**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG HỖ TRỢ SẮP XẾP THỜI KHOÁ BIỂU TỰ ĐỘNG ỨNG DỤNG THUẬT TOÁN DI TRUYỀN**

Giảng viên hướng dẫn : ThS. NGUYỄN LÊ MINH

Sinh viên thực hiện : NGUYỄN ĐÌNH HOÀNG

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :58

Tp. Hồ Chí Minh, năm 2021

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG HỖ TRỢ SẮP XẾP THỜI KHOÁ BIỂU TỰ ĐỘNG ỨNG DỤNG THUẬT TOÁN DI TRUYỀN**

Giảng viên hướng dẫn : ThS. NGUYỄN LÊ MINH

Sinh viên thực hiện : NGUYỄN ĐÌNH HOÀNG

Lớp : CÔNG NGHỆ THÔNG TIN

Khoá :58

Tp. Hồ Chí Minh, năm 2021

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Mã sinh viên:** 5851071026 **Họ tên SV:** Nguyễn Đình Hoàng

**Khóa:** 58 **Lớp:** Công nghệ thông tin

1. **Tên đề tài:** Xây dựng hệ thống hỗ trợ sắp xếp thời khoá biểu tự động ứng dụng thuật toán di truyền
2. **Mục đích, yêu cầu**
   1. **Mục đích:** hỗ trợ việc sắp xếp thời khoá biểu tự động cho đối tượng trường đại học giao thông vận tải áp dụng thuật toán di truyền.
   2. **Yêu cầu:**

* **Yêu cầu chức năng:**
* Quản lý thông tin sinh viên, giảng viên, lớp học, phòng học, môn học, học phần, thời khoá biểu.
* Thêm xoá sửa cập nhật các thông tin trên.
* Ứng dụng thuật toán di truyền cho việc tự động sắp xếp thời khoá biểu.
* Chức năng đăng nhập, mã hoá mật khẩu.
* **Yêu cầu phi chức năng:**
* Thiết kế giao diện quản lý.

1. **Nội dung thực hiện:**
2. **Công nghệ, công cụ và ngôn ngữ lập trình:** 
   * + **Công nghệ:** .Net, gRPC service
     + **Công cụ:** Visual Studio 2019, SQL Sever
     + **Ngôn ngữ lập trình:** C#
3. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng:**
   * + Website có thể quản lý được một trung tâm giáo dục cơ bản.
     + Quản lý tốt các thông tin theo yêu cầu trên.
     + Ứng dụng được thuật toán di truyền cho việc sắp xếp thời khoá biểu.
     + Website ứng dụng dùng cho các trung tâm giáo dục dễ quản lý.
4. **Kế hoạch thực hiện**

*(trong phần này mô tả kế hoạch thời gian dự kiến cho các mục nội dung chính nêu trên tính theo đơn vị tuần)*

* + - Thiết kế và xây dựng cơ sở dữ liệu. (2 tuần)
    - Thiết kế UX-UI cho window application và website (1 tuần)
    - Xây dựng model back-end (1 tuần)
    - Xây dựng sever kết nối database ( >2 tuần)
    - Tìm hiểu về thuật toán di truyền (1 tuần)
    - Code winform và webform (>2 tuần)
    - Code các chức năng quản lý(>3 tuần)
    - Code thuật toán di truyền (>2 tuần)
    - Kiểm thử (1 tuần)
    - Viết báo cáo (2 tuần)

1. **Giảng viên và cán bộ hướng dẫn**

Họ tên: ThS. NGUYỄN LÊ MINH

Đơn vị công tác: Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh

Điện thoại: Email: nlminh@utc2.edu.vn

|  |  |
| --- | --- |
| **Ngày tháng 03 năm 2021**  **Trưởng BM Công nghệ Thông tin**  **Ths. Trần Phong Nhã** | **Đã giao nhiệm vụ TKTN**  **Giảng viên hướng dẫn**  **Ths. Nguyễn Lê Minh** |

# LỜI CẢM ƠN

Lời đầu tiên, tôi xin cảm ơn đến gia đình tôi. Đặc biệt, tôi xin cảm ơn cha mẹ. Cha mẹ là người nuôi nấng, dưỡng dục tôi nên người, luôn là chỗ dựa tinh thần vững chắc và ủng hộ cho tôi mỗi khi con gặp khó khăn trong học đường và cuộc sống. Xin gửi lời cảm ơn đến những người thân trong gia đình đã quan tâm, động viên, khuyến khích, giúp đỡ tôi trong những lúc khó khăn nhất.

Tôi xin chân thành cảm ơn trường Đại học Giao Thông Vận Tải phân hiệu tại Thành Phố Hồ Chí Minh, khoa Công nghệ Thông tin đã tạo điều kiện cho tôi thực hiện đề tài này.

Tôi xin gửi lời cảm ơn đến tất cả các thầy, các cô khoa Công nghệ Thông Tin, trường Đại học Giao Thông Vận Tải phân hiệu tại Thành Phố Hồ Chí Minh. Cảm ơn các thầy cô đã dạy dỗ và truyền đạt những kiến thức bổ ích cho tôi trong suốt thời gian qua, giúp tôi có những kiến thức cần thiết để hoàn thành đề tài luận văn này cũng như tri thức giúp tôi vào đời.

Đặc biệt, tôi xin gửi lời cảm ơn chân thành đến với thầy Nguyễn Lê Minh, người đã hướng dẫn, giúp đỡ và tạo điều kiện cho tôi thực hiện đề tài.

Tôi cũng xin gửi lời cảm ơn đến thầy cố vấn học tập Trần Phong Nhã đã quan tâm giúp đỡ tôi trong thời gian vừa qua.

Trong suốt quá trình làm việc, tôi đã nỗ lực cố gắng rất nhiều để hoàn thành đề tài này một cách tốt nhất, những thiếu sót do hạn chế về mặt kiến thức cũng như kinh nghiệm là điều khó tránh khỏi. Vì vậy, tôi cũng mong nhận được những ý kiến đóng góp của quý thầy cô và các bạn để đề tài này có thể phát triển hoàn thiện hơn.

Cuối cùng, tôi xin gửi lời cảm ơn chân thành đến tất cả mọi người.

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

***Tp. Hồ Chí Minh, ngày ….… tháng ….… năm ….…***

**Giảng viên hướng dẫn**

**Ths. Nguyễn Lê Minh**

**MỤC LỤC**

[NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP i](#_Toc76809377)

[LỜI CẢM ƠN iii](#_Toc76809378)

[MỤC LỤC v](#_Toc76809379)

[DANH MỤC HÌNH viii](#_Toc76809380)

[DANH MỤC BẢNG x](#_Toc76809381)

[DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT xi](#_Toc76809382)

[PHẦN GIỚI THIỆU 12](#_Toc76809383)

[1. ĐẶT VẤN ĐỀ 12](#_Toc76809384)

[2. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ 12](#_Toc76809385)

[3. MỤC TIÊU ĐỀ TÀI 14](#_Toc76809386)

[4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU 14](#_Toc76809387)

[5. NỘI DUNG NGHIÊN CỨU 14](#_Toc76809388)

[6. NHỮNG ĐÓNG GÓP CHÍNH CỦA ĐỀ TÀI 16](#_Toc76809389)

[7. BỐ CỤC LUẬN VĂN 16](#_Toc76809390)

[**CHƯƠNG 1: MÔ TẢ BÀI TOÁN 17**](#_Toc76809391)

[1.1. ĐẶC TẢ YÊU CẦU 17](#_Toc76809392)

[1.1.1. Yêu cầu chức năng 17](#_Toc76809393)

[1.1.2. Yêu cầu phi chức năng 18](#_Toc76809394)

[1.2. CÁC PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN 19](#_Toc76809395)

[1.2.1. Các phương pháp truyền thống 19](#_Toc76809396)

[1.2.2. Các phương pháp hiện nay 19](#_Toc76809397)

[1.2.3. Đánh giá phương pháp 19](#_Toc76809398)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 20**](#_Toc76809399)

[2.1. TỔNG QUAN VỀ BÀI TOÁN LẬP LỊCH 20](#_Toc76809400)

[2.2. TÌM HIỂU VỀ BÀI TOÁN LẬP LỊCH 20](#_Toc76809401)

[2.3. GIẢI THUẬT DI TRUYỀN 20](#_Toc76809402)

[2.3.2. Các thành phần trong giải thuật di truyền 22](#_Toc76809403)

[2.3.3. Các tham số trong giải thuật di truyền 24](#_Toc76809404)

[2.3.4. Điều kiện dừng giải thuật 24](#_Toc76809405)

[2.3.5. Ứng dụng giải thuật di truyền 24](#_Toc76809406)

[2.4. NGÔN NGỮ LẬP TRÌNH VÀ HỆ CƠ SỞ DỮ LIỆU 25](#_Toc76809407)

[2.4.2. .Net Core Framework 27](#_Toc76809408)

[2.4.3. ASP.NET Core MVC 27](#_Toc76809409)

[2.4.4. Window form application 29](#_Toc76809410)

[2.4.5. gRPC service 29](#_Toc76809411)

[**CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP 31**](#_Toc76809412)

[3.1. QUY TRÌNH SẮP XẾP THỜI KHÓA BIỂU 31](#_Toc76809413)

[3.1.1. Quy trình đề xuất áp dụng hệ thống sắp xếp thời khóa biểu tự động 31](#_Toc76809414)

[3.1.2. Nội dung quy trình 32](#_Toc76809415)

[3.1.3. Mục đích, phạm vi áp dụng: 32](#_Toc76809416)

[3.1.4. Yêu cầu và ràng buộc thời khóa biểu: 32](#_Toc76809417)

[3.2. THIẾT KẾ VÀ CÀI ĐẶT THUẬT TOÁN 34](#_Toc76809418)

[3.2.2. Biểu diễn mô hình cá thể 35](#_Toc76809419)

[3.2.5. Các phép toán di truyền 38](#_Toc76809420)

[3.2.6. Tính độ thích nghi của cá thể 41](#_Toc76809421)

[3.2.7. Trọng số các loại vi phạm ràng buộc 47](#_Toc76809422)

[3.2.8. Điều kiện dừng quá trình tiến hóa 47](#_Toc76809423)

[3.3. THIẾT KẾ CHỨC NĂNG HỆ THỐNG 48](#_Toc76809424)

[3.3.1. Sơ đồ use case module web-app hỗ trợ cán bộ – sinh viên 48](#_Toc76809425)

[3.3.2. Sơ đồ use case mức 0 module phần mềm sắp xếp TKB 48](#_Toc76809426)

[3.3.3. Sơ đồ use case chức năng quản lý và sắp xếp thời khóa biểu 49](#_Toc76809427)

[3.4. THIẾT KẾ MÔ HÌNH DỮ LIỆU 50](#_Toc76809428)

[3.4.1. Đặc tả dữ liệu 50](#_Toc76809429)

[3.4.2. Mô hình cơ sở dữ liệu 44](#_Toc76809430)

[3.5. THIẾT KẾ MÔ HÌNH TUẦN TỰ SẮP XẾP THỜI KHÓA BIỂU 46](#_Toc76809431)

[3.6. CÀI ĐẶT CHƯƠNG TRÌNH 47](#_Toc76809432)

[3.6.1. Xây dựng hệ thống 47](#_Toc76809433)

[3.6.2. Giao diện chức năng 48](#_Toc76809434)

[**CHƯƠNG 4: KIỂM THỬ VÀ ĐÁNH GIÁ 59**](#_Toc76809435)

[4.1. KIỂM THỬ HỆ THỐNG 59](#_Toc76809436)

[4.1.1. Cấu hình máy tính chuẩn bị cho các kịch bản thử nghiệm 59](#_Toc76809437)

[4.1.2. Kịch bản thử nghiệm 1 59](#_Toc76809438)

[4.1.3. Kịch bản thử nghiệm 2 60](#_Toc76809439)

[4.1.4. Kịch bản thử nghiệm 3 62](#_Toc76809440)

[4.2. ĐÁNH GIÁ HỆ THỐNG 63](#_Toc76809441)

[PHẦN KẾT LUẬN 64](#_Toc76809442)

[1. KẾT LUẬN 64](#_Toc76809443)

[2. HƯỚNG PHÁT TRIỂN 65](#_Toc76809444)

[PHỤ LỤC 66](#_Toc76809445)

[1. HƯỚNG DẪN CÀI ĐẶT 66](#_Toc76809446)

[TÀI LIỆU THAM KHẢO 69](#_Toc76809447)

# 

# DANH MỤC HÌNH

[Hình 1.1 Lưu đồ quy trình phối hợp xếp TKB học kì tại Đại học GTVT 2](#_bookmark10)

[Hình 1.2 Sơ đồ các bước xây dựng đề tài 3](#_bookmark14)

[Hình 2.1 Sơ đồ tổng quan mô tả thuật giải di truyền 10](#_bookmark35)

[Hình 2.2 Mô hình bánh xe Roulette 12](#_bookmark40)

[Hình 2.3 Quá trình chuyển đổi MSIL code thành native code 14](#_bookmark48)

[Hình 2.4 Các phiên bản .NET 15](#_bookmark49)

[Hình 2.5 Mô hình MVC 17](#_bookmark52)

[Hình 2.6 Sơ đồ giao tiếp giữa gRPC server và gRPC client 19](#_bookmark55)

[Hình 3.1 Quy trình sắp xếp thời khóa biểu đề xuất 20](#_bookmark59)

[Hình 3.2 Lưu đồ thuật toán di truyền áp dụng cho hệ thống 23](#_bookmark65)

[Hình 3.3 Cấu trúc thời khóa biểu theo phòng học 24](#_bookmark67)

[Hình 3.4 Mô hình biểu diễn cấu trúc hoàn chỉnh của một NST 25](#_bookmark68)

[Hình 3.5 Mô hình biểu diễn quần thể NST 25](#_bookmark70)

[Hình 3.6 Tạo cá thể bằng cách mã hóa dữ liệu 26](#_bookmark72)

[Hình 3.7 Phương pháp lai ghép đồng nhất 28](#_bookmark76)

[Hình 3.8 Phương pháp đột biến tương hổ 29](#_bookmark78)

[Hình 3.9 Sơ đồ use case module web-app hỗ trợ cán bộ – sinh viên 36](#_bookmark92)

[Hình 3.10 Sơ đồ use case các chức năng chính của module phần mềm sắp TKB ...](#_bookmark94)37

[Hình 3.11 Sơ đồ use case các chức năng chi tiết sắp xếp và quản lý TKB...](#_bookmark96)...........38

[Hình 3.12 Mô hình dữ liệu ER của hệ thống 44](#_bookmark123)

[Hình 3.13 Mô hình cơ sở dữ liệu SQL 45](#_bookmark124)

[Hình 3.14 Sơ đồ tuần tự chức năng sắp xếp thời khóa biểu 46](#_bookmark126)

[Hình 3.15 Sơ đồ tổng thể dự án của đề tài 47](#_bookmark129)

[Hình 3.16 Giao diện đăng nhập phần mềm hỗ trợ xếp thời khóa biểu 48](#_bookmark132)

[Hình 3.17 Giao diện trang chính phần mềm hỗ trợ xếp thời khóa biểu 48](#_bookmark133)

[Hình 3.18 Giao diện quản lý đơn vị 49](#_bookmark134)

[Hình 3.19 Giao diện quản lý phòng học 49](#_bookmark135)

[Hình 3.20 Giao diện quản lý cán bộ 50](#_bookmark136)

[Hình 3.21 Giao diện quản lý học phần 50](#_bookmark137)

[Hình 3.22 Giao diện quản lý nhóm học phần theo học kì, năm học 51](#_bookmark138)

[Hình 3.23 Giao diện tổng hợp KHHT và phân nhóm học phần tự động 51](#_bookmark139)

[Hình 3.24 Giao diện hiển thị TKB theo học phần 52](#_bookmark140)

[Hình 3.25 Giao diện hiển thị TKB của cán bộ 53](#_bookmark141)

[Hình 3.26 Giao diện hiển thị TKB của phòng học 53](#_bookmark142)

[Hình 3.27 Giao diện sắp xếp TKB tự động 54](#_bookmark143)

[Hình 3.28 Giao diện thiết lập thông số cho thuật toán 55](#_bookmark144)

[Hình 3.29 Giao diện thiết lập hệ số ràng buộc 55](#_bookmark145)

[Hình 3.30 Trang hiển thị danh mục học phần 56](#_bookmark147)

[Hình 3.31 Trang hiển thị kế hoạch học tập toàn khóa của sinh viên 57](#_bookmark148)

[Hình 3.32 Trang cập nhật kế hoạch học tập của sinh viên 57](#_bookmark149)

[Hình 3.33 Giao diện thêm kế hoạch học tập của sinh viên 58](#_bookmark150)

[Hình 3.34 Trang quản lý thời khóa biểu của cán bộ 58](#_bookmark151)

[Hình 4.1 Cấu hình máy tính cho các kịch bản thử nghiệm 59](#_bookmark155)

[Hình 4.2 Thông số chương trình cho kịch bản thử nghiệm 1 59](#_bookmark159)

[Hình 4.3 Kết quả kịch bản thử nghiệm 1 60](#_bookmark161)

[Hình 4.4 Thông số chương trình cho kịch bản thử nghiệm 2 61](#_bookmark165)

[Hình 4.5 Kết quả thử nghiệm 2 61](#_bookmark167)

[Hình 4.6 Thông số chương trình cho kịch bản thử nghiệm 3 62](#_bookmark171)

[Hình 4.7 Kết quả thử nghiệm 3 63](#_bookmark173)

# DANH MỤC BẢNG

[Bảng 1.2 Quy trình thực hiện đề tài luận văn 4](#_bookmark15)

[Bảng 3.1 Các thuộc tính bảng dữ liệu CanBo 39](#_bookmark100)

[Bảng 3.2 Các thuộc tính bảng SinhVien 39](#_bookmark102)

[Bảng 3.3 Các thuộc tính bảng dữ liệu DonVi 40](#_bookmark104)

[Bảng 3.4 Các thuộc tính của bảng PhongHoc 40](#_bookmark106)

[Bảng 3.5 Các thuộc tính bảng KhuNhaHoc 40](#_bookmark107)

[Bảng 3.6 Các thuộc tính bảng HocPhan 41](#_bookmark109)

[Bảng 3.7 Các thuộc tính bảng NhomHocPhan 41](#_bookmark110)

[Bảng 3.8 Các thuộc tính bảng Thu 41](#_bookmark112)

[Bảng 3.9 Các thuộc tính bảng Tiet 42](#_bookmark113)

[Bảng 3.10 Các thuộc tính bảng ThoiGianHoc 42](#_bookmark114)

[Bảng 3.11 Các thuộc tính bảng HocKi\_NamHoc 42](#_bookmark115)

[Bảng 3.12 Các thuộc tính bảng KHHT 43](#_bookmark117)

[Bảng 3.13 Các thuộc tính bảng GiangDay 43](#_bookmark119)

[Bảng 3.14 Các thuộc tính bảng ThoiKhoaBieu 43](#_bookmark121)

# DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| Ký hiệu / chữ viết tắt | Diễn giải |
| ASP | Active Server Pages |
| CSDL | Cơ sở dữ liệu |
| GAs | Genetic Algorithms – Giải thuật di truyền |
| gRPC | Google Remote Procedure Call |
| KHHT | Kế hoạch học tập |
| HP | Học phần |
| F(x) | Fitness - Độ thích nghi của cá thể x trong quần thể P |
| FRB(x) | Độ thích nghi ràng buộc của cá thể x trong quần thể P |
| NST | Nhiễm sắc thể |
| P | Population – Quần thể cá thể |
| PĐT | Phòng đào tạo |
| SQL | Structured Query Language |
| TKB | Thời khóa biểu |

# PHẦN GIỚI THIỆU

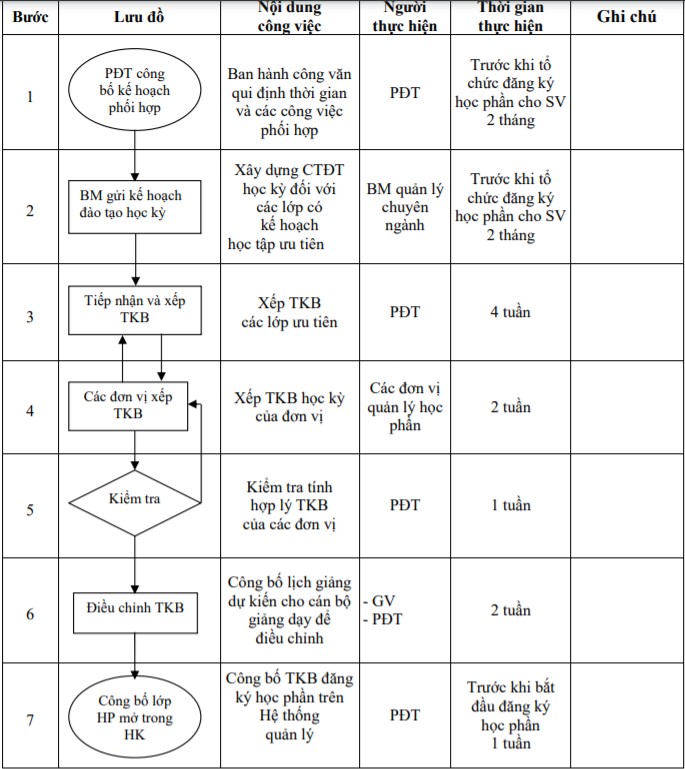
## ĐẶT VẤN ĐỀ

Ở Việt Nam hiện nay, các trường Đại học đang dần chuyển sang hình thức đào tạo tín chỉ. Mặc dù hình thức đào tạo này có nhiều ưu điểm hơn so với đào tạo niên chế, tuy nhiên việc xếp thời khóa biểu vẫn là một gánh nặng thực sự cho các trường, đặc biệt là các trường có quy mô đào tạo lớn. Để giải quyết bài toán xếp thời khóa biểu 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. Có rất nhiều các ràng buộc được đặt ra trong bài toán này như ràng buộc về đối tượng tham gia (Cán Bộ Giảng Viên, Lớp Học, Sinh Viên), ràng buộc về tài nguyên phục vụ giảng dạy (Phòng Học Lý Thuyết, Phòng Thực Hành, …), 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 trên 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. Phương pháp này có nhiều đặc điểm nổi trội như không đòi hỏi tri thức, 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. Trên thế giới hiện nay, giải thuật di truyền kết hợp với tin học được ứng dụng để giải quyết những bài toán tối ưu một cách rất hiệu quả.

Vì vậy, việc nghiên cứu và ứng dụng giải thuật di truyền (Genetic Algorithm - GA) tạo nên phần mềm sắp xếp thời khóa biểu tự động để giải quyết hiệu quả bài toán xếp thời khóa biểu nói trên là việc làm cần thiết.

## LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ

* + Trên thế giới: Một điều đáng ngạc nhiên là số lượng các phần mềm TKB trên thế giới không nhiều. Các phần mềm sắp tiêu biểu: Mimosa (www.mimosasoftware.com) – sắp xếp TKB cho các trường đại học hệ tín chỉ, Gp-untils [(www.grupet.at](http://www.grupet.at/)) – mô hình sắp TKB theo tuần, phù hợp cho các trường phổ thông, aSc Timetable ([www.asctimetables.com](http://www.asctimetables.com/)) – phần mềm xếp TKB phổ thông,..
  + Tại Việt Nam: Hầu hết tại Việt Nam các trường học các cấp đều sắp TKB theo hướng thủ công hoặc sử dụng các phần mềm hỗ trợ sắp xếp TKB bằng cách liệt kê,tìm kiếm cũng có một số phần mềm sắp xếp TKB tự động: Schoolne, uniScheGA.
  + Tại trường Đại học GTVT:

**Hình 1.1 Lưu đồ quy trình phối hợp xếp TKB học kì tại Đại học GTVT**

Hiện nay tại trường đại học GTVT việc sắp xếp TKB sẽ được phối hợp thực hiện giữa phòng Đào tạo và các Khoa/Bộ môn phụ trách giảng dạy học phần. Mỗi khoa sẽ có cách sắp xếp TKB riêng và được kiểm tra bởi phòng đào tạo. Do quy trình cần sự phối hợp giữa các khoa và phòng đào tạo nên phải mất rất nhiều thời gian mới tạo ra được một TKB hoàn chỉnh.

## MỤC TIÊU ĐỀ TÀI

Tìm hiểu và ứng dụng giải thuật di truyền vào bài toán xếp thời khóa biểu hệ tín chỉ, nhằm đưa ra phương án sắp xếp thỏa mãn tất cả các ràng buộc đặt ra đồng thời khai thác hiệu quả các nguồn lực đào tạo của nhà trường với thời gian ngắn.

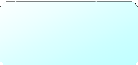
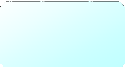
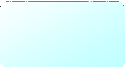
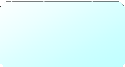
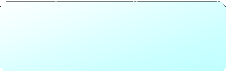
Xây dựng phần mềm giúp sắp xếp thời khóa biểu tự động và nhanh chóng giảm thiểu tối đa thời gian và công sức so với việc thực hiện thủ công. Phần mềm có giao diện hài hòa, dễ nhìn, linh hoạt, năng động hơn và đáp ứng được yêu cầu ngày càng cao trong công tác quản lý cán bộ giảng viên, học phần… và nhu cầu của người sử dụng.

## ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

* Tìm hiểu bài toán lập lịch và các hướng giải quyết truyền thống.
* Nghiên cứu các đặc điểm, đặc trưng của giải thuật di truyền, các thành phần cơ bản của giải thuật di truyền như khởi động quần thể ban đầu, đánh giá độ thích nghi của cá thể, các toán tử di truyền (chọn lọc, lai ghép, đột biến), điều kiện dừng.
* Ứng dụng giải thuật di truyền xây dựng phần mềm lập thời khóa biểu theo học chế tín chỉ cho trường đại học với các ràng buộc và những yêu cầu cơ bản.

