tối ưu thông thường tìm kiếm từ một định , trong khi đó GAs luôn hoạt động trên tập hợp định ( điểm tối ưu ) , điều này là một ưu điểm của GAs giúp tăng cơ hội tiếp cận tối ưu toàn cục và tránh hội tụ sớm tại điểm cục bộ địa phương.

- GAS đánh giá hàm mục tiêu để phục vụ quá trình tìm kiếm , vì vậy có thể ứng dụng cho bất kì bài toán tối ưu nào ( liên tục hay rời rạc ).

- GAs thuộc lớp các thuật toán xác suất , các thao tác cơ bản của GAs dựa trên khả năng tích hợp ngẫu nhiên trong quá trình xử lý.

Vì vậy trong bài toán xếp lịch này, chúng tôi áp dụng giải thuật di truyền để tìm kiếm lời giải tối ưu.

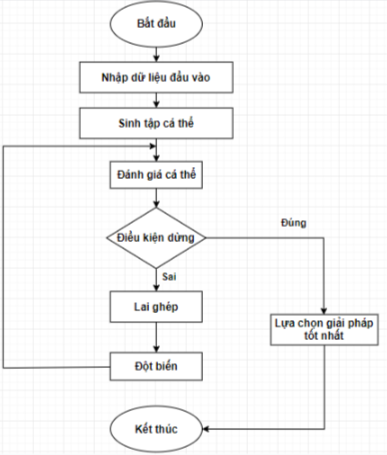
Thuật toán di truyền là thuật toán tối ưu ngẫu nghiên dựa trên cơ chế chọn lọc tự nhiên và tiến hóa di truyền. Thuật toán này hình thành trên quan niệm cho rằng, quá trình tiến hóa tự nhiên là quá trình hoàn hảo nhất và tự nó đã mang tính tối ưu. Trong lĩnh vực tối ưu hóa, thuật toán di truyền đã phát triển nhanh chóng và được áp dụng trong nhiều lĩnh vực khác nhau như bài toán người du lịch (TSP), bài toán điều phối xe (VRP), bài toán xử lí ảnh, bài toán nhận dạng hệ thống và điều khiển... [1].

Giải thuật di truyền (hay giải thuật tiến hóa nói chung) là một trong những phát triển quan trọng của những nhà nghiên cứu về tính toán ứng dụng cuối thế kỷ trước trong việc giải xấp xỉ các bài toán tối ưu toàn cục. Việc khai thác nguyên lí tiến hóa như là một định hướngheuristics đã giúp cho giải thuật di truyền giải quyết hiệu quả các bài toán tối ưu (với các lời giải chấp nhận được) mà không cần sử dụng các điều kiện truyền thống (liên tục hay khảvi) như là điều kiện tiên quyết.Một trong những đặc tính quan trọng của giải thuật di truyền là làm việc theo quần thể các giải pháp. Việc tìm kiếm bây giờ được thực hiện song song song trên nhiều điểm (multipoints).Tuy nhiên, đây không phải là là thuật toán tìm kiếm đa điểm đơn thuần vì các điểm có tươngtác với nhau theo nguyên lí tiến hóa tự nhiên. Trong ngữ cảnh sử dụng giải thuật di truyền,người ta có thể dùng khái niệm “cá thể” tương đương với khái niệm “giải pháp” [1].

Có một số khái niệm đặc trưng trong giải thuật di truyền như sau:

* Cá thể, nhiễm sắc thể. Một cá thể trong giải thuật di truyền biểu diễn một giải pháp của bài toán. Giới hạn trong giải thuật di truyền, ta quan niệm một cá thể có một nhiễm sắc thể (NST). Do đó khái niệm cá thể và nhiễm sắc thể trong giải thuật di truyền coi như là tương đương. Một NST được tạo thành từ nhiều gen, mỗi gen có thể có các giá trị khác nhau để quy định một tính trạng nào đó. Trong giải thuật GA, một gen được coi như một phần tử trong chuỗi NST.
* Quần thể là một tập hợp các cá thể có cùng một số đặc điểm nào đó. Trong giải thuật di truyền ta quan niệm quần thể là một tập các giải pháp của một bài toán.

Sơ đồ khối của thuật toán như sau:



Hình 2. 1 Sơ đồ khối giải thuật di truyền đơn giản (GA). [1]

Các bướccơ bản của giải thuật di truyền được mô tả như sau:

Bước 1: Nhập dữ liệu đầu vào

Bước 2: Sinh tập các cá thể

Bước 3: Chọn các thể tốt dựa vào việc đánh giá các cá thể. Nếu điều kiện dừng đúng thì kết thúc thuật toán.

Bước 4: Lai ghép các các thể tốt chọn được ở bước 3.

Bước 5: Thực hiện đột biến nhiễm sắc thể.

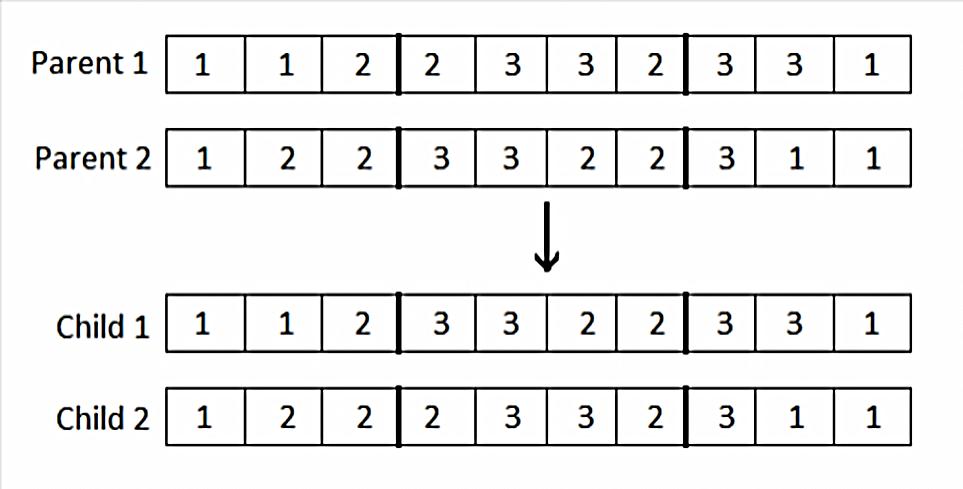
Bước 6: Quay lại bước 3.

**Biểu diễn giải pháp:** Trong giải thuật di truyền, mỗi thành viên của quần thể là một chuỗi nhị phân có độ dài là 1, tương ứng với một mã hóa nghiệm của bài toán. Mỗi chuỗi đó được xem như là một nhiễm sắc thể. Trong hầu hết các trường hợp, quần thể này đều được sinh ra một cách ngẫu nhiên. Sau khi sinh ra ngẫu nhiên các quần thể sẽ được đem đi đánh giá và chọn lọc.

**Lựa chọn**: Việc lựa chọn các cá thể được thực hiện khi chúng ta cần một số lượng cá thể để thực hiện sinh sản ra thế hệ sau. Mỗi cá thể có một giá trị thích nghi – gọi là fitness. Giá trị này được dùng để quyết định xem việc ta lựa chọn cá thể nào. Thông thường, cá thể tốt nhất sẽ có số điểm thấp nhất hoặc lớn nhất. Một số phương pháp lựa chọn thường dùng bao gồm:

* *Roulette wheel*: Dựa trên xác suất (tỷ lệ thuận với giá trị hàm thích nghi) để lựa chọn cá thể.
* *Giao đấu (nhị phân)*: Chỉ định ngẫu nhiên 2 cá thể, sau đó chọn cá thể tốt hơn trong hai cá thể đó.

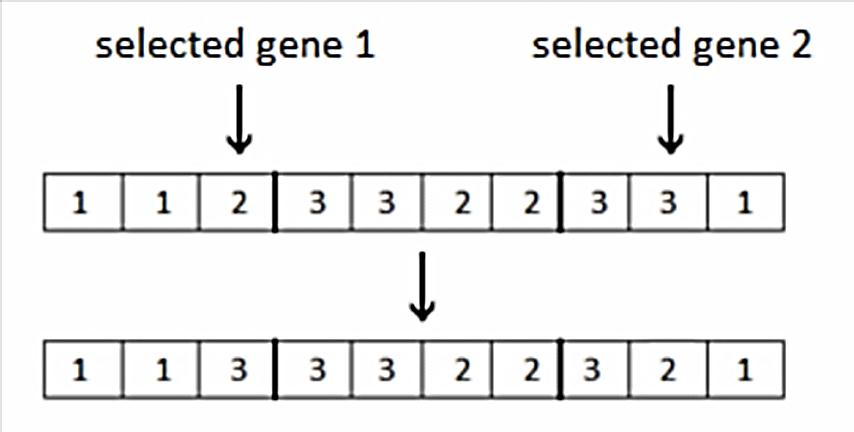
**Lai ghép**: Toán tử lai ghép được áp dụng với mục đích sinh ra các cá thể con mới từ các cá thể cha mẹ, thừa hưởng các tính trạng tốt từ cha mẹ. Trong ngữ cảnh tìm kiếm thì toán tử lai ghép thực hiện tìm kiếm xung quanh khu vực của các giải pháp biểu diển bởi các cá thể cha mẹ. Quá trình lai ghép được sử dụng để tạo ra cá thể mới. Chọn một cách ngẫu nhiên một phân đoạn gen trên hai NST được lựa chọn, thực hiện tráo đổi để sinh ra hai NST con.



Hình 2. 2 Minh họa quá trình lai ghép trong thuật toán di truyền

**Đột biến**: Tương tự với lai ghép, đột biến cũng là toán tử mô phỏng về hiện tượng đột biến trong sinh học. Kết quả của đột biến là thường sinh ra các cá thể mới khác biệt so với cá thể cha mẹ. Trong ngữ cảnh tìm kiếm, toán tử đột biến nhằm đưa quá trình tìm kiếm ra khỏi khu vực cục bộ địa phương.

Ví dụ, quá trình đột biến được thực hiện bằng cách tráo đổi ngẫu nhiên giá trị tại cùng một vị trí gen trên hai NST con được tạo ra ở giai đoạn lai ghép.



Hình 2. 3 Minh họa quá trình đột biến trong thuật toán di truyền [2]

**Điều kiện kết thúc:**

Thoát ra quá trình tiến hóa quần thể, dựa vào bài toán mà có các cách kết thúc vấn đề khác nhau một khi đạt đến mức yêu cầu. Một vài trường hợp thông thường như sau:

* Kết thúc theo kết quả: một khi đạt đến mức giá trị yêu cầu thì chấm dứt ngay quá trình thực hiện.
* Kết thúc dựa vào số thế hệ: chọn số thế hệ, quá trình sẽ dừng đúng ngay số thế hệ đã qui định trước, không cần biết kết quả như thế nào.
* Tính theo thời gian: không cần biết đã bao nhiêu thế hệ hay kết quả nào, chỉ dựa vào số giờ qui định mà kết thúc.
* Tổ hợp: dùng nhiều phương án khác nhau cho vấn đề, chẳng hạn như: chạy theo số thế hệ xong sau đó đánh giá cho chạy theo kết quả, hoặc ngược lại.

# **CHƯƠNG 3: BÀI TOÁN XẾP LỊCH DẠY VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## 3.1 Phân tích thiết kế hệ thống

### 3.1.1 Nhiệm vụ cơ bản

* Quản lý danh sách môn học
* Quản lý danh sách lớp học phần
* Quản lý danh sách giảng viên
* Quản lý danh sách phòng học

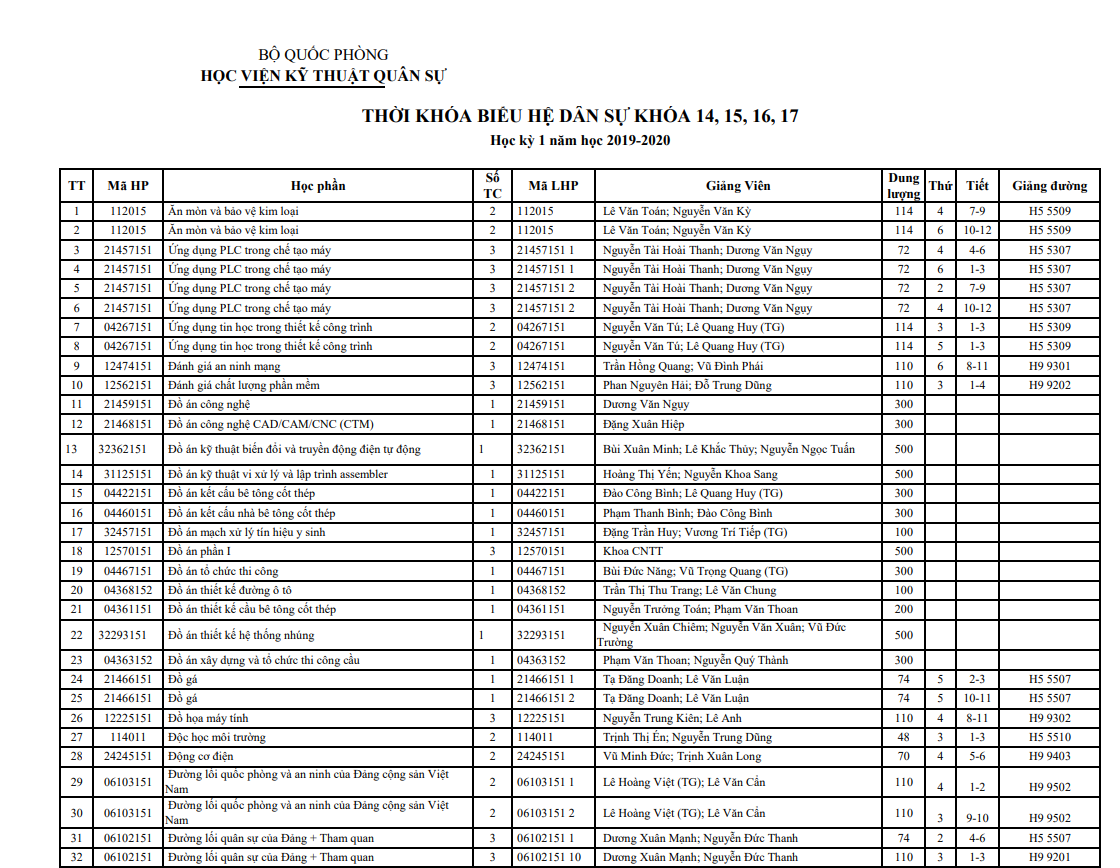
### 3.1.2 Quy trình xếp lịch dạy theo đào tạo tín chỉ và quy tắc quản lý

Quy trình xử lý:

* Dự kiến danh sách giảng viên giảng dạy
* Vào mỗi đầu kỳ học, các khoa sẽ thống kê và lập danh sách các giảng viên giảng dạy trong kỳ học và gửi danh sách giảng viên dự kiến lên phòng đào tạo.
* Lập danh sách môn học học dự kiến
* Khoa sẽ lập danh sách các môn học dự kiến cần giảng dạy trong học kỳ này và từ đó lên kế hoạch phân công dạy phù với với giảng viên gửi **lịch phân công dạy(MB1)** lên phòng đào tạo
* Thống kê và lập danh sách lượng sinh viên
* Bộ phận quản lý điểm sinh viên sẽ thống kê và lập danh sách lượng sinh viên theo học các môn trong học kỳ tới. Sau khi lập xong, danh sách sẽ được gửi danh sách lượng sinh viên lên phòng đào tạo.
* Việc thống kê và lập danh sách lượng sinh viên được bộ phân quản lý điểm sinh viên thực hiện trên danh sách sinh viên của từng ngành từng khóa, số lượng sinh viên sẽ được tính như sau: số sinh viên sẽ là tổng số sinh viên của các ngành có môn học tương ứng cộng thêm số lượng sinh viên đã học môn học đó mà chưa qua.
* Thống kê và lập danh sách giảng đường dự kiến
* Phòng tổ chức hành chính sẽ thống kê và lập danh sách các giảng đường dự kiến có thể học trong kỳ tới và sau đó gửi danh sách giảng đường dự kiến lên cho phòng đào tạo.
* Lập danh sách lớp môn học dự kiến
* Sau khi lập xong hai danh sách lượng sinh viên và danh sách môn học, khoa và bộ phân quản lý điểm sinh viên sẽ gửi danh sách môn học dự kiến lại cho phòng đào tạo, phòng đào tạo sẽ lập danh sách lớp môn học.
* Sau khi có được đủ ba danh sách bao gồm: danh sách lớp môn học, danh sách giáo viên dự kiến, danh sách phòng học dự kiến, phòng đào tạo tiến hành xếp lịch dạy.

### 3.1.3 Mẫu biểu

Bảng 3. 1Thời khóa biểu dân dự khóa 14,15, 16,17



Bảng 3. 2 Bảng phân công dạy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Môn | Số TC | Giảng viên | Ghi chú |
| 1 | Xử lý tín hiệu số(CNTT) | 3 | Vi Bảo Ngọc |  |
| 2 | Công nghệ đa phương tiện | 3 | Trần Cao Trưởng |  |
| 3 | Phương pháp nguyên cưu IT | 4 | Vi Bảo Ngọc  Trần Cao Trường Nguyễn Trung Tín |  |
| 4 | Phát triển trò chơi trực tuyến | 3 | Nguyễn Trung Tín |  |
| 5 | Lập trình trò chơi và mô phỏng | 3 | Hà Trí Trung |  |
| 6 | Lý thuyết hệ điều hành | 3 | Hà Trí Trung |  |
| 7 | Công nghệ XML và WEB ngữ nghĩa | 3 | Nguyễn Trung Tín |  |
| 8 | Thiết kế trò chơi số | 3 | Nguyễn Trung Tín |  |
| 9 | Lý thuyết hệ điều hành | 3 | Hà Chí Trung |  |
| 10 | Trí tuệ nhân tạo | 3 | Trần Cao Trưởng |  |
| 11 | Quản lý dự án hệ thống thông tin | 3 | Hà Đại Dương |  |
| 12 | Khai phá dữ liệu | 3 | Hồ Nhật Quang |  |
| 13 | Công nghệ Client/Server | 3 | Hồ Nhật Quang |  |
| 14 | Công nghệ phần mềm | 3 | Nguyễn Quốc Khánh |  |
| 15 | Thiết kế giao diện người sử dụng | 3 | Nguyễn Thị Hiền |  |
| 16 | Lập trình nâng cao | 3 | Nguyên Thị Hiền |  |
| 17 | Công nghệ lập trình tích hợp | 3 | Nguyễn Quốc Khánh  Vũ Văn Trường |  |
| 18 | Ngôn ngữ lập trình 1 | 2 | Hồ Nhật Quang  Vũ Văn Trường |  |
| 19 | Ngôn ngữ lập trình 2 | 2 | Nguyễn Quốc Khánh  Nguyễn Thị Hiền  Phan Văn Việt |  |
| 20 | Phân tích và mô hình hóa phần mềm | 3 | Phan Văn Việt |  |
| 21 | Phát triển phần mềm di động | 3 | Phan Văn Việt |  |
| 22 | Thiết kế và xây dựng phần mềm | 3 | Phan Văn Việt |  |
| 23 | Đánh giá chất lượng phần mềm | 3 | Hà Đại Dương |  |
| 24 | Cơ sở dữ liệu | 4 | Nguyễn Hoài Anh  Đỗ Thị Mai Hường  Nguyễn Văn Giang |  |
| 25 | Cơ sở dữ liệu nâng cao | 3 | Nguyễn Mậu Uyên |  |
| 26 | Cơ bản về diên toán đám mây | 3 | Trần Văn An  Nguyễn Văn Giang |  |
| 27 | Công nghệ web nâng cao | 3 | Trần Văn An |  |
| 28 | Các hệ thống phân tán | 3 | Hoa Tất Thắng |  |
| 29 | Phân tích và thiết kế hệ thống | 3 | Nguyễn Hoài Anh |  |
| 30 | Phân tích thiết kế giải thuật | 3 | Hoa Tất Thắng  Nguyễn Mậu Uyên  Tống Minh Đức |  |
| 31 | Phát triển hệ thống quản trị doanh nghiệp | 3 | Trần Văn An |  |
| 32 | Thực tập cơ sở dữ liệu | 3 | BM HTTT |  |
| 33 | Thực tập kỹ thuật lập trình | 3 | BM HTTT |  |
| 34 | Nhập môn cơ sở dữ liệu lớn | 3 | Nguyễn Văn Việt |  |
| 35 | Hệ thống thống tin quản lý | 3 | Hà Đại Dương |  |
| 36 | Đảm bảo an toàn thông tin | 3 | Phan Việt Anh  Trần Hồng Quang  Nguyễn Việt Hùng  Cao Văn Lợi |  |
| 37 | Thực tập công nghệ | 2 | Nguyễn Văn Quân |  |
| 38 | Chuyên đề nâng cao lập trình trên các cấu trúc dữ liệu | 2 | Nguyễn Mạnh Hùng |  |
| 39 | Đánh giá an ninh mạng | 4 | Trần Hồng Quang |  |
| 40 | An ninh mạng | 2 | Nguyễn Trung Thành |  |
| 41 | Thương mại điện tử | 3 | Nguyễn Kim Thanh |  |
| 42 | Lập trình mạng | 3 | Nguyễn Quang Uy |  |
| 43 | Quản trị mạng | 3 | Nguyễn Trung Thành |  |
| 44 | Kỹ thuật liên mạng | 3 | Lưu Hồng Dũng |  |
| 45 | Thiết kế mạng | 3 | Nguyễn Trung Thành |  |

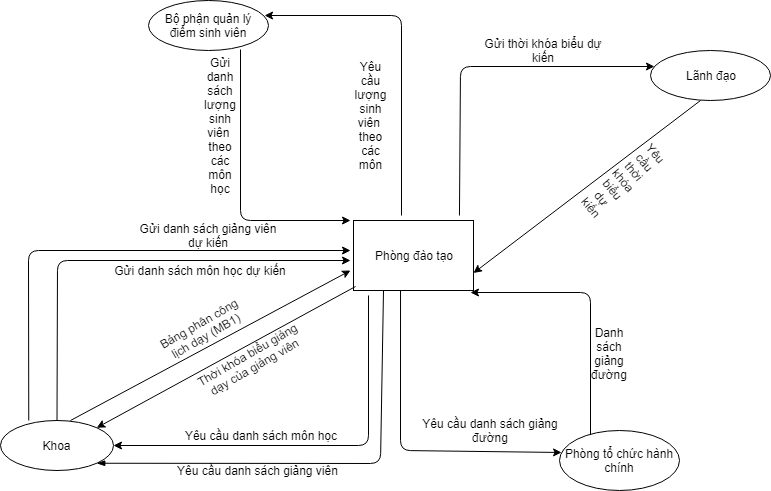
### 3.1.4 Mô hình tiến trình nghiệp vụ

- Ký hiệu

* Bộ phận trong hệ thống

|  |
| --- |
| Bộ phần đào tạo |
| Lập danh sách lớp học phần  Gửi thời khóa biểu dự kiến lên lãnh đạo |

* Tác nhân tác động vào hệ thống
* Luồng thông tin
* Mô hình



Hình 3. 1 Mô hình tiến trình nghiệp vụ 1

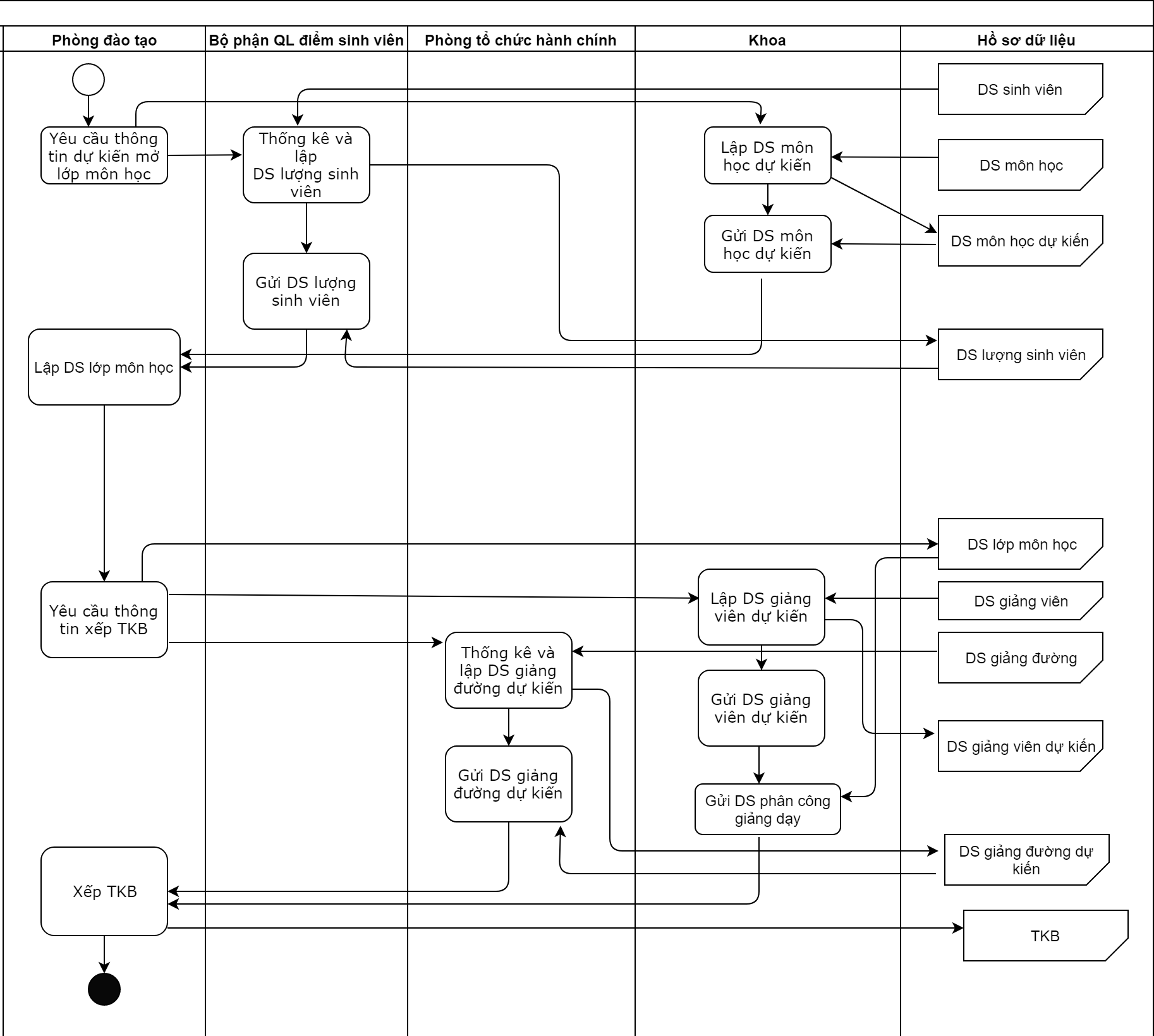
### 3.1.5 Biểu đồ hoạt động

- Ký hiệu

* Bắt đầu/ Kết thúc



* Công việc
* Điều khiện rẽ nhãnh
* Giấy tờ giao dịch
* Kho dữ liệu
* Luồng công việc/luồng dữ liệu
* Biểu đồ hoạt đông



Hình 3. 2 Biểu đồ hoạt động

### 3.1.6 Mô hình hóa chức năng nghiệp vụ chi tiết

#### 3.1.6.1 Xác định chức năng

* Các bước thực hiện:
* Bước 1: Đọc quy trình xử lý, gạch chân động từ + bổ ngữ, ta được danh sách các chức năng có thể có của hệ thống.
* Bước 2: Trong danh sách các chức năng của bước 1, tìm và loại bỏ những chức năng trùng lặp.
* Bước 3: Trong danh sách những chức năng ở bước 2, gom nhóm những chức năng đơn giản lại.
* Bước 4: Trong danh sách những chức năng ở bước 3, tìm và loại bỏ những chức năng không có ý nghĩa đối với hệ thống.
* Bước 5: Chuẩn hóa lại tên cho hợp lý.
* Áp dụng:
* Từ quy trình xử lý, ta xác định được các động từ + bổ ngữ sau:

1. Thống kê và lập danh sách các giảng viên giảng dạy
2. Gửi danh sách giảng viên dự kiến
3. Lập danh sách các môn học dự kiến
4. Lên kế hoạch phân công lịch dạy
5. Gửi lịch phân công giảng dạy
6. Thống kê và lập danh sách lượng sinh viên
7. Gửi danh sách lượng sinh viên
8. Việc thống kê và lập danh sách lượng sinh viên
9. Thống kê và lập danh sách các giảng đường
10. Gửi danh sách giảng đường dự kiến
11. Lập danh sách lớp môn học
12. Xếp thời khóa biểu

* Các chức năng trùng lập của phần trên là:

Sau khi loại bỏ những chức năng trùng lặp trên, ta còn:

1. Thống kê và lập danh sách các giảng viên giảng dạy
2. Gửi danh sách giảng viên dự kiến
3. Lập danh sách các môn học dự kiến
4. Lên kế hoạch phân công lịch dạy
5. Gửi lịch phân công giảng dạy
6. Thống kê và lập danh sách lượng sinh viên
7. Gửi danh sách lượng sinh viên
8. Thống kê và lập danh sách lượng sinh viên
9. Thống kê và lập danh sách các giảng đường
10. Lập danh sách lớp môn học
11. Xếp thời khóa biểu

* Gom nhóm những chức năng đơn giản ở trên ta có:

|  |  |
| --- | --- |
| Chức năng đơn giản được gom nhóm | Tên sau khi gom |
| 1.Thống kế và lập danh sách các giảng viên giảng dạy  2.Gửi danh sách giảng viên dự kiến | Lập, gửi danh sách giảng viên dự kiến |
| 4.Lên kế hoạch phân công lịch dạy  5.Gửi lịch phân công giảng dạy | Gửi lịch phân công giảng dạy |
| 6.Thống kê và lập danh sách lượng sinh viên  7.Gửi danh sách lượng sinh viên | Lập, gửi danh sách lượng sinh viên |
| 9.Thống kê và lập danh sách các giảng đường  10.Gửi danh sách giảng đường dự kiến | Lập, gửi danh sách giảng đường dự kiến |

Sau khi gom nhóm ta có nhưng chức năng sau:

1. Lập, gửi danh sách giảng viên dự kiến
2. Lập danh sách các môn học dự kiến
3. Gửi lịch phân công giảng dạy
4. Lập, gửi danh sách lượng sinh viên
5. Lập, gửi danh sách giảng đường dự kiến
6. Lập danh sách lớp môn học
7. Xếp thời khóa biểu

Tìm và loai bỏ những chức năng không có ý nghĩa đối với hệ thống

Những chức năng sau khi đã loại bỏ là:

1. Lập, gửi danh sách giảng viên dự kiến
2. Lập danh sách các môn học dự kiến
3. Gửi lịch phân công giảng dạy
4. Lập, gửi danh sách lượng sinh viên
5. Lập, gửi danh sách giảng đường dự kiến
6. Lập danh sách lớp môn học
7. Xếp thời khóa biểu

Chuẩn hóa tên:

1. Lập, gửi danh sách giảng viên dự kiến
2. Lập danh sách môn học dự kiến
3. Gửi lịch phân công giảng dạy
4. Lập, gửi danh sách lượng sinh viên
5. Lập, gửi danh sách giảng đường dự kiến
6. Lập danh sách lớp môn học
7. Xếp thời khóa biểu

#### 3.1.6.2 Gom nhóm chức năng

### 3.1.7 Vẽ sơ đồ phân rã chức năng

* Ký hiệu
* Chức năng

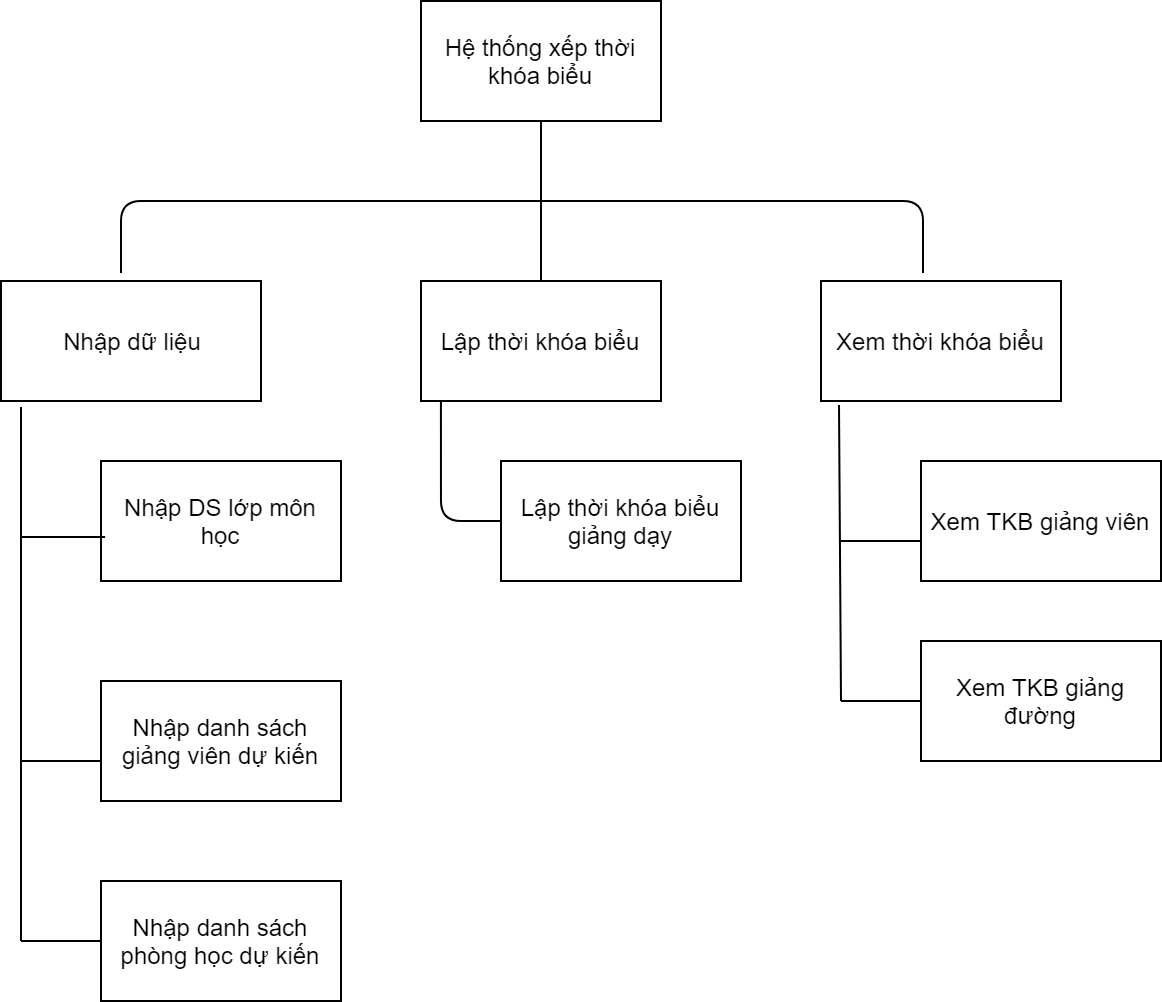
|  |
| --- |
| Nhập dữ liệu |

* Phân cấp



* Sơ đồ

Vì nội dung đồ án chỉ đề cập về vấn đề xếp lịch nên phần dự kiến đào tạo em sẽ bỏ qua trong các phần thiết kế tiếp theo của đồ án, em chỉ sử dụng nhưng dữ liệu cần thiết đó là danh sách lớp môn học, danh sách giảng đường dự kiến, danh sách giảng viên dự kiến.



Hình 3. 3 Biểu đồ phân rã chức năng

### 3.1.7 Danh sách hồ sơ dữ liệu sử dụng

- Danh sách lớp môn học

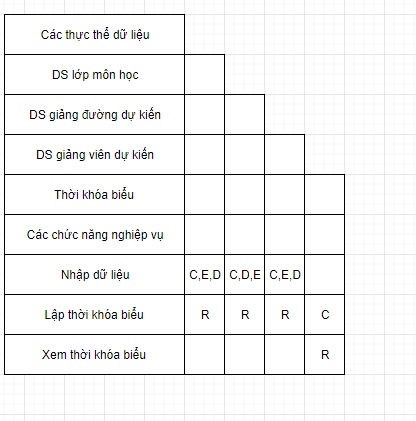
- Danh sách giảng đường dự kiến

- Danh sách giảng viên dự kiến

- Thời khóa biểu

### 3.1.8 Ma trận thực thể chức năng

Bảng Ma trận thực thể chức năng



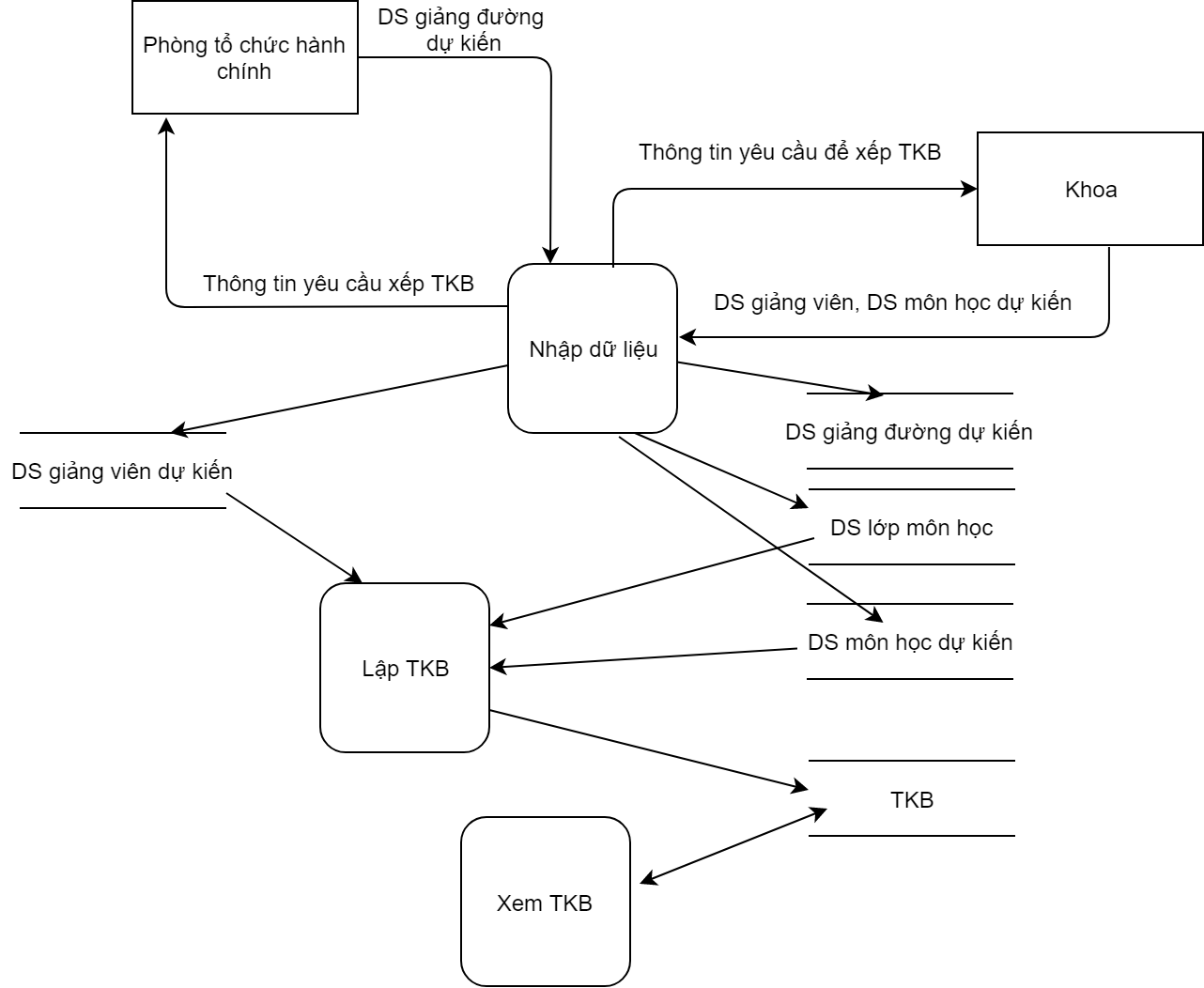
Bảng 3. 3 Ma trận thực thể 1

Ký hiệu:

C: Create, R: Read, E: Edit, D: Delete

### 3.1.9 Sơ đồ luồng dữ liệu

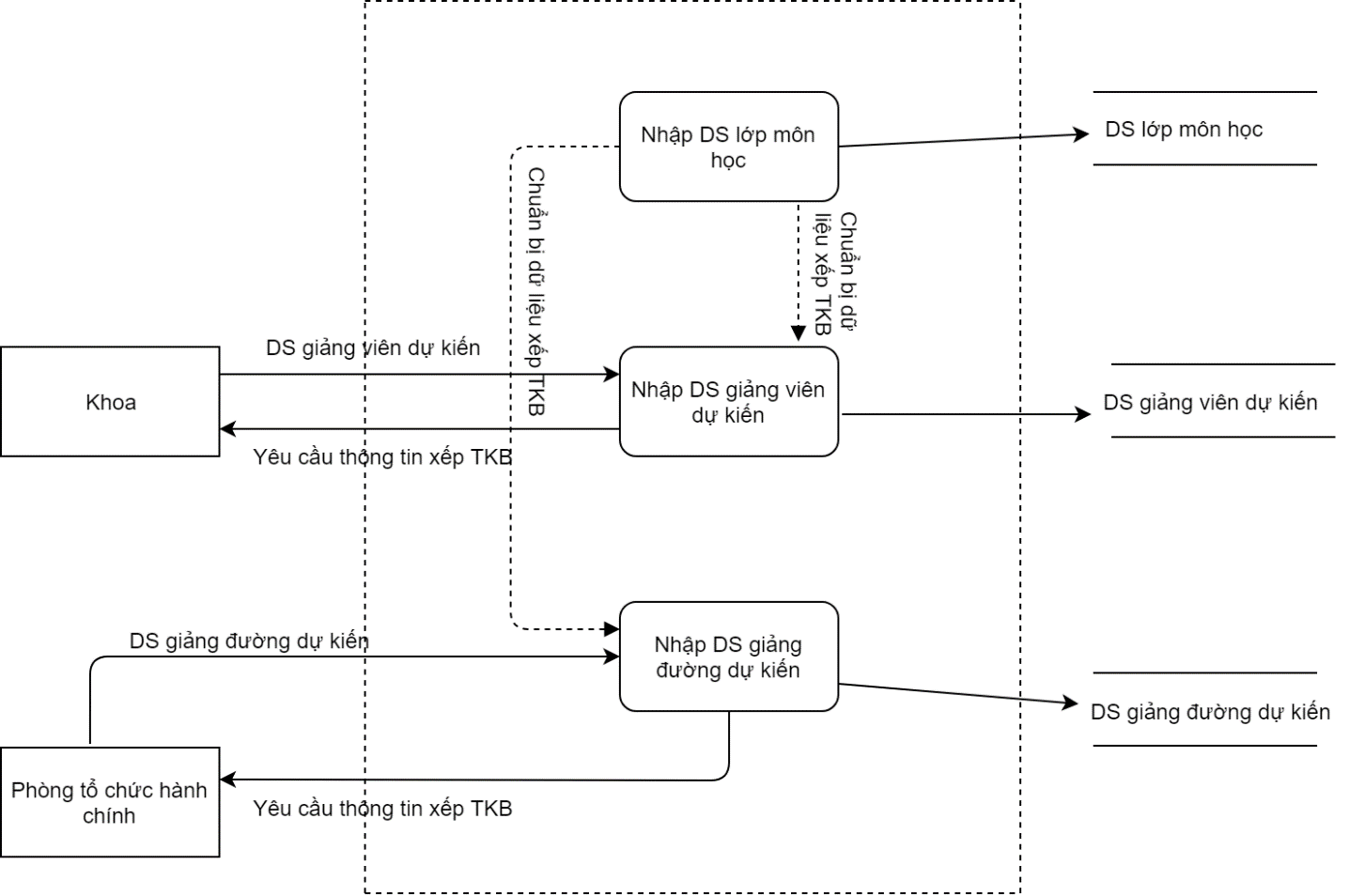
#### 3.1.9.1 DFD0



Hình 3. 4 DFD mức 0

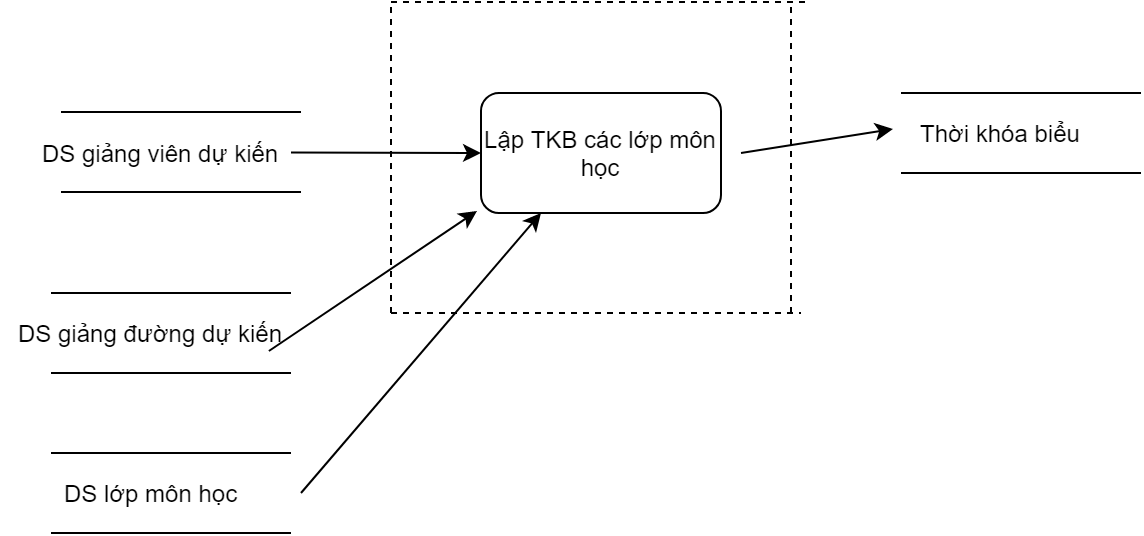
#### 3.1.9.2 Biểu đồ luồng dữ liệu mức 1

Tiến trình nhập dữ liệu



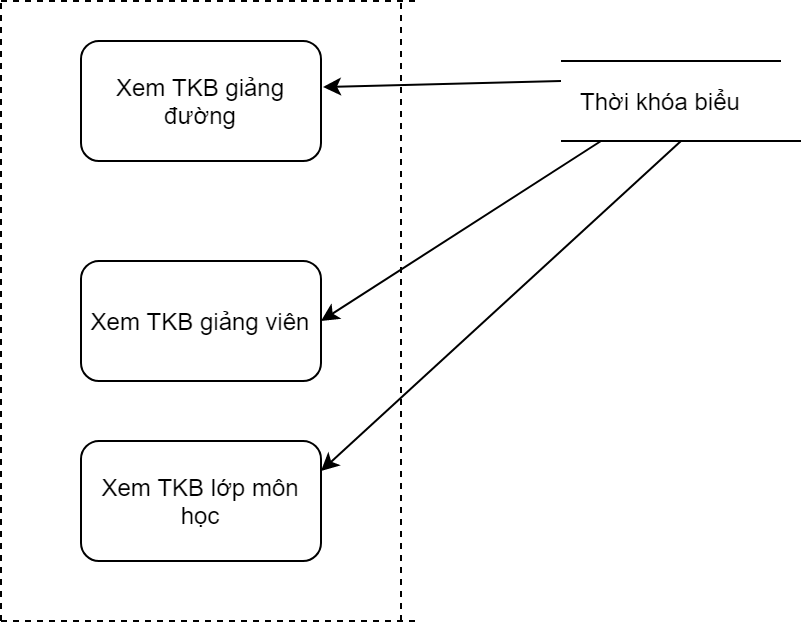
Hình 3. 5 Sơ đồ luông dữ liệu mức

* Tiến trình xếp thời khoá biểu



Hình 3. 6 Sơ đồ tiến trình xếp TKB

* Tiến trình xem thời khóa biểu



Hình 3. 7 Sơ đồ tiến trình xem TKB

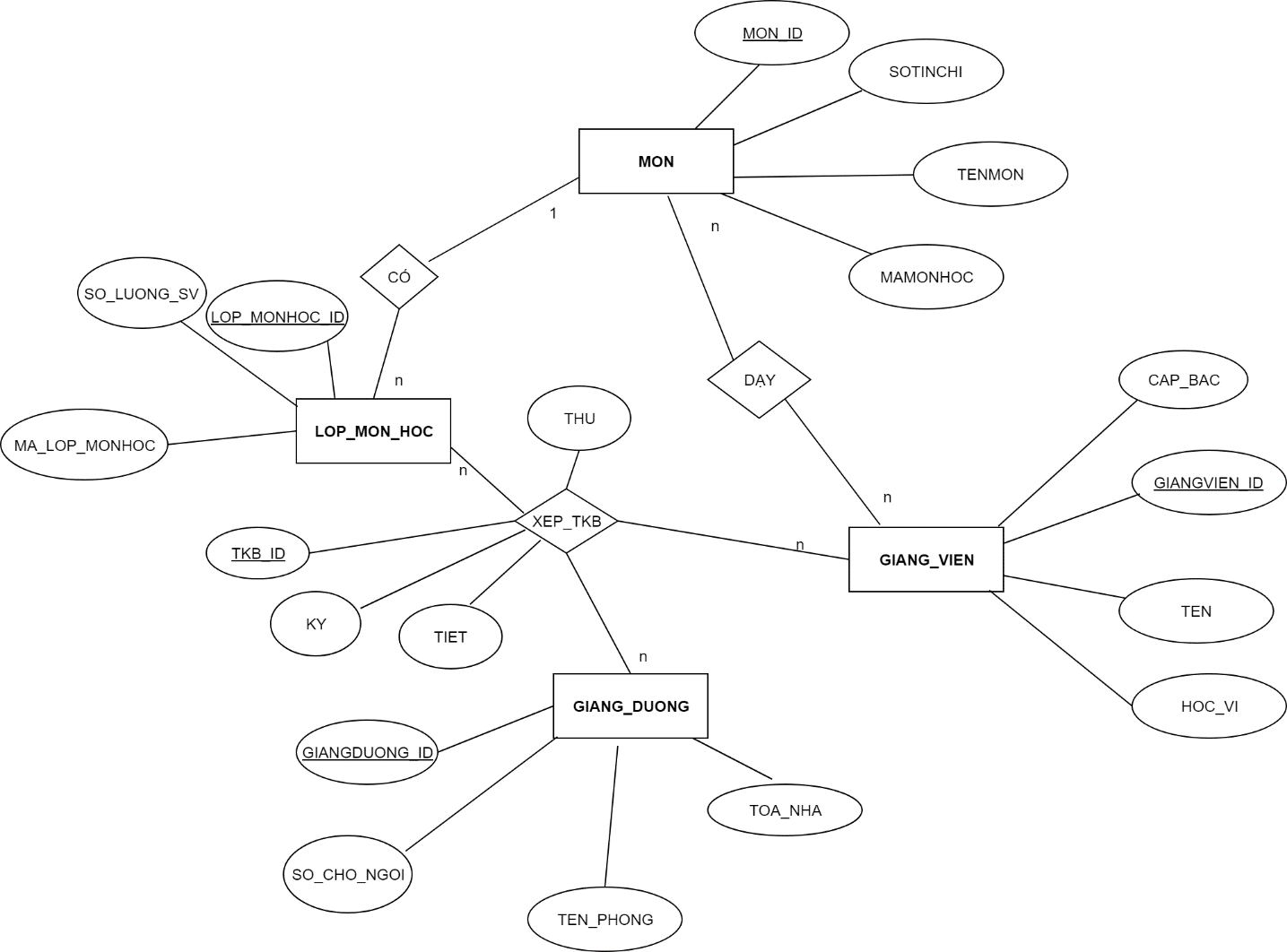
3.1.10 Mô hình liên kết thực thể(ER)

Xác định các kiểu thực thể, các thuộc tính và thuộc tính khóa của thực thể

Bảng 3. 4 Các thực thể, thuộc tính và khóa

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Kiểu thực thể | Thuộc tính | Thuộc tính khóa |
| 1 | Môn | Môn id, Mã môn học, Số tín chỉ | Môn id |
| 2 | Giảng viên | Giảng viên id, Tên Giảng viên, học vị, cấp bậc | Giảng viên id |
| 3 | Giảng đường | Giảng đường id, Tòa nhà, Tên phòng, Số chỗ ngồi | Giảng đường id |
| 4 | Lớp môn học | Lớp môn học ID, Số lượng sinh viên, lớp tên | Lớp môn học id |

Mô hình ER



Bảng 3. 5 Mô hình ER

**Tách các quan hệ:**

* Môn học

MON(MON\_ID, MAMONHOC,TENMON, SOTINCHI)

* Giảng viên

GIANGVIEN(GIANGVIEN\_ID, TEN, HOC\_VI, CAP\_BAC)

* Giảng đường

GIANGDUONG(GIANGDUONG\_ID, SO\_CHO\_NGOI, TOA\_NHA, TEN\_PHONG)

* Lớp môn học

LOP\_MON\_HOC(LOP\_MONHOC\_ID, SO\_LUONG\_SV, MA\_LOP\_MONHOC)

Biểu diễn cho các mỗi quan hệ

* Giảng viên DAY môn học thuộc dạng quan hệ nhiều với nhiều.

GV\_DAY\_MON(GIANG\_VIEN\_ID, MON\_ID)

Tạo ra một bản với hai khóa phụ lấy từ hai khóa của hai thực thể hình thành mối quan hệ

* Mon CO các lớp môn học thuộc dạng quan hệ một nhiều với một ở phía môn và nhiều ở phía lớp môn học

LOP\_MONHOC(LOP\_MONHOC\_ID, MA\_LOP\_MONHOC, SO\_LUONG\_SV, MON\_ID)

Thực thể lớp môn học lấy khóa chính của thực thể môn về làm thuộc tính

* XEP\_TKB cho lớp môn học, giảng viên, và giảng đường thuộc dạng quan hệ nhiều nhiều

TKB(TKB\_ID, THU, CA, LOP\_MONHOC\_ID, GIANGDUONG\_ID, GIANGVIEN\_ID, KY)

Tạo ra một bản có khóa chính và thuộc tính riêng đồng thời lấy khóa của cả ba thực thể tham gia vào quan hệ thuộc tính

Cơ sở dữ liệu

Các bảng dữ liệu

Bảng 3. 6 LOP\_MONHOC

Tên bảng: lophocphan

Dùng để lưu thông tin về các lớp môn học

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Kích cỡ | Mô tả |
| 1 | idlophocphan | Int | (10) | Id lớp môn học |
| 2 | Idmonhoc | Int | (10) | Id môn học |
| 3 | Sosinhvien | Int | (11) | Số lượng sinh viên của lớp |
| 4 | Malophocphan | Varchar | (255) | Mã lớp học phần |

Bảng 3. 7 MONHOC

Tên bảng: monhoc

Dùng để lưu thông tin về các môn học

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Kích cỡ | Mô tả |
| 1 | idmonhoc | Int | (10) | Id môn học |
| 2 | mamon | Varchar | (255) | Mã môn học |
| 3 | tenmon | Varchar | (255) | Tên môn học |
| 4 | sotinchi | int | (11) | Số tín chỉ |

Bảng 3. 8 GIANGDUONG

Tên bảng: monhoc

Dùng để lưu thông tin về các giảng đường

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Kích cỡ | Mô tả |
| 1 | idgiangduong | Int | (10) | Id giảng đường |
| 2 | toanha | Varchar | (255) | Tên tòa nhà |
| 3 | tenphong | Varchar | (255) | Tên phòng |
| 4 | sochongoi | int | (11) | Số chỗ ngồi |

Bảng 3. 9 GIANGVIEN

Tên bảng: giangvien

Dùng để lưu thông tin về các giảng viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Kích cỡ | Mô tả |
| 1 | idgiangvien | Int | (10) | Id giảng viên |
| 2 | ten | Nvarchar | (255) | Tên môn học |
| 3 | capbac | Varchar | (255) | Cấp bậc |
| 4 | hocvi | Varchar | (255) | Học vị |

Bảng 3. 10 DAY

Tên bảng: phancongday

Dùng để lưu thông tin về phan cong day

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Kích cỡ | Mô tả |
| 1 | idmonhoc | Int | (10) | Id môn học |
| 2 | idgiangvien | Int | (10) | Id giảng viên |

Bảng 3. 11TKB

Tên bảng: thoikhoabieu

Dùng để lưu thông tin về các lớp môn học

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Kích cỡ | Mô tả |
| 1 | idlophocphan | Int | (10) | Id môn học |
| 2 | idgiangvien | Int | (10) | Mã môn học |
| 3 | idgiangduong | Int | (10) | Tên môn học |
| 4 | thu | Int | (10) | Thứ |
| 5 | tiet | Int | (10) | Tiết |
| 6 | ky | Varchar | (255) | Kỳ |

## 3.2 Áp dụng giải thuật di truyền và bài toán

### 3.2.1 Các yêu cầu cơ bản của thời khóa biểu theo đào tạo tín chỉ

- Thời khóa biểu của giảng viên không trùng lặp: Thỏa mãn điều này có nghĩa là tại một thời điểm chỉ cho phép giáo viên dạy một lớp môn học tại một phòng học xác định nào đó.

Phòng học được sắp xếp để đảm bảo làm sao cho sức chứa của phòng học phải lớn hơn hoặc bằng tổng số sinh viên của lớp môn học tại phòng đó

Từ các yêu cầu cơ bản trên ta có các ràng buộc cho bài toán thời khóa biểu tín chỉ

**Ràng buộc cứng:**

* Không giảng viên nào dạy 2 lớp học phần trong cùng thời gian( vì tại một thời điểm một giảng viên chỉ dạy được 1 lớp)
* Giảng viên phải dạy đúng lớp và đúng môn học được giao ( vì giảng viên không được dạy lớp và môn học không được phân công hay dạy lớp và môn học không đúng chuyên môn)

Ví dụ: Giảng viên Nguyễn Trung Tín được phân công dạy 2 môn “Thiết kế trò chơi số” và “Công nghệ XML và WEB ngữ nghĩa”. Như vậy, không được phân công giảng viên Nguyễn Trung Tín dạy môn “Cơ sở dữ liệu”

* Mỗi giảng viên dạy 1 môn nào đó phải đủ số lớp theo phân công ( Vì mỗi giảng viên không được dạy thiếu số lớp được phân công.

Ví dụ: Giảng viên Nguyễn Trung Tín được phân công dạy môn “Công nghệ XML và WEB ngữ nghĩa” cho 2 lớp CNG&MP14 và CNDL14 thì phải phân công giảng viên Nguyễn Trung Tín dạy đủ 2 lớp đó, không được chỉ dạy một trong hai lớp CNG&MP14 hoặc CNDL14.

* Mỗi giảng viên phải dạy đủ số môn ( Vì mỗi giang viên không được dạy thiếu số môn được phân công)

Ví dụ: Giảng viên Nguyễn Trung Tín được phân công dạy môn “Thiết kế trò chơi số” và “Công nghệ XML và WEB ngữ nghĩa”. Như vậy giảng viên Nguyễn Trung Tín phải dạy đủ 2 môn đó, không được chỉ dạy một trong hai môn “Thiết kế trò chơi số” hoặc “Công nghệ XML và WEB ngữ nghĩa”.

* Mỗi giảng viên dạy môn học nào đó phải dạy đủ số tiết ( vì mỗi giảng viên dạy một môn học cho một lớp nào đó phải không được thiếu số tiết như phân công)
* Các lớp học đúng thời gian được phân công ( vì các lớp không được học sai thời gian phân công)
* Không lớp học phần nào học cùng một giảng đường trong cùng một thời gian (vì mỗi giảng đường chỉ được chứa 1 lớp trong 1 khoảng thời gian)
* Trong mỗi buổi học, các tiết học của cùng một môn học liên tục(không tách rơi)

**Ràng buộc mềm:**

* Khoảng cách giữa các tiết dạy của một giảng viên dạy một môn học cho 1 lớp là ít nhất có thể
* Số tiết dạy 1 ngày của giảng viên không quá nhiều
* Lịch học trên giảng đường là kín lịch học

**Hàm mục tiêu:**

* Số ngày lên lớp của giảng viên trong một tuần là ít nhất (giúp tối ưu thời gian đi lại của giảng viên)

Ta có thể thấy nếu vi phạm các ràng buộc cứng sẽ làm cho thời khóa biểu không thể chấp nhận được, và đó sẽ không phải là một thời khóa biểu thực sự. Còn nếu vi phạm ràng buộc mềm thì thời khóa biểu vẫn được coi là thời khóa biểu nhưng nó không được hợp lý lắm và sẽ cso một số người không thích kiểu lập thời khóa biểu này. Tuy nhiên với chương trình này chúng ta sẽ cố gắng làm sao đảm bảo không vi pahmj các ràng buộc cứng còn các ràng buộc mềm nếu giải quyết được thì càng tốt còn nếu không thì cũng có thể coi là chấp nhận được.

### 3.2.2 Mô tả bài toán

Từ phân tích thông tin khảo sát về việc xếp lịch dạy cho giảng viên của khoa Công nghệ thông tin, bài toán xếp lịch dạy được phát biểu như sau.

**Dữ liệu đầu vào (Input)**

* Bảng phân công lịch dạy(A) của giáo viên gồm: giáo viên nào dạy môn gì
* Bảng danh sách môn học(S): mã môn, tên môn, số tín chỉ môn học
* Bảng danh sách giảng viên(P):
* Bảng danh sách giảng đường (R)
* Bảng danh sách lớp học phần(C)
* Danh sách các ràng buộc cứng, ràng buộc mềm

**Dữ liệu đầu ra (Output)**

* Lịch dạy của giáo viên(X) sao cho số ngày lên lớp của mỗi giáo viên là ít.

**Phương pháp:** Dùng giải thuật di truyền

## 3.2.3 Mô hình toán học của bài toán xếp lịch

Mô hình toán học để mô tả các ràng buộc và hàm mục tiêu của bài toán được trình bày như sau:

* Tập S các môn học (Subjects)
* Tập T các tiết học trong 1 ngày (Time)
* Tập D các ngày học trong 1 tuần (Day)
* Tập C các lớp học phần (ví dụ: 12559151 , 12325151 1, 12325151 2,…..(Class)
* Tập P các giảng viên
* Tập R các giảng đường
* Mảng A (assignment) biểu diễn bảng phân công dạy của giảng viên với các môn học và lớp học phần.

ap,s,c =1)

* Mảng L (learning) biểu diễn thời gian học có thể được đăng ký của mỗi lớp vào giảng đường.

lc,t,d,r = (2) (

* Mảng 1 chiều N (Number) biểu diễn số tín chỉ của các môn học. Trong đó Ns là số tín chỉ của môn học s.
* Biến quyết định x*p,s,c,t,d,r* biểu diễn mối quan hệ giữa giảng viên p với môn học s và lớp c vào tiết t của ngày d tại giảng đường r.

x*p,s,c,t,d,r* = (3)

* Biến biểu diễn giờ dạy của mỗi giảng viên trong mỗi ngày.

(4)

**Ràng buộc cứng:**

* HC1: Không giảng viên nào dạy 2 lớp học phần trong cùng thời gian

≤ 1 p∈P, t∈T, d∈D, r ∈R (5)

* HC2: Giảng viên phải dạy đúng lớp học phần và đúng môn học được giao

(6)

* HC3: Mỗi giảng viên dạy 1 môn nào đó phải đủ số lớp theo phân công

p∈P, s∈S (7)

* HC4: Mỗi giảng viên phải dạy đủ số môn

p∈P (8)

* HC5: Mỗi giảng viên dạy một môn học cho một lớp nào đó phải dạy đủ số tiết

p∈P (9)

* HC7: Các lớp học đúng thời gian được phân công

(10)

**Ràng buộc mềm:**

* SC1: Số ngày lên lớp của giảng viên trong một tuần là ít nhất.

(13)

**3.2.4 Thiết kế giải thuật di truyền cho bài toán lập lịch giảng dạy**

## 3.2.4.1 Thiết kế giải thuật

Trong phần này chúng tôi sẽ mô tả thiết kế giải thuật di truyền để giải quyết bài toán xếp lịch dạy cho giảng viên khoa Công nghệ thông tin – học viện Kỹ Thuật Quân Sự. Giải thuật bắt đầu bằng việc khởi tạo ngẫu nhiên một quần thể các giải pháp (lịch dạy). Quần thể sẽ được tiến hóa qua nhiều thế hệ. Qua mỗi thế hệ, cá thể tốt nhất được lưu giữ cho lần tiến hóa tiếp theo. Nếu điều kiện dừng được thỏa mãn thì phương án tối ưu nhất trong các lần tiến hóa được lựa chọn là bảng phân công lịch dạy của giảng viên.

* Bước 1. Nhập dữ liệu đầu vào.
* Bước 2. Sinh tập k các cá thể. Chỉ giữ lại những cá thể thỏa mãn tất cả các ràng buộc cứng, nếu có cá thể không thỏa mãn ít nhất một ràng buộc cứng sẽ bị loại.
* Bước 3. Chọn cá thể tốt. Những cá thể thỏa mãn ràng buộc cứng được sắp xếp theo thứ tự ưu tiên của ràng buộc mềm, chọn ra 50% cá thể tốt nhất. Những cá thể được chọn sẽ được xem xét với hàm mục tiêu, để chọn ra 50% cá thể có hàm mục tiêu tốt nhất. Nếu tập cá thể chỉ còn 1 cá thể thì dừng thuật toán.
* Bước 4. Thực hiện lai ghép các cá thể được chọn ở Bước 3 với nhau
* Bước 5. Thực hiện đột biến m cá thể trong bước 4.
* Bước 6. Quay lại Bước 3.

3.4.2.2. Biểu diễn quần thể

Mỗi cá thể là một bảng phân công giảng dạy của khoa. Mỗi cá thể là một ma trận 6 chiều (ma trận X): giảng viên, môn học, lớp, ngày dạy (thứ), tiết dạy, giảng đường. Nếu phần tử X[p,s,c,d,t,r]=1 có nghĩa là giảng viên p dạy môn s cho lớp c vào tiết t của ngày d tại giảng đường r và X[p,s,c,d,t,r]=0 trong trường hợp ngược lại. Việc sinh ra một cá thể theo các bước sau nhằm đảm bảo mỗi cá thể X sinh ra sẽ thỏa mãn các ràng buộc cứng.

Bước 1. Khởi tạo mảng X ban đầu với giá trị rỗng

Bước 2. Duyệt với từng phần tử trong mảng A. Nếu A[p,s,c] =0, nghĩa là giảng viên p được phân công dạy môn s cho lớp c, ta sẽ sinh các phần tử trong mảng X liên quan đến bộ giá trị (p,s,c) để đảm bảo giảng viên p dạy đúng môn (s) đúng lớp (c) được phân công (ràng buộc HC3, HC4). Việc sinh các phần tử của mảng X liên quan đến bộ giá trị (p,s,c) theo các bước sau nhằm thỏa mãn các ràng buộc HC1, HC2, HC5, HC6.

Tạo ra tập hợp T là các tiết mà lớp c có thể tham dự theo buổi học của lớp đó. T={1, 2, 3, 4,5,6, 7,8,9,10,11,12}. Lấy số tín chỉ của môn học đang xét. Xét từ 1 đến hết số tín chỉ của môn học đó. Với mỗi lần xét ta tạo một biến KT bằng true. Trong khi biến KT đang bằng true, thực hiện việc lặp liên tục và đồng thời kiểm tra việc thỏa mãn các ràng buộc HC1, HC2 và HC6 bằng cách chọn t là một giá trị ngẫu nhiên trong tập hợp T. Chọn d là một ngày ngẫu nhiên trong tập các ngày học của một tuần. Nếu X[p’,s’,c,t,d,r] = 1 có nghĩa là tiết t của ngày d tại giản đường r đã có lịch dạy cho lớp c, ta gán KT bằng false và ở một thời điểm lớp c đã có lịch học hoặc giáo viên p đã có lịch dạy thì sẽ không xếp lịch cho lớp hoặc giáo viên đó nữa. Ngược lại, ta gán X[p,s,c,t,dr] bằng 1 (Xét tiết học cho giảng viên và lớp) .

Thuật toán:

BEGIN

T=tập hợp các tiết mà lớp c có thể học. T={1, 2, 3, 4,5,6,7,8,9,10,11,12}

Ns = số tín chỉ của môn học s

For s=1 to Ns  /\*HC5\*/

BEGIN

KT=true

Repeat /\*HC1, HC2\*/

d=random (2,3,4,5,6)

t=random(T) /\*HC6\*/

if (kiemtralichhoc(c,t,d)=true or kiemtralichday (p,t,d)=true) then KT=false

else begin X[p,s,c,t,d,r]=1; KT = true; end;

until KT=true

END

END

## 3.4.2.3 Đánh giá và lựa chọn cá thể

* Loại các cá thể không thỏa mãn ít nhất 1 ràng buộc cứng
* Chọn 50% cá thể có ràng buộc mềm tốt nhất. Sau đó, chọn 50% trong số các cá thể này mà có hàm mục tiêu tốt nhất.

## 3.4.2.4 Lai ghép

Phương pháp lai ghép: Mỗi cá thể tách làm 2 phần: lịch dạy buổi sáng và lịch dạy buổi chiều. Ghép lịch dạy buổi sáng của cá thể X với lịch dạy buổi chiều của cá thể Y, và lịch dạy buổi chiều của cá thể X ghép với lịch dạy buổi sáng của cá thể Y. Lý do chọn phương pháp lai ghép, khi lai ghép lịch dạy buổi sáng và lịch dạy buổi chiều như vậy thì sẽ giúp cho cá thể con sinh ra ít bị xung đột các ràng buộc cứng hơn.

3.4.2.5 Đột biến

Phương pháp đột biến: Chọn ra m cá thể bất kỳ để thực hiện đột biến. Lấy ra một bộ (p,s,c) bất kỳ trong cá thể đó. Thay đổi lịch dạy của bộ (p,s,c) này.

Trong chương này, chúng tôi đã trình bày mô hình toán học của bài toán lập lịch, giải thuật di truyền và thiết kế giải thuật di truyền cho bài toán lập lịch giảng dạy. Trong chương tiếp theo, chúng tôi sẽ xây dựng phần mềm xếp lịch dạy cho giảng viên.

# **CHƯƠNG IV. KẾT QUẢ THỰC NGHIỆM**

## 4.1. Chuẩn bị dữ liệu

Dữ liệu thử nghiệm là lịch phân công giảng dạy học kỳ I, năm học 2018-2019 của khoa Công nghệ thông tin – học viện Kỹ Thuật Quân Sự. Dữ liệu này chúng tôi đã trình bày trong Chương 1.

Dữ liệu bao gồm:

* Danh sách giáo viên
* Danh sách môn học
* Danh sách lớp học
* Bảng phân công giảng dạy của học kỳ
* Bảng lịch học của các lớp
* Các ràng buộc của lịch dạy
* Hàm mục tiêu của lịch dạy

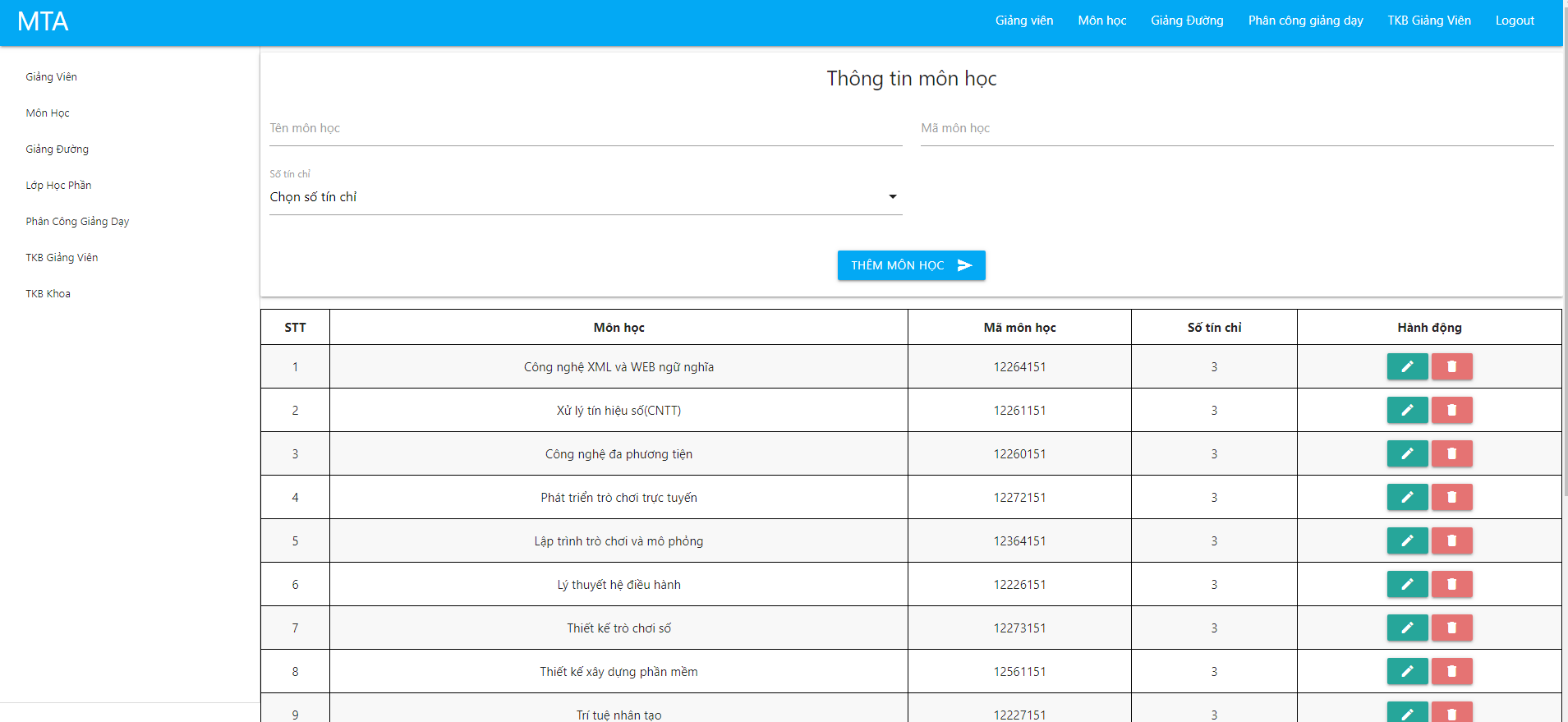
- Menu chính:

## 4.2 Một số chức năng vào giao diện của ứng dụng

### 4.2.1 Chức năng nhập dữ liệu

#### 4.2.1.1 Chức năng nhập lớp môn học

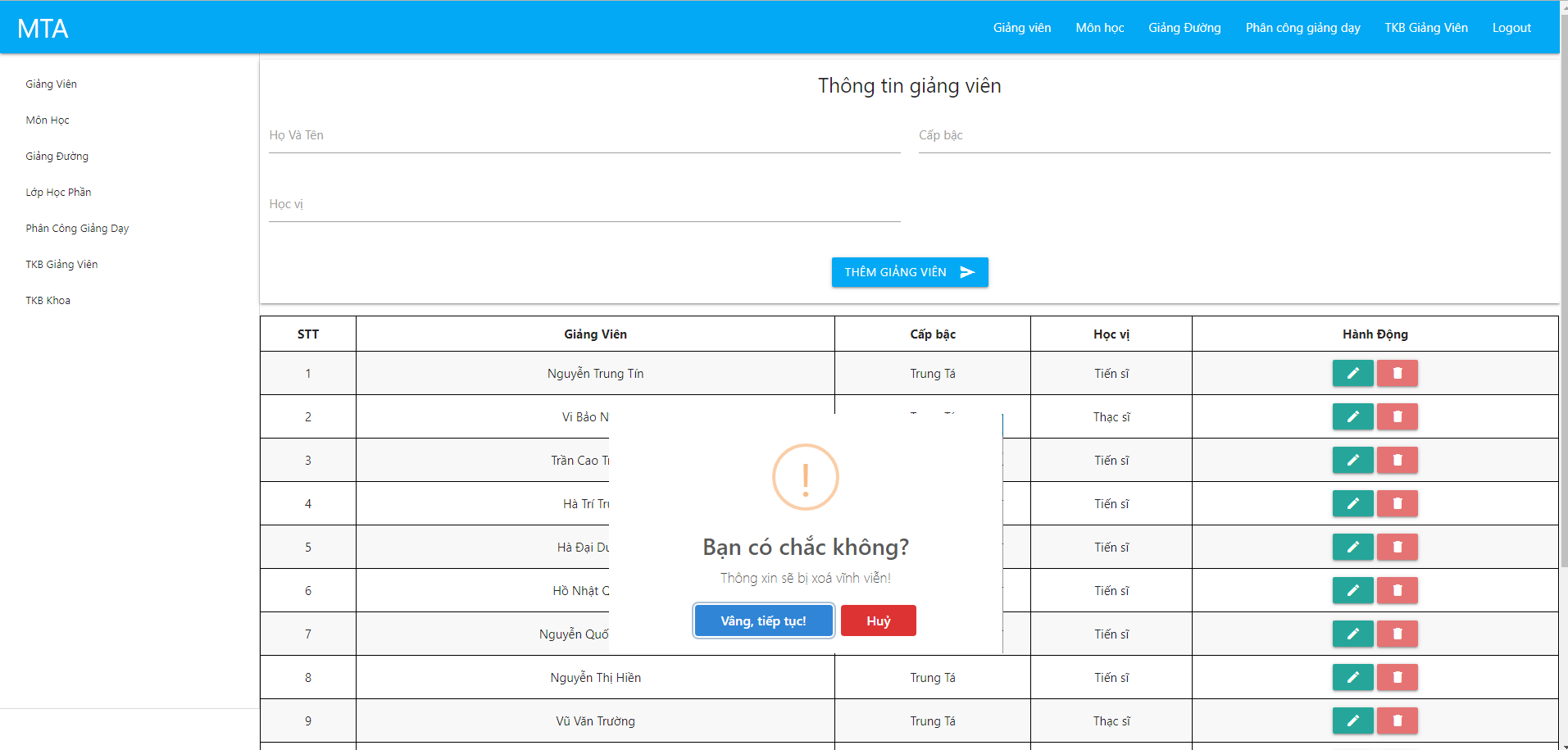
Nhập các lớp môn học cho quá trình xếp lịch thời khóa biểu tại đây. Sử dụng một lưới hiển thị kết nối tới cơ sở dữ liệu để hiệu chỉnh, xóa và thêm mới các lớp môn học. Menu tự động hiển thị khi di chuột tới trường dữ liệu tương ứng, có thể hiệu chỉnh trực tiếp trên lưới dữ liệu hoặc sử dụng các textbox và combobox bên dưới. Sử dụng các nút bên dưới để kết thúc hoặc áp dụng các thay đổi vào cơ sở dữ liệu thực tế.



Hình 4. 1Trang nhập môn học dự kiến

#### 4.2.1.2 Chức năng nhập giáo viên dự kiến

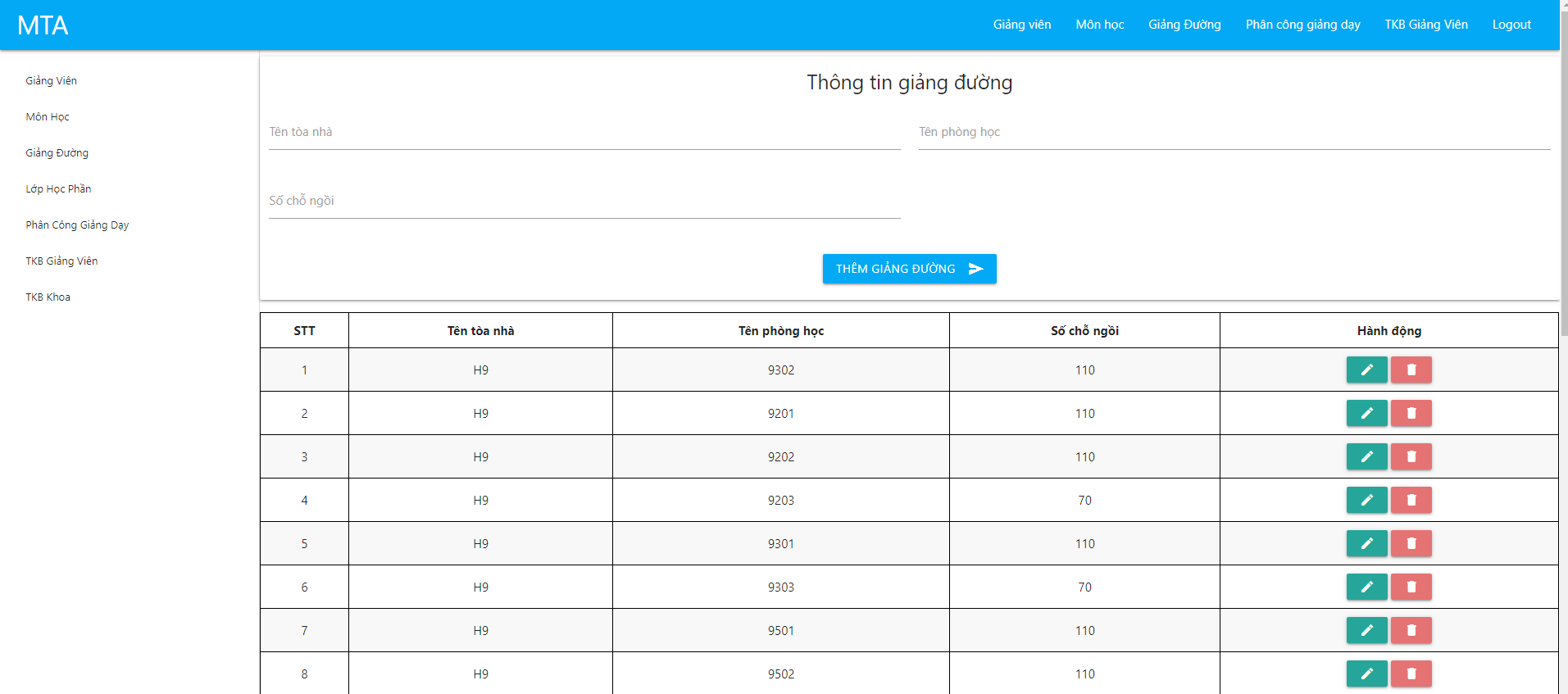
Dùng lưới dữ liệu để hiện thị các bảng trong cơ sở dữ liệu, có thể tương tác với các dữ liệu trên lưới một cách trực quan và dễ sử dụng. Tại đây có thể nhập các môn mà giáo viên có khả năng dậy đồng thời cho phép đăng ký các ca bận của giáo viên trong tuần.



Hình 4. 2 Trang nhập giảng viên dự kiến

#### 4.2.1.3 Chức năng nhập phòng học dự kiến

Trang này chỉ để nhập mới hoặc sửa chứa các thông tin về giảng đường

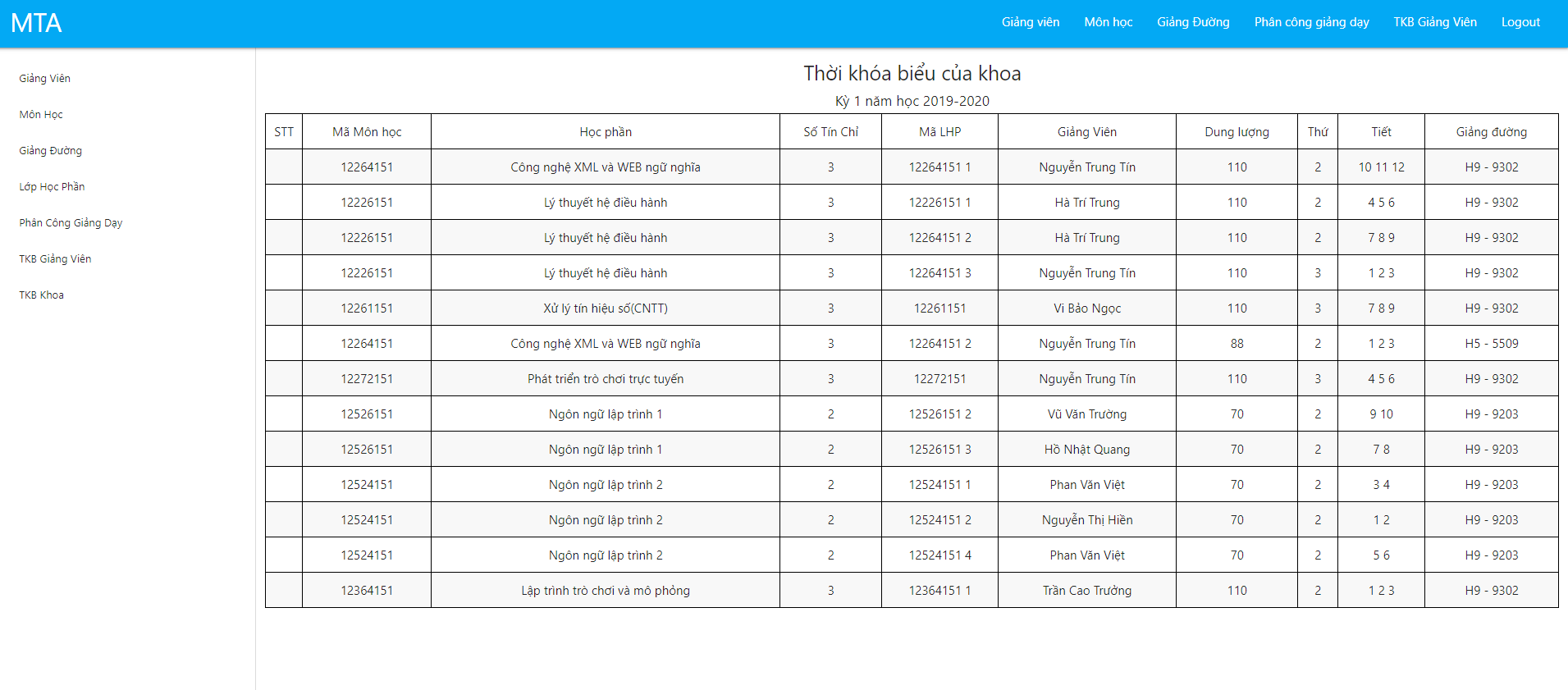


Hình 4. 3 Trang nhập giảng đường dự kiến

### 4.2.2 Chức năng hiển thị thời khóa biểu

#### 4.2.2.1 Xem thời khóa biểu giảng viên

#### 4.2.2.2 Xem thời khóa biểu các lớp môn học Tại đây hiển thị toàn bộ thời khóa biểu của các lớp môn học trong một kỳ.



Hình 4. 4 Trang xem TKB các môn học

Có thể thấy trong tất cả các lần chạy thực nghiệm các giải pháp được đưa ra thỏa mãn mọi ràng buộc cứng. Nhược điểm của ứng dụng là một vài giảng viên có các thời gian dạy giữa các tiết trong ngày chưa được tối ưu hoàn toàn.

**KẾT LUẬN**

Trong bài toán này, chúng tôi đưa ra vấn đề xếp lịch dạy cho giảng viên khoa Công Nghệ Thông Tin –học viện Kỹ Thuật Quân Sự. Để giải quyết vấn đề này, chúng tôi đã mô hình hóa toán học bài toán.

Mục tiêu chính của bài toán là xếp lịch dạy cho các giảng viên sao cho thỏa mãn tối ưu các ràng buộc. Đầu tiên chúng tôi khảo sát các thông tin liên quan đến việc lập lịch giảng dạy, phân tích các yêu cầu khi lập lịch để xác định các ràng buộc cứng, ràng buộc mềm và hàm mục tiêu của bài toán. Tiếp theo, chúng tôi mô hình hóa toán học tập các ràng buộc và hàm mục tiêu để thiết kế giải thuật di truyền tìm lời giải tối ưu bài toán. Chúng tôi trình bày cách xây dựng ứng dụng thực hiện bài toán này.

Kết quả thực nghiệm cho thấy, thuật toán hoạt động hiệu quả, đưa ra lời giải trong thời gian chấp nhận được và thỏa mãn tất cả các ràng buộc cứng. Tuy nhiên, có một số thời gian biểu trong lịch chưa thực sự được tối ưu.

Trong tương lai, chúng tôi có thể phát triển theo hướng phân tích thêm một số ràng buộc mở rộng cho bài toán. Bên cạnh đó, chúng tôi cũng nghiên cứu phương pháp để xây dựng lịch dạy cho giảng viên tối ưu hơn.

**TÀI LIỆU THAM KHẢO**

[1] Goldberg, David E. *Genetic Algorithms in Search*, Optimization and Machine Learning. Doctor of Philosophy. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA. 1989

[2] Phan Việt Anh, Bùi Thu Lâm. *Giải thuật di truyền và ứng dụng trong hỗ trợ lập lịch điều hành công tác bệnh viện*. Thạc sĩ, Tiến sĩ. Khoa Công nghệ thông tin – học viện Kỹ thuật quân sự. Khoa Công nghệ thông tin – Đại học kỹ thuật Lê Quý Đôn, 2012

[3] Trương Văn Hiếu. *Kết hợp thuật giải di truyền phân nhóm và tìm kiếm cục bộ cho bài toán xếp thời khóa biểu*. Luận văn thạc sĩ. Học viện Bưu chính viễn thông, 2010.

[4] <https://viblo.asia/p/tim-hieu-ve-node-js-co-ban-ojaqG0dGEKwZ>