**TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT HƯNG YÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 5**

**NGHIÊN CỨU GIẢI THUẬT DI TRUYỀN VÀ ỨNG DỤNG VÀO BÀI TOÁN SẮP XẾP THỜI KHÓA BIỂU Ở TRƯỜNG THPT**

|  |  |
| --- | --- |
| Giáo Viên Hướng Dẫn: | **Nguyễn Hoàng Điệp** |
| Sinh Viên Thực Hiện: | **Trần Thị Khuyên**  **Nguyễn Thị Quỳnh Nga** |
| Lớp: | **TK6SE** |
| Khóa: | **2008-2012** |

**Hưng Yên, tháng 12 năm 2011**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

|  |
| --- |
| **Giáo viên hướng dẫn**  **Nguyễn Hoàng Điệp** |

………………………………………………………………………………………

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

………………………………………………………………………………………

|  |
| --- |
| Giáo Viên Phản Biện |

# MỤC LỤC

[MỤC LỤC IV](#_Toc313031967)

[DANH MỤC HÌNH VII](#_Toc313031968)

[LỜI NÓI ĐẦU VIII](#_Toc313031969)

[CHƯƠNG I. LÝ THUYẾT VỀ GIẢI THUẬT DI TRUYỀN (GENETIC ALGORITHM - GA) 1](#_Toc313031970)

[1.1. Lịch sử của giải thuật di truyền. 1](#_Toc313031971)

[1.2. Tóm tắt giải thuật di truyền 1](#_Toc313031972)

[1.3. Cách biểu diễn bài toán trong giải thuật di truyền (hay chọn cách biểu diễn cấu trúc dữ liệu cho bài toán) 6](#_Toc313031973)

[1.3.1. Biểu diễn Gen bằng chuỗi nhị phân. 6](#_Toc313031974)

[1.3.1.1. Mảng integer nén để tối ưu truy xuất. 6](#_Toc313031975)

[1.3.1.2. Biểu diễn số thực bằng chuỗi nhị phân 7](#_Toc313031976)

[1.3.2. Biểu diễn gen bằng chuỗi số thực. 9](#_Toc313031977)

[1.3.3. Biểu diễn gen bằng cấu trúc cây. 10](#_Toc313031978)

[1.4. Nguyên lý về xác định tính thích nghi. 10](#_Toc313031979)

[1.4.1. Độ thích nghi tiêu chuẩn. 11](#_Toc313031980)

[1.4.2. Độ thích nghi xếp hạng (rank method). 12](#_Toc313031981)

[1.5. Mã hóa(encoding). 14](#_Toc313031982)

[1.5.1. Giới thiệu (Introduction) 14](#_Toc313031983)

[1.5.2. Mã hóa bằng số nhị phân (Binary Encoding) 15](#_Toc313031984)

[1.5.3. Mã hóa vị trí (Permutation Encoding) 15](#_Toc313031985)

[1.5.4. Mã hóa theo giá trị (Value Encoding) 15](#_Toc313031986)

[1.5.5. Cây mã hóa (Tree Encoding) 16](#_Toc313031987)

[1.6. Các phương pháp chọn(Selection). 17](#_Toc313031988)

[1.6.1. Chọn lọc Roulette(Roulette Wheel Selection). 17](#_Toc313031989)

[1.6.2. Chọn lọc xếp hạng(Rank Selection). 17](#_Toc313031990)

[1.6.3. Chọn lọc cạnh tranh( Tournament Selection). 18](#_Toc313031991)

[1.7. Các phương pháp lai tạo(crossover) và đột biến(mutation). 18](#_Toc313031992)

[1.7.1. Binary Encoding(mã hóa nhị phân) 18](#_Toc313031993)

[1.7.1.1. Lai ghép(Crossover) 18](#_Toc313031994)

[1.7.1.2. Đột biến(Mutation ) 19](#_Toc313031995)

[1.7.2. Permutation Encoding(Mã hóa vị trí) 20](#_Toc313031996)

[1.7.2.1. Crossover 20](#_Toc313031997)

[1.7.2.2. Mutation 20](#_Toc313031998)

[1.7.3. Value Encoding(mã hóa theo giá trị) 20](#_Toc313031999)

[1.7.3.1. Crossover 20](#_Toc313032000)

[1.7.3.2. Mutation 20](#_Toc313032001)

[1.7.4. Tree Encoding(Cây mã hóa) 21](#_Toc313032002)

[1.7.4.1. Crossover 21](#_Toc313032003)

[1.7.4.2. Mutation 21](#_Toc313032004)

[1.8. Các toán tử trong giải thuật di truyền 22](#_Toc313032005)

[1.8.1. Toán tử Chọn lọc 22](#_Toc313032006)

[1.8.2. Toán tử lai ghép 23](#_Toc313032007)

[1.8.3. Toán tử đột biến 24](#_Toc313032008)

[1.8.4. Toán tử sinh sản 24](#_Toc313032009)

[1.9. Các tham số cần sử dụng trong giải thuật di truyền. 24](#_Toc313032010)

[1.10. Điều kiện kết thúc thuật giải di truyền. 24](#_Toc313032011)

[1.11. Nguyên lý hoạt động của giải thuật di truyền. 25](#_Toc313032012)

[1.12. Ứng dụng của thuật giải di truyền. 25](#_Toc313032013)

[CHƯƠNG II. BÀI TOÁN SẮP XẾP THỜI KHÓA BIỂU 26](#_Toc313032014)

[2.1. Tìm hiểu chung 26](#_Toc313032015)

[2.1.1. Các đặc tính của bài toán lập lịch 26](#_Toc313032016)

[2.1.2. Bài Toán sắp xếp Thời Khoá Biểu ở trường Trung học phổ thông 26](#_Toc313032017)

[2.1.2.1. Giới thiệu bài toán 26](#_Toc313032018)

[2.1.2.2. Dữ liệu bài toán 27](#_Toc313032019)

[2.1.2.3. Các ràng buộc của bài toán 27](#_Toc313032020)

[2.1.3. Một số bước cơ bản để giải quyết bài toán lập lịch thời khoá biếu 28](#_Toc313032021)

[2.2. Áp dụng giải thuật di truyền vào bài toán sắp xếp thời khóa biểu 29](#_Toc313032022)

[2.2.1. Giai đoạn 1: Xếp lịch học các lớp 29](#_Toc313032023)

[2.2.1.1. Chọn mô hình cá thể. 29](#_Toc313032024)

[2.2.1.2. Tạo quần thể ban đầu 30](#_Toc313032025)

[2.2.1.3. Độ thích nghi - chọn cá thể 31](#_Toc313032026)

[2.2.1.4. Lai ghép và đột biến 31](#_Toc313032027)

[2.2.2. Giai đoạn 2: Xếp lịch học cho toàn trường. 32](#_Toc313032028)

[2.2.2.1. Chọn mô hình cá thể 32](#_Toc313032029)

[2.2.2.2. Tạo quần thể ban đầu 33](#_Toc313032030)

[2.2.2.3. Độ thích nghi - chọn cá thể 33](#_Toc313032031)

[2.2.2.4. Thuật toán chọn lọc, lai ghép và đột biến 34](#_Toc313032032)

[2.2.2.5. Chọn điểm dừng thuật toán 34](#_Toc313032033)

[CHƯƠNG III. PHÂN TÍCH THIẾT KẾ BÀI TOÁN 35](#_Toc313032034)

[3.1. Phân tích thiết kế với UML 35](#_Toc313032035)

[3.1.1. Biểu đồ Use Case 35](#_Toc313032036)

[3.1.2. Biểu đồ Tuần tự 36](#_Toc313032037)

[3.1.2.1. Biểu đồ Tuần tự chức năng Đăng nhập 36](#_Toc313032038)

[3.1.2.2. Biểu đồ tuần tự chức năng “sắp xếp TKB” 36](#_Toc313032039)

[3.1.2.3. Biểu đồ tuần tự chức năng “cập nhật thông tin TKB” 37](#_Toc313032040)

[3.1.2.4. Biểu đồ tuần tự chức năng “xem thời khóa biểu theo giáo viên” 37](#_Toc313032041)

[3.1.2.5. Biểu đồ tuần tự chức năng “xem thời khóa biểu theo lớp” 37](#_Toc313032042)

[3.1.3. Biểu đồ Lớp 38](#_Toc313032043)

[3.2. Thiết kế cơ sở dữ liệu bài toán 38](#_Toc313032044)

[3.2.1. Cơ sở dữ liệu 38](#_Toc313032045)

[3.2.2. Mô tả bảng trong Cơ sở dữ liệu 38](#_Toc313032046)

[3.2.2.1. Bảng tblToMon 39](#_Toc313032047)

[3.2.2.2. Bảng tblGiaoVien 39](#_Toc313032048)

[3.2.2.3. Bảng tblMonhoc 39](#_Toc313032049)

[3.2.2.4. Bảng tblHocKy 40](#_Toc313032050)

[3.2.2.5. Bảng tblLop 40](#_Toc313032051)

[3.2.2.6. Bảng tblGiangDay 40](#_Toc313032052)

[CHƯƠNG IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 42](#_Toc313032053)

[4.1. Kết quả đạt được 42](#_Toc313032054)

[4.2. Hạn chế, hướng phát triển 42](#_Toc313032055)

[4.2.1. Hạn chế 42](#_Toc313032056)

[4.2.2. Hướng phát triển 42](#_Toc313032057)

[TÀI LIỆU THAM KHẢO 43](#_Toc313032058)

# DANH MỤC HÌNH

[Hình 1: Sơ đồ tổng quát của giải thuật di truyền 5](#_Toc313031744)

[Hình 2: Biểu đồ Use Case mức tổng quát 35](#_Toc313031745)

[Hình 3: Biểu đồ tuần tự chức năng “Đăng nhập” 36](#_Toc313031746)

[Hình 4: Biểu đồ tuần tự chức năng “sắp xếp TKB” 36](#_Toc313031747)

[Hình 5: Biểu đồ tuần tự chức năng “cập nhật thông tin TKB” . 37](#_Toc313031748)

[Hình 6: Biểu đồ tuần tự chức năng “xem thời khóa biểu theo giáo viên” 37](#_Toc313031749)

[Hình 7: Biểu đồ tuần tự chức năng “xem thời khóa biểu theo lớp” 38](#_Toc313031750)

[Hình 8: Biểu đồ lớp mức Tổng quát 38](#_Toc313031751)

[Hình 9: Cơ sở dữ liệu 38](#_Toc313031752)

# LỜI NÓI ĐẦU

Với khả năng hiện nay, máy tính đã giúp con người giải quyết được rất nhiều bài toán khó mà trước kia thường bó tay. Mặc dù vậy vẫn còn một số lớn các bài toán thú vị nhưng chưa có thuật giải hợp lý để giải chúng. Trong số đó các bài toán tối ưu thường gặp trong thực tiễn.

Trước kia để giải những bài toàn tối ưu người ta thường dùng những phương pháp cổ điển như: leo đồi, mô phỏng luyện thép… Với những bài toán có không gian tìm kiếm nhỏ. Thì những phương pháp trên có thể giải quyết tốt. Nhưng với không gian tìm kiếm lớn, thì những phương pháp trên không hiệu quả. Vì vậy, điều kiện đòi hỏi chúng ta phải có những phương pháp mới để có thể giải quyết tốt những bài toán dạng trên. Ngày nay để giải bài toán tối ưu, chúng ta có thể dùng ”giải thuật di truyền” .

“Giải thuật di truyền” được phát triển dựa trên sự mô phỏng quá trình tiến hóa của sinh học. Được bắt đầu bằng Nils Aall Baricelli mô phỏng quá trình tiến hóa trong trò chơi năm 1954. Sau đó đến Alex Fraser xuất bản cuốn sách Artificial Selection (chọn lọc nhân tạo). Nhưng John Holland mới là người đầu tiên thực sự đặt tên cho giải thuật là “giải thuật di truyền” bằng việc xuất bản cuốn sách năm 1975. Từ đây giải thuật đã có tên là “giải thuật di truyền”. Và cùng với đó là sự phát triển mạnh mẽ hoàn thiện lý thuyết” giải thuật di truyền”. Và ứng dụng của giải thuật trong những bài toán thực tế. Qua quá trình tìm hiểu, em thấy những ứng dụng của “giải thuật di truyền” rất hay. Nên sau khi cô Nguyên Hoàng Điệp giao một số đề tài gợi ý chúng em đã quyết định chọn “giải thuật di truyền” làm đề tài cho đồ án 5 của mình.

Trong quá trình thực tập, em xin cảm ơn sự hướng dẫn tận tình của cô Nguyễn Hoàng Điệp và cùng Thầy Cô trong khoa đã giúp đỡ chúng em hoàn thiện đề tài này. Mặc dù chúng em đã cố gắng, nhưng do thời gian và kiến thức còn hạn chế. Nên trong đồ án còn nhiều sai sót. Vì vậy, chúng em mong sự góp ý của Thầy Cô và các bạn để hoàn thiện tốt hơn đề tài này.

Ngày 30 tháng 12 năm 2011

Sinh viên thực hiện

**Trần Thị Khuyên**

**Nguyễn Thị Quỳnh Nga**

# CHƯƠNG I. LÝ THUYẾT VỀ GIẢI THUẬT DI TRUYỀN (GENETIC ALGORITHM - GA)

## Lịch sử của giải thuật di truyền.

Trước tiên ý niệm về thuật giải di truyền đã được một số nhà sinh vật học đưa ra từ những năm 50-60, thế kỷ XX. Alex Fraser là người tiên phong nêu lên sự tương đồng giữa sự tiến hóa của sinh vật và chương trình tin học giả tưởng về genetic algorithm. Tuy nhiên, John Henry Holland mới là người triển khai ý tưởng và phương thức giải quyết vấn đề dựa theo sự tiến hóa của con người. Từ những bài giảng, bài báo của mình, ông đã đúc kết các ý tưởng vào trong cuốn sách đầu tay *Adaptation in Natural and Artifical systems (mô phỏng theo tự nhiên và hệ thống nhân tạo )*, xuất bản năm 1975. Dựa trên lý thuyết cơ bản về GA của Holland, Keneth De Jong đã triển khai, chứng minh và những thành quả do ông thực hiện đã góp phần quan trọng trong việc tạo ra nền tảng toán học cho lý thuyết thuật giải di truyền. Và sau này là John koza đã tiếp nối, phát triển giải thuật di truyền.

Lần đầu tiên Holland nghiên cứu các thuật giải này, chúng hoàn toàn không có tên. Do nguồn gốc của phương pháp này là từ các gen di truyền, Holland đã đặt tên cho nó là “thuật giải di truyền“.

## Tóm tắt giải thuật di truyền

Thuật giải di truyền (GA) là kỹ thuật chung giúp giải quyết vấn đề bài toán bằng cách mô phỏng sự tiến hóa của con người hay của sinh vật nói chung (dựa trên thuyết tiến hóa muôn loài của Darwin) trong điều kiện qui định sẵn của môi trường. GA là một thuật giải, nghĩa là mục tiêu của GA không nhằm đưa ra lời giải chính xác tối ưu mà là đưa ra lời giải tương đối tối ưu.

Theo đề xuất ban đầu của giáo sư John Holland, một vấn đề-bài toán đặt ra sẽ được mã hóa thành các chuỗi bit với chiều dài cố định. Nói một cách chính xác là các thông số của bài toán sẽ được chuyển đổi và biểu diễn lại dưới dạng các chuỗi nhị phân. Các thông số này có thể là các biến của một hàm hoặc hệ số của một biểu thức toán học. Người ta gọi các chuỗi bit này là mã *genome* ứng với mỗi cá thể, các genome đều có cùng chiều dài. Nói ngắn gọn, một lời giải sẽ được biểu diễn bằng một chuỗi bit, cũng giống như mỗi cá thể đều được quy định bằng gen của cá thể đó vậy. Như vậy, đối với thuật giải di truyền, một cá thể chỉ có một gen duy nhất và một gen cũng chỉ phục vụ cho một cá thể duy nhất.

Ban đầu, ta sẽ phát sinh một số lượng lớn, giới hạn các cá thể có gen ngẫu nhiên. Nghĩa là phát sinh một tập hợp các chuỗi bit ngẫu nhiên. Tập các cá thể này được gọi là quần thể ban đầu (initial population). Sau đó, dựa trên một hàm nào đó, ta sẽ xác định được một giá trị gọi là độ thích nghi - Fitness. Giá trị này, có thể hiểu chính là độ "tốt" của lời giải. Vì phát sinh ngẫu nhiên nên độ "tốt" của lời giải hay tính thích nghi của các cá thể trong quần thể ban đầu là không xác định.

Để cải thiện tính thích nghi của quần thể, người ta tìm cách tạo ra quần thể mới. Có hai thao tác thực hiện trên thế hệ hiện tại để tạo ra một thế hệ khác với độ thích nghi tốt hơn. Thao tác đầu tiên là sao chép nguyên mẫu một nhóm các cá thể tốt từ thế hệ trước rồi đưa sang thế hệ sau (selection). Thao tác này đảm bảo độ thích nghi của thế hệ sau luôn được giữ ở một mức độ hợp lý. Các cá thể được chọn thông thường là các cá thể có độ thích nghi cao nhất.

Thao tác thứ hai là tạo các cá thể mới bằng cách thực hiện các thao tác *sinh sản* trên một số cá thể được chọn từ thế hệ trước – thông thường cũng là những cá thể có độ thích nghi cao. Có hai loại thao tác sinh sản : một là lai tạo (crossover), hai là đột biến (mutation). Trong thao tác lai tạo, từ gen của hai cá thể được chọn trong thế hệ trước sẽ được phối hợp với nhau (theo một số quy tắc nào đó) để tạo thành hai gen mới.

Thao tác chọn lọc và lai tạo giúp tạo ra thế hệ sau. Tuy nhiên, nhiều khi do thế hệ khởi tạo ban đầu có đặc tính chưa *phong phú* và chưa phù hợp nên các cá thể không rải đều được hết không gian của bài toán . Từ đó, khó có thể tìm ra lời giải tối ưu cho bài toán. Thao tác đột biến sẽ giúp giải quyết được vấn đề này. Đó là sự biến đổi ngẫu nhiên một hoặc nhiều thành phần gen của một cá thể ở thế hệ trước tạo ra một cá thể hoàn toàn mới ở thế thệ sau. Nhưng thao tác này chỉ được phép xảy ra với tần suất rất thấp (thường dưới 0.01), vì thao tác này có thể gây xáo trộn và làm mất đi những cá thể đã chọn lọc và lai tạo có tính thích nghi cao, dẫn đến thuật toán không còn hiệu quả.

Thế hệ mới được tạo ra lại được xử lý như thế hệ trước (xác định độ thích nghi và tạo thế hệ mới) cho đến khi có một cá thể đạt được giải pháp mong muốn hoặc đạt đến thời gian giới hạn.

Tóm lại: Một thuật giải di truyền (hay một chương trình tiến hóa bất kỳ) giải một bài toán cụ thể phải gồm năm thành phần sau đây:

* Cách biểu diễn nhiễm sắc thể cho lời giải bài toán.
* Cách khởi tạo quần thể ban đầu.
* Hàm lượng giá đóng vai trò môi trường, đánh giá các lời giải theo mức độ thích nghi của chúng.
* Các phép toán di truyền.
* Các tham số khác(kích thước quần thể,Pc , Pm …)

Lược đồ GA:

Input: một bài toán tối ưu max f(x) trong không gian x € X.

Output: một nghiệm tốt của f, x0 € X f(x0) đạt lân cận max

Method

1. Khởi tạo một quần thể ban đầu với n cá thể.

2. Lặp m buớc, mỗi bước phát sinh một quần thể mới theo quy trình sau.

2.1. Lai ghép:

- Chọn ngẫu nhiên một cặp hai cá thể cha mẹ B và M theo xác xuất Pl

- Sinh hai cá thể mới C1 và C2 từ B và M.

- Thay thế C1 và C2 cho B và M.

2.2. Đột biến:

- Chọn ngẫu nhiên một cá thể X theo xác xuất Pd

- Đột biến cá thể X.

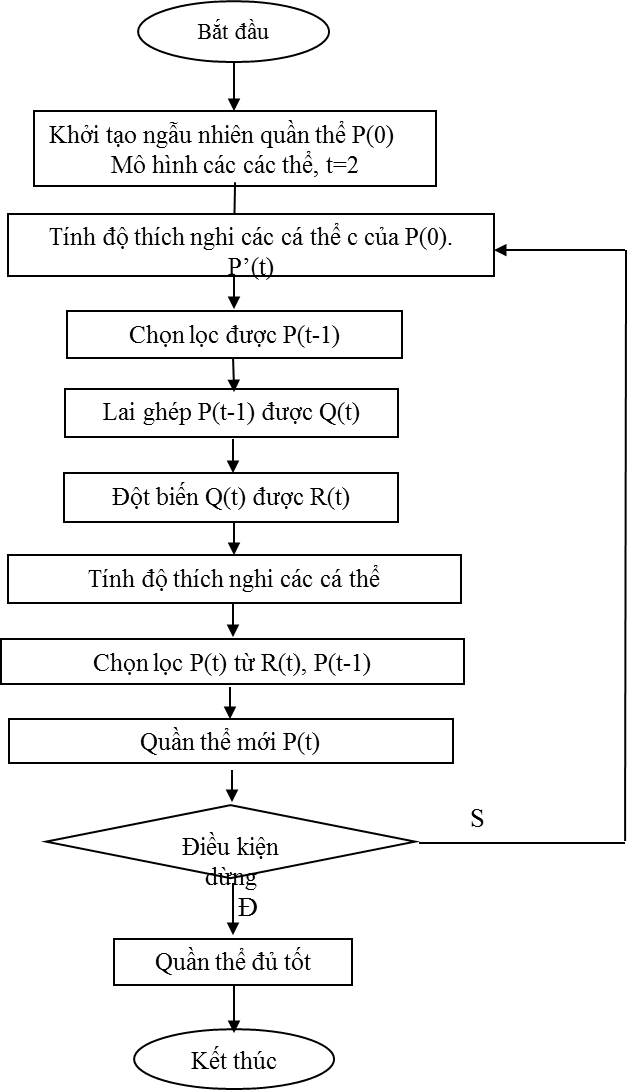
2.3. Lặp nhận:

- Tính lại độ thích nghi của các cá thể.

- Chọn các cá thể có độ thích nghi tốt đưa vào quá trình mới.

3. Lấy nghiệm.

End.



Hình 1:Sơ đồ tổng quát của giải thuật di truyền

## Cách biểu diễn bài toán trong giải thuật di truyền (hay chọn cách biểu diễn cấu trúc dữ liệu cho bài toán)

Để áp dụng giải một bài toán bằng giải thuật di truyền, thao tác quan trọng nhất – là phải biết chọn cấu trúc dữ liệu phù hợp. Để giải bài toán trong giải thuật di truyền, ta thường chọn sử dụng một trong 3 loại cấu trúc dữ liệu sau: Chuỗi nhị phân, chuỗi số thực và cấu trúc cây. Trong đó chuỗi nhị phân và chuỗi số thực thường được sủ dụng nhiều hơn.

### Biểu diễn Gen bằng chuỗi nhị phân.

Quy tắc biểu diễn gen qua chuỗi nhị phân : Chọn chuỗi nhị phân ngắn nhất nhưng đủ thể hiện được tất cả kiểu gen.Để biểu diễn chuỗi nhị phân, ta thường dùng các cách sau : Mảng byte, mảng bit biểu diễn bằng mảng byte, mảng bit biểu diễn bằng mảng INTEGER.Mảng byte và mảng bit bây giờ ít sử dụng. Đối với máy tính ngày nay, người ta thường dùng mảng integer để tối ưu truy xuất. Vì vậy ở đây em chỉ giới thiệu về mảng integer.

VD: Nhiễm sắc thể x ta biểu diễn bằng 1 chuỗi 15 bit

X=(010100110010101)2

#### Mảng integer nén để tối ưu truy xuất.

Trong các máy tính ngày nay, thông thường thì đơn vị truy xuất hiệu quả nhất không còn là byte nữa mà là một bội số của byte. Đơn vị truy xuất hiệu quả nhất được gọi là độ dài từ (word length). Hiện nay, các máy Pentium đều có độ dài từ là 4 byte. Do đó, nếu ta tổ chức chuỗi nhị phân dưới dạng byte sẽ làm chậm phần nào tốc độ truy xuất. Để hiệu quả hơn nữa, ta sử dụng mảng kiểu INTEGER. Lưu ý kiểu INTEGER có độ dài phụ thuộc vào độ dài từ của máy tính mà trình biên dịch có thể nhận biết được. Chẳng hạn với các version PASCAL,C trên hệ điều hành DOS, kích thước của kiểu INTEGER là 2 byte. Trong khi đó, với các version PASCAL, C trên Windows 9x như Delphi, Visual C++ thì độ dài của kiểu INTEGER là 4 byte. Do đó, để chương trình của chúng ta chạy tốt trên nhiều máy tính khác nhau, ta dùng hàm sizeof(<tên kiểu>). Hàm này sẽ trả ra độ dài của kiểu dữ liệu ta đưa vào.

#### Biểu diễn số thực bằng chuỗi nhị phân

Tuy có nhiều chọn lựa nhưng thông thường, để biểu diễn một số thực x, người ta chỉ dùng công thức đơn giản, tổng quát sau :

Giả sử ta muốn biểu diễn số thực x nằm trong khoảng [min, max] bằng một chuỗi nhị phân A dài L bit. Lúc đó, ta sẽ chia miền [min, max] (lượng hóa) thành 2L-1 vùng. Trong đó, kích thước một vùng là :

Người ta gọi g là độ chính xác của số thực được biểu diễn bằng cách này (vì g quy định giá trị thập phân nhỏ nhất của số thực mà chuỗi nhị phân dài L bit có thể biểu diễn được). Giá trị của số thực x được biểu diễn qua chuỗi nhị phân sẽ được tính như sau:

*x = min + Decimal(<A>)\*g.*

trong đó Decimal(<A>) là hàm để tính giá trị thập phân nguyên dương của chuỗi nhị phân A theo quy tắc đếm. Hàm này được tính theo công thức sau:

*Decimal(<A>) = aL-1.2L-1 + … + a2. 22 + a1.21 + a0.20*

Với ai là bit thứ i trong chuỗi nhị phân tính từ phải sang trái (bit phải nhất là bit 0)

VD: Bài toán tối ưu số

Tìm giá trị lớn nhất của hàm f(x) = x\*sin(10\*pi\*x) + 1 với x € [-1,2]

Sử dụng vectơ bit làm nhiễm sắc thể để biểu diễn giá trị thực của biến x. Chiều dài vectơ phụ thuộc vào độ chính xác cần có, trong thí dụ này, ta tính chính xác đến 6 số lẻ.

Miền giá trị của x có chiều dài 2 - (-1) = 3; với yêu cầu về độ chính xác 6 số lẻ như thế phải chia khoảng [-1, 2] thành ít nhất 3\*106 khoảng có kích thước bằng nhau. Điều này có nghĩa là cần có 22 bit cho vevtơ nhị phân (nhiễm sắc thể):

2097152 = 221 < 3 000000 < 222 = 4194304

* Ánh xạ chuỗi nhị phân (b21b20…b0) từ cơ số 2 sang cơ số 10:

(<b21b20…b0>)2 = ()2 =x’

* Tìm số thực x tương ứng

x = -1 + x’\* 

với -1 là lân cận dưới của miền giá trị và 3 là chiều dài của miền.

Thí dụ, nhiễm sắc thể (1000101110110101000111) biểu diễn số 0.637197 vì x’ = (1000101110110101000111)2 = 228896710 và x = -1.0 + 2288967\* 3/4194303 = 0.637197

Ta cần cực đại hóa hàm sau đây:

f(x1, x2) = 21.5 + x1\* sin(4\*pi\*x1) + x2 \* sin(10\*pi\*x2)

Với -3.0 ≤ x1 ≤ 12.1 và 4.1 ≤ x2 ≤ 5.8

Giả sử ta cần độ chính xác đến 4 số lẻ đối với mỗi biến. Miền của biến x1 có chiều dài 12.1 – (-3) = 15.1; điều kiện chính xác đòi hỏi đoạn [-3, 12.1] cần được chia thành các khoảng có kích thước bằng nhau, ít nhất là 15.1 \* 10000 khoảng. Điều này có nghĩa là cần 18 bit làm phần đầu tiên của nhiễm sắc thể:

217 ≤ 151000 ≤ 218

Miền của biến x2 có chiều dài 5.8 – 4.1 = 1.7; điều kiện chính xác đòi hỏi đoạn [4.1, 5.8] cần được chia thành các khoảng có kích thước bằng nhau, ít nhất là 1.7 \* 10000 khoảng. Điều này có nghĩa là cần 15 bit kế tiếp của nhiễm sắc thể:

214 ≤ 17000 ≤ 215

Chiều dài toàn bộ nhiễm sắc thể (vectơ lời giải) lúc này là m =15+18 = 33 bit; 18 bit đầu tiên mã hóa x1, và 15 bit còn lại (từ 19 đến 33) mã hóa x2.

Ta hãy xét một nhiễm sắc thể làm thí dụ:

(010001001011010000111110010100010)

* + 18 bit đầu tiên, 010001001011010000 , biểu diễn

x1 = -3.0 + decimal(0100010010110100002) \* = -3.0 + 70352 \*  = -3.0 + 4.052426

* + 15 bit kế tiếp 111110010100010, biểu diễn

x2 = 4.1 + decimal(1111100101000102)\* = 4.1 + 31906 \*  = 4.1 + 1.655330 = 5.755330

Như vậy, nhiễm sắc thể (010001001011010000111110010100010) Tương ứng với <x1, x2> = <1.052426, 5.755330>

Độ thich nghi của nhiễm sắc thể này là: f(1.052426, 5.755330) = 20.252640

### Biểu diễn gen bằng chuỗi số thực.

Đối với những vấn đề bài toán có nhiều tham số, việc biểu diễn gen bằng chuỗi số nhị phân đôi lúc sẽ làm cho kiểu gen của cá thể trở nên quá phức tạp. Dẫn đến việc thi hành các thao tác trên gen trở nên kém hiệu quả. Khi đó, người ta sẽ chọn biểu diễn kiểu gen dưới dạng một chuỗi số thực. Tuy nhiên, chọn biểu diễn kiểu gen bằng chuỗi số thực, bạn cần lưu ý quy tắc sau: *Quy tắc biểu diễn kiểu gen bằng chuỗi số thực: Biểu diễn kiểu gen bằng số thực phải đảm bảo tiết kiệm không gian đối với từng thành phần gen.*

Quy tắc này lưu ý chúng ta phải tiết kiệm về mặt không gian bộ nhớ đối với các từng thành phần gen. Giả sử nghiệm của bài toán được cấu thành từ 3 thành phần, thành phần X thực có giá trị trong khoảng [1.0, 2.0], thành phần Y nguyên trong khoảng [0,15] và thành phần Z trong khoảng [5,8]. Thì chúng ta rất không nên chọn biểu diễn kiểu gen bằng một chuỗi 3 thành phần số thực. Vì như chúng ta đã biết, ít nhất mỗi số thực được phải được biểu diễn bằng 6 byte. Chỉ với 3 số thực, ta đã tốn hết 18 byte. Như vậy với trường hợp cụ thể này, ta nên chọn biểu diễn bằng chuỗi nhị phân, trong đó dùng khoảng 10(bit) cho thành phần X (độ chính xác khoảng 0.001), 4 bit cho thành phần Y và 2 bit cho thành phần Z. Tổng cộng chỉ chiếm có 16 bit = 2 byte. Chúng ta đã tiết kiệm được rất nhiều bộ nhớ!

### Biểu diễn gen bằng cấu trúc cây.

Một loại cây thường được sử dụng trong thuật giải di truyền là dạng cây hai nhánh (ở đây chúng tôi dùng chữ hai nhánh để phân biệt với loại cây nhị phân – thường dùng trong sắp xếp và tìm kiếm).

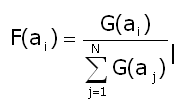
## Nguyên lý về xác định tính thích nghi.

“Tính tốt của một cá thể (lời giải) trong một quần thể chỉ là một cơ sở để xác định tính thích nghi của cá thể (lời giải) đó”. Nguyên lý này ban đầu có vẻ hơi bất ngờ một khi chúng ta đã hiểu những ý tưởng chung của thuật giải di truyền. Thật đơn giản, người leo lên ngọn đồi cao nhất trong thế hệ hiện tại vẫn có khả năng bị”kẹt” trong các thế hệ sau cũng như một lời giải chưa tốt ở thế hệ hiện tại vẫn còn khả năng tiềm tàng dẫn đến lời giải tối ưu. Tuy vậy, thường thì lời giải tốt ở thế hệ hiện tại sẽ có xác suất dẫn đến lời giải tối ưu cao hơn những lời giải xấu hơn. Do đó, người ta vẫn xem độ tốt của lời giải là một yếu tố căn bản để xác định tính thích nghi của lời giải. Thông thường, độ thích nghi của lời giải cũng chính là xác suất để cá thể đó được chọn lọc hoặc lai ghép khi tiến hành sinh ra thế hệ kế tiếp. Ta sẽ lần lượt tìm hiểu 3 phương pháp để xác định tính thích nghi của một cá thể.

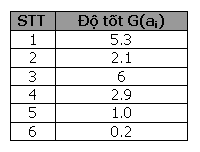
### Độ thích nghi tiêu chuẩn.

Hàm mục tiêu là hàm dùng để đánh giá độ tốt của một lời giải hoặc cá thể. Hàm mục tiêu nhận vào một tham số là gen của một cá thể và trả ra một số thực. Tùy theo giá trị của số thực này mà ta biết độ tốt của cá thể đó (chẳng hạn với bài toán tìm cực đại thì giá trị trả ra càng lớn thì cá thể càng tốt, và ngược lại, với bài toán tìm cực tiểu thì giá trị trả ra càng nhỏ thì cá thể càng tốt).

Giả sử trong một thế hệ có N cá thể, cá thể thứ i được ký hiệu là ai. Hàm mục tiêu là hàm G. Vậy độ thích nghi của một cá thể ai tính theo độ thích nghi tiêu chuẩn là



Chẳng hạn, xét một thế hệ gồm có 6 cá thể với độ tốt (giá trị càng lớn thì cá thể càng tốt) lần lượt cho trong bảng sau

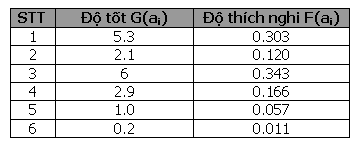


Theo công thức trên, tổng tất cả G của 6 phần tử là : 17.5

Như vậy, độ thích nghi của phần tử a1: F(a1) = 5.3 / 17.5 » 0.303

Độ thích nghi của phần tử a2: F(a2) = 2.1 / 17.5 = 0.12

Ta có bảng kết quả cuối cùng như sau :



**Nhận xét**: độ thích nghi luôn có giá trị biến thiên trong khoảng [0,1]. Hơn nữa, vì độ thích nghi sẽ ứng với khả năng được chọn lọc trong việc sinh ra thế hệ sau nên người ta thường chọn cách tính sao cho độ thích nghi cuối cùng là một xác suất, nghĩa là tổng độ thích nghi của các cá thể phải nhỏ hơn hoặc bằng 1.

### Độ thích nghi xếp hạng (rank method).

Cách tính độ thích nghi tiêu chuẩn như trên chỉ thực sự hiệu quả đối với những quần thể có độ tốt tương đối đồng đều giữa các cá thể. Nếu, vì một lý do nào đó – có thể do chọn hàm mục tiêu không tốt - có một cá thể có độ tốt quá cao, tách biệt hẳn các cá thể còn lại thì các cá thể của thế hệ sau sẽ bị “hút” về phía cá thể đặc biệt đó. Do đó, sẽ làm giảm khả năng di truyền đến thế sau của các cá thể xấu, tạo nên hiện tượng di truyền cục bộ, từ đó có thể làm giảm khả năng dẫn đến lời giải tốt nhất (vì cá thể đặc biệt đó chưa chắc đã dẫn đến lời giải tốt nhất).

Phương pháp xác định độ thích nghi xếp hạng sẽ loại bỏ hiện tượng di truyền cục bộ này. Phương pháp này không làm việc trên giá trị độ lớn của hàm mục tiêu G mà chỉ làm việc dựa trên thứ tự của các cá thể trên quần thể sau khi đã sắp xếp các cá thể theo giá trị hàm mục tiêu G. Chính vì vậy mà ta gọi là độ thích nghi xếp hạng. Phương pháp này sẽ cho ta linh động đặt một trọng số để xác định sự tập trung của độ thích nghi lên các cá thể có độ tốt cao, mà vẫn luôn đảm bảo được quy luật: cá thể có độ thích nghi càng cao thì xác suất được tồn tại và di truyền càng cao.

Một cách ngắn gọn, ta có độ thích nghi (hay xác suất được chọn) của cá thể thứ i được tính theo công thức sau:

F(i) = p\*(1-p)i-1

với p là một hằng số trong khoảng [0,1].

Công thức trên được xây dựng dựa trên quy tắc được trình bày ngay sau đây và chúng ta sẽ xem phần giải thích quy tắc này như một tư liệu tham khảo.

**QUY TẮC**

1) Sắp xếp các cá thể của quần thể giảm dần theo thứ tự của giá trị hàm mục tiêu.

2) Chọn một con số p trong khoảng [0,1]. Đây chính là trọng số xác định độ “hút” của các cá thể tốt.

3) Mỗi lượt chọn chỉ chọn một cá thể. Trong một lượt chọn, lần lượt xét các cá thể theo thứ tự đã sắp. Nếu xét đến cá thể thứ i mà cá thể đó được chọn thì lượt chọn kết thúc, ta thực hiện lượt chọn kế tiếp. Ngược lại, nếu cá thể thứ i không được chọn, ta xét đến cá thể thứ i+1. Ta quy ước rằng, khi đã xét đến một cá thể, thì xác suất để chọn cá thể đó (trong thao tác chọn lọc hoặc lai tạo) luôn là p. Rất hiển nhiên, khi đã xét đến một cá thể thì xác suất (XS) để KHÔNG chọn cá thể đó sẽ là 1-p.

Ta ký hiệu a[i] là cá thể thứ i. Từ quy tắc trên, suy ra để a[i] được xét đến thì :

+ a[i-1] đã phải được xét đến

+ nhưng a[i-1] phải KHÔNG được chọn.

Do đó, XS a[i] được xét đến (chứ không phải XS để được chọn!)

= XS a[i-1] được xét \* XS a[i-1] KHÔNG được chọn.

= XS a[i-1] được xét \* (1-p)

Trong đó, XS a[1] được xét =1 vì cá thể đầu tiên luôn được xét đến.

Bây giờ ta sẽ xây dựng công thức tổng quát để tính XS a[i] được xét đến dựa theo p.

XS a[1] được xét = 1 = (1-p)0

XS a[2] được xét = XS a[1] được xét \* (1-p) = 1\*(1-p) = (1-p)1

XS a[3] được xét = XS a[2] được xét \* (1-p) = (1-p)1 \* (1-p) = (1-p)2

XS a[4] được xét = XS a[3] được xét \* (1-p) = (1-p)2 \* (1-p) = (1-p)3

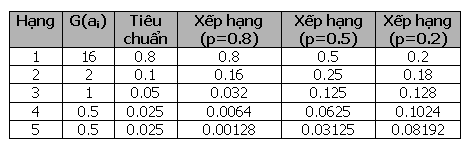
...

Nói tóm lại :

XS a[i] được xét = XS a[i-1] được xét \* (1-p) = (1-p)i-2 \* (1-p) = (1-p)i-1

Như vậy XS a[i] được chọn = XS a[i] được xét \* p = (1-p)i-1\*p

Để thấy được tính linh động của phương pháp này, bạn hãy quan sát giá trị thích nghi ứng với mỗi giá trị p khác nhau trong bảng sau :



Giá trị p càng nhỏ thì độ giảm của tính thích nghi càng nhỏ. Dựa vào đặc tính này, ta có thể dễ dàng kiểm soát được tính “hút” của các cá thể tốt trong quần thể bằng cách tăng hoặc giảm trị p tương ứng.

## Mã hóa(encoding).

### Giới thiệu (Introduction)

Mã hóa nhiễm sắc thể là bước đầu tiên để giẩi quyết vấn đề bằng giải thuật di truyền. Mã hóa là vấn đề quan trọng nhất.

### Mã hóa bằng số nhị phân (Binary Encoding)

Mã hóa bằng số nhị phân là phương pháp chính. Bởi vì là phương pháp đầu tiên GA dung để mã hóa và nó đơn giản.

Mỗi nhiễm sắc thể được biểu diễn bằng chuỗi bit 0 hoặc 1.

|  |  |
| --- | --- |
| Nhiễm sắc thể A | 101100101100101011100101 |
| Nhiễm sắc thể B | 111111100000110000011111 |

*Ví dụ mã hóa nhiễm sắc thể bằng chuỗi nhị phân*

Mã hóa số thực biểu diễn cho nhiễm sắc thể với số lượng gen nhỏ. Còn với lượng gen lớn, phương pháp này thường không giải quyết được nhiều vấn đề tự nhiên và các phép sửa chữa sau lai ghép và đột biến.

### Mã hóa vị trí (Permutation Encoding)

Những vấn đề dựa trên thứ tự có thể dùng mã hóa vị trí, ví dụ như bài toán người du lịch hoặc thao tác thứ tự vấn đề .

Trong mã hóa vị trí mỗi nhiễm sắc thể được biểu diễn bằng chuỗi số nguyên theo một vị trí trình tự nhất định.

|  |  |
| --- | --- |
| Nhiễm sắc thể A | 1  5  3  2  6  4  7  9  8 |
| Nhiễm sắc thể B | 8  5  6  7  2  3  1  4  9 |

*Ví dụ mã hóa nhiễm sắc thể theo vị trí*

Mã hóa vị trí có thể được dùng trong nhiều vấn đề có tính trình tự. Một vài phép lai ghép và đột biến đòi hỏi sự nhất quán, cho một vài vấn đề.

### Mã hóa theo giá trị (Value Encoding)

Mã hóa theo giá trị có thể dùng trong nhiều vấn đề , ở một vài giá trị phức tạp(ví dụ: giá trị thực). Dùng mã hóa nhị phân để giải quyết vấn đề này rất khó.

Trong mã hóa theo giá trị, mỗi nhiễm sắc thể được biểu diễn theo trình tự dựa trên giá trị. Phương pháp này dùng giải quyết nhiều vấn đề, ví dụ : Số thực, ký tự hoặc đối tượng không xác định.

|  |  |
| --- | --- |
| Nhiễm sắc thể A | 1.2324  5.3243  0.4556  2.3293  2.4545 |
| Nhiễm sắc thể B | ABDJEIFJDHDIERJFDLDFLFEGT |
| Nhiễm sắc thể C | (back), (back), (right), (forward), (left) |

*Ví dụ mã hóa nhiễm sắc thể theo giá trị*

Mã hóa theo giá trị giải quyết tốt cho nhiều vấn đề đặc biệt. Tuy nhiên phương pháp này thường cần để phát triển một vài vấn đề lai ghép mới và đột biến cụ thể.

### Cây mã hóa (Tree Encoding)

Cây mã hóa dùng trong chương trình tiến hóa hoặc biểu thức. cho lập trình tiến hóa. Trong cây mã hóa mỗi nhiễm sắc thể là một cây , ví dụ hàm và lệnh trong ngôn ngữ lập trình.

|  |  |
| --- | --- |
| Nhiễm sắc thể A | Nhiễm sắc thể B |
| treec1 | treec2 |
| ( +  x  ( /  5  y ) ) | ( do\_until  step  wall ) |

*Ví dụ mã hóa nhiễm sắc thể bằng cây*

Cây mã hóa dùng trong chương trình tiến hóa hoặc cấu trúc khác có thể mã hóa bằng cây. Ngôn ngữ lập trình LISP thường dùng tìm kết quả, chương trình sự kiện trong LISP biểu diễn bằng cây một cách dễ dàng, vì vậy lai ghép và đột biến có thể dễ dùng và đáng tin cậy .

## Các phương pháp chọn(Selection).

Chọn lọc cá thể thông qua kết quả, hay mục đích của vấn đề dựa trên mức độ thích nghi của cá thể. Vì vậy, đánh giá độ thích nghi của cá thể để tìm ra cá thể tốt nhất. Thông thường, đặt mỗi vấn đề nhỏ tương ứng với một giá trị điểm thích nghi , kết quả đánh giá gồm tổng các số điểm đó. Cá thể tốt nhất sẽ có điểm thấp nhất hoặc lớn nhất.

Theo thuyết Darwin, cá 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. Sau đây là vài phương pháp trong số đó.

### Chọn lọc Roulette(Roulette Wheel Selection).

Các cá thể được chọn theo độ thích nghi của chúng. Nhiễm sắc thể tốt hơn có cơ hội cao hơn để tham dự vào thế hệ tiếp theo.

Thuật giải chọn lọc roulette(Davis, [1991,8]) như sau:

* Tính tổng độ thích nghi của mọi thành viên trong quần thể; gọi kết quả là độ thích nghi tổng cộng(total fitness).
* Phát sinh n, một số ngẫu nhiên giữa 0 và độ thích nghi tổng cộng(total fitness).
* Trở về thành viên đầu tiên của quần thể có độ thích nghi lớn hơn hay bằng n , bổ sung vào độ thích nghi của các thành viên đứng trước trong quần thể.

### Chọn lọc xếp hạng(Rank Selection).

Phương pháp này sẽ sắp hạng cá thể dựa trên độ thích nghi của chúng. Cá thể xấu nhất sẽ có giá trị 1, kế tiếp là 2… Và cá thể tốt nhất có độ thích nghi N(N là số các nhiễm sắc thể trong quần thể).

### Chọn lọc cạnh tranh( Tournament Selection).

* Chọn lọc cạnh tranh 2(2- Tournament Selection)

Hai nhiễm sắc thể khác nhau được chọn ngẫu nhiên và được so sánh với nhiễm sắc thể tồn tại. Nếu nhiễm sắc thể I1 không tốt hơn nhiễm sắc thể I2 nghĩa là : f(I1)≤ f(I2), thì nhiễm sắc thể I1 chết đi và bị loại ra khỏi quần thể(liên kết được phá vỡ 1 cách tùy ý). Quá trình này lặp lại đến hết N nhiễm sắc thể còn lại.

* Chọn lọc cạnh tranh 3(3- Tournament Selection)

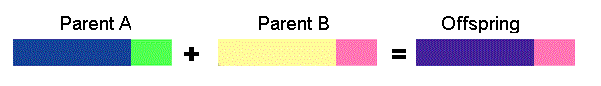
Ba nhiễm sắc thể khác nhau được chọn ngẫu nhiên và được so sánh với nhiễm sắc thể tồn tại. Nếu chúng ta có: f(I1) ≤ f(I2) và f(I1) ≤ f(I3), thì nhiễm sắc thể I1 chết đi và bị loại ra khỏi quần thể(liên kết được phá vỡ 1 cách tùy ý). Quá trình này lặp lại đến hết N nhiễm sắc thể còn lại.

## Các phương pháp lai tạo(crossover) và đột biến(mutation).

Lai ghép và đột biến là hai phép cơ bản được thực hiện trong giải thuật di truyền trên nhiều vấn đề. Kiểu và thực thi của phép thực hiện trên mã hóa và ngoài ra trên vấn đề. Có nhiều phuơng pháp lai ghép và đột biến. Ở đây chúng ta chỉ miêu tả một số thường dùng.

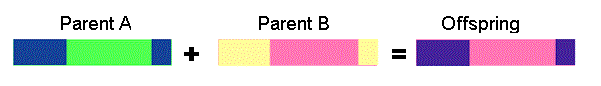
### Binary Encoding(mã hóa nhị phân)

#### Lai ghép(Crossover)

Lai ghép ở một vị trí(Single point crossover) – Từ hai nhiễm sắc thể cha mẹ ban đầu ta cắt ở một vị trí sau đó ghép lại với nhau thành nhiễm sắc thể con

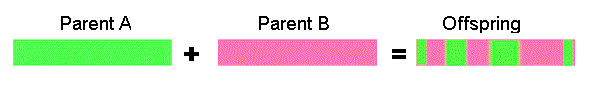
11001011+11011111 = 11001111

Lai ghép ở hai vị trí(Two point crossover)– Từ hai nhiễm sắc thể cha mẹ ban đầu ta cắt ở hai vị trí sau đó ghép chúng với nhau thành nhiễm sắc thể con.



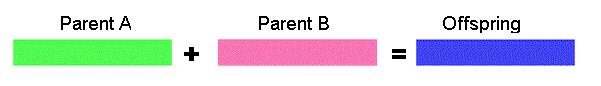
11001011 + 11011111 = 11011111

Lai ghép đồng dạng(Uniform crossover) – Những bit được copy ngẫu nhiên từ nhiễm sắc thể cha thứ nhất sang nhiễm sắc thể cha thứ hai và ngược lại.



11001011 + 11011101 = 11011111

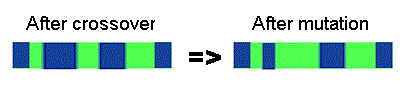
Lai ghép số học(Arithmetic crossover) – Một vài phép tính số học được thực hiện khi lai ghép để tạo ra nhiễm sắc thể con.(AND,OR,NOT…)



11001011 + 11011111 = 11001001 (AND)

#### Đột biến(Mutation )

Chèn bit(Bit inversion) – chọn một số bit sau đó chèn vào nhiễm sắc thể cha, tạo ra nhiễm sắc thể mới.



11001001 =>  10001001

### Permutation Encoding(Mã hóa vị trí)

#### Crossover

Single point crossover(Lai ghép một vị trí) – Chọn một vị trí lai ghép, sau đó sao ghép hai nhiễm sắc thể cha mẹ ở vị trí đã chọn, ta hãy tự điều chỉnh cho phù hợp. Chú ý: ở đây có nhiều đường tạo ra nhiễm sắc thể con sau phép lai ghép này.

(1 2 3 4 5 6 7 8 9) + (4 5 3 6 8 9 7 2 1) = (1 2 3 4 5 6 8 9 7)

#### Mutation

Thay đổi thứ tự(Order changing )- Chọn hai vị trí sau đó đổi vị trí cho nhau.

(1 2 3 4 5 6 8 9 7) => (1 8 3 4 5 6 2 9 7)

### Value Encoding(mã hóa theo giá trị)

#### Crossover

Tất cả các phương pháp lai ghép trong mã hóa nhị phân đều có thể dùng được ở đây.

#### Mutation

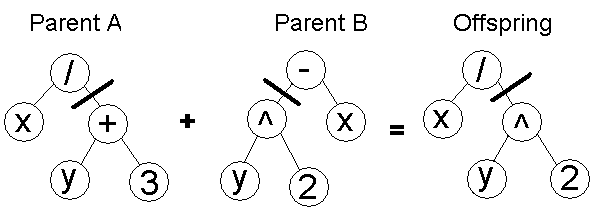
Ta thay đổi giá trị thực của một hoặc vài giá trị trong nhiễm sắc thể.

(1.29 5.68  2.86  4.11  5.55) => (1.29  5.68  2.73  4.22  5.55)

### Tree Encoding(Cây mã hóa)

#### Crossover

Tree crossover – Ta chọn một vị trí trong nhiễm sắc thể cha mẹ