## NỘI DUNG NGHIÊN CỨU

* + Nghiên cứu lý thuyết
* Nghiên cứu tài liệu, ngôn ngữ và công nghệ liên quan xây dựng hệ thống.
* Tổng hợp các tài liệu lý thuyết về giải thuật di truyền.
* Biểu diễn bài toán xếp thời khóa biểu hệ tín chỉ trong trường đại học sử dụng mô hình giải thuật di truyền.
  + Nghiên cứu thực nghiệm
* Phân tích và thiết kế hệ thống xếp thời khóa biểu hệ tín chỉ theo quy trình xây dựng ứng dụng phần mềm.
* Xây dựng hệ thống xếp thời khóa biểu hệ tín chỉ sử dụng giải thuật di truyền.
  + Quy trình thực hiện đề tài



Xây dựng hệ thống hỗ trợ sắp TKB tự động bằng GAs

Bắt đầu đề tài

Tìm hiểu lý thuyết và yêu cầu đề

Phân tích yêu cầu & hệ thống

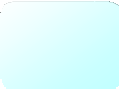
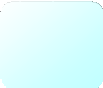
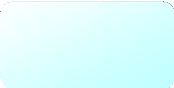
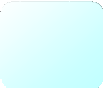
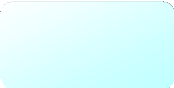
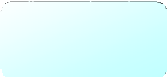
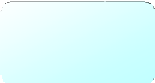
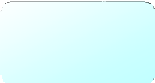
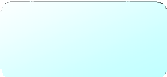
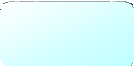
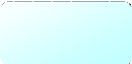
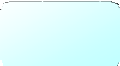
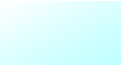
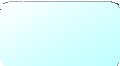
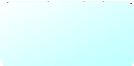
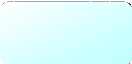
Thiết kế

Xây dựng chương trình

Viết tài liệu báo cáo

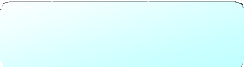
**Hình 1.2 Sơ đồ các bước xây dựng đề tài**

Tìm hiểu quy trình sắp TKB trong thực tế



|  |  |  |
| --- | --- | --- |
| **Chi tiết các bước thực hiện** | **Thời gian** | **Diễn giải** |
| Bắt đầu đề tài  Nhận đề tài  Lập kế hoạch thực hiện | 1 tuần | Nhận đề tài luận văn văn và tiến hành lập kế hoạch thực hiện trong tuần đầu tiên. |
| Tìm hiểu lý thuyết và yêu cầu đề  Tìm hiểu quy Nghiên cứu các  trình sắp TKB Tìm hiểu bài Nghiên cứu giải công nghệ thưc trong thực tế toán lập lịch thuật di truyền hiện đề tài | 3 tuần | Tìm hiểu các cơ sở lý thuyết liên quan đến yêu cầu của đề tài. |
| Phân tích yêu cầu & hệ thống  Phân tích bài toán Phân tích các chức  lập lịch TKB theo năng của hệ thống  học phần - tín chỉ | 2 tuần | Phân tích các yêu cầu của đề tài, từ đó khái quát hóa các chức năng, mục tiêu của hệ thống. |
| Thiết kế  Thiết kế thuật toán GAs Thiết kế các mô Thiết kế giao cho bài toán sắp TKB hình hệ thống Thiết kế CSDL diện chức năng theo học phần - tín chỉ hệ thống | 3 tuần | Phân tích, thiết kế, xây dựng giải thuật di truyền cho bài toán sắp xếp TKB. |
| Xây dựng chương trình  Lập trình Kiểm thử | 6 tuần | Lập trình xây dựng hệ thống và kiểm thử các chức năng. |
| Viết tài liệu báo cáo | 1 tuần | Tài liệu báo cáo được viết song song với các công việc trên. Và được chỉnh sửa hoàn chỉnh trong tuần cuối cùng. |

**Bảng 1.1 Quy trình thực hiện đề tài luận văn**



1. **NHỮNG ĐÓNG GÓP CHÍNH CỦA ĐỀ TÀI**

* Nhận thức đầy đủ về thế mạnh của giải thuật di truyền trong việc giải các bài toán tối ưu.
* Đề ra được giải pháp và ứng dụng các vấn đề của giải thuật di truyền vào việc giải quyết bài toán xếp thời khóa biểu hệ tín chỉ.
* Xây dựng hệ thống phần mềm nhằm phục vụ cho việc xếp thời khóa biểu hệ tín chỉ tại một số trường Đại học.

## BỐ CỤC LUẬN VĂN

Nội dung chính của luận được chia thành 3 phần như sau:

* + Phần giới thiệu
  + Phần nội dung
    - Chương 1: Mô tả bài toán
    - Chương 2: Cơ sở lý thuyết
    - Chương 3: Thiết kế và cài đặt giải pháp
    - Chương 4: Kiểm thử và đánh giá
  + Phần kết luận

# CHƯƠNG 1: MÔ TẢ BÀI TOÁN

## ĐẶC TẢ YÊU CẦU

Đề tài sẽ thực hiện nghiên cứu quy trình sắp xếp thời khóa biểu tại các trường đại học trong thực tế để hiểu được các nội dung trong việc quản lý dữ liệu, phân chia thời khóa biểu. Từ đó, phân tích khả năng có thể tiết kiệm được thời gian, công sức khi hệ thống áp dụng vào thực tế hay không.

Hệ thống sẽ bao gồm hai module chính:

* + - Module web-app: dành cho sinh viên và giảng viên. Đây là module để thu thập các dữ liệu về KHHT của sinh viên, lịch bận/rảnh của giảng viên. Đồng thời cũng là module nhận dữ liệu thời khóa biểu sau khi được sắp xếp để hiển thị với giảng viên và sinh viên.
    - Module phần mềm sắp xếp thời khóa biểu tự động: dành cho admin. Đây là module chính dùng để sắp xếp thời khóa biểu tự động và quản lý các dữ liệu đầu vào liên quan. Dữ liệu KHHT sinh viên và lịch bận/rảnh của giáo viên thu thập được từ module web-app sẽ được tổng hợp và chuẩn bị cho việc sắp xếp thời khóa biểu.

## Yêu cầu chức năng

* + - 1. **Module web-app**
         * Đăng nhập: Hệ thống sẽ nhận diện được người dùng đăng nhập là sinh viên hoặc cán bộ / giảng viên để hiển thị giao diện phù hợp.
         * Đối với người dùng là sinh viên:

+ Cập nhật KHHT: Thêm KHHT mới, chuyển đổi học kì – năm học cho KHHT, xóa KHHT.

+ Xem thời khóa biểu học phần theo học kì – năm học.

+ Xuất thời khóa biểu ra file Excel, PDF hoặc in ấn thời khóa biểu.

* + - * + Đối với người dùng là cán bộ – giảng viên:

+ Cập nhật lịch bận/rảnh.

+ Xem lịch giảng dạy cá nhân theo học kì – năm học.

+ Chỉnh sửa lịch giảng dạy (trong thời gian cho phép).

+ Xem thời khóa biểu học phần theo học kì – năm học.

+ Xuất thời khóa biểu ra file Excel, PDF hoặc in ấn thời khóa biểu.

* + - 1. **Module window-form-app**
         * Đăng nhập: Người dùng đăng nhập là admin hoặc cán bộ phụ trách sắp xếp thời khóa biểu.
         * Chức năng cập nhật dữ liệu:

+ Cập nhật danh mục phòng học.

+ Cập nhật danh sách đơn vị.

+ Cập nhật danh sách cán bộ – giảng viên.

+ Cập nhật danh sách học phần.

+ Cập nhật danh sách nhóm học phần theo học kì – năm học.

+ Cập nhật danh sách phân công giảng dạy theo học kì – năm học.

* + - * + Chức năng tổng hợp KHHT sinh viên và phân nhóm học phần.
        + Chức năng sắp xếp thời khóa biểu tự động:

+ Thiết lập thông số cho thuật toán.

+ Thiết lập thông số ràng buộc vi phạm.

+ Khởi tạo dữ liệu và sắp xếp thời khóa biểu tự động.

+ Xem và lưu trữ thời khóa biểu sau khi sắp xếp.

* + - * + Chức năng quản lý thời khóa biểu

+ Quản lý thời khóa biểu theo học kì – năm học.

+ Xem/xuất file excel thời khóa biểu theo phòng học.

+ Xem/xuất file excel thời khóa biểu theo học phần.

+ Xem/xuất file excel thời khóa biểu theo giảng viên.

+ Cập nhật thời khóa biểu thủ công.

## Yêu cầu phi chức năng

* Phát triển bằng ngôn ngữ C#, sử dụng .Net Core Framework
* Database sử dụng SQL Server 2014 trở lên
* Giao diện thân thiện với người sử dụng
* Có tính ổn định hạn chế xảy ra lỗi ở mức thấp nhất
* Thời gian load dữ liệu nhanh
* Tìm kiếm dễ dàng và chính xác

## CÁC PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN

Để tạo ra được hệ thống sắp xếp thời khóa biểu tự động, trước tiên cần phải nghiên cứu và phân tích các phương pháp giải quyết bài toán lập lịch. Từ đó lựa chọn giải pháp phù hợp cho bài toán.

## Các phương pháp truyền thống

* + - * Phương pháp liệt kê
      * Phương pháp tìm kiếm ngẫu nhiên

## Các phương pháp hiện nay

* + - * Giải thuật leo đồi
      * Giải thuật luyện kim
      * Giải thuật di truyền (GAs)
      * Giải thuật tô màu đồ thị

## Đánh giá phương pháp

Đặc trưng của GAs so với các phương pháp truyền thống:

* + - * GAs làm việc với sự mã hoá của tập thông số chứ không làm việc với các giá trị của các thông số.
      * Hầu hết các kĩ thuật tối ưu thông thường tìm kiếm từ một đỉnh, trong khi đó GAs luôn hoạt động trên tập hợp đỉnh (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.

Các giải thuật leo đồi và luyện kim có rất nhiều nhược điểm và thường không trả về được kết quả như mong đợi. Các giải thuật di truyền và tô màu đồ thị có nhiều ưu điểm hơn vì thế hiện nay hai phương pháp này được sử dụng nhiều nhất để giải quyết các bài toán tối ưu trong đó có bài toán xếp thời khóa biểu.

GAs có thể cung cấp lời giải tiềm năng cho một bài toán xác định để người sử dụng lựa chọn. Vì vậy trong luận văn này, tôi sử dụng giải thuật di truyền để giải quyết bài toán xếp thời khóa biểu hệ đào tạo tín chỉ cho trường đại học.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. TỔNG QUAN VỀ BÀI TOÁN LẬP LỊCH

Lập lịch có thể được định nghĩa là một bài toán tìm kiếm chuỗi tối ưu để thực hiện một tập các hoạt động chịu tác động của một tập các ràng buộc cần phải được thỏa mãn. Người lập lịch thường cố gắng thử đến mức tối đa sự sử dụng các cá thể, máy móc và tối thiểu thời gian đòi hỏi để hoàn thành toàn bộ quá trình nhằm sắp xếp lịch. Vì thế bài toán lập lịch là một vấn đề rất khó để giải quyết. Hiện nay có nhiều khả năng để phát triển các kỹ thuật hiện tại để giải quyết bài toán này. Những kỹ thuật đó bao gồm: các tiếp cận Trí tuệ nhân tạo như hệ thống tri thức cơ sở (knowledge-based systems), bài toán thoả mãn ràng buộc, hệ chuyên gia, mạng Nơron và các tiếp cận của các Nghiên cứu hoạt động: lập trình tính toán, lập trình động, tìm kiếm nhánh và đường biên, kỹ thuật mô phỏng, tìm kiếm Tabu và phương pháp nút cổ chai.

## TÌM HIỂU VỀ BÀI TOÁN LẬP LỊCH

* + 1. **Các đặc tính của bài toán lập lịch**
       - Tài nguyên: Đó là các nguồn dữ liệu đầu vào của bài toán. Các tài nguyên này có thể phục hồi hoặc không.
       - Tác vụ: Được đánh giá qua các tiêu chuẩn thực hiện như thời gian thực hiện, chi phí, mức tiêu thụ nguồn tài nguyên.
       - Ràng buộc: Đây là những điều kiện cần thỏa mãn để bài toán có thể đưa ra lời giải tốt nhất.
       - Mục tiêu: Đánh giá độ tối ưu của lịch trình lời giải của bài toán.

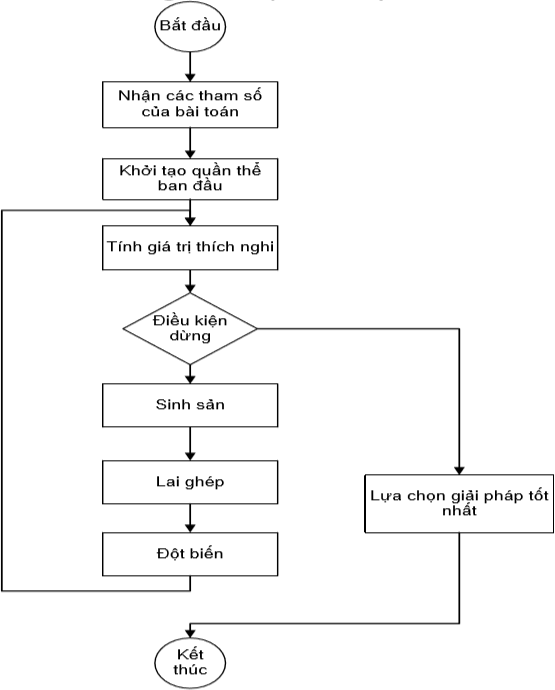
Khi các mục tiêu được thỏa mãn thì các ràng buộc cũng phải được thỏa mãn

## GIẢI THUẬT DI TRUYỀN

* + 1. **Tổng quan về giải thuật di truyền**

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 kỉ trước trong việc giải quyết các bài toán tối ưu toàn cục. Việc khai thác nguyên lí tiến hóa như là một định hướng heuristics 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 khả vi) như là điều kiện tiên quyết.

Các nhiễm sắc thể có thể là một chuỗi tuyến tính, trong nhiễm sắc thể có thể có các đơn vị nhỏ hơn đó là gen. Mỗi gen đại diện một thuộc tính, tính chất và có vị trí nhất định trong nhiễm sắc thể. Quần thể (population) là một tập hợp hữu hạn xác định các cá thể, trong thuật giải di truyền quần thể là một tập các cá thể biểu diễn một tập các lời giải. Các phép toán chọn lọc (selection), lai ghép (crossover), đột biến (mutation) được thực hiện trên quần thể để tạo ra một quần thể mới.



Đúng

Sai

sau:

**Hình 2.1 Sơ đồ tổng quan mô tả thuật giải di truyền**

Một bài toán được giải bằng thuật giải di truyền thông thường phải qua các bước

* Biểu diễn lời giải của bài toán (hay nhiễm sắc thể) bằng chuỗi nhị phân, chuỗi

ký tự, số thập phân, …

* + Khởi tạo quần thể ban đầu gồm N cá thể một cách ngẫu nhiên.
  + Xây dựng hàm thích nghi làm tiêu chuẩn đánh giá các cá thể theo độ thích nghi của chúng.
  + Xác định xác suất lai tạo, xác suất đột biến, …
  + Xây dựng các phép toán lai tạo, chọn lọc, đột biến.

## Các thành phần trong giải thuật di truyền

* + - 1. **Biểu diễn giải thuật**

Đây là một trong những công việc quan trọng trong thiết kế giải thuật di truyền, quyết định việc áp dụng các toán tử tiến hóa. Một trong những biểu diễn truyền thống của GA là biểu diễn nhị phân. Với phép biểu diễn này, giải pháp cho một bài toán được biểu diễn như là một vector bit, còn gọi là nhiễm sắc thể. Mỗi nhiễm sắc thể bao gồm nhiều gen, trong đó một gen đại diện cho một tham số thành phần của giải pháp. Một kiểu biểu diễn khác cũng thường dùng là biểu diễn số thực. Với phép biểu diễn này, các toán tử tiến hóa sẽ thực hiện trực tiếp trên các giá trị số thực (genes).

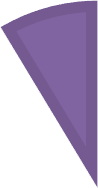
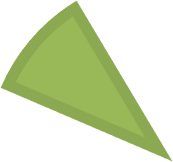
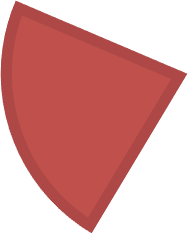
* + - 1. **Đánh giá cá thể**

Chắc chắn rằng việc chọn lựa cá thể sẽ thông qua kết quả, hay mục đích của bài toán. Dựa trên mức độ thích nghi của cá thể, bao gồm những vướng mắc mà cá thể gặp phải. Thông thường, đặt mỗi ràng buộc của bài toán tương ứng với một giá trị điểm thích nghi, kết quả đánh giá thông qua một hàm mục tiêu (Fitness) gồm tổng các số điểm đó. Cá thể tốt nhất sẽ có số điểm thấp nhất hoặc lớn nhất.

* + - 1. **Chọn lọc cá thể**

Theo thuyết tiến hóa của Darwin, nhiễm sắc thể tốt nhất sẽ tồn tại và tạo ra các cá thể con mới. Có nhiều phương pháp để chọn các nhiễm sắc thể tốt nhất.

* + - * + Chọn lọc Roulette (Roulette Wheel Selection): Mỗi cá thể trong quần thể sẽ chiếm một khe trong vòng tròn Roulette có độ rộng tỉ lệ với giá trị hàm mục tiêu của cá thể. Mỗi lần quay vòng tròn Roulette chúng ta được ngẫu nhiên một cá thể.
        + Chọn lọc xếp hạng (Rank Selection): Dựa vào giá trị hàm mục tiêu mỗi thể sẽ tiến hành xếp hạng chúng theo thứ tự. Tùy theo yêu cầu và phân tích bài toán, cá thể có giá trị hàm mục tiêu thấp nhất hoặc lớn nhất sẽ đứng đầu và được lựa chọn.
        + Chọn lọc cạnh tranh (Tournament Selection): Chỉ định ngẫu nhiên 2 (hoặc nhiều hơn 2) cá thể trong quần thể. Lựa chọn cá thể có giá trị hàm mục tiêu tốt nhất trong cac cá thể đó.



**POPULATION**

1st individual 2nd individual 3rd individual 4th individual

**1.2, 9%**

**1.4, 10%**

**3.2, 23%**

**8.2, 58%**

**Hình 2.2 Mô hình bánh xe Roulette**

* + - 1. **Toán tử lai ghép**

Lai ghép nhằm nâng cao kết quả cá thể, do đó, toán tử lai ghép sẽ tạo điều kiện cho tiến trình hội tụ nhanh hay chậm. Còn tùy thuộc vào cách tổ chức và phân bố các nhiễm sắc thể mà chúng ta có xác suất lai ghép nhanh hay chậm. Sau đây là vài phương pháp lai ghép thông dụng trong kỹ thuật di truyền:

* Lai ghép đồng nhất (Uniform Crossover)
* Lai ghép ánh xạ từng phần (PMX Partial Mapped Crossover)
* Lai ghép có trật tự (OX Order Crossover)
* Lai ghép dựa trên vị trí (Position Based Crossover)
* Lai ghép dựa trên thứ tự (Order Base Crossover)
* Lai ghép có chu trình (CX Cycle Crossover)
* Lai ghép thứ tự tuyến tính (LOX Linear Order Crossover)
  + - 1. **Toán tử đột biến**

Cũng giống như lai ghép, toán tử đột biến làm tăng nhanh quá trình hội tụ, nhưng tăng một cách đột ngột, cũng có khi sẽ không gây tác dụng gì một khi không thành công. Không ai có thể đánh giá được phương pháp đột biến nào tốt hơn, do đó có một vài phương pháp đơn giản, cũng có vài trường hợp khá phức tạp. Người ta thường chọn một trong những phương pháp sau:

* Đột biến đảo ngược (Inversion Mutation)
* Đột biến chèn (Insertion Mutation)
* Đột biến thay thế (Displacement Mutation)
* Đột biến tương hỗ (Reciprocal Exchange Mutation)
* Đột biến chuyển dịch (Shift Mutation)

## Các tham số trong giải thuật di truyền

***Xác suất lai ghép – crossover rate – pc:*** Là tham số cho biết tần suất thực hiện toán tử lai ghép. Nếu không có lai ghép, cá thể con sẽ chính là bản sao của cá thể “cha mẹ”. Nếu xác suất lai ghép bằng 100%, khi đó mọi cá thể con đều được tạo ra qua quá trình lai ghép.

***Xác suất đột biến – mutation rate – pm:*** Là tham số cho biết tần suất đột biến của nhiễm sắc thể. Nếu không có đột biến, thế hệ con được tạo ra ngay sau giai đoạn lai ghép mà không bị thay đổi. Ngược lại, một hoặc một số phần của nhiễm sắc thể sẽ bị thay đổi. Nếu xác suất đột biến là 100%, toàn bộ nhiễm sắc thể sẽ bị thay đổi. Nếu tham số này bằng 0%, không có gì bị thay đổi hết.

***Kích thước quần thể - population size – Ps:*** Là tham số cho biết có bao nhiêu cá thể (NST) trong 1 thế hệ của quần thể. Nếu chọn kích thước quần thể quá nhỏ thì tính đa dạng của quần thể bị hạn chế và ảnh hưởng đến kết quả, còn nếu quá lớn sẽ làm hao phí tài nguyên của máy tính và làm chậm quá trình tiến hóa. [5]

## Điều kiện dừng giải thuật

Thoát ra quá trình tiến hóa quần thể, dựa vào bài toán mà có các cách kết thúc vấn đề khác nhau, một khi đã 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.

## Ứng dụng giải thuật di truyền

Giải thuật di truyền được sử dụng cho những bài toán khó, và đã được ứng dụng thành công cho một số bài toán như: lập kế hoạch, điều khiển tương thích, chương trình trò chơi, các bài toán vận tải, bài toán người đi du lịch, …

Để ứng dụng giải thuật di truyền vào việc giải quyết một bài toán nào đó cần phải thực hiện một số công việc quan trọng sau:

1. Lựa chọn cách biểu diễn mô hình NST sao cho mỗi NST có thể chứa đựng được một lời giải của bài toán.
2. Xây dựng hàm đánh giá độ thích nghi cho từng NST. Đây là bước khó khăn và ảnh hưởng lớn đến tính hiệu quả của giải thuật.
3. Lựa chọn các toán tử di truyền phù hợp, trong đó tập trung cho ba toán tử chính là chọn lọc, lai ghép và đột biến.
4. Xác định các tham số của giải thuật di truyền như kích thước quần thể, xác suất lai ghép, xác suất đột biến.
5. Xác định điều kiện dừng cho quá trình tiến hóa.

## NGÔN NGỮ LẬP TRÌNH VÀ HỆ CƠ SỞ DỮ LIỆU

* + 1. **Ngôn ngữ lập trình C#**

C# (C Sharp) là một ngôn ngữ lập trình hướng đối tượng đa năng vô cùng mạnh mẽ được phát triển bởi Microsoft vào năm 2000 và thiết kế chủ yếu bởi Anders Hejlsberg

- kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC. Microsoft phát triển C# dựa trên C++ và Java, là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java. [7]

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành. Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language). Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành.

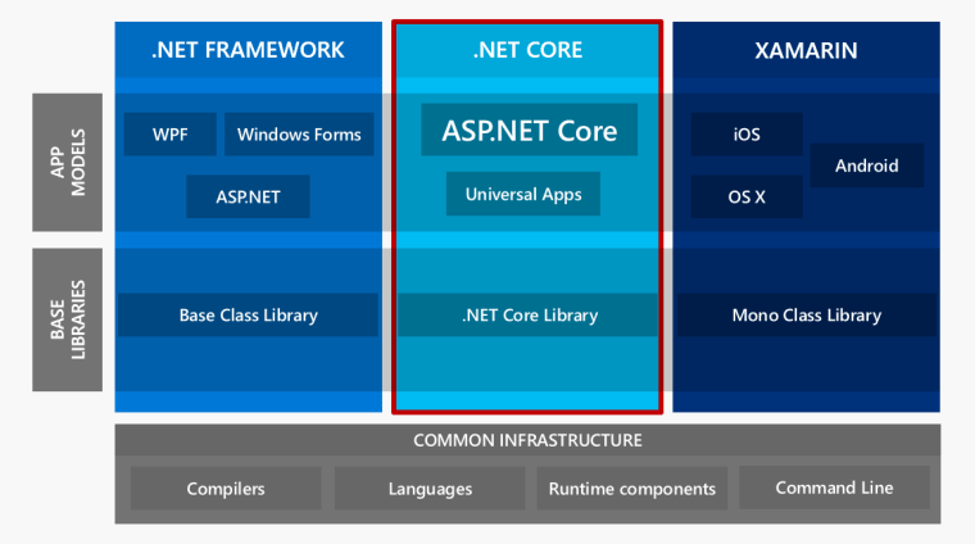
**Hình 2.3 Quá trình chuyển đổi MSIL code thành native code**

năng.

* **Đặc trưng của ngôn ngữ lập trình C#:**
* C# là ngôn ngữ đơn giản, mạnh mẽ.
* C# là ngôn ngữ đa năng và hiện đại.
* C# là một ngôn ngữ lập trình hướng đối tượng đồng thời hỗ trợ lâp trình chức
* C# là một ngôn ngữ ít từ khóa.
* C# kết hợp chặt chẽ với nền tảng .NET - một khung nền tảng rất mạnh của

Microsoft.

* **Nền tảng .NET:**



**Hình 2.4 Các phiên bản .NET**

* .NET Framework được Microsoft đưa ra chính thức từ năm 2002. .NET Framework chỉ hoạt động trên Windows. Những nền tảng ứng dụng như WPF, Winforms, ASP. NET (1-4) hoạt động dựa trên .NET Framework.
* Mono là phiên bản cộng đồng nhằm mang .NET đến những nền tảng ngoài Windows. Mono được phát triển chủ yếu nhằm xây dựng những ứng dụng với giao diện người dùng và được sử dụng rất rộng rãi: Unity Game, Xamarin…
* Cho đến năm 2013, Microsoft định hướng đi đa nền tảng và phát triển .NET core. .NET core hiện được sử dụng trong các ứng dụng Universal Windows platform và ASP.NET Core. Từ đây, C# có thể được sử dụng để phát triển các loại ứng dụng đa nền

tảng trên các hệ điều hành khác nhau (Windows, Linux, MacOS,…)

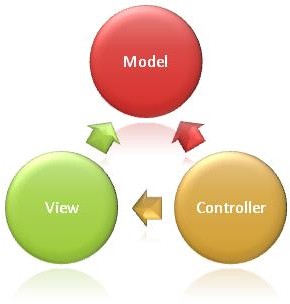
## .Net Core Framework

.NET Core là một nền tảng phát triển đa mục đích, mã nguồn mở được duy trì bởi Microsoft và cộng đồng .NET trên GitHub. Đó là nền tảng chéo (hỗ trợ Windows, macOS và Linux) và có thể được sử dụng để xây dựng các ứng dụng thiết bị, đám mây và IoT. .NET Core bao gồm các phần sau:

* + - * .NET Core runtime: Cung cấp một hệ thống kiểu, tải lắp ráp, trình thu gom rác, interop gốc và các dịch vụ cơ bản khác. Các thư viện khung .NET Core cung cấp các kiểu dữ liệu nguyên thủy, các kiểu thành phần ứng dụng và các tiện ích cơ bản.
      * ASP.NET Core runtime: Cung cấp khung để xây dựng các ứng dụng kết nối internet, điện toán đám mây hiện đại, chẳng hạn như ứng dụng web, ứng dụng IoT và phụ trợ di động.
      * .NET Core SDK và trình biên dịch ngôn ngữ (Roslyn và F #) cho phép trải nghiệm nhà phát triển .NET Core.
      * Dotnet command, được sử dụng để khởi chạy các ứng dụng .NET Core và các lệnh CLI. Nó chọn thời gian chạy và lưu trữ thời gian chạy, cung cấp chính sách tải lắp ráp và khởi chạy các ứng dụng và công cụ.
* **Đặc điểm của .NET Core:**
  + - * Đa nền tảng: Chạy trên các hệ điều hành Windows, macOS và Linux.
      * Nhất quán trên các kiến trúc: có thể chạy mã nguồn của bạn với cùng một hành vi trên nhiều kiến trúc hệ thống, bao gồm x64, x86 và ARM.
      * Các công cụ dòng lệnh: Bao gồm các công cụ dòng lệnh dễ sử dụng, có thể được sử dụng để phát triển cục bộ và trong các tình huống tích hợp liên tục.
      * Triển khai linh hoạt: có thể cài đặt song song (cài đặt toàn người dùng hoặc toàn hệ thống). Có thể được sử dụng với các container Docker
      * Tương thích: .NET Core tương thích với .NET Framework, Xamarin và Mono, thông qua .NET Standard.
      * Nguồn mở: Nền tảng .NET Core là nguồn mở, sử dụng giấy phép MIT và Apache 2. .NET Core là một dự án .NET Foundation.

## ASP.NET Core MVC

ASP.NET Core MVC là một nền tảng mã nguồn mở, nhẹ, giúp tối ưu hóa hiệu năng của ứng dụng với ASP.NET Core. ASP.NET Core MVC cung cấp các tính năng dựa trên mô hình xây dựng website động cho phép phân chia rõ ràng các khối lệnh. Nó cung cấp cho bạn toàn quyền kiểm soát đánh dấu, hỗ trợ phát triển với TDD-friendly và sử dụng các tiêu chuẩn web mới nhất.



**Hình 2.5 Mô hình MVC**

**Models:** Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server. Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gởi chúng đến view, ứng dụng không cần phải có tầng model và các lớp liên quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model).

**Views**: Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.

**Controllers**: Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gởi lên (query-string values) và gởi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.

**Ưu điểm:**

**-** Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller.

* Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
* Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến. Để có nhiều thông tin hơn, bạn nên xem phần Front Controller trên web site MSDN.
* Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD).
* Nó hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng.

## Window form application

Windows Form Application hay viết tắt là WinForm là thuật ngữ chỉ việc phát triển các ứng dụng giao diện người dùng bằng cách sử dụng các thành phần xây dựng sẵn (buil in component) còn được gọi là các điều khiển. Hay nói cách khác Windows Forms là một API (Application Programming Interface) cho phép tạo GUI (Graphical User Interface) cho các ứng dụng chạy trên desktop.

Windows Form cho phép người phát triển tạo ra các giao diện người dùng sử dụng các thành phần khác nhau (components). Các thành phần này còn được gọi là các điều khiển (controls). Những điều khiển này cho phép chúng ta thu thập thông tin từ người dùng cũng như trình bày các thông tin để người dùng có thể xem. [3]

* + - * Một Form được chạy trên một máy tính cục bộ (local machine) và một form có thể truy cập đến các tài nguyên khác nhau như bộ nhớ, các thư mục, các tệp tin, các cơ sở dữ liệu…
      * Do đó Windows Form phù hợp cho các ứng dụng desktop như các ứng dụng quản lý thông tin, các ứng dụng tương tác trực tiếp với người dùng.
      * Vai trò của Windows Form:

+ Các Form có thể chứa các điều khiển (các thành phần) khác nhau.

+ Xử lý dữ liệu được nhập bởi người dùng.

+ Hiển thị (trình bày) các thông tin tới người dùng.

+ Kết nối đến các nguồn CSDL khác nhau trên các máy tính cục bộ hoặc máy tính khác

## gRPC service

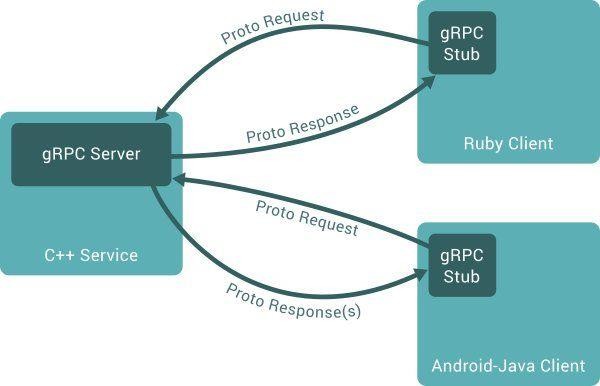
**Inter Process Communications** (**IPC**): là quá trình trao đổi dữ liệu qua lại giữa các tiến trình với nhau, các tiến trình này có thể hoạt động trên cùng một hệ thống, hoặc hoạt động trên nhiều hệ thống khác nhau.

**Remote Procedure Call** (**RPC**) là một phương pháp dùng để trao đổi dữ liệu. RPC khiến cho việc thực hiện IPC dễ dàng, giống như một lời gọi hàm bình thường. RPC có thể được thực hiện giữa hai tiến trình trên cùng một máy tính, hoặc giữa các máy tính khác nhau trong mạng.

**Google Remote Procedure Call (gRPC):** là một platform cải tiến dành cho RPC của Google. gRPC mặc định sử dụng **Protocol Buffers** như là một ngôn ngữ để định

nghĩa các service và là định dạng để giao tiếp giữa các service với nhau. gRPC cấp cho người dùng cơ chế cho phép các service sử dụng những platform khác nhau như Java, NodeJS, Go, Python, Ruby… có thể tương tác với nhau một cách dễ dàng mà không cần thêm công sức để viết các thư viện trung gian. Công nghệ gRPC được xây dựng dựa trên HTTP/2. Những cải tiến của HTTP/2 so với các phiên bản tiền nhiệm cho phép các kết nối http hoạt động tốt hơn, hiệu quả hơn.

**Protocol Buffers** là ngôn ngữ để định nghĩa các service và định dạng, kiểu dữ liệu để giao tiếp giữa các service với nhau. Mỗi định nghĩa Protocol Buffers được đặt trong một file với đuôi là proto.



**Hình 2.6 Sơ đồ giao tiếp giữa gRPC server và gRPC client**

# CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

* 1. **QUY TRÌNH SẮP XẾP THỜI KHÓA BIỂU**
     1. **Quy trình đề xuất áp dụng hệ thống sắp xếp thời khóa biểu tự động**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sinh viên** | **Phòng đào tạo** | | | **Khoa/Bộ môn quản lý học phần** | | **Cán bộ giảng viên** | |
| Đăng kí KHHT  Sắp xếp TKB cá nhân và đăng kí học phần | Tổng hợp KHHT sinh viên | | |  |  | Đăng kí lịch bận/rảnh | |
|  | Phân chia danh sách nhóm HP dự kiến |  |
|  | Điều chỉnh và phân công GV giảng dạy | |
| Kiểm tra và tổng hợp dữ liệu  Ràng buộc xếp TKB  Sắp xếp TKB tự động bằng hệ thống | | |
|  | TKB học phần dự kiến |  |
|  |  |  |
|  | | |
|  |  | | Kiểm tra và điều chỉnh TKB | |  |  |
| Công bố TKB học phần mở trong học kì | | |  | |  | |

**Hình 3.1 Quy trình sắp xếp thời khóa biểu đề xuất**

* + 1. **Nội dung quy trình**
       - Phòng đào tạo tổng hợp KHHT theo từng học phần của tất cả sinh viên trong

học kì.

* + - * Dựa vào số lưọng sinh viên đăng kí của học phần, phòng đào tạo tiến hành lập

danh sách phân chia nhóm học phần dự kiến và gửi đến các Khoa/bộ môn phụ trách giảng dạy các học phần có liên quan.

* + - * Các Khoa/bộ môn phụ trách phân công giảng viên giảng dạy từng học phần. Tùy vào nhân sự hiện tại, khoa/bộ môn phụ trách có thể đề xuất chia lại nhóm học phần cho phù hợp. Trong thời gian này, cán bộ giảng viên đăng kí lịch bận/rảnh trên website của hệ thống.
      * Phòng đào tạo kiểm tra và tổng hợp dữ liệu về: danh sách nhóm học phần, danh sách phân công giảng viên giảng dạy, danh sách phòng học, lịch bận giảng viên và phòng học, các ràng buộc xếp thời khóa biểu.
      * Phòng đào tạo tiến hành sắp thời khóa biểu tự động bằng hệ thống.
      * Phòng đào tạo kiểm tra tính hợp lý thời khóa biểu và điều chỉnh thủ công.
      * Phòng đào tạo công bố lịch giảng cho giảng viên điều chỉnh.
      * Phòng đào tạo công bố thời khóa biểu chính thức lên hệ thống website cho sinh viên sắp xếp thời khóa biểu cá nhân và đăng kí học phần.

Tất cả những bước của quy trình trên tại phòng đào tạo đều được thực hiện trên phần mềm hỗ trợ sắp TKB tự động.

## Mục đích, phạm vi áp dụng:

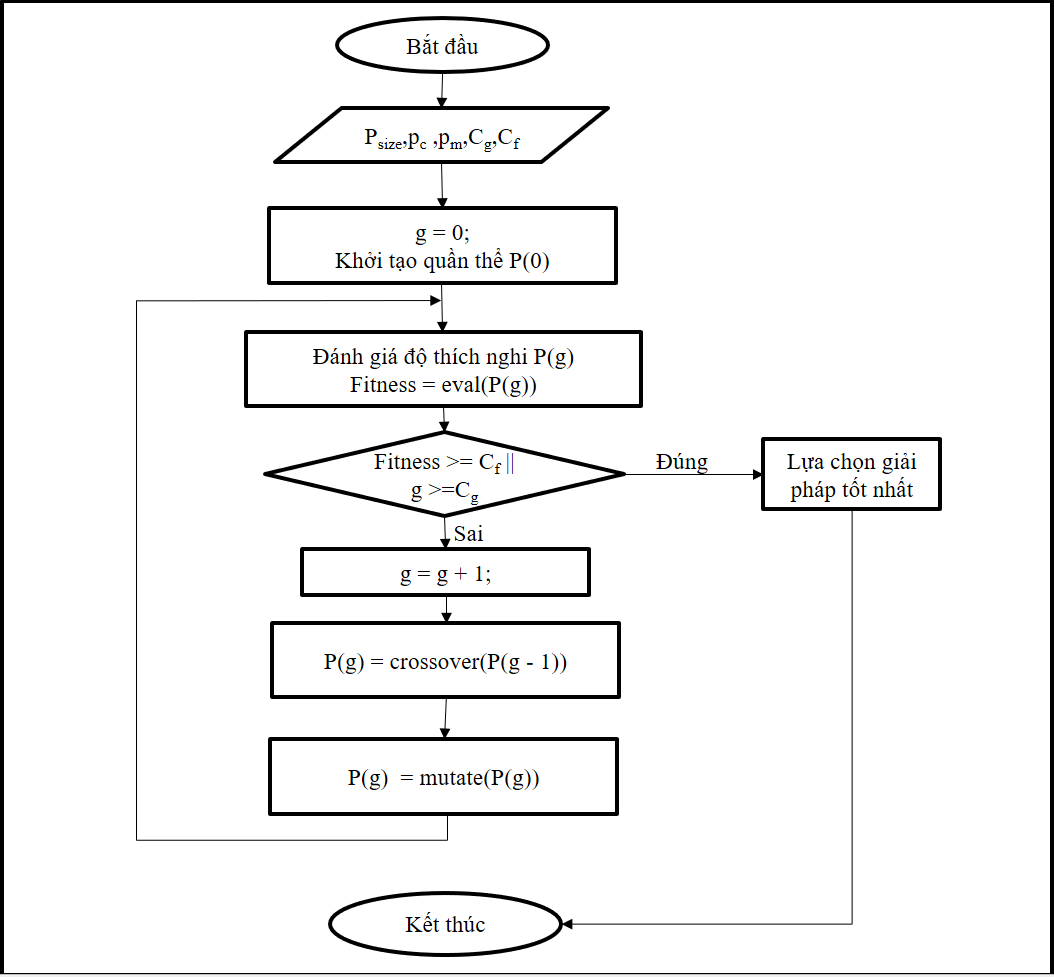
* + - * Lập kế hoạch giảng dạy từng học kỳ.
      * Đảm bảo đáp ứng nhu cầu học tập của sinh viên.
      * Đảm bảo tính hợp lý thời khóa biểu của giảng viên.
      * Đảm bảo tổ chức giảng dạy tốt các học phần.
      * Áp dụng cho phòng đào tạo và các bộ môn quản lý học phần.

## Yêu cầu và ràng buộc thời khóa biểu:

* Yêu cầu:
  + - * Lập kế hoạch mở lớp đúng với nhu cầu của sinh viên đã nhập trong KHHT.
      * Xếp thời khóa biểu hợp lý cho từng giảng viên, đảm bảo cho giảng viên có thể giảng dạy cho nhiều hệ đào tạo khác nhau.
      * Đảm bảo cho việc giảng dạy các lớp học phần thực hiện tốt không mâu thuẫn.
      * Tùy theo điều kiện của đơn vị, khuyến khích hạn chế xếp TKB vào thứ bảy.
* Ràng buộc xếp thời khóa biểu:
  + - * Ràng buộc về trùng lịch phòng học.
      * Ràng buộc về trùng lịch giảng viên.
      * Ràng buộc về số nhóm của giảng viên được phân công giảng dạy.
      * Ràng buộc về tiết học và số tiết học của nhóm học phần.
      * Ràng buộc về sức chứa của phòng học.
      * Ràng buộc về buổi học của nhóm học phần.
      * Ràng buộc về lịch bận của giảng viên.

## THIẾT KẾ VÀ CÀI ĐẶT THUẬT TOÁN

* + 1. **Lưu đồ thuật toán di truyền áp dụng vào bài toán**

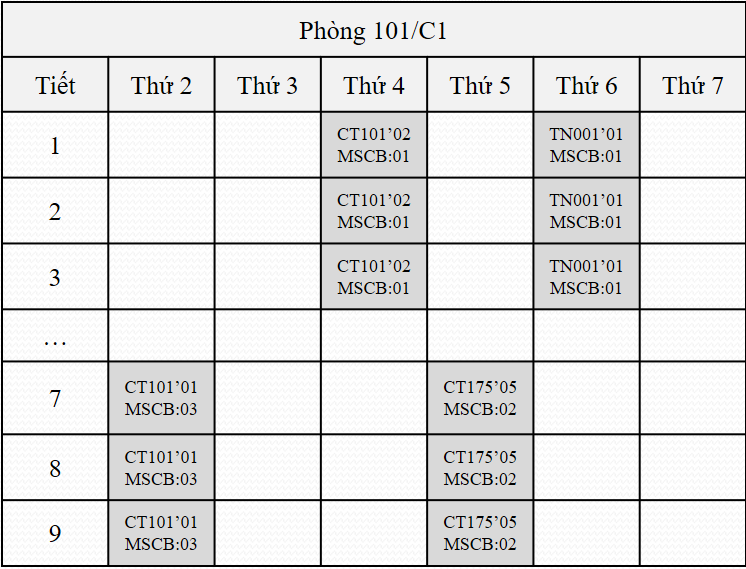


**Hình 3.2 Lưu đồ thuật toán di truyền áp dụng cho hệ thống**

Trong đó:

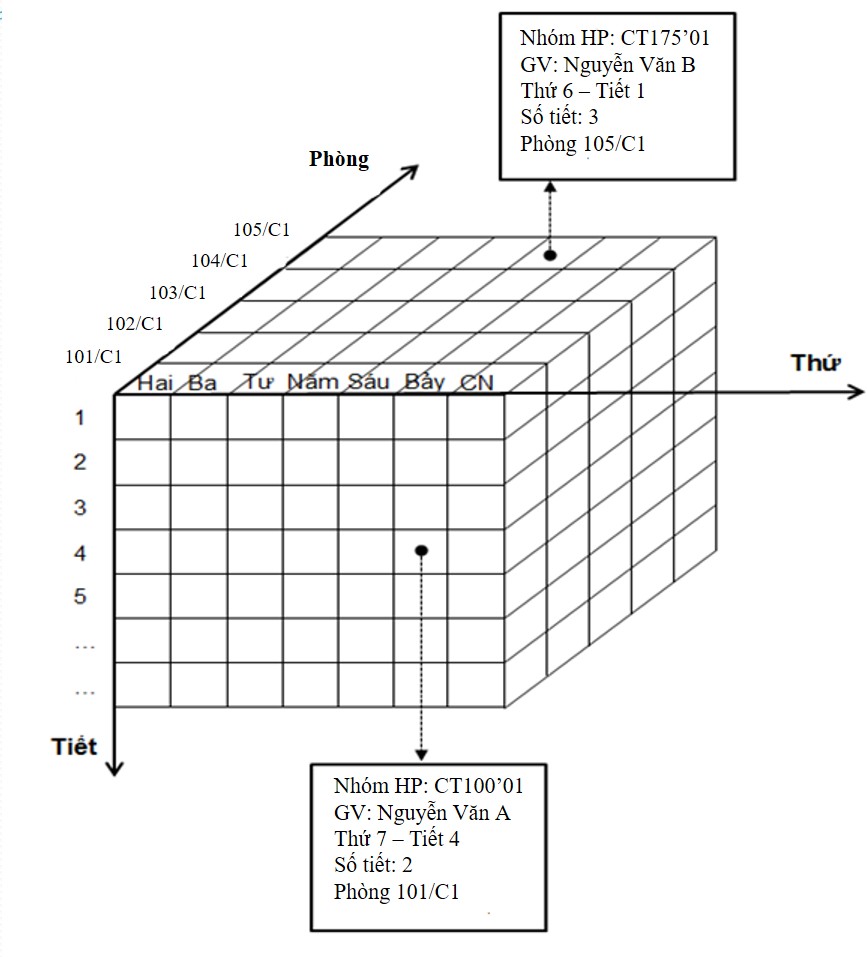
* + - * Psize: Kích thước quần thể, pc: xác suất lai ghép, pm: xác suất đột biến, Cg: số thế hệ, Cf: hệ số tỉ lệ thích nghi chấp nhận được.
      * Fitness: tỉ lệ thích nghi của quần thể.
      * g: thế hệ quần thể.
      * P(g): quần thể ở thế hệ thứ g.

## Biểu diễn mô hình cá thể



**Hình 3.3 Cấu trúc thời khóa biểu theo phòng học**

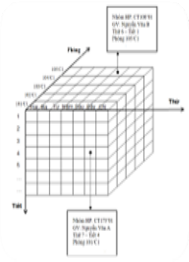
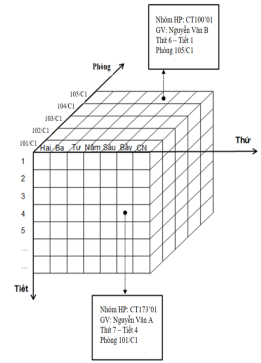
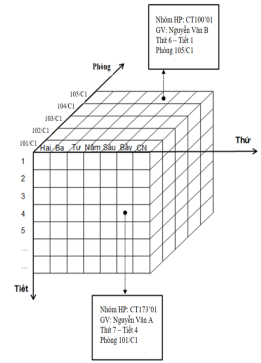
Mỗi một cá thể hay NST dùng để chứa một phương án xếp thời khóa biểu. Mỗi NST có thể xem là một mảng 3 chiều: Chiều thứ nhất (trục y) biểu diễn các tiết học trong ngày, chiều thứ hai (trục x) biểu diễn các ngày trong tuần, chiều thứ ba (trục z) biểu diễn các phòng học. Ứng với một điểm I = (x,y,z) sẽ chứa một giá trị bao gồm nhóm học phần và giảng viên giảng dạy, điểm I được gọi là một gene trong NST.



**Hình 3.4 Mô hình biểu diễn cấu trúc hoàn chỉnh của một NST**

* + 1. **Biểu diễn mô hình quần thể**

Quần thể là tập hợp các NST. Ngoài việc lưu trữ danh sách các NST, quần thể còn chứa thêm các thông tin khác như kích thước quần thể, độ thích nghi mỗi NST.



NST 1

Fitness=0.5

NST 2

Fitness=0.7

...

NST n

Fitness=0.1

Population

Size

**Hình 3.5 Mô hình biểu diễn quần thể NST**

* + 1. **Khởi tạo quần thể**

Khởi tạo quần thể là bước đầu trong thuật giải di truyền, thuật toán có hội tụ nhanh hay chậm đến giá trị tối ưu cũng phụ thuộc vào quần thể khởi tạo ban đầu. Khi khởi tạo quần thể phải khởi tạo tập dữ liệu dữ liệu ban đầu, bao gồm tập danh sách nhóm học phần, khởi tạo tập giảng viên phân công giảng dạy, tập danh sách phòng học, tập dữ liệu về thời gian, …

*Thuật toán khởi tạo quần thể:*

Procedure **Population()**

**Input:** popSize //Số lượng cá thể yêu cầu trong quần thể

**Output:** Population //Quần thể các cá thể

**Begin**

Population = new Individual[popSize] While (i ≤ N) do

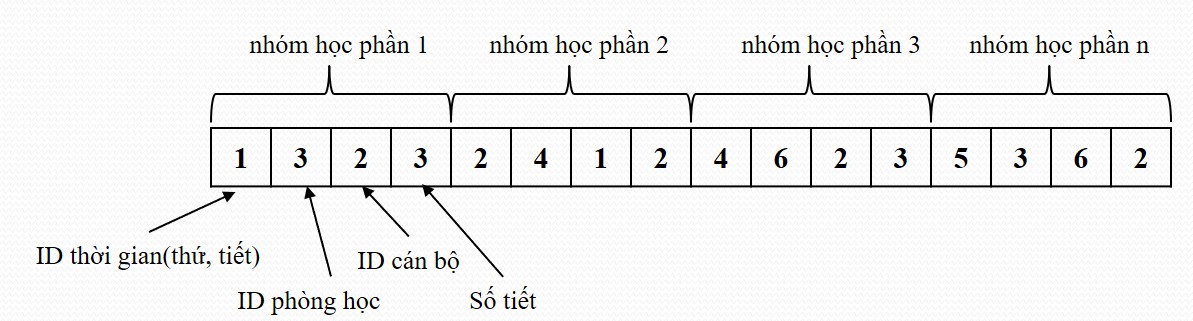
Individual individual = Individual() //tạo một cá thể mới Population[i] = individual //đặt cá thể mới vào quần thể i++

Endwhile

**Return** Population

**End**

Trong đó, **Individual()** là hàm tạo ra cá thể mới, nó được thực hiện trên ý tưởng ứng với mỗi nhóm học phần (Gi) trong tập dữ liệu nhóm học phần (Groups), tiến hành chia ngẫu nhiên: thời gian học, phòng học, giảng viên giảng dạy. Một cá thể mới được tạo ra bằng cách mã hóa (lưu ID) các dữ liệu trên thành một chuỗi số nguyên.



**Hình 3.6 Tạo cá thể bằng cách mã hóa dữ liệu**

ID thời gian: sẽ bao gồm ID thứ (thứ 2 đến thứ 7) và ID tiết (tiết 1 đến tiết 9) – bắt đầu bằng 1 (thứ 2, tiết 1) và cuối cùng là 54 (thứ 7, tiết 9).

ID cán bộ, ID phòng học: đại diện cho cán bộ phân công giảng dạy và phòng học. Số tiết: cho biết số tiết học trong một buổi của nhóm học phần đó.

***Ví dụ:*** Nhóm học phần 1 học vào tiết 1,2,3 - thứ 2 tại phòng có ID là 3, do cán bộ có ID là 2 giảng dạy.

*Thuật toán khởi tạo cá thể:*

Procedure **Individual()**

**Input:** Request timetable //các tập dữ liệu yêu cầu để xếp TKB

**Output:** TKB (individual – cá thể)

**Begin**

individual = new integer[Groups.length]; integer index = 0;

Foreach Gi ∈{Groups} do

individual[index] = Random(Timeslots).getID(); index++;

individual[index] = Random(Rooms).getID(); index++;

individual[index] = Random(Professors).getID(); index++;

individual[index] = Gi.periods; index++;

EndFor

**Return** individual;

**End**

## Các phép toán di truyền

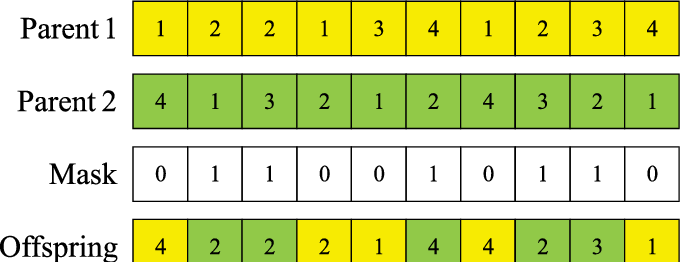
* + - 1. **Toán tử chọn lọc**

Ta sử dụng toán tử chọn lọc xếp hạng để giải quyết bài toán. Với cách làm này các NST trong quần thể được sắp xếp giảm dần theo độ thích nghi của chúng.

* + - 1. **Toán tử lai ghép**

Do bài toán có cấu trúc NST khá phức tạp, vì vậy ta chọn toán tử lai ghép đồng nhất để áp dụng. Ý tưởng của phương pháp lai ghép, với mỗi giá trị của mặt nạ, nếu mặt

nạ có giá trị là 1 thì cá thể con sẽ nhận gen của cha (mẹ), ngược lại là gen của mẹ (cha).



**Hình 3.7 Phương pháp lai ghép đồng nhất**

*Thuật toán lai ghép*:

Procedure **crossoverPopulation() Input:** Population: Po

**Output:** newPopulation: Pn

**Begin**

For each individual in population: parent1 = individual; newPopulation = new array; If crossoverRate > random():

parent2 = selectParent(); //Random select individual offspring = crossover(parent1, parent2); newPopulation.push(offspring);

Else: newPopulation.push(parent1); End if

End loop;

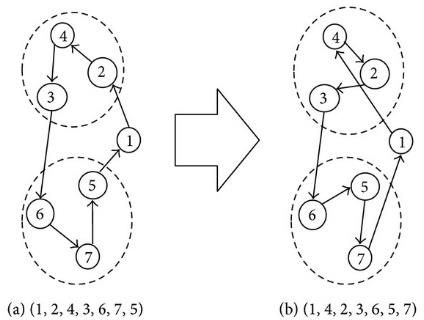
**Return** Pn;

**End**

Để thuật toán trở nên đơn giản ta tạo mặt nạ nhị phân bằng hàm random(). Nếu random() < 0.5 thì xem là 0, ngược lại thì xem là 1.

* + - 1. **Toán tử đột biến**

Trong bài toán, nhiễm sắc thể đại diện cho lời giải của bài toán và mỗi gen trong nhiễm sắc thể có một xác suất đột biến là p, ví dụ: p = 0.03 tức với 100 cá thể trong quần thể thì có 0.03\*100 = 3 cá thể sẽ bị đột biến trong mỗi thế hệ, và quá trình đột biến được thực hiện bằng phương pháp đột biến tương hỗ bằng cách hoán vị 2 gen bất kỳ trong một nhiễm sắc thể.



**Hình 3.8 Phương pháp đột biến tương hổ**

*Thuật toán đột biến:*

Procedure **mutationPopulation() Input:** Population: Po

**Output:** newPopulation: Pn

**Begin**

For each individual in population: For each gen in individual

If mutationRate > random():

newgen = individual.getGenRandom(); individual.setGen(gen.index,newgen);

End if End loop

newPopulation.push(individual); End loop

**Return** Pn;

**End**

## Tính độ thích nghi của cá thể

* + - 1. **Tính độ thích nghi của cá thể**

Việc đánh giá độ thích nghi của cá thể được căn cứ vào số lần vi phạm các ràng buộc. Để thực hiện, đầu tiên ta tính độ thích nghi của cá thể dựa trên từng ràng buộc, sau đó dựa vào hàm đánh giá cá thể ta sẽ tính được độ thích nghi của cá thể đó.

* **Công thức tính độ thích nghi mỗi ràng buộc của cá thể:**

FRB= log(𝑅+𝑥)

𝑅

(R ∈ N\*, x ∈ N)

Trong đó: R – hệ số ràng buộc xếp TKB, x – số vi phạm ràng buộc FRB = 0 - Ràng buộc đã thỏa mãn, không còn vi phạm.

* **Công thức tính độ thích nghi cá thể:**

F = 1 - log(1 + ∑ 𝐹 ) = 1 – log(1 + ∑𝑛 log(𝑅𝑖+𝑥𝑖))

𝑅𝐵

𝑖=0

𝑅𝑖

∑ 𝐹𝑅𝐵 : Tổng của tất cả độ thích nghi dựa trên ràng buộc.

∑ 𝐹𝑅𝐵 = 0 => F = 1 - log(1) = 1 => không có vi phạm ràng buộc, TKB đạt giải pháp tối ưu.

*Thuật toán tính độ thích nghi ràng buộc:*

Function Độ\_thích\_nghi\_RB (Cathe) Begin

Count = 0 {Biến đếm số lần vi phạm} For each gen in Cathe

Begin

If (gen vi phạm ràng buộc RB) Then Count = Count + 1

End Return log(𝑅+𝐶𝑜𝑢𝑛𝑡)

𝑅

End

* + - 1. **Tính độ thích nghi dựa vào ràng buộc trùng lịch phòng học**

Vi phạm ràng buộc trùng lịch phòng học xảy ra khi hai nhóm học phần khác nhau được sắp lịch học cùng một phòng tại cùng một thời điểm. Ví dụ:

* TKB nhóm CT101’01: phòng 101/C1, Thứ 2, tiết bắt đầu: 1, số tiết: 3.
* TKB nhóm CT175’01: phòng 101/C1, Thứ 2, tiết bắt đầu: 3, số tiết: 3.

Nhóm CT175’01 đã vi phạm trùng lịch phòng học với nhóm CT101’01 ở tiết 3

*Thuật toán:*

Function Fitness\_schedule\_room Begin

Count = 0 (biến đếm số vi phạm ràng buộc) For each classA in timetable

For each classB in timetable

If classA.room = classB.room & classA.time = classB.time Count++

End if End loop

End loop

Return log(𝑅+𝐶𝑜𝑢𝑛𝑡)

𝑅

End

* + - 1. **Tính độ thích nghi dựa vào ràng buộc trùng lịch cán bộ giảng dạy**

Vi phạm ràng buộc trùng lịch cán bộ giảng dạy xảy ra khi hai nhóm học phần khác nhau được sắp cùng một cán bộ giảng dạy tại cùng một thời điểm. Ví dụ:

* TKB nhóm CT101’01: Thứ 2, tiết bắt đầu: 1, số tiết: 3, CB: Nguyen Van A
* TKB nhóm CT175’01: Thứ 2, tiết bắt đầu: 3, số tiết: 3, CB: Nguyen Van A Nhóm CT175’01 đã vi phạm trùng lịch giảng viên với nhóm CT101’01 ở tiết 3

*Thuật toán:*

Function Fitness\_schedule\_teacher Begin

Count = 0 (biến đếm số vi phạm ràng buộc) For each classA in timetable

For each classB in timetable

If classA.teacher = classB.teacher & classA.time = classB.time Count++

End if

End loop

Return

End

* + - 1. **Tính độ thích nghi dựa vào ràng buộc số nhóm phân công giảng dạy**

Một học phần được phân chia thành nhiều nhóm học phần dựa vào số lượng sinh viên đăng kí trong KHHT. Bộ môn phụ trách học phần có trách nhiệm phân công số nhóm giảng dạy cho cán bộ phù hợp để đảm bảo tránh cho việc một cán bộ được xếp giảng dạy nhiều nhóm học phần, còn cán bộ khác thì không được xếp lịch dạy.

Vi phạm ràng buộc số nhóm phân công giảng dạy xảy ra khi số nhóm học phần được xếp của giảng viên nhiều hơn số nhóm học phần đã được phân công.

*Thuật toán:*

Function Fitness\_teaching\_group\_numbers Begin

Count = 0 (biến đếm số vi phạm ràng buộc)

For each teacher in teachers sonhom = 0;

For each class in timetable

If class.teacher = teacher sonhom++

End if End loop

If teacher.sonhom < sonhom : Count++ end if

End loop

Return log(𝑅+𝐶𝑜𝑢𝑛𝑡)

𝑅

End

* + - 1. **Tính độ thích nghi dựa vào ràng buộc sức chứa phòng học**

Vi phạm ràng buộc sức chứa phòng học xảy ra khi một nhóm học phần được xếp có sĩ số lớn hơn sức chứa của phòng học.

*Thuật toán:*

Function Fitness\_capacity\_room Begin

Count = 0 (biến đếm số vi phạm ràng buộc) For each classA in timetable

If class.size > class.room.capacity Count++

End if

End loop

Return log(𝑅+𝐶𝑜𝑢𝑛𝑡)

𝑅

End

* + - 1. **Tính độ thích nghi dựa vào ràng buộc tiết học**

Vì thời gian học được chia làm 2 buổi: buổi sáng: từ tiết đến tiết 5, buổi chiều: từ tiết 6 đến tiết 9. Do đó TKB các nhóm học phần không được bắt đầu vào tiết 5 và tiết 9, hoặc thời gian học không được vượt quá tiết 5 và tiết 9.

Ví dụ: TKB nhóm CT101’02: Thứ 2, tiết bắt đầu 4, số tiết: 3.

*Thuật toán:*

Function Fitness\_period Begin

Count = 0 (biến đếm số vi phạm ràng buộc) For each classA in timetable

TietBD = classA.TietBD

TietKT = classA.sotiet+classA.tietBD

If (TietBD<=5&&TietKT>5)|||(TietBD<=9&&TietKT>9) Count++

End if

End loop

Return log(𝑅+𝐶𝑜𝑢𝑛𝑡)

𝑅

End

* + - 1. **Tính độ thích nghi dựa vào ràng buộc buổi học**

Vi phạm ràng buộc buổi học xảy ra khi một học phần có nhiều buổi học (2 buổi trở lên), các buổi học được xếp vào cùng một ngày hoặc không cùng một cán bộ giảng dạy cho các buổi đó.

*Thuật toán:*

Function Fitness\_session Begin

Count = 0 (biến đếm số vi phạm ràng buộc)

For each group in Groups //lặp từng nhóm HP trong danh sách nhóm HP If group.session = 1 : continue; //bỏ qua nhóm HP có 1 buổi học

For each classA in timetable(group)

//lặp từng lớp học trong TKB của nhóm HP For each classB in timetable(group)

If classA.day = classB.day || classA.teacher != classB.teacher Count++

End if End loop

End loop

End loop

Return log(𝑅+𝐶𝑜𝑢𝑛𝑡)

𝑅

End

* + - 1. **Tính độ thích nghi dựa vào ràng buộc trùng lịch bận cán bộ**

Vi phạm ràng buộc trùng lịch bận cán bộ xảy ra khi TKB của cán bộ có lịch giảng dạy trùng với lịch bận đã đăng kí với hệ thống trước đó.

*Thuật toán:*

Function Fitness\_schedule\_busy\_teacher Begin

Count = 0 (biến đếm số vi phạm ràng buộc) For each classA in timetable

Teacher = classA.teacher

For each time in Teacher.timebusy If(classA.time = time)

Count++ End if

End loop

End loop

Return log(𝑅+𝐶𝑜𝑢𝑛𝑡 )

𝑅

End

## Trọng số các loại vi phạm ràng buộc

Trong các loại ràng buộc, mỗi loại ràng buộc có một tính chất riêng. Có loại vi phạm ràng buộc thường xuyên xảy ra và có loại ít xảy ra hơn. Có loại dễ đạt được và có loại khó đạt được hơn. Vì vậy, nếu xem xét các ràng buộc này một cách bình đẳng thì chắc chắn hệ thống sẽ không xác định được vi phạm nào cần xử lý trước và vi phạm nào cần xử lý sau, điều này dẫn đến việc có thể không tìm ra được phương án xếp thời khóa biểu thỏa mãn yêu cầu.

**Nguyên tắc 1:** Ràng buộc nào cần đạt được trước thì đặt trọng số thấp, ràng buộc nào cần đạt được sau thì đặt trọng số cao.

**Nguyên tắc 2:** Ràng buộc nào khó đạt được hơn thì đặt trọng số thấp, dễ đạt được hơn thì đặt trọng số cao.

**Nguyên tắc 3:** Ràng buộc mềm nên đặt trọng số cao hơn so với ràng buộc cứng. Vì các ràng buộc mềm là những ràng buộc thêm, nếu đạt được thì càng tốt còn không thì cũng có thể chấp nhận được.

## Điều kiện dừng quá trình tiến hóa

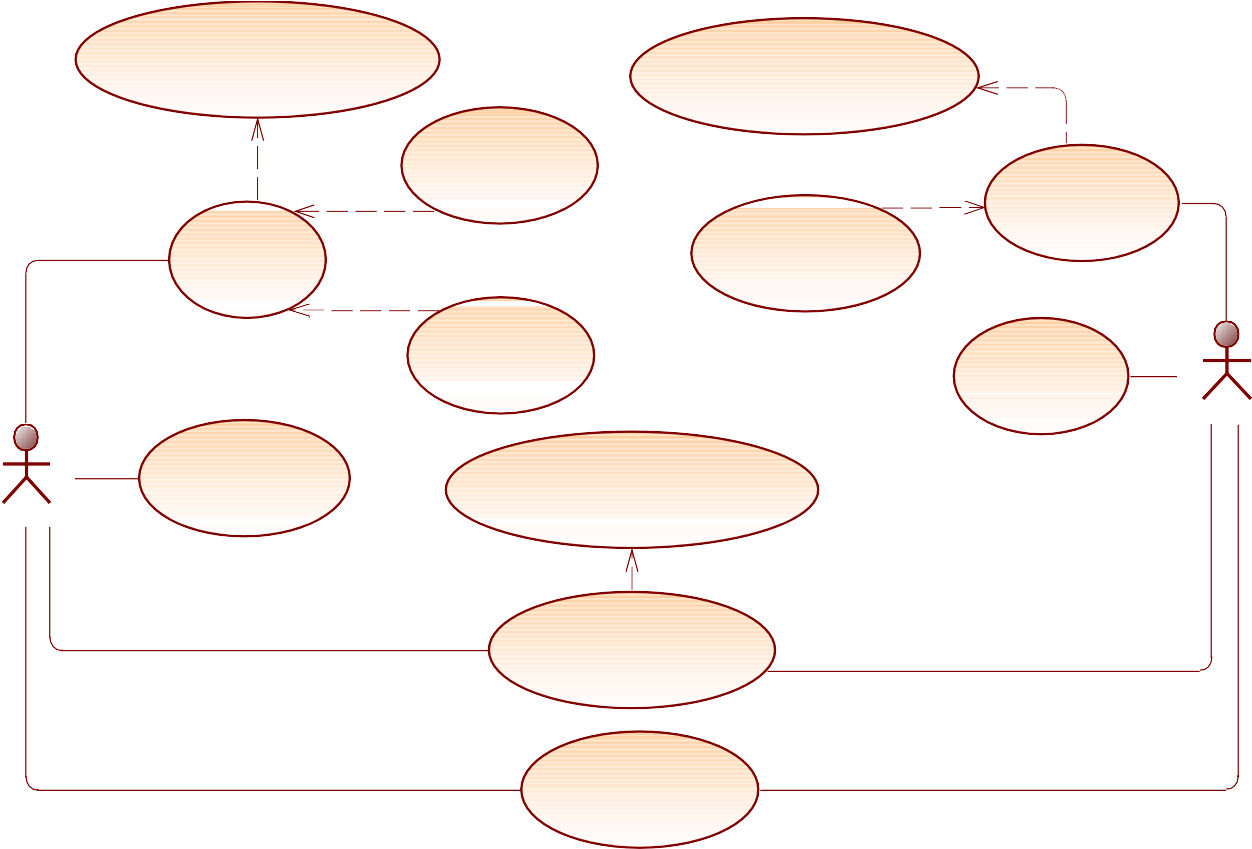
**Điều kiện 1:** Hệ thống được thiết kế cho phép người dùng có thể chủ động trong việc quyết định số lượng vi phạm ràng buộc của TKB (x) thông qua việc kết hợp các hệ số ràng buộc vi phạm (R) và hệ số tỉ lệ thích nghi chấp nhận được (Cfitness). Nếu số x < R và tỉ lệ thích nghi của quần thể lớn hơn hoặc bằng Cfitness, khi đó quá trình tiến hóa sẽ kết thúc.

**Điều kiện 2:** Dựa vào hệ số số thế hệ (Cgeneration), quá trình tiến hóa sẽ kết thúc khi vượt qua Cgeneration bất chấp kết quả đạt được có thỏa mãn yêu cầu.

Khi kết hợp cả 2 điều kiện: Cfitness = 1.0 và Cgeneration = MAX thì quần thể sẽ tiến hóa đến khi nào tìm được phương án tối ưu nhất.

## THIẾT KẾ CHỨC NĂNG HỆ THỐNG

* + 1. **Sơ đồ use case module web-app hỗ trợ cán bộ – sinh viên**



Tìm kiếm KHHT theo học kì - năm học

Tìm kiếm TKB theo học kì - năm học

<<include>>

<<include>>

Thêm HP vào KHHT

<<extend>>

<<extend>>

Xem TKB giảng dạy

Cập nhật KHHT

Cập nhật lịch giảng dạy

<<extend>>

Xóa HP khỏi KHHT

Cập nhật lịch bận

Can bo

Xem KHHT toàn khóa

Sinh viên

Tìm kiếm dữ liệu theo học kì - năm học

<<include>>

Xem danh mục TKB học phần

Đăng nhập

**Hình 3.9 Sơ đồ use case module web-app hỗ trợ cán bộ – sinh viên**

Sơ đồ use case mô tả các chức năng của module web-app hỗ trợ cán bộ – sinh viên. Trong đó:

* Actor: Sinh viên – có thể truy cập đến các chức năng: Xem KHHT toàn khóa, Cập nhật KHHT (thêm, xóa học phần), xem danh mục TKB học phần.
* Actor: Cán bộ – có thể truy cập đến các chức năng: Cập nhật lịch bận, Xem TKB giảng dạy, Cập nhật lịch giảng dạy, xem danh mục TKB học phần.

Cả hai actor bắt buộc phải đăng nhập trước khi truy cập đến hệ thống.

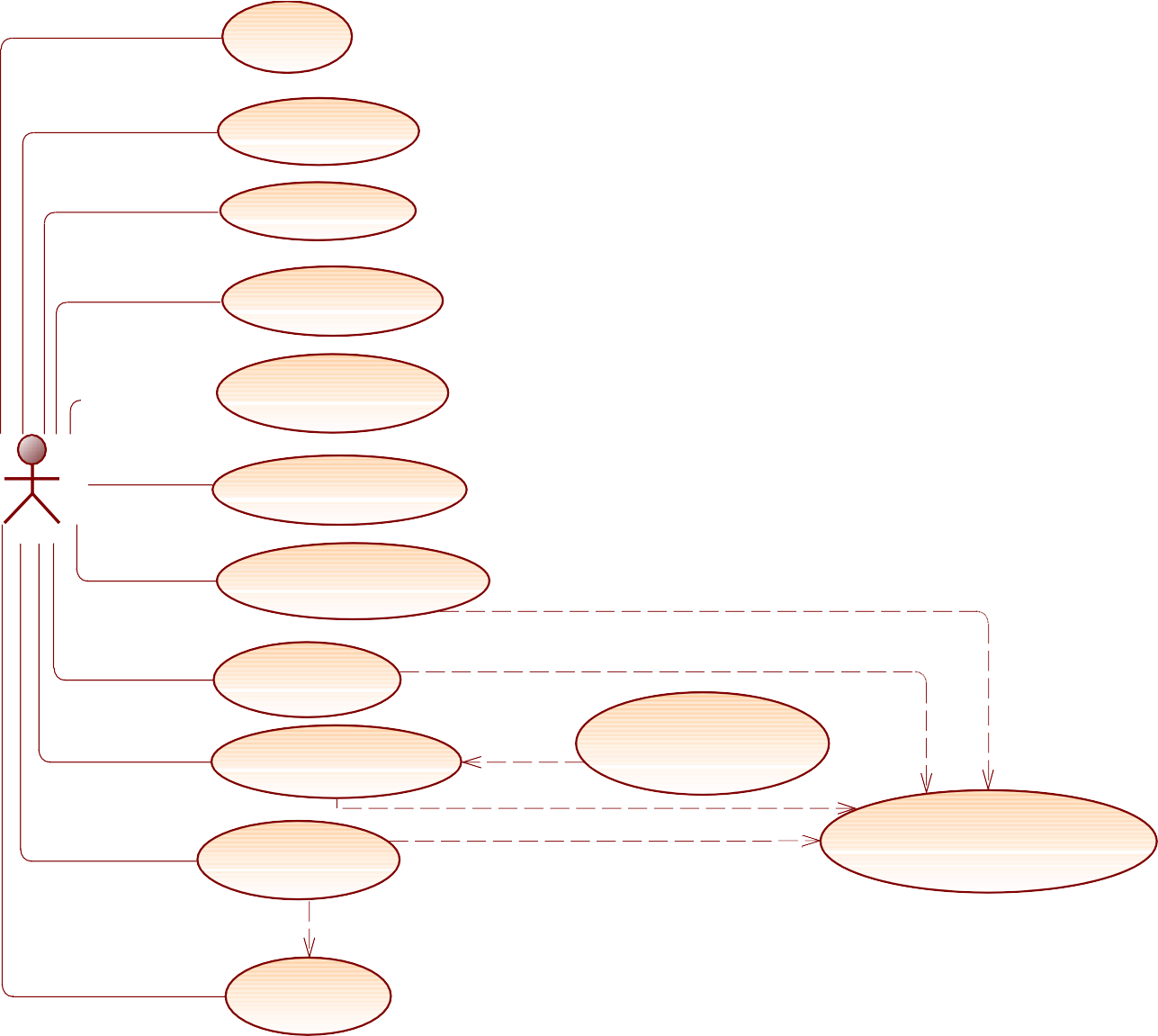
## Sơ đồ use case mức 0 module phần mềm sắp xếp TKB

* Actor: Admin/cán bộ phụ trách giáo vụ.

Actor phải đăng nhập trước khi truy cập hệ thống. Các chức năng quản lý dữ liệu (cán bộ, đơn vị, phòng học, học phần, thời gian học, nhóm học phần) đều có chức năng

chung: thêm, sửa, xóa dữ liệu, xuất file excel.

Các chức năng quan trọng: tổng hợp KHHT, chia nhóm học phần tự động, quản lý nhóm học phần, phân công giảng dạy, quản lý thời khóa biểu, sắp xếp thời khóa biểu.



Đăng nhập

Quản ly ́ dư ̃ liệu cán bô ̣

Quản ly ́ dư ̃ liệu đơn vi ̣

Quản ly ́ dư ̃ liệu học phần

Quản ly ́ dư ̃ liệu phòng học

Quản ly ́ dư ̃ liệu thời gian học

Admin

Quản ly ́ dư ̃ liệu nhóm học phần

<<include>>

<<include>>

Phân công giảng dạy

Tổng hợp KHHT sinh viên

<<extend>>

Chia nhóm học phần tự động

<<include>>

Quản ly ́ thời khóa biểu

<<include>>

Tìm kiếm dư ̃ liệu theo học kì - năm học

<<include>>

Sắp xếp TKB

**Hình 3.10 Sơ đồ use case các chức năng chính của module phần mềm sắp TKB**

* + 1. **Sơ đồ use case chức năng quản lý và sắp xếp thời khóa biểu**
* Actor: Admin / Cán bộ phụ trách giáo vụ.

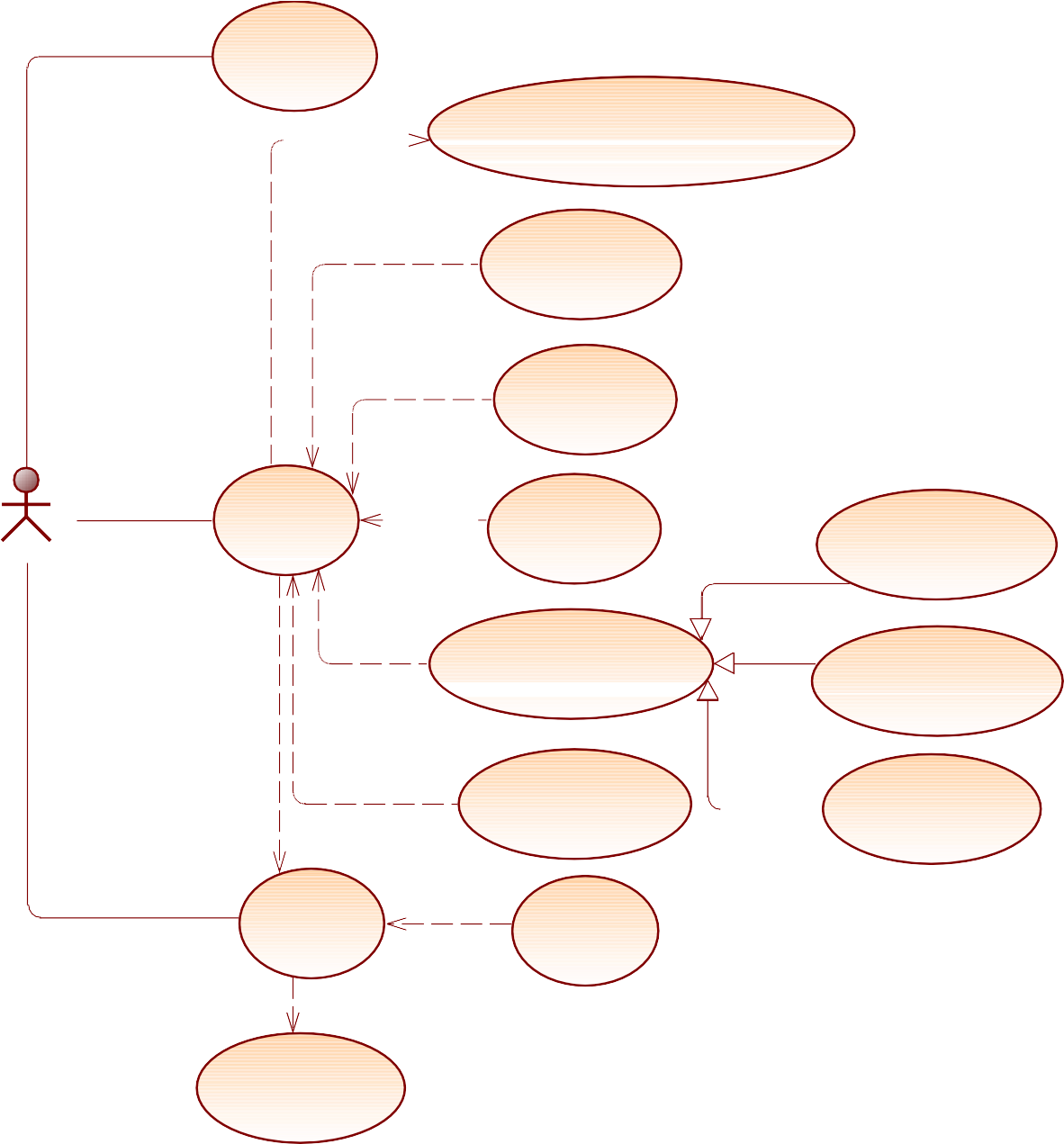
Actor phải đăng nhập trước khi truy cập chức năng này.

**Chức năng quản lý thời khóa biểu:**

* Điều kiện tiên quyết: Tìm kiếm dữ liệu TKB theo học kì – năm học.
* Các chức năng mở rộng: Xem danh mục TKB, thêm mới, xóa, cập nhật, xuất file excel.

**Chức năng sắp xếp thời khóa biểu:**

* Điều kiện tiên quyết: Load dữ các tập dữ liệu đầu vào.
* Chức năng mở rộng: Lưu TKB (nếu thỏa mãn điều kiện đặt ra)



Đăng nhập

Tìm kiếm dư ̃ liệu TKB theo học kì - năm học

<<include>>

<<extend>>

Thêm TKB mới

<<extend>>

Cập nhật TKB

Admin

Quản lý TKB

<<ext end>>

Xóa TKB

Xem TKB theo học phần

Xem danh mục TKB

<<extend>>

Xem TKB theo phòng học

<<include>>

<<extend>>

Xuất file danh mục TKB

Xem TKB theo cán bộ

Sắp xếp TKB

<<extend>>

Lưu trữ TKB

<<include>>

Load dữ liệu đầu vào

**Hình 3.11** **Sơ đồ use case các chức năng chi tiết sắp xếp và quản lý TKB**

* 1. **THIẾT KẾ MÔ HÌNH DỮ LIỆU**
     1. **Đặc tả dữ liệu**
        1. **Bảng dữ liệu CanBo**

**Mô tả:** Dùng để lưu trữ các thông tin cá nhân của sinh viên và cán bộ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_canbo (\*) | Int |  | Primary key | ID cán bộ |
| Ten\_canbo | Nvarchar |  |  | Họ tên cán bộ |
| GioiTinh | Int | X |  | Giới tính: 0 – nữ, 1 – nam |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NgaySinh | Date | X |  | Ngày |
| SDT | Varchar | X |  | Số điện thoại |
| Email | Varchar | X |  | Email |
| Username | Varchar |  |  | Tài khoản đăng nhập |
| Password | Varchar |  |  | Mật khẩu đăng nhập |
| (\*) bảng CanBo tham gia vào việc sắp xếp TKB, nên khóa chính được lưu bằng kiểu  int để phục vụ việc truy xuất dữ liệu và mã hóa chuỗi gene | | | | |

**Bảng 3.1 Các thuộc tính bảng dữ liệu CanBo**

* + - 1. **Bảng dữ liệu SinhVien**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| MSSV | Varchar |  | Primary key | Mã số sinh viên |
| Ten\_SinhVien | Nvarchar |  |  | Họ tên sinh viên |
| GioiTinh | Int | X |  | Giới tính: 0 – nữ, 1 – nam |
| NgaySinh | Date | X |  | Ngày |
| SDT | Varchar | X |  | Số điện thoại |
| Email | Varchar | X |  | Email |
| Username | Varchar |  |  | Tài khoản đăng nhập |
| Password | Varchar |  |  | Mật khẩu đăng nhập |

**Bảng 3.2 Các thuộc tính bảng SinhVien**

* + - 1. **Bảng dữ liệu DonVi và LoaiDonVi**

**Mô tả:** Bảng DonVi và LoaiDonVi lưu trữ thông tin của một đơn vị (Khoa, Bộ Môn, …)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| Ma\_DonVi | Varchar |  | Primary Key | Mã đơn vị |
| Ten\_DonVi | NVarchar | X |  | Tên đơn vị |
| DiaChi | Nvarchar | X |  | Địa chỉ |
| Email | Varchar | X |  | Email |
| Sdt | Varchar | X |  | Số điện thoại |
| GioiThieu | NVarchar | X |  | Giới thiệu đơn vị |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NgayThanhLap | Date | X |  | Ngày thành lập đơn vị |

**Bảng 3.3 Các thuộc tính bảng dữ liệu DonVi**

* + - 1. **Bảng dữ liệu KhuNhaHoc và PhongHoc**

**Mô tả:** KhuNhaHoc mô tả các thông tin của khu nhà học, mỗi khu nhà học sẽ bao gồm nhiều phòng học.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_PhongHoc(\*) | Int |  | Primary key | Mã định danh phòng học |
| TenPhongHoc | Varchar | X |  | Tên phòng học (số  phòng/tên khu nhà học) |
| SucChua | Int | X |  | Sức chứa phòng học |
| (\*) bảng PhongHoc tham gia vào việc sắp xếp TKB, nên khóa chính được lưu bằng  kiểu int để phục vụ việc truy xuất dữ liệu và mã hóa chuỗi gene. | | | | |

**Bảng 3.4 Các thuộc tính của bảng PhongHoc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_KhuNhaHoc | Int |  | Primary key | Mã định danh khu nhà học |
| TenKhuNhaHoc | Varchar | X |  | Tên khu nhà học |
| ViTri | Nvarchar | X |  | Vị trí |

**Bảng 3.5 Các thuộc tính bảng KhuNhaHoc**

* + - 1. **Bảng dữ liệu HocPhan và NhomHocPhan**

**Mô tả:** Lưu trữ thông tin về học phần và nhóm học phần được chia mỗi học kì – năm học dựa vào số lượng sinh viên đăng kí học phần trong KHHT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| MaHocPhan | Varchar |  | Primary key | Mã học phần |
| TenHocPhan | NVarchar |  |  | Tên học phần |
| SoTinChi | Int |  |  | Số tín chỉ |
| SoTietLT | Int | X |  | Số tiết lý thuyết |
| SoTietTH | Int | X |  | Số tiết thực hành |
| SoTietMoiTuan(\*) | Int |  |  | Số tiết học qui định mỗi  tuần |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SoBuoiMoiTuan(\*) | Int |  |  | Số buổi học qui định  mỗi tuần |
| SiSo\_QuiDinh(\*) | Int |  |  | Sỉ số qui định |
| TienQuyet | Varchar | X |  | Tiên quyết |
| HocKi\_hoc | Varchar | X |  | Học kì học |
| Is\_schedule | int |  |  | Là học phần có sắp học  hay không? |
| (\*) các thuộc tính qui định dùng để chia nhóm học phần tự động | | | | |

**Bảng 3.6 Các thuộc tính bảng HocPhan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_NhomHocPhan(\*) | Int |  | Primary key | Mã định danh nhóm  HP |
| Ten\_NhomHocPhan | Varchar |  |  | Tên nhóm học phần |
| SoTietHoc(\*\*) | Int |  |  | Số tiết học trong tuần |
| SoBuoiHoc(\*\*) | Int |  |  | Số buổi học |
| SiSo(\*\*) | Int |  |  | Sỉ số nhóm học phần |
| (\*) bảng NhomHocPhan tham gia vào việc sắp xếp TKB, nên khóa chính được lưu bằng kiểu int để phục vụ việc truy xuất dữ liệu và mã hóa chuỗi gene.  (\*\*) Dữ liệu các thuộc tính này sẽ được lấy từ các thuộc tính tương ứng của bảng HocPhan khi chia nhóm học phần tự động, tuy nhiên cũng có thể điều chỉnh lại thủ  công. | | | | |

**Bảng 3.7 Các thuộc tính bảng NhomHocPhan**

* + - 1. **Bảng dữ liệu Thu, Tiet, ThoiGianHoc, và HocKi\_NamHoc**

**Mô tả:** Các bảng Thu, Tiet, ThoiGianHoc mô tả thời gian học trong tuần, HocKi\_NamHoc dùng để lưu trữ thông tin về học kì trong năm học.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_thu | Int |  | Primary key | Mã định danh thứ |
| Thu | NVarchar |  |  | Thứ trong tuần |

**Bảng 3.8 Các thuộc tính bảng Thu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_tiet | Int |  | Primary key | Mã định danh tiết |
| Tiet | Varchar |  |  | Tiết trong ngày |
| TG\_BatDau | Varchar | X |  | Thời gian bắt đầu tiết |
| TG\_KetThuc | Varchar | X |  | Thời gian kết thúc tiết |
| SoPhut | Int | X |  | Số phút |

**Bảng 3.9 Các thuộc tính bảng Tiet**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_ThoiGianHoc(\*) | Int |  | Primary key | Mã định danh thời gian  học |
| ID\_thu | Int |  |  | Thứ trong tuần |
| ID\_tiet | Int |  |  | Tiết trong ngày |
| Is\_schedule | Int |  |  | Có sắp lịch học vào tiết  này hay không? |
| (\*)bảng ThoiGianHoc tham gia vào việc sắp xếp TKB, nên khóa chính được lưu bằng  kiểu int để phục vụ việc truy xuất dữ liệu và mã hóa chuỗi gene. | | | | |

**Bảng 3.10** **Các thuộc tính bảng ThoiGianHoc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_HocKi\_NamHoc | Int |  | Primary key | Mã định danh học kì  trong năm học |
| HocKi | Int |  |  | Học kì trong năm |
| NamHoc | Varchar |  |  | Năm học |
| TG\_BatDau | Varchar | X |  | Thời gian bắt đầu |
| TG\_KetThuc | Varchar | X |  | Thời gian kết thúc |

**Bảng 3.11** **Các thuộc tính bảng HocKi\_NamHoc**

* + - 1. **Bảng KHHT**

**Mô tả:** Lưu trữ thông tin KHHT của tất cả sinh viên đã đăng KHHT.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID\_HocKi\_NamHoc | Int |  | Primary key | học kì đăng kí |
| MSSV | Varchar |  | Primary key | Sinh viên đăng kí |
| MaHocPhan | Varchar |  | Primary key | Học phần đăng kí |
| CaiThien | Int |  |  | La HP cai thien diem?  (0-không, 1-có) |

**Bảng 3.12** **Các thuộc tính bảng KHHT**

* + - 1. **Bảng GiangDay**

**Mô tả:** Thông tin phân công giảng dạy trong học kì sẽ được lưu ở bảng này, thuộc tính SoNhom qui định số nhóm học phần mà giảng viên sẽ phải dạy học phần đó trong học kì, đây là một ràng buộc trong bài toán.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_canbo | Int |  | Primary key | Cán bộ giảng dạy |
| MaHocPhan | Varchar |  | Primary key | Học phần giảng dạy |
| ID\_hocki\_namhoc | Varchar |  | Primary key | Học kì giảng dạy |
| SoNhom | Int |  |  | Số nhóm giảng dạy |

**Bảng 3.13** **Các thuộc tính bảng GiangDay**

* + - 1. **Bảng ThoiKhoaBieu**

**Mô tả:** Lưu thông thời khóa biểu của các nhóm học phần trong một học kì.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ**  **liệu** | **Null?** | **Key** | **Mô tả** |
| ID\_ThoiKhoaBieu | Int |  | Primary key | Mã định TKB |
| ID\_canbo | Int |  | Forgein key | Cán bộ giảng dạy |
| ID\_NhomHocPhan | Int |  | Forgein key | Nhóm học phần |
| ID\_phonghoc | Int |  | Forgein key | Phòng học |
| ID\_thu | Int |  | Forgein key | Thứ |
| ID\_tiet | Int |  | Forgein key | Tiết bắt đầu học |
| So\_Tiet | Int |  |  | Số tiết học |
| ID\_hocki\_namhoc | Varchar |  | Forgein key | Học kì giảng dạy |
| TuanHoc | Varchar |  |  | Tuần học trong học kì |

**Bảng 3.14 Các thuộc tính bảng ThoiKhoaBieu**

* + 1. **Mô hình cơ sở dữ liệu**

KHU\_NHA\_HOC



ID\_KhuNhaHoc Integer Ten\_KhuNhaHoc Text

Co ID\_PhongHoc TenPhongHoc

PHONGHOC

<pi> Integer <M> Text

THOI\_KHOA\_BIEU

ID\_ThoiKhoaBieu <pi> Integer

<M>

Relationship\_5

ViTri

Text

SucChua Identifier\_1 <pi>

Integer

hoc tai

SoTietHoc TuanHoc

Identifier\_1 <pi>

Integer

Variable multibyte

tiet bat dau

ID\_Tiet

TIET

<pi> Integer <M>

HOCKI\_NAMHOC

ID\_HocKi\_NamHoc <pi> Integer

<M>

Tiet TG\_BatDau

Integer Date

Thuoc

HocKi NamHoc TG\_BatDau TG\_KetThuc TuanHoc

Integer Variable binary Date

Date

Variable multibyte

TG\_KetThuc Identifier\_1 <pi>

Date

tiet trong ngay

0,n

Identifier\_1 <pi>

ThoiGianHoc ID\_ThoiGianHoc <pi> Integer

<M>

MSSV

Ten\_SinhVien NgaySinh GioiTinh DiaChi

SDT

SINHVIEN

<pi> Characters (10) Variable multibyte Date

Boolean

Variable multibyte Variable multibyte (11)

<M>

0,n

0,n

KHHT

CaiThien Boolean

0,n

GiangDay

giang day

Is\_Schedule

Identifier\_1 <pi>

THU

Boolean

thu trong tuan

Password

Variable multibyte

SoNhom\_GD Integer

ID\_Thu <pi> Integer

<M>

Username

Variable multibyte

thu

Thu

Variable multibyte

Email Identifier\_1 <pi>

Variable multibyte

0,n

HOCPHAN

0,n

0,n

CANBO

Identifier\_1 <pi>

0,n

NHOM\_HOCPHAN

ID\_Nhom\_HocPhan <pi> Integer <M>

MaHocPhan TenHocPhan SoTinChi SoTietLT SoTietTH SoTietMoiTuan SoBuoiMoiTuan

SiSo\_QuiDinh

<pi> Variable multibyte (10) <M> Variable multibyte

Integer Integer Integer Integer Integer

Integer

ID\_CanBo Ten\_CanBo NgaySinh GioiTinh DiaChi SDT

Email Username

<pi> Integer

Variable multibyte Date

Boolean

Variable multibyte Variable multibyte (11) Variable multibyte Variable multibyte

<M>

0,n

LichBan\_CanBo LyDo Variable multibyte

Tong\_TietHoc

Tong\_BuoiHoc SiSo

Identifier\_1 <pi>

Integer

Integer Integer

BaoGom

TienQuyet HocKi\_Hoc Is\_Schedule

Identifier\_1 <pi>

Variable multibyte Variable multibyte Boolean

Password Identifier\_1 <pi>

Variable multibyte

0,n

ID\_Loai\_DonVi Ten\_Loai\_DonVi MoTa

Identifier\_1 <pi>

LOAI\_DONVI

<pi> Integer

Variable multibyte Variable multibyte

<M>

CoLoai\_DonVi

Ma\_DonVi Ten\_DonVi DiaChi Email

SDT

GioiThieu

PhuTrach

DONVI

<pi> Variable characters (10) <M> Variable multibyte

Variable multibyte Variable multibyte Variable multibyte (11) Variable multibyte

0,n

TG\_BatDau Date TG\_KetThuc Date

CongTac

CHUCVU

ID\_ChucVu <pi> Integer

<M>

0,n

Ten\_ChucVu Variable multibyte

MoTa Variable multibyte

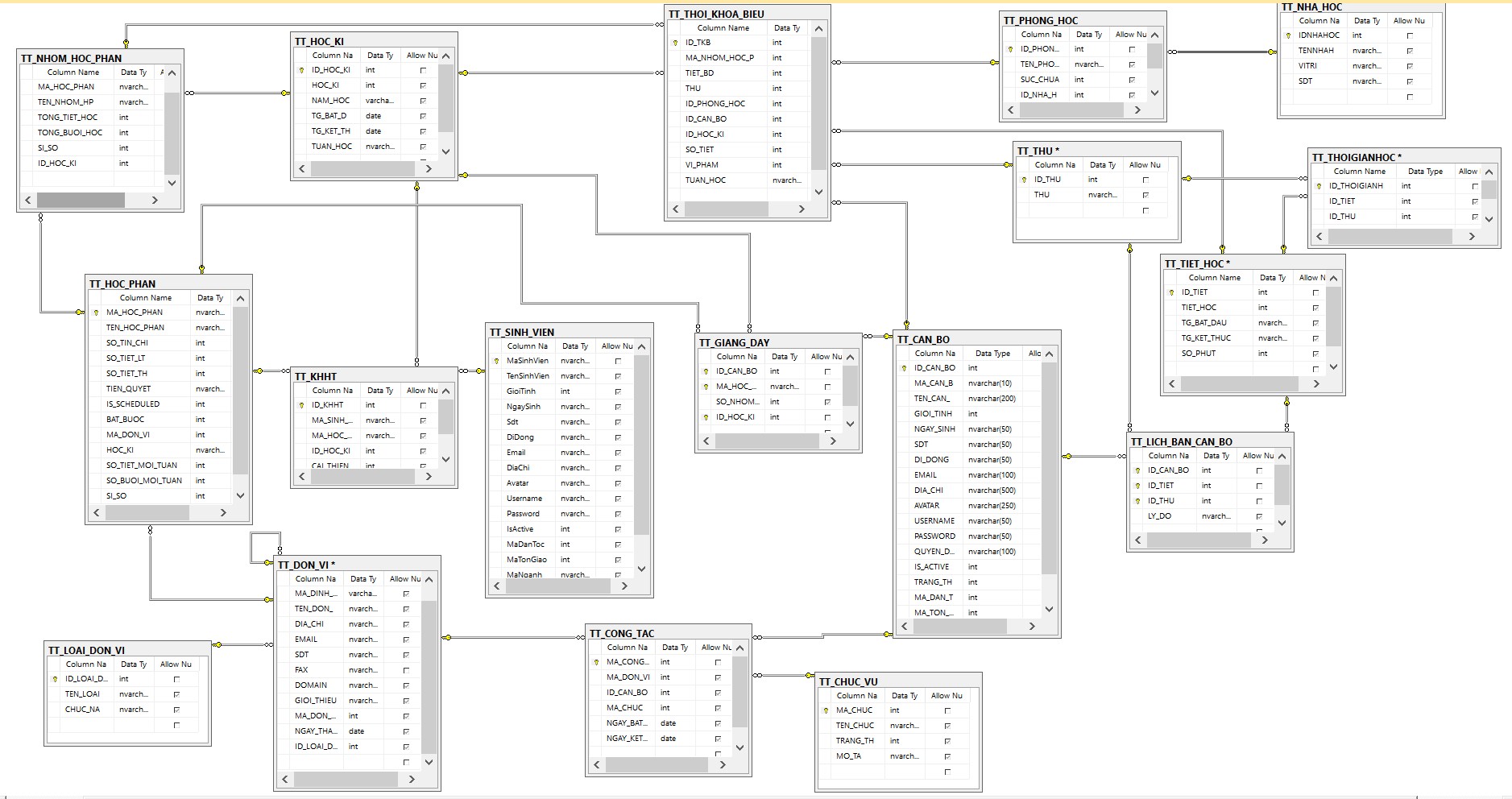
Identifier\_1 <pi>

NgayThanhLap Identifier\_1 <pi>

Date

DonViPhuTrach

**Hình 3.12 Mô hình dữ liệu ER của hệ thống**

**Hình 3.13 Mô hình cơ sở dữ liệu SQL**

* 1. **THIẾT KẾ MÔ HÌNH TUẦN TỰ SẮP XẾP THỜI KHÓA BIỂU**

SapTKB

GiaoDien\_SapTKB

DieuKhien\_SapTKB

XuLy\_SapTKB

Database

Admin

Gọi chức năng load dữ liệu đầu vào

Gửi yêu cầu load dữ liệu đầu vào

Gọi hàm LoadData()

Gọi hàm AccessData()

Hiển thị kết quả load dữ liệu thành công

Gửi danh sách dữ liệu đầu vao

Danh sách các dữ liệu đầu vào

Tập các dữ liệu đầu vào

Gọi chức năng thiết lập thông số GA và ràng buộc

Gởi thông số và yêu cầu thiết lập thông số

Gọi hàm SetGA()

SetGA()

Kết quả thiết lập

Gửi kết quả thiết lập

Hiển thị thông báo kết quả thiết lập

Gọi chức năng sắp xếp TKB

Gởi yêu cầu bắt đầu sắp xếp TKB

Gọi hàm StartGA()

InitPopulation()

loop

[Fitness >= F || g > Cg]

Fitness = evalPopulation()

crossoverPopulation()

mutatePopulation()

g+=1;

Hiển thị danh sách các phương án TKB tốt nhất

Danh sách các phương án TKB

Tập các phương án TKB

Gọi chức năng xem phương án TKB tốt nhất

ởi yêu cầu xem phương án TKB tốt nhất (index = 0)

g

gọi hàm getTimetable(index)

Hiển thị TKB tốt nhất

Phương án TKB tốt nhất

Timetable

getTimetable(index)



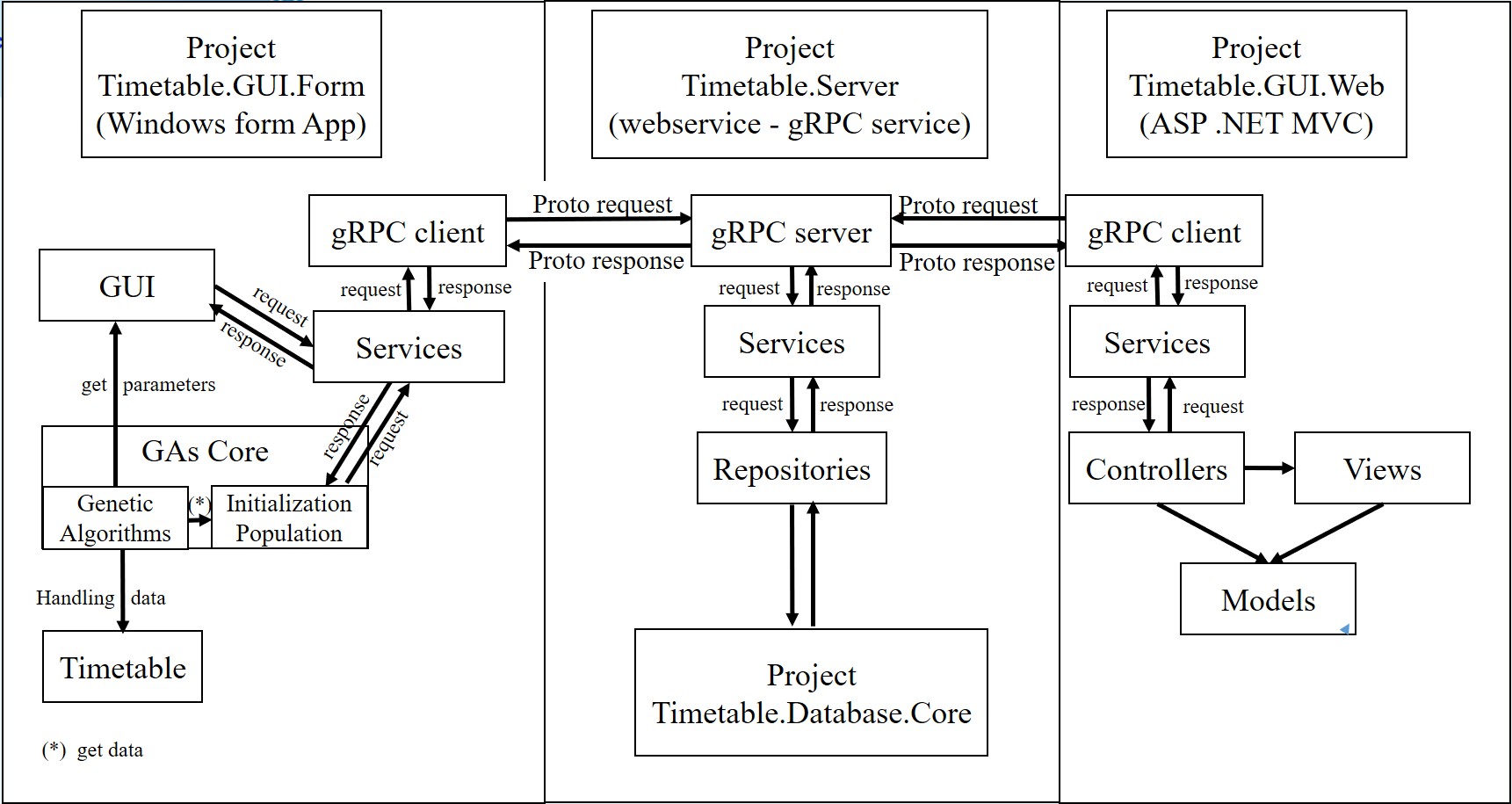
**Hình 3.14 Sơ đồ tuần tự chức năng sắp xếp thời khóa biểu**

* 1. **CÀI ĐẶT CHƯƠNG TRÌNH**
     1. **Xây dựng hệ thống**

Dựa trên tích, thiết kế hệ thống như đã trình bày ở trên. Tôi đề xuất xây dựng hệ thống hỗ trợ quản lí và xếp TKB tự động bằng thuật toán di truyền nhằm tối ưu công sức cũng như thời gian dành cho việc lập TKB.

Hệ thống được xây dựng bằng ngôn ngữ lập trình C#, trên nền tảng .NET Core Framework, hệ quản trị CSDL là SQL server. Hệ thống bao gồm 3 module chính:

1. Module Database: Sử dụng công nghệ gRPC service làm web service quản lý việc truy xuất cơ sở dữ liệu từ database, nhận – thực hiện – phản hồi các yêu cầu từ 2 module còn lại.
2. Module web-app: Sử dụng công nghệ ASP .NET Core MVC. Module hỗ trợ sinh viên quản lí đăng kí KHHT, hỗ trợ cán bộ quản lí cập nhật lịch giảng dạy. xem danh sách TKB các học phần.
3. Module app xếp TKB tự động: Sử dụng công nghệ Window Form Application kết hợp với giải thuật di truyền để quản lí TKB, xếp TKB tự động. Hổ trợ quản lí và chia nhóm học phần tự động, quản lí các dữ liệu đầu vào để phục vụ cho việc xếp TKB.



**Hình 3.15 Sơ đồ tổng thể dự án của đề tài**

* + 1. **Giao diện chức năng**
       1. **Giao diện phần mềm hỗ trợ xếp thời khóa biểu tự động**

Phần mềm hỗ trợ xếp TKB tự động chỉ dành cho cán bộ quản lý học vụ.

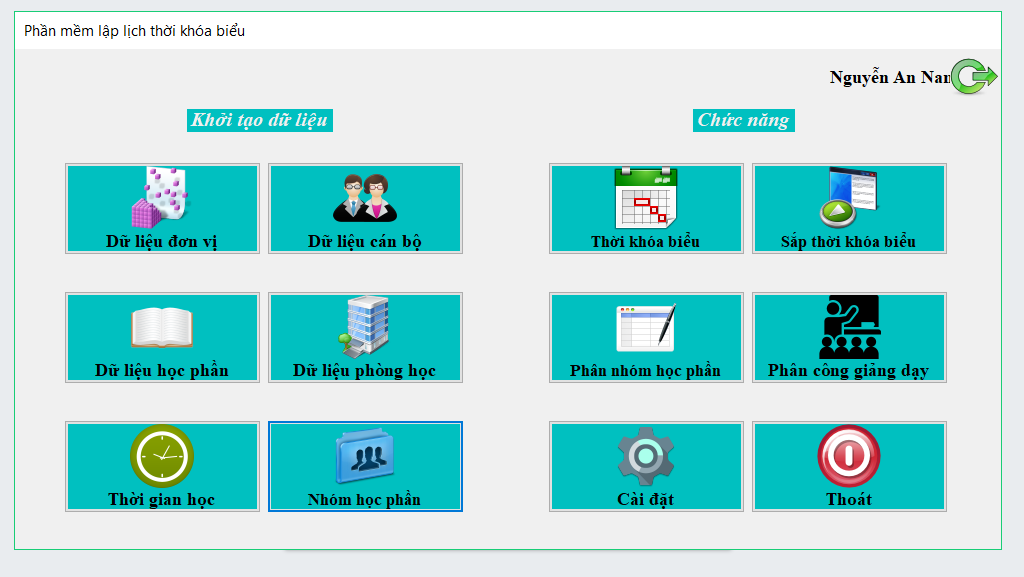
* ***Đăng nhập***

Chỉ người dùng có quyền quản lý hoặc quản lý học vụ mới có thể đăng nhập vào hệ thống.



**Hình 3.16 Giao diện đăng nhập phần mềm hỗ trợ xếp thời khóa biểu**

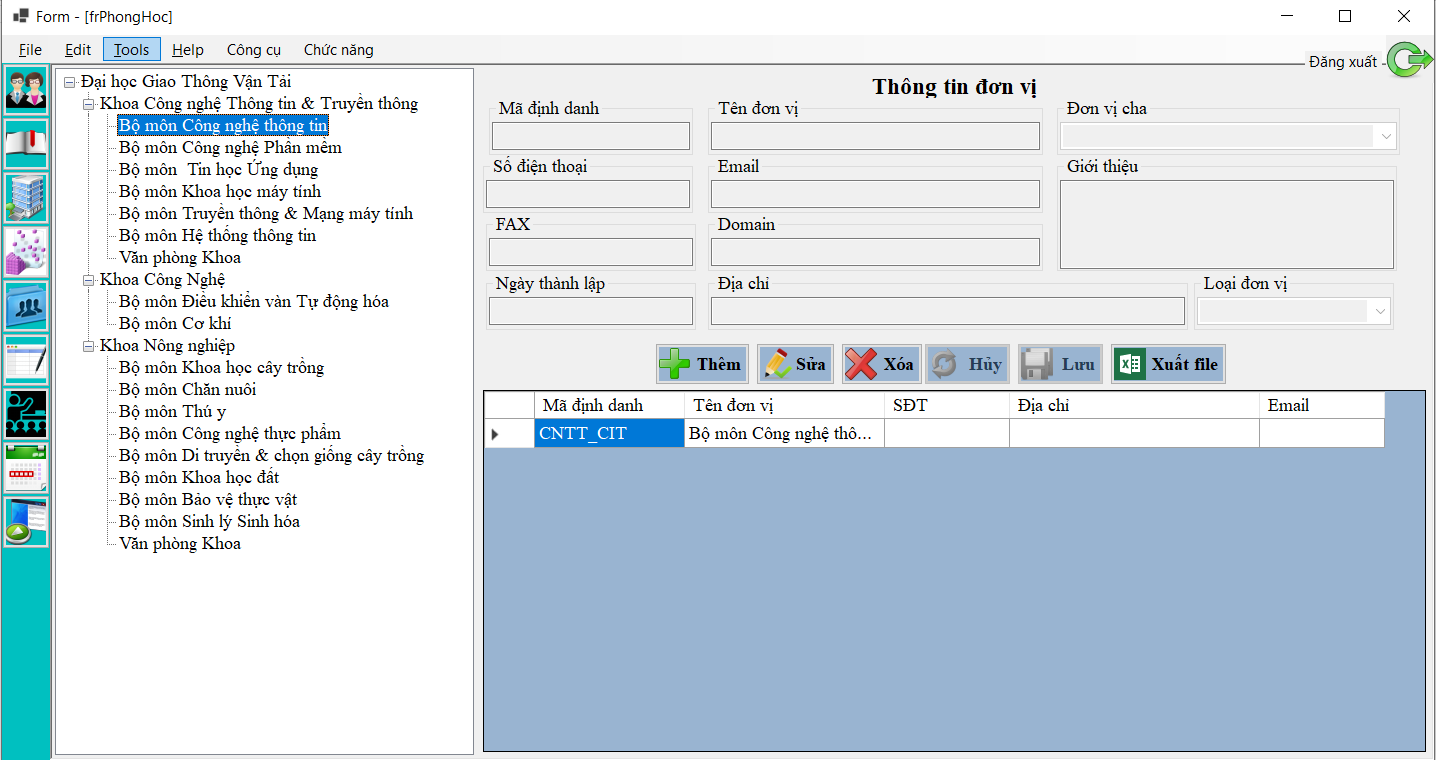
* ***Trang chủ***



**Hình 3.17 Giao diện trang chính phần mềm hỗ trợ xếp thời khóa biểu**

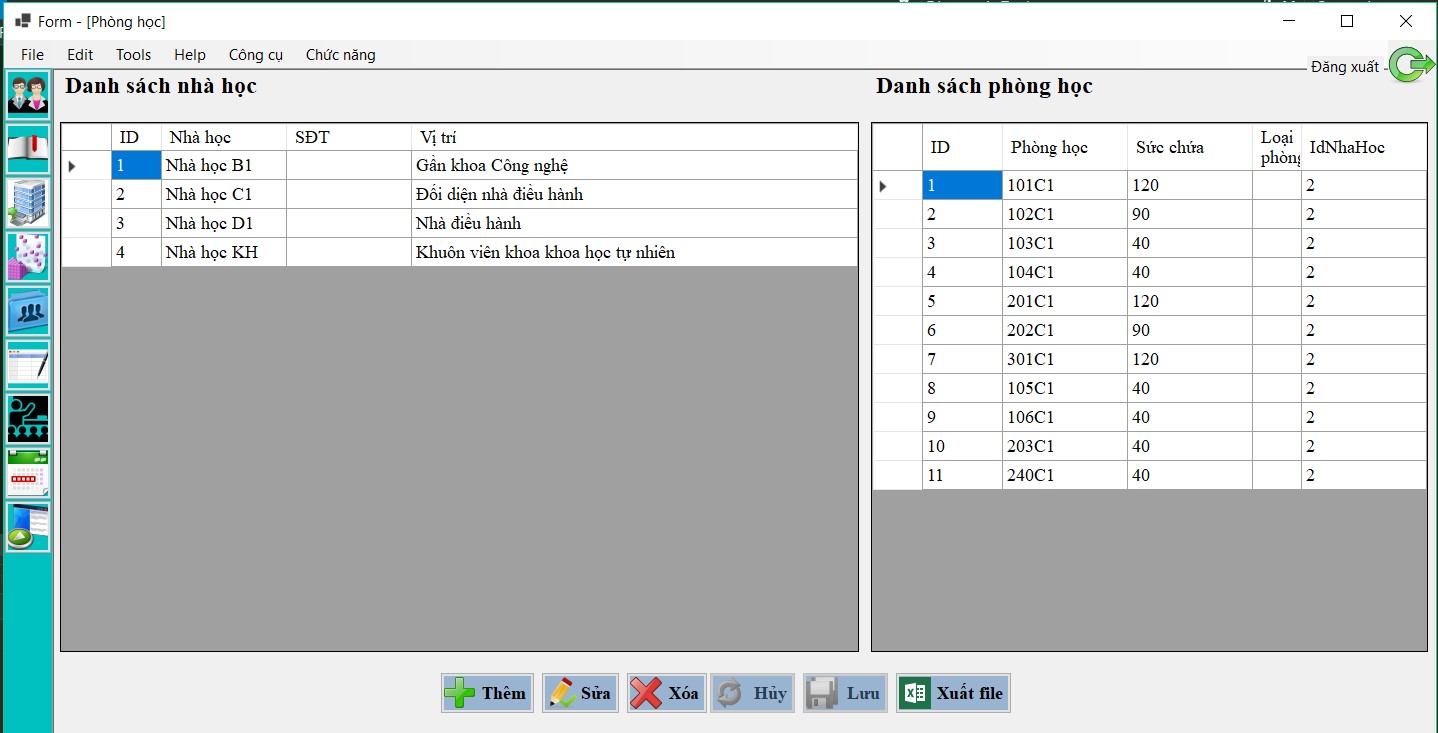
***Quản lý đơn vị***

Giao diện quản lý đơn vị bao gồm các chức năng thêm, sửa, xóa đơn vị, xuất danh sách đơn vị sang excel. Hiển thị danh sách đơn vị theo dạng cây giúp dễ dàng nhận biết đơn vị quản lý hoặc danh sách đơn vị trực thuộc của một đơn vị.



**Hình 3.18 Giao diện quản lý đơn vị**

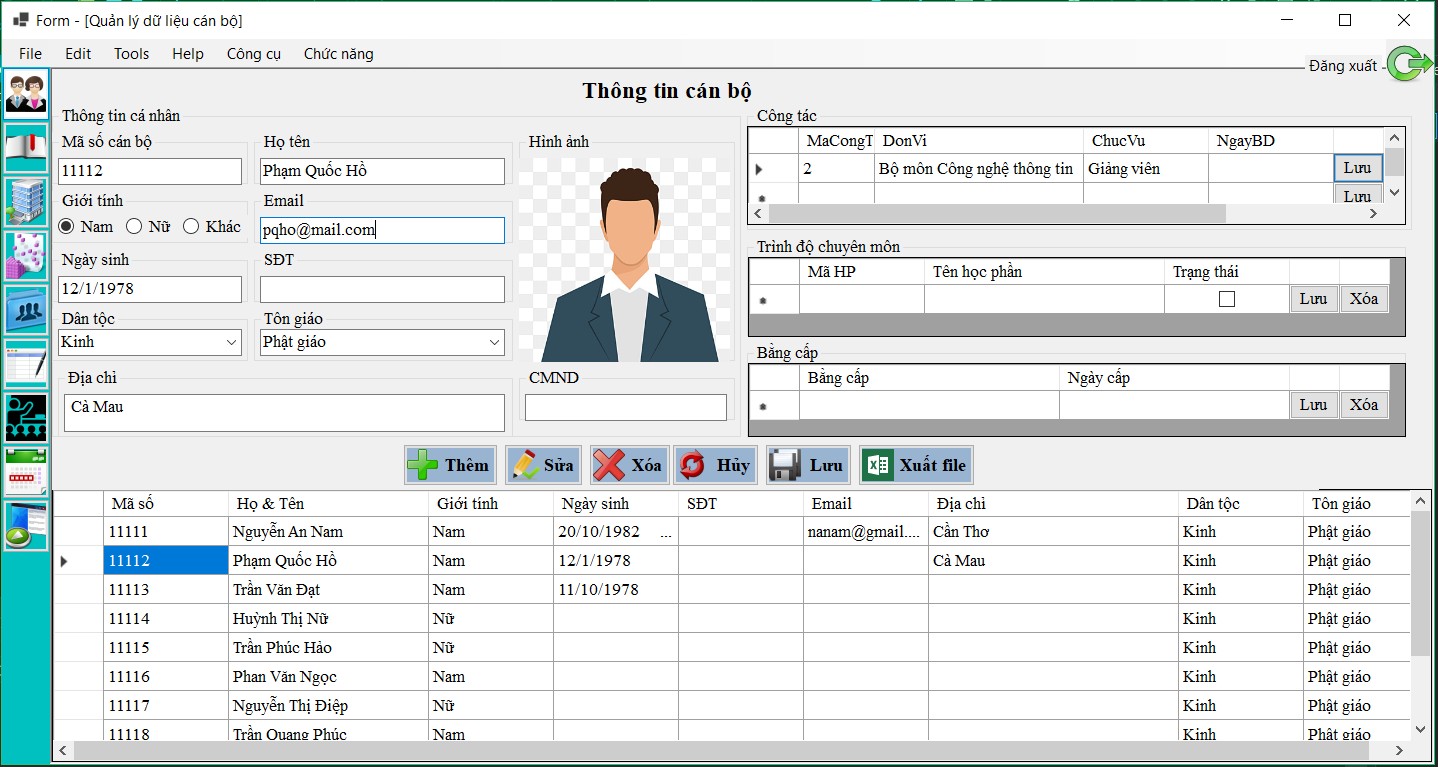
* ***Quản lý phòng học***

Giao diện quản lý phòng học bao gồm giao diện hiển thị danh sách nhà học và hiển thị danh sách phòng học thuộc nhà học. Các chức năng: Thêm, sửa, xóa phòng học, xuất danh sách phòng học sang excel.

**Hình 3.19 Giao diện quản lý phòng học**

* ***Quản lý cán bộ***

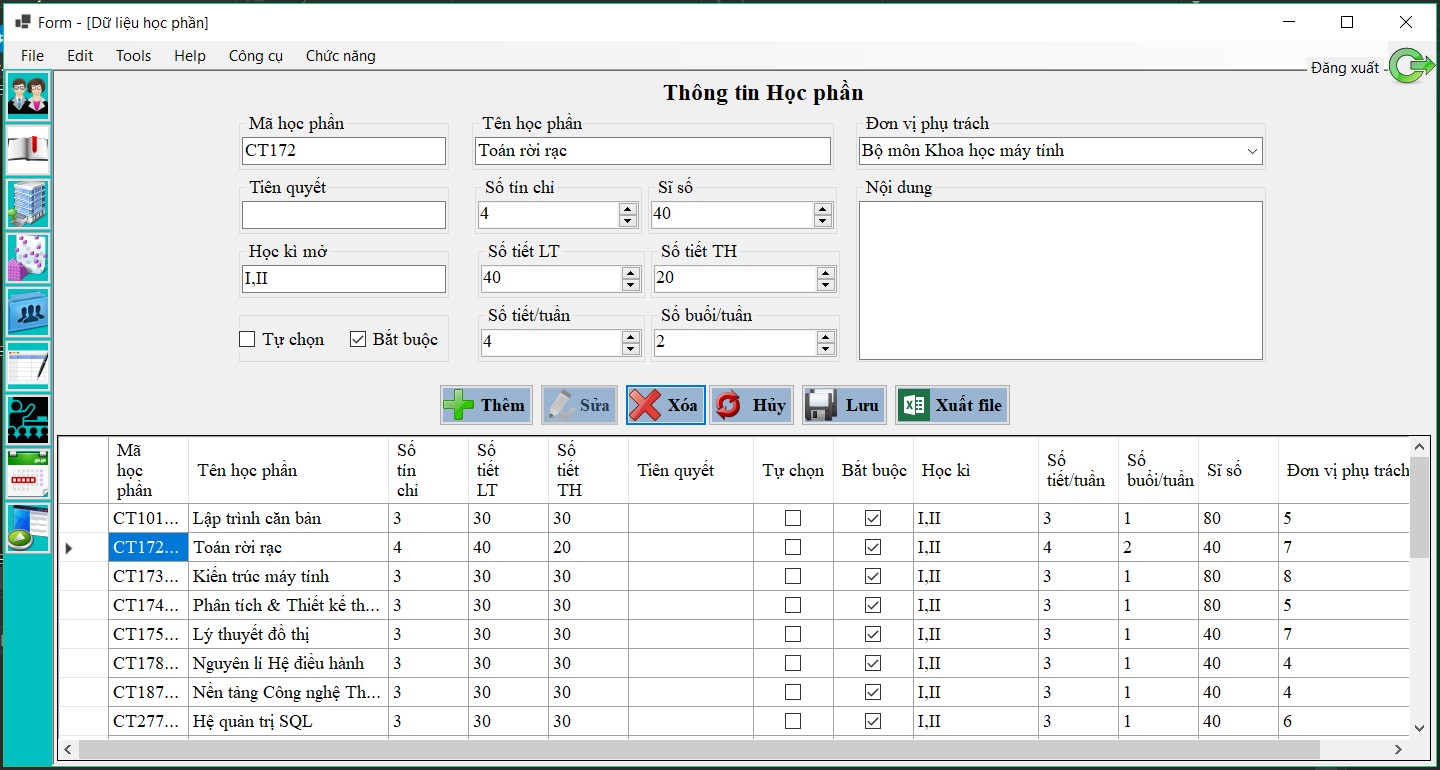
Giao diện quản lý cán bộ hiển thị danh sách cán bộ và thông tin chi tiết cán bộ: thông tin cá nhân, danh sách đơn vị công tác và chức vụ, trình độ chuyên môn và bằng cấp. Các chức năng: Thêm, sửa, xóa cán bộ và xuất danh sách sang file excel.



**Hình 3.20 Giao diện quản lý cán bộ**

* ***Quản lý học phần***

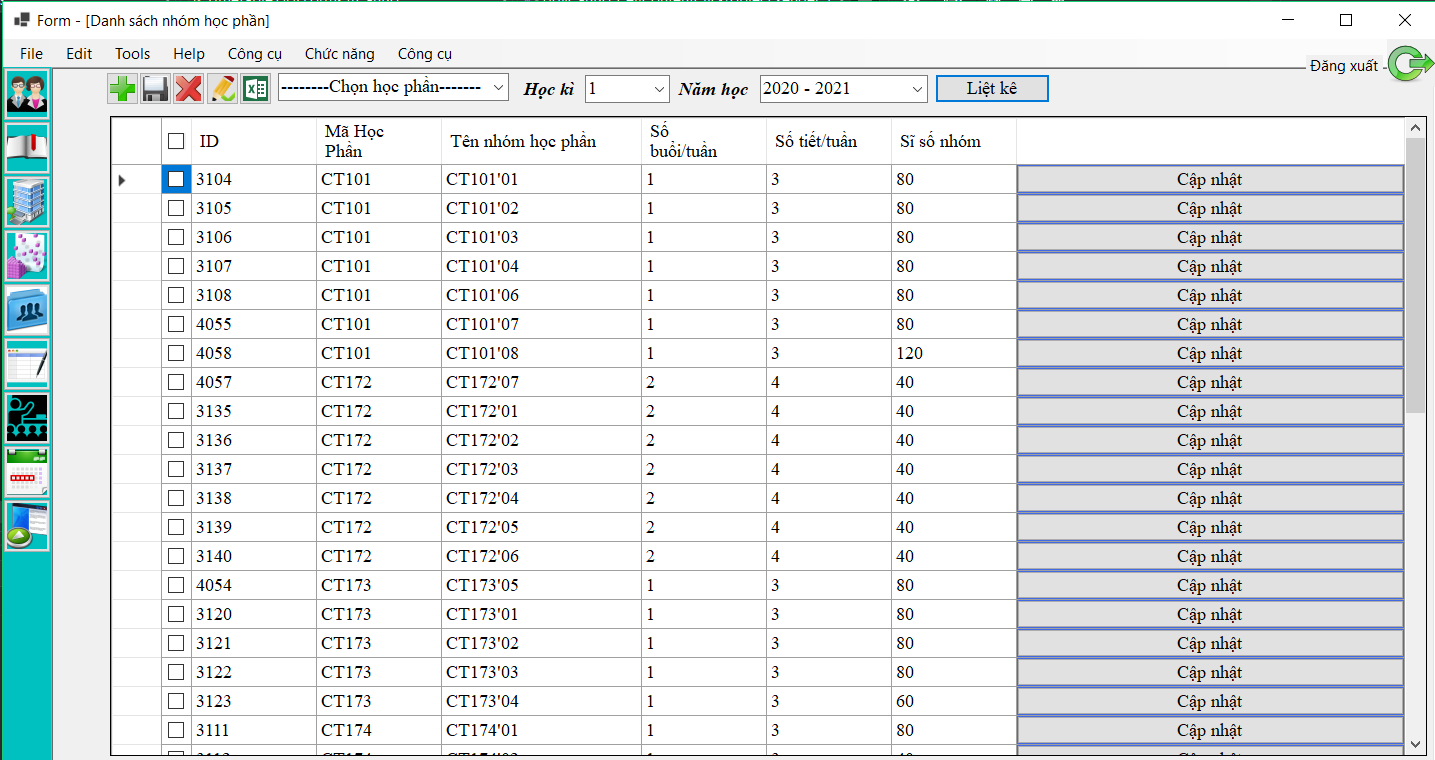
Giao diện quản lý học phần hiển thị danh sách tất cả học phần và thông tin chi tiết học phần. Các chức năng: Thêm, sửa, xóa học phần và xuất danh sách học phần sang file excel.



**Hình 3.21 Giao diện quản lý học phần**

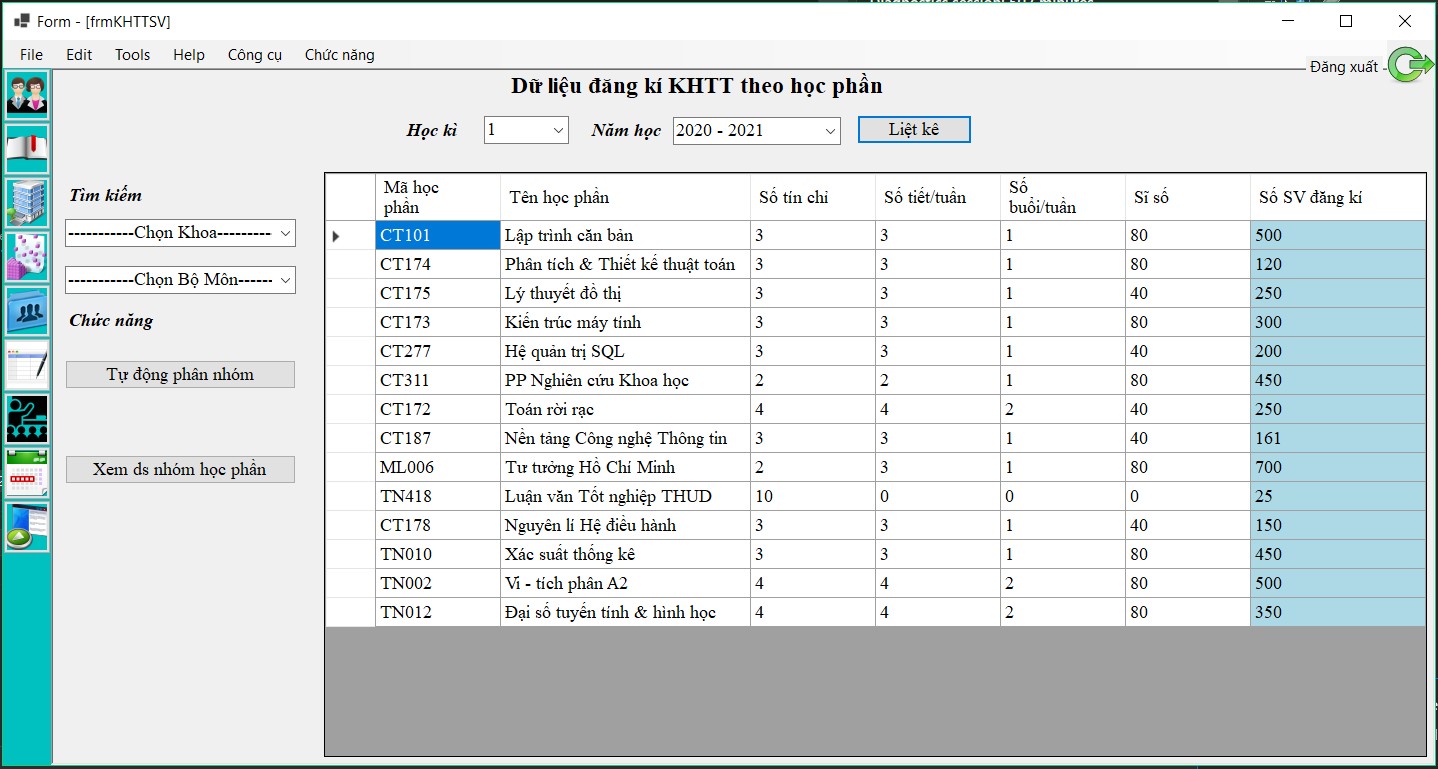
* ***Quản lý nhóm học phần***

Giao diện quản lý nhóm học phần hiển thị danh sách nhóm học phần đã chia theo học kì, năm học. Các chức năng: Thêm, sửa, xóa nhóm học phần và xuất danh sách nhóm học phần theo học kì sang file excel, tìm danh sách nhóm theo học phần.



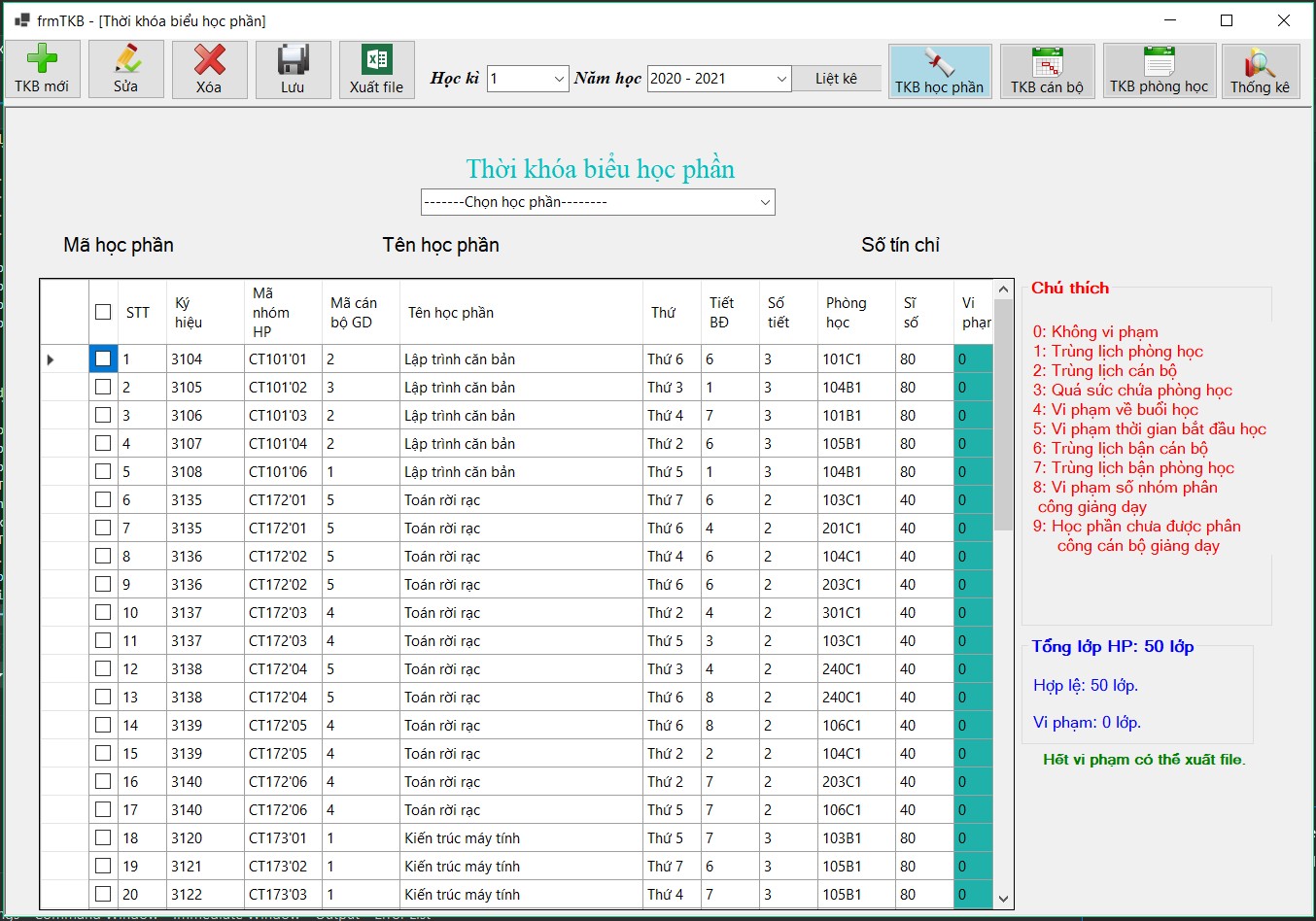
**Hình 3.22 Giao diện quản lý nhóm học phần theo học kì, năm học**

* ***Tổng hợp KHHT và phân nhóm tự động***

Giao diện hiển thị danh sách tổng hợp KHHT của sinh viên đã đăng kí theo học kì. Dựa vào các thuộc tính của học phần: Số tiết, số buổi, sĩ số và số lượng sinh viên đăng kí - hệ thống sẽ tự động chia nhóm cho học kì đó.

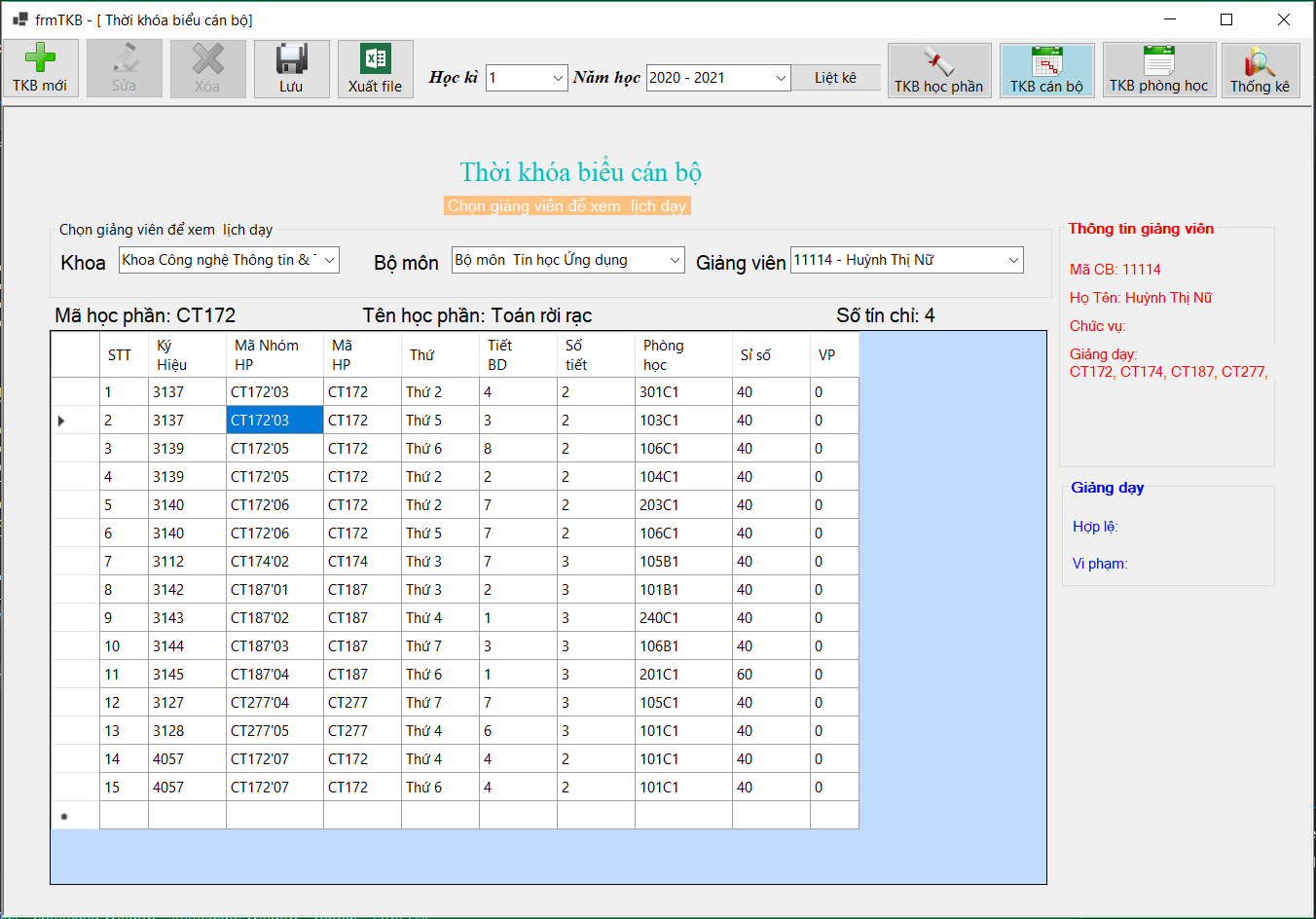
**Hình 3.23 Giao diện tổng hợp KHHT và phân nhóm học phần tự động**

* ***Quản lý TKB***

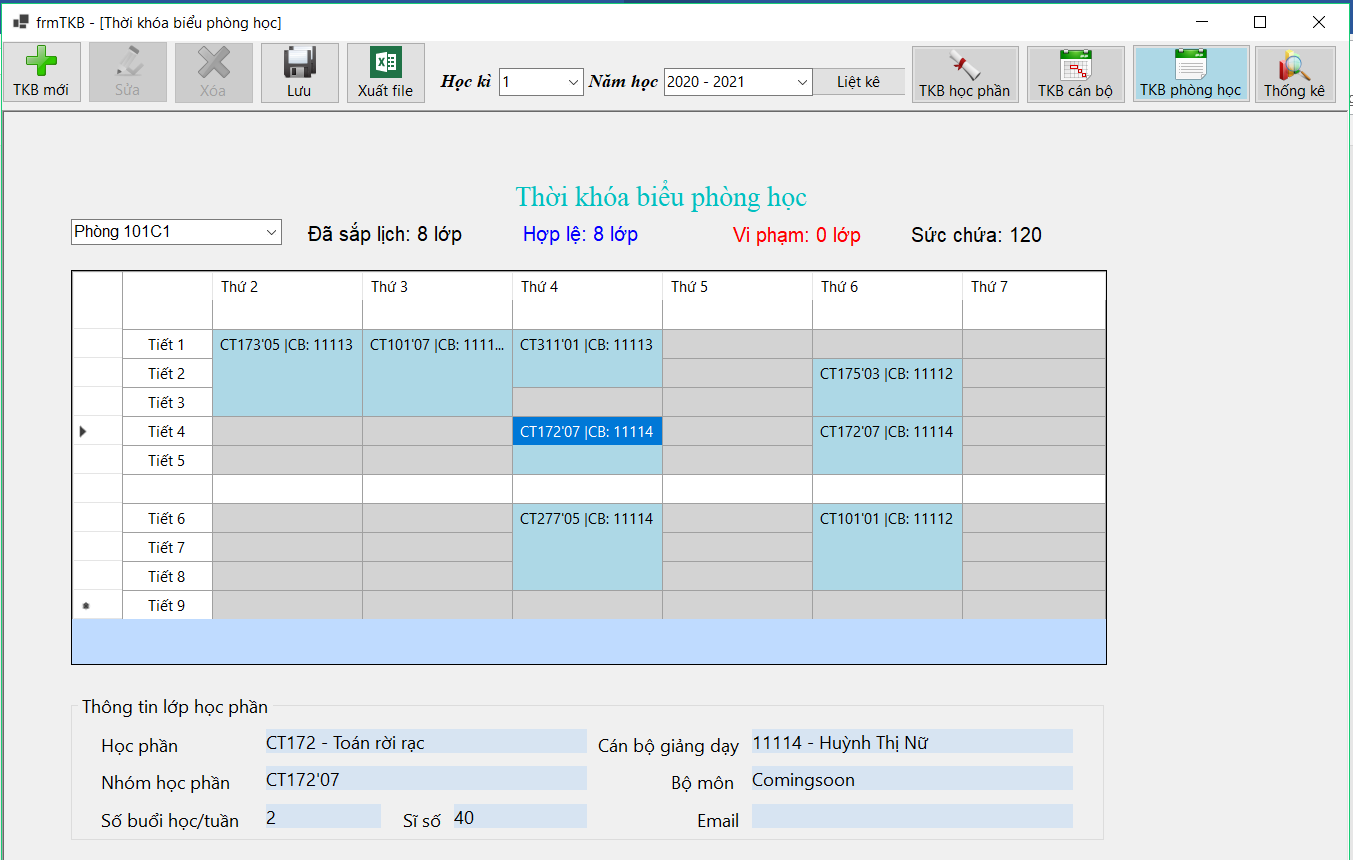
Giao diện quản lý TKB hiển thị TKB theo học kì, năm học. Các chức năng: hiển thị TKB theo học phần, hiển thị TKB của cán bộ, hiển thị TKB của phòng học, chỉnh sửa TKB thủ công (thêm mới, sửa, xóa, xuất file excel).

**Hình 3.24 Giao diện hiển thị TKB theo học phần**

Tại giao diện hiển thị TKB theo học phần, người dùng có thể xem được TKB danh sách toàn bộ các nhóm học phần hoặc theo từng học phần. Ngoài ra, tại giao diện này người dùng có thể nắm được số nhóm học phần được sắp lịch có vi phạm ràng các ràng buộc hay không.



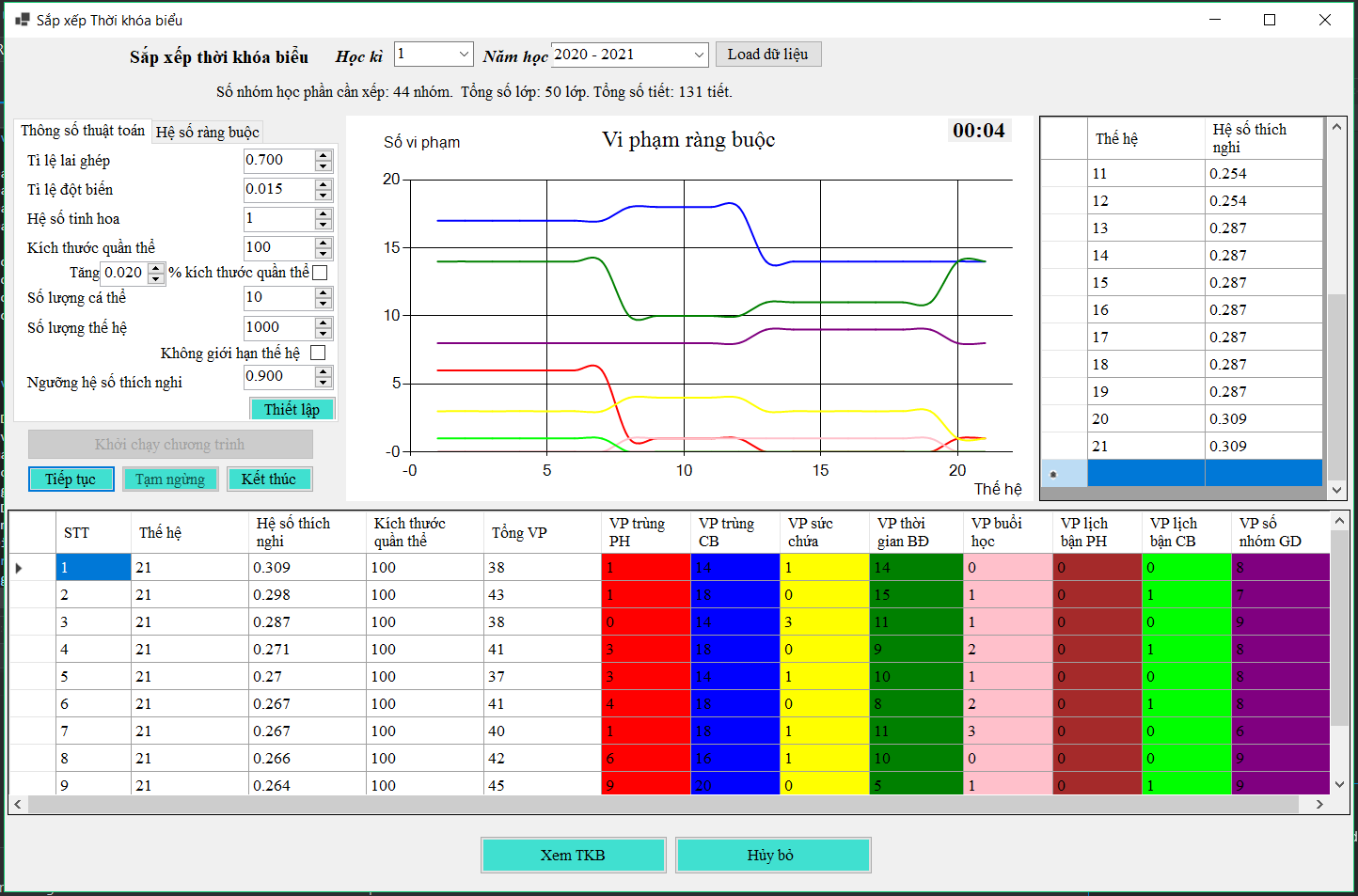
**Hình 3.25 Giao diện hiển thị TKB của cán bộ**



**Hình 3.26 Giao diện hiển thị TKB của phòng học**

* ***Trang sắp xếp TKB tự động***

Đây là chức năng quan trọng nhất của chương trình. Giao diện sắp xếp TKB theo học kì, năm học. Các chức năng: Load các dữ liệu đầu vào liên quan đến việc sắp TKB, cài đặt các thông số cho giải thuật, cài đặt các hệ số ràng buộc vi phạm.

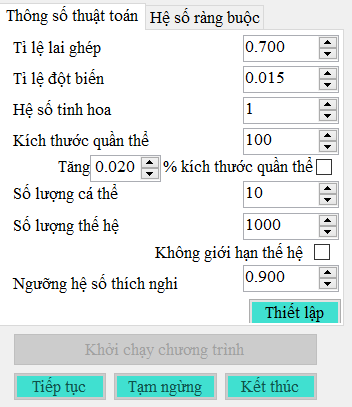


**Hình 3.27 Giao diện sắp xếp TKB tự động**

* *Thiết lập thông số cho thuật toán GAs*

Tốc độ xử lý, điều kiện dừng của chương trình và việc tìm ra lời giải của bài toán phụ thuộc vào việc thiết lập các thông số cho thuật toán GAs. Thông số thuật toán gồm 2 nhóm chính:

* Nhóm tham số đầu vào của thuật toán: tỉ lệ lai ghép, tỉ lệ đột biến, hệ số tinh hoa (số cá thể có tỉ lệ thích nghi cao nhất được giữ lại qua mỗi thế hệ), kích thước quẩn thể.
* Nhóm tham số điều kiện dừng quá trình tiến hóa: số lượng thế hệ, ngưỡng hệ số thích nghi.

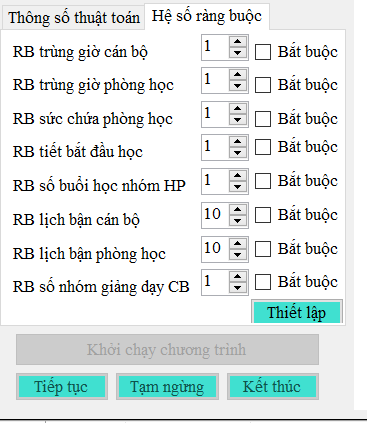


**Hình 3.28 Giao diện thiết lập thông số cho thuật toán**

* *Thiết lập hệ số ràng buộc*

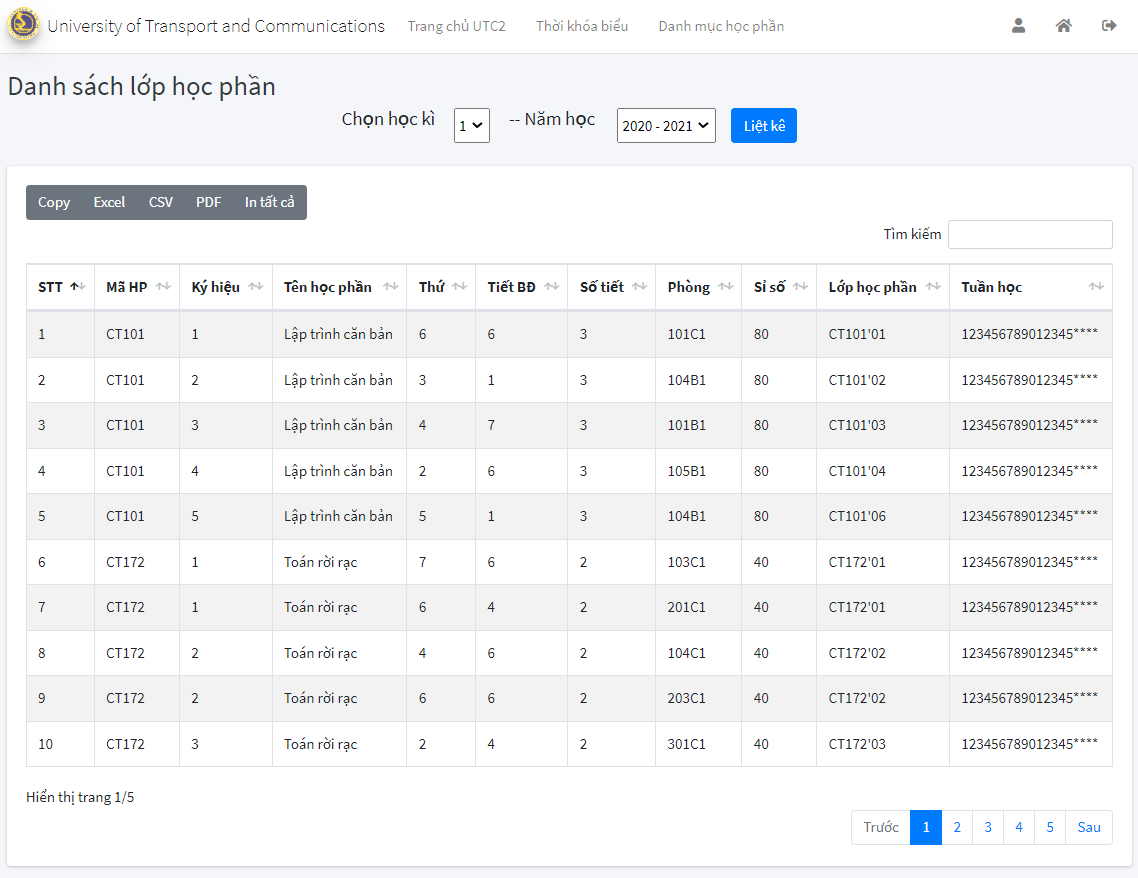
Hệ số ràng buộc qui định về mức độ quan trọng của ràng buộc đó trong bài toán, người dùng có thể ước lượng được số vi phạm mỗi ràng buộc có thể chấp nhận thông qua tính toán tỉ lệ thích nghi dựa trên mỗi ràng buộc. Theo đó

Với: x – số vi phạm ràng buộc, R - hệ số ràng buộc. F - ngưỡng hệ số thích nghi chấp nhận được.



**Hình 3.29 Giao diện thiết lập hệ số ràng buộc**

* + - 1. **Giao diện website hỗ trợ sinh viên và giảng viên**
* ***Trang danh mục học phần (TKB các học phần)***



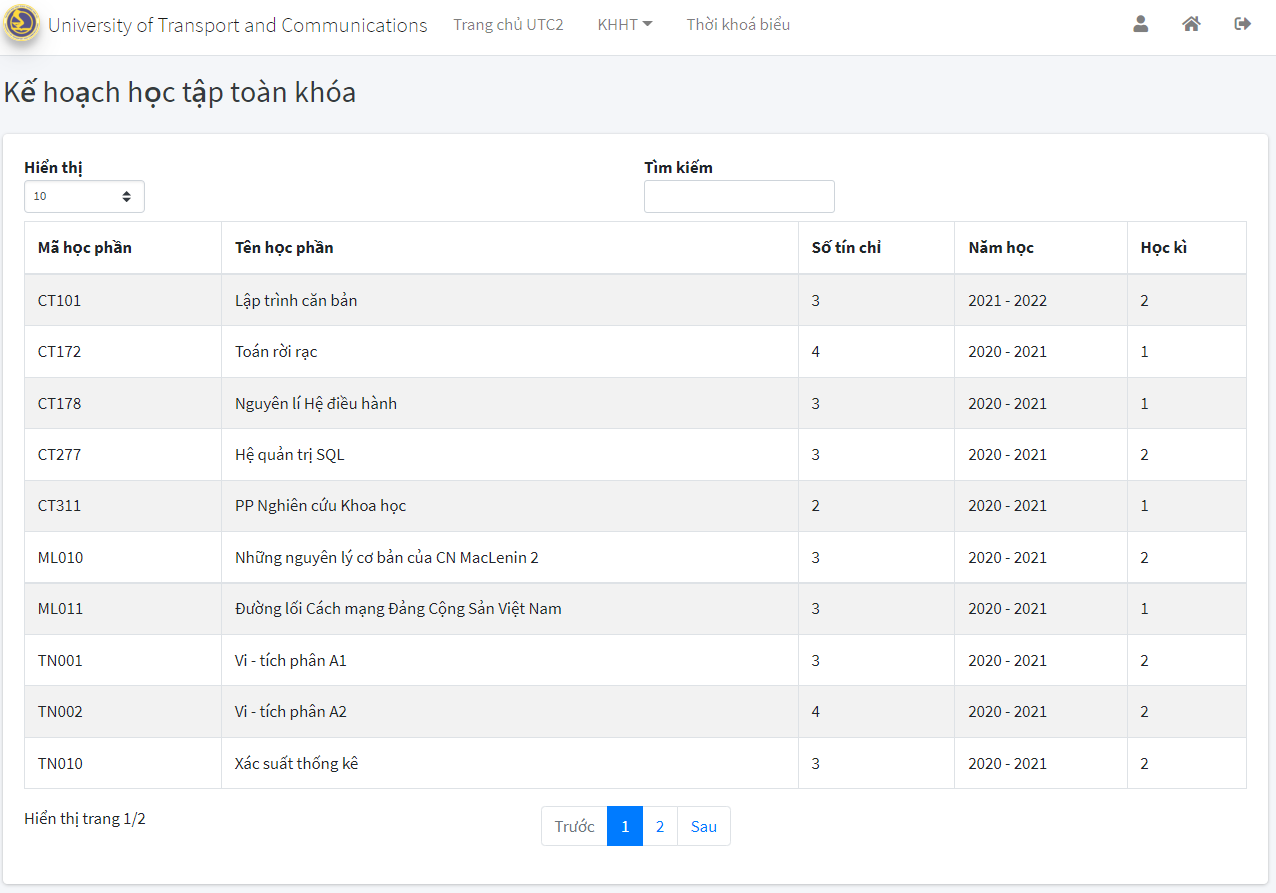
**Hình 3.30 Trang hiển thị danh mục học phần**

Giao diện hiển thị danh sách TKB học phần theo học kì, năm học. Các chức năng: Tìm kiếm TKB theo từng học phần, xuất danh sách TKB sang các file excel, csv, pdf hoặc in trực tiếp. Đối tượng truy cập: sinh viên, cán bộ.

* ***Trang xem KHHT toàn khóa***

Đối tượng truy cập: Sinh viên.

Sinh viên xem KHHT toàn khóa của cá nhân tại đây.

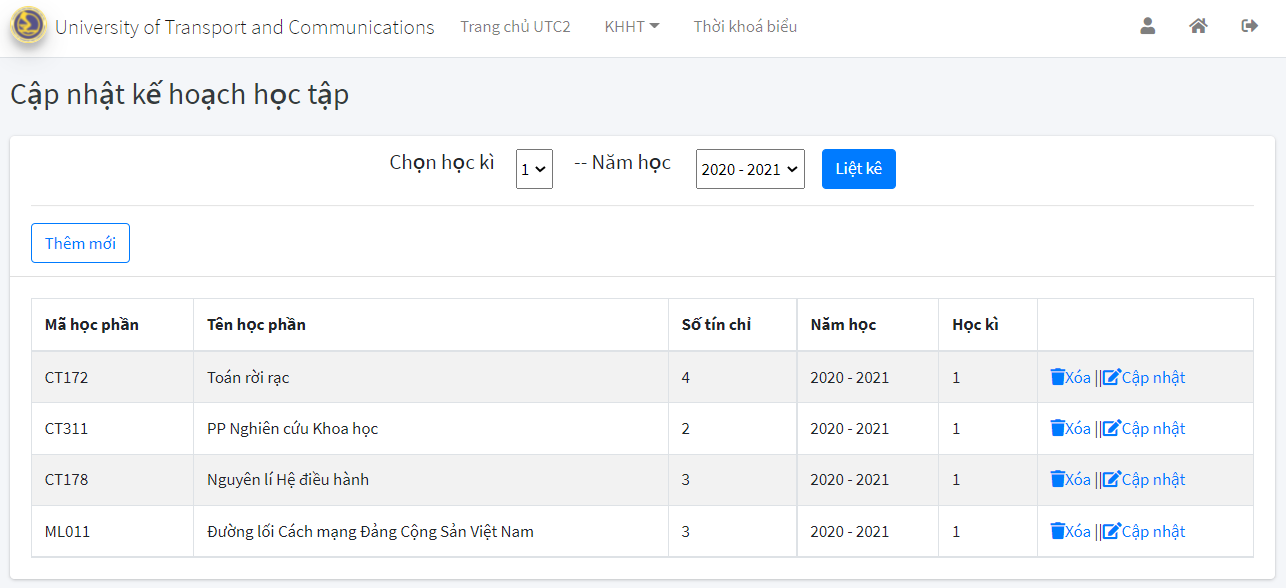


**Hình 3.31 Trang hiển thị kế hoạch học tập toàn khóa của sinh viên**

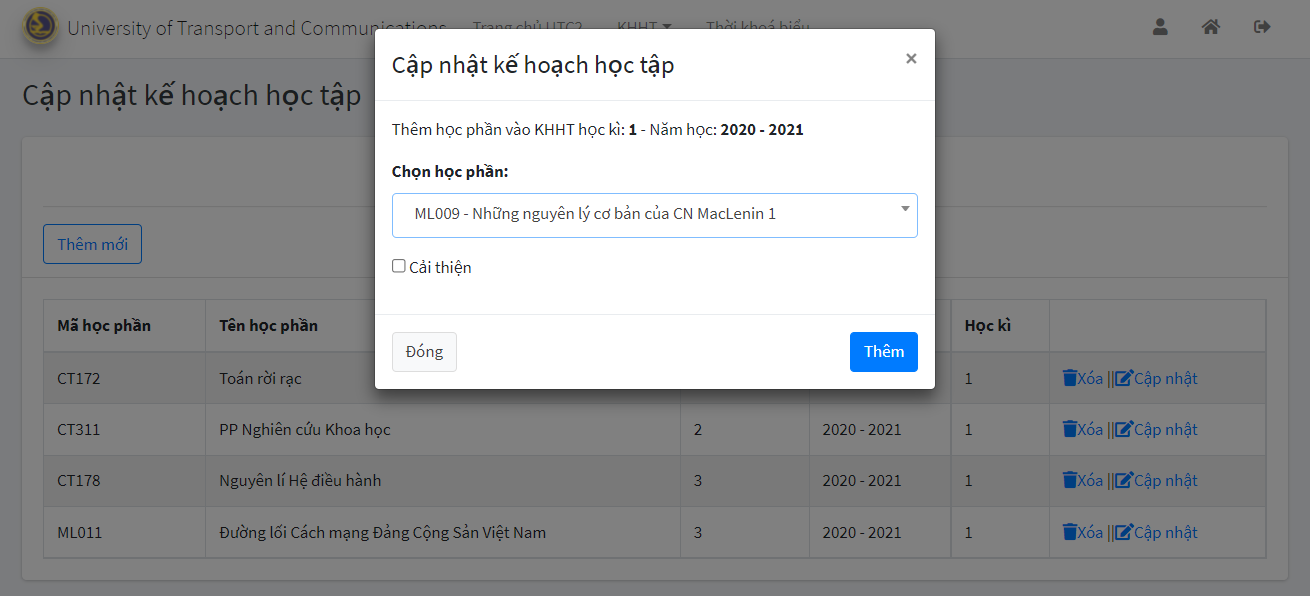
* ***Trang cập nhật KHHT***

Đối tượng truy cập: Sinh viên.

Sinh viên cập nhật KHHT cá nhân theo năm học – học kì. Tại đây sinh viên có thể thêm, xóa học phần KHHT hoặc cập nhật lại học kì cho KHHT đã có.



**Hình 3.32 Trang cập nhật kế hoạch học tập của sinh viên**

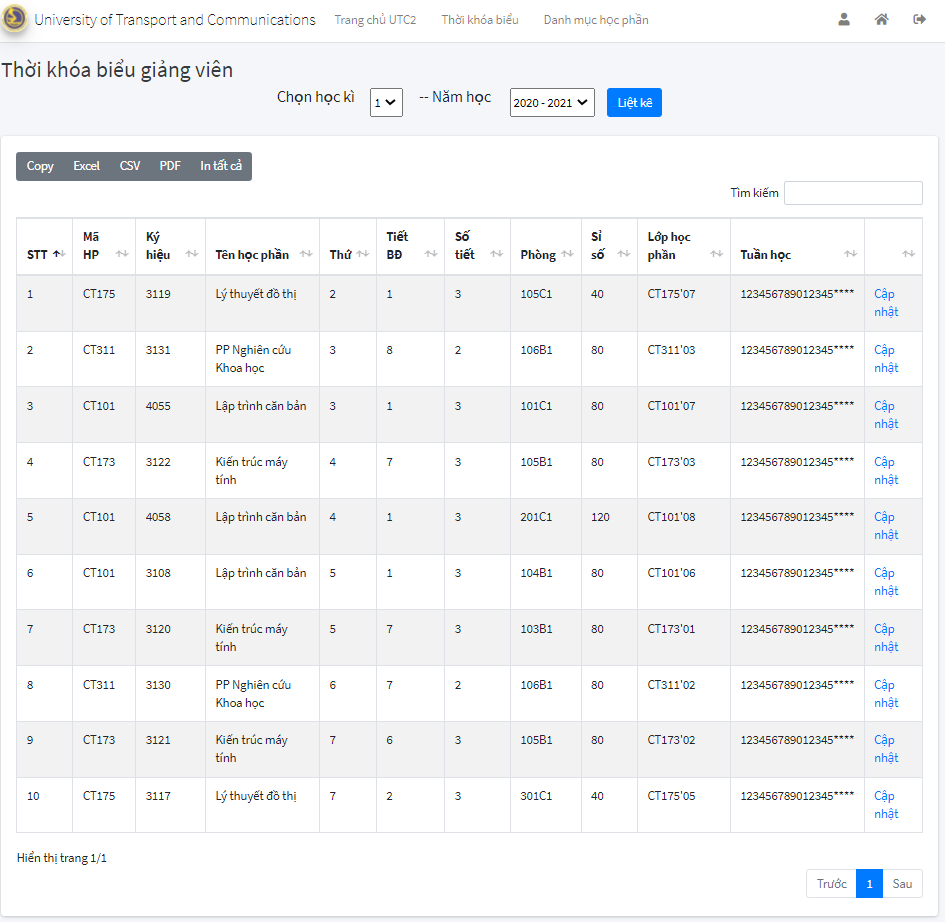


**Hình 3.33 Giao diện thêm kế hoạch học tập của sinh viên**

* ***Trang quản lý TKB giảng viên***

Đối tượng truy cập: Cán bộ giảng viên

Hiển thị TKB của cán bộ theo học kì, năm học. Các chức năng: cập nhật lịch giảng dạy (trong thời gian cho phép), in và xuất TKB, tìm kiếm.



**Hình 3.34 Trang quản lý thời khóa biểu của cán bộ**

# CHƯƠNG 4: KIỂM THỬ VÀ ĐÁNH GIÁ

## KIỂM THỬ HỆ THỐNG

* + 1. **Cấu hình máy tính chuẩn bị cho các kịch bản thử nghiệm**

Các kịch bản thử nghiệm đều chạy cùng trên một máy tính có cấu hình như sau: CPU: Intel Core i3 6100U, 2.3 GHz, RAM: 8GB, tốc độ bus 2133MHz, HĐH: Windows 10

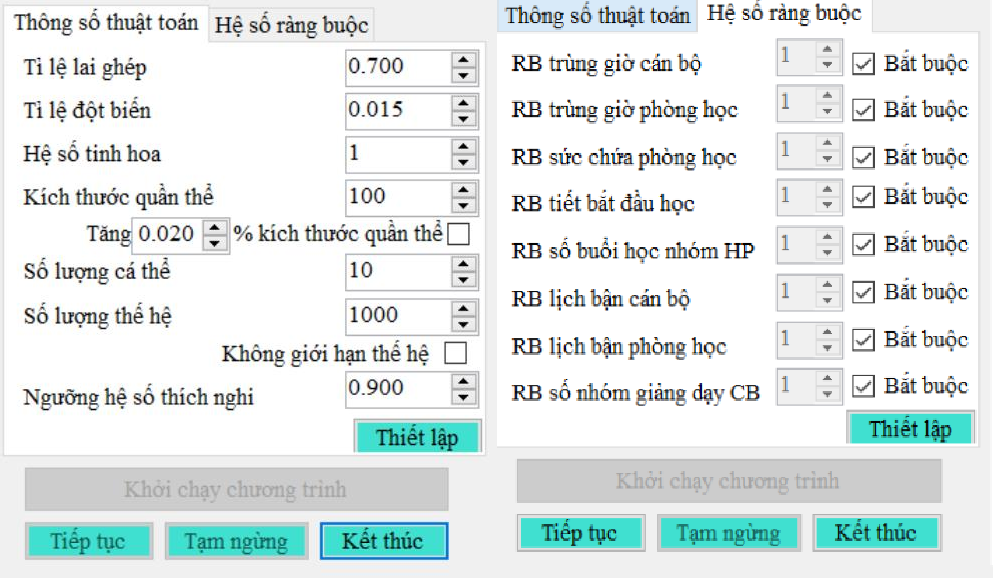


**Hình 4.1 Cấu hình máy tính cho các kịch bản thử nghiệm**

* + 1. **Kịch bản thử nghiệm 1**
       1. **Mục tiêu và dữ liệu đầu vào**

Kịch bản 1 thử nghiệm với tập dữ liệu đơn giản. Số nhóm cần xếp: 14 nhóm, tổng số buổi học: 17 buổi, tổng số tiết học: 45 tiết, 5 giảng viên được phân công giảng dạy, 10 phòng học. Tiến hành sắp lịch cho 17 buổi học của 14 nhóm học phần.

* + - 1. **Các thông số**



**Hình 4.2 Thông số chương trình cho kịch bản thử nghiệm 1**

* + - 1. **Kết quả**



**Hình 4.3 Kết quả kịch bản thử nghiệm 1**

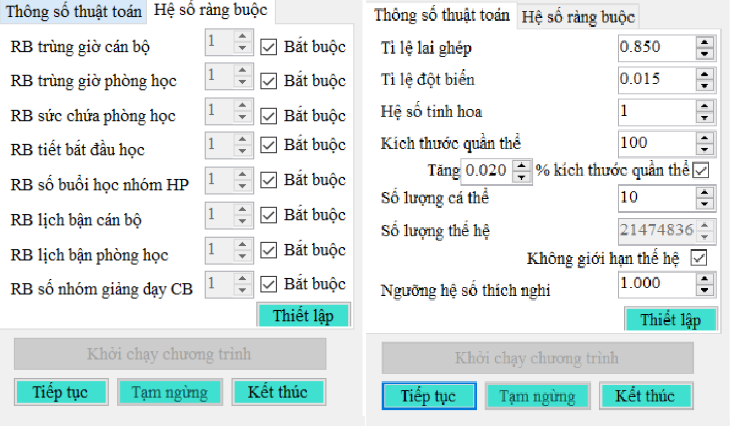
Nhận xét: với tập dữ liệu đơn giản như trên thì hệ thống chỉ mất khoảng 2 giây để đưa ra được phương án TKB tối ưu.

## Kịch bản thử nghiệm 2

* + - 1. **Mục tiêu và dữ liệu đầu vào**

Kịch bản 2 thử nghiệm với tập dữ liệu phức tạp hơn. Tổng số nhóm cần xếp: 44 nhóm, tổng số buổi học: 50 buổi, tổng số tiết học: 131 tiết, 5 giảng viên được phân công giảng dạy, 10 phòng học. Tiến hành sắp lịch cho 50 buổi của 44 nhóm học phần.

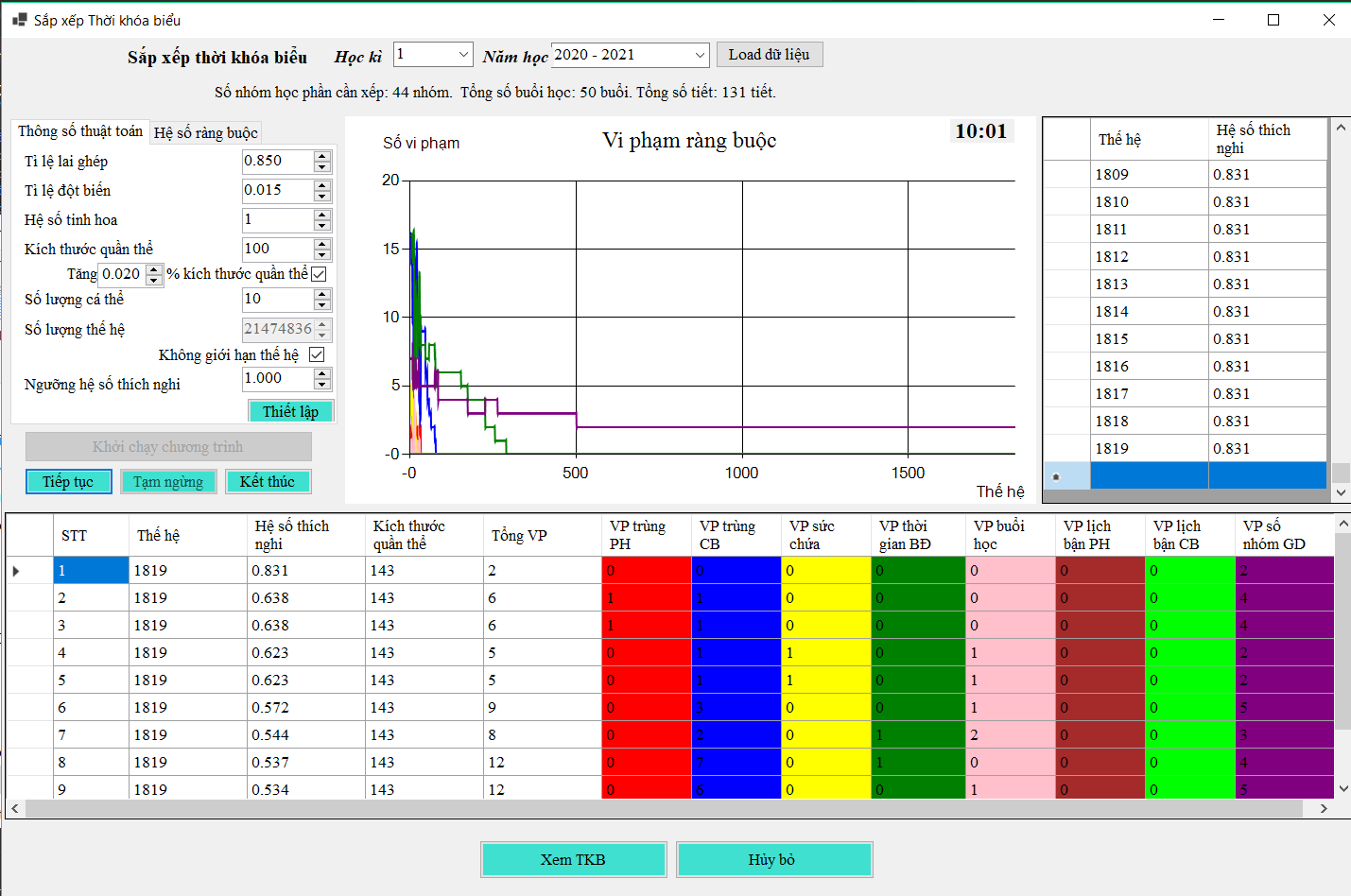
* + - 1. **Các thông số**



**Hình 4.4 Thông số chương trình cho kịch bản thử nghiệm 2**

Các thông số thiết lập với mục đích tìm ra phương án tối ưu nhất, không vi phạm các ràng buộc. Các ràng buộc ở kịch bản 2 được xem là các ràng buộc cứng, chương trình chỉ dừng lại khi tìm được phương án tối ưu nhất hoặc vượt quá thế hệ qui định.

* + - 1. **Kết quả**



Có thể tiếp tục chương

trình

**Hình 4.5 Kết quả thử nghiệm 2**

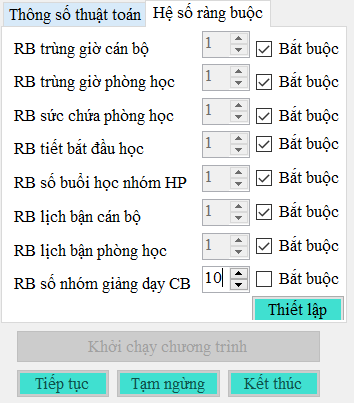
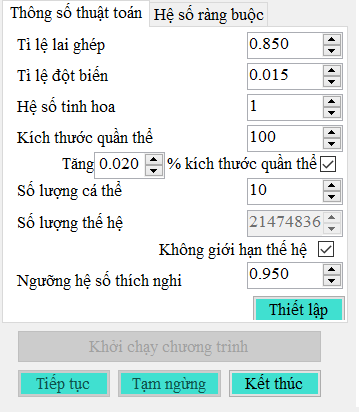
Nhận xét: Với tập dữ liệu như trên, chương trình trãi qua 10 phút vẫn chương tìm ra được phương án tối ưu nhất. Qua kết quả ta thấy ràng buộc số nhóm giảng dạy khó đạt được nhất.

## Kịch bản thử nghiệm 3

* + - 1. **Mục tiêu và dữ liệu đầu vào**

Kịch bản thử nghiệm 3 giữ nguyên tập dữ liệu của kịch bản 2. Thiết lập lại các thông số giải thuật và hệ số ràng buộc. Tiến hành lập lịch để tìm ra phương án thỏa mãn các điều kiện đầu vào.

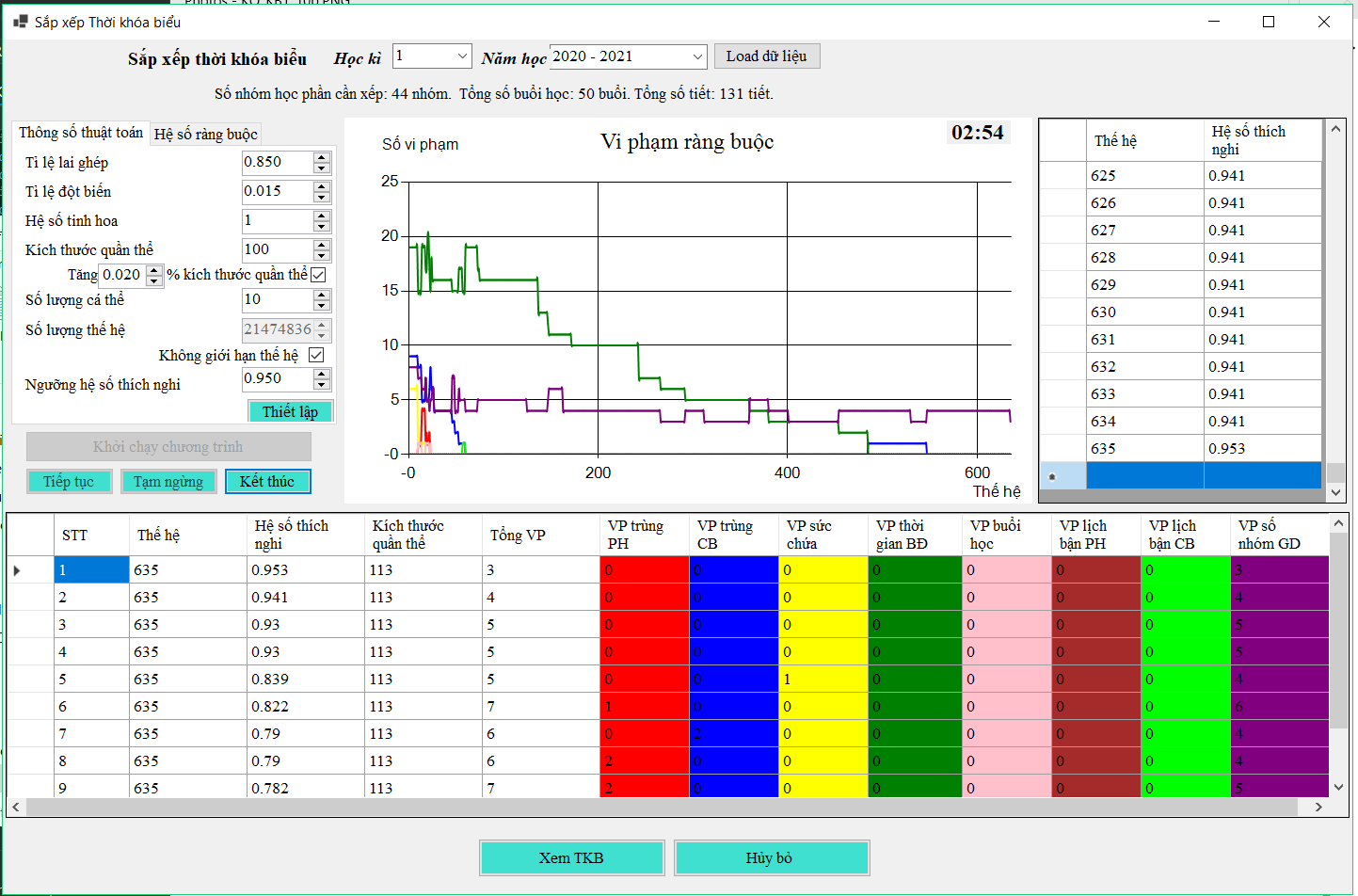
* + - 1. **Các thông số**



**Hình 4.6 Thông số chương trình cho kịch bản thử nghiệm 3**

Các thông số ở kịch bản 3 được thiết lập với mục đích chấp nhận vi phạm ràng buộc số nhóm giảng dạy trong một giới hạn cho phép, khi đó ràng buộc số nhóm giảng dạy trở thành ràng buộc mềm.

* + - 1. **Kết quả**



**Hình 4.7 Kết quả thử nghiệm 3**

Nhận xét: Cùng tập dữ liệu với kịch bản thử nghiệm 2, tuy nhiên khi thiết lập lại thông số thì chương trình chỉ mất 2 phút 54 giây để tìm ra phương án TKB thỏa mãn các yêu cầu đặt ra.

Tuy nhiên so với kết quả ở kịch bản 2 thì số vi phạm ràng buộc nhóm giảng dạy ở kịch bản 3 lại nhiều hơn.

## ĐÁNH GIÁ HỆ THỐNG

Chương trình dễ dàng chuyển đổi trạng thái các ràng buộc một cách linh hoạt, giúp người sử dụng có thể kiểm soát số lượng vi phạm ràng buộc trong các phương án sắp xếp TKB nếu không tìm được phương án tối ưu. Thời gian thực hiện chương trình tùy vào tập dữ liệu và các thông số thiết lập cho chương trình. Quá trình thử nghiệm hệ thống thực tế cho thấy nếu số lớp cần xếp tăng lên thì thời gian thực hiện sẽ tăng nhanh theo hàm mũ.

# PHẦN KẾT LUẬN

## KẾT LUẬN

Qua quá trình thực hiện luận văn, em đã có kiến thức đầy đủ về giải thuật di truyền, biết cách vận dụng giải thuật di truyền vào việc giải quyết một số bài toán tối ưu, đặc biệt là các bài toán tối ưu có không gian tìm kiếm lớn.

Vận dụng được giải thuật di truyền để xây dựng hệ thống phần mềm hỗ trợ xếp thời khóa biểu hệ tín chỉ cho trường đại học:

* + - * Xây dựng được các module quản lý dữ liệu đầu vào liên quan đến việc sắp thời khóa biểu.
      * Hệ thống hỗ trợ tổng hợp kế hoạch học tập của sinh viên và phân nhóm học phần tự động giúp tiết kiệm thời gian.
      * Hệ thống đáp ứng tốt tất cả các ràng buộc được nêu ra trong luận văn bao gồm các ràng buộc cứng và mềm.
      * Hệ thống chạy ổn định, có giao diện đẹp, có biểu đồ minh họa trực quan trong quá trình xếp thời khóa biểu.
      * Hệ thống cung cấp nhiều lựa chọn phương án sắp thời khóa biểu.
      * Kết quả xếp thời khóa biểu được trình bày đa dạng bao gồm thời khóa biểu của theo học phần, từng giảng viên, từng phòng học
      * Dữ liệu thời khóa biểu có thể dễ dàng xuất sang Microsoft Excel và có thể tra cứu từ website hỗ trợ.
      * Website hỗ trợ nhập kế hoạch học tập cho sinh viên, cập nhật nhật lịch giảng dạy cho cán bộ.

Tuy nhiên, hệ thống cũng còn có một số tồn tại sau:

* + - * Yêu cầu của bài toán xếp thời khóa biểu trong thực tế khá đa dạng trong khi hệ thống hiện tại chỉ đáp ứng được các yêu cầu cơ bản được nêu ra.
      * Hệ thống chưa đáp ứng được sắp xếp thời khóa biểu cho các lớp học phần đặc biệt: luận văn, niên luận, thực tập, các học phần thể chất, các học phần thực hành. Các nhóm học phần này sẽ được thêm thủ công sau khi việc sắp thời khóa biểu tự động hoàn tất.
      * Tốc độ truy xuất dữ liệu và thực thi hệ thống trên tập dữ liệu lớn chưa được như mục tiêu đề ra ban đầu.
      * Hiện tại việc phân công giảng dạy sẽ được gửi về Khoa/ bộ môn tiến hành phân công và gửi kết quả về cho phòng admin/quản lý học vụ nhập dữ liệu thủ công dẫn đến mất thời gian và có thể sai xót trong quá trình nhập liệu.

## HƯỚNG PHÁT TRIỂN

Để hệ thống ngày càng hoàn thiện hơn và chạy nhanh hơn cần phải tiếp tục nghiên cứu và phát triển hệ thống theo các hướng sau:

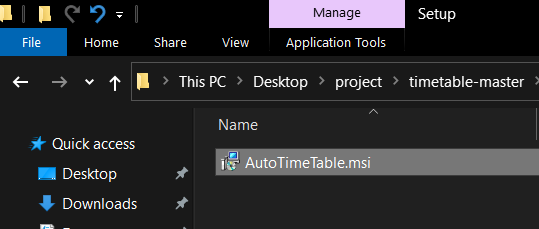
* Phát triển hệ thống:
  + - * Cải thiện tốc độ truy xuất dữ liệu.
      * Tích hợp thêm các ràng buộc xếp thời khóa biểu ngoài các ràng buộc cơ bản được đề cập trong luận văn. Khi thực hiện người sử dụng có thể tùy ý lựa chọn các ràng buộc mà mình cần đáp ứng.
      * Tích hợp thêm tính năng phân công giảng dạy trên web và phân quyền cho những người phụ trách Khoa/bộ môn có quyền truy cập và thực hiện chức năng này để tối ưu hóa thời gian và lưu trữ dữ liệu chính xác hơn.
      * Tích hợp thêm tính năng thống kê và phân tích các phương án sau khi sắp xếp, đưa ra gợi ý nếu chưa tìm được phương án tối ưu.
* Hoàn thiện giải thuật di truyền:
  + - * Song song hóa giải thuật GA để xử lý đồng thời trên nhiều máy tính nhằm tăng tốc độ thực hiện chương trình.
      * Tiếp tục hoàn thiện hàm đánh giá độ thích nghi để tùy theo ngữ cảnh có thể tự điều chỉnh các trọng số sao cho phù hợp nhất.
      * Nghiên cứu kết hợp giải thuật di truyền với các kỹ thuật khác như mạng Nơron, lôgic mờ, …để nâng cao hiệu quả cho hệ thống.

# PHỤ LỤC

1. **HƯỚNG DẪN CÀI ĐẶT**

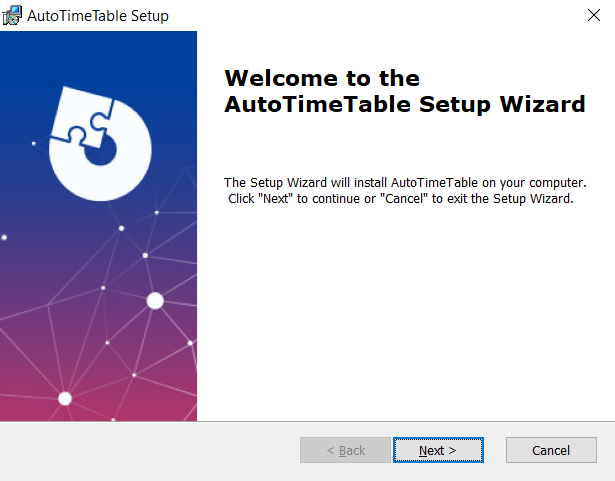
Để tiến hành cài đặt chương trình ta thực hiện các bước như sau:

* + - * Bước 1: Click vào file setup



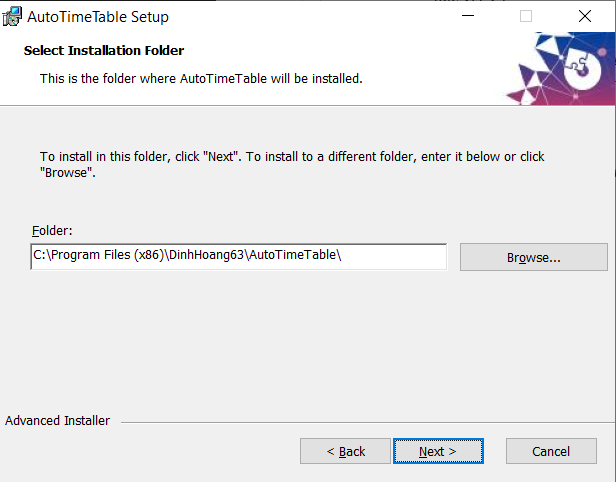
**Hình phụ lục 1: File Setup**

* + - * Bước 2: Tiến hành chạy file Setup. Chọn next để tiếp tục cài đặt.



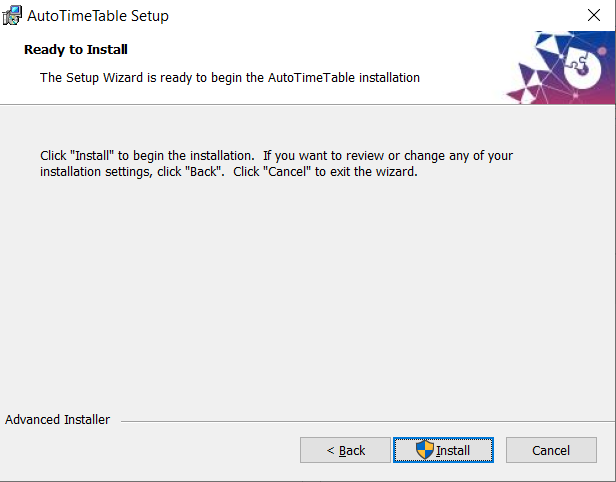
**Hình phụ lục 2: Chạy file Setup**

* + - * Bước 3: Chọn nơi cài đặt ứng dụng



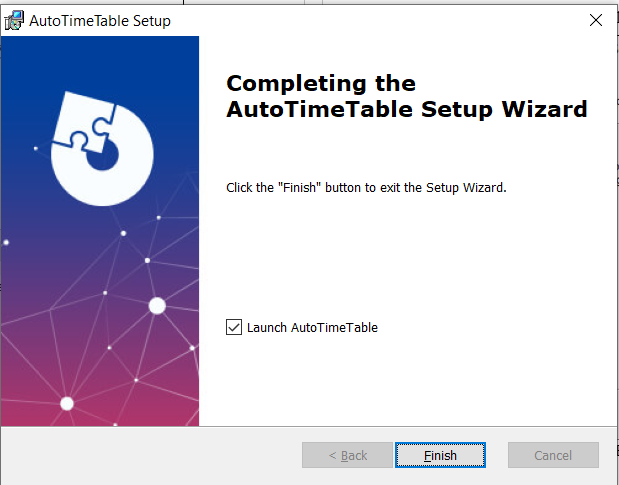
**Hình phụ lục 3: Nơi cài đặt ứng dụng**

* + - * Bước 4: Chọn Install để cài đặt ứng dụng



**Hình phụ lục 4: Cài đặt ứng dụng**

* + - * Bước 5: Cài đặt thành công ứng dụng. Chọn Lauch AutoTimeTable để chạy ứng dụng.



**Hình phụ lục 5: Hoàn thành cài đặt**

# TÀI LIỆU THAM KHẢO

1. H. X. Huấn, N. V. Thắng, *Một giải pháp tiến hóa cho bài toán thời khóa biểu*, Tạp chí Tin học và Điều khiển học T. 17, S.2 (2001).
2. Lee Jacobson, Burak Kanber, *Genetic Algorithms in Java Basics*, Appress.
3. N. H. Hòa (chủ biên), H. V. Tú, H. M. Trí, *Giáo trình Lập trình .NET*, NXB Đại học GTVT.
4. N. Đ. Thúc (2001), *Trí tuệ nhân tạo-Lập trình tiến hóa*, NXB Giáo Dục.
5. P. A. Tuấn, *Ứng dụng giải thuật di truyền để xếp thời khóa biểu hệ tín chỉ cho trường đại học*, Luận văn thạc sĩ kỹ thuật (2012), Đại học Đà Nẵng.
6. S. N. Sivanandam, S. N. Deepa (2008), *Introduction to Genetic Algorithms*, Springer.
7. Wikipedia: *https://vi.wikipedia.org/wiki/C\_Sharp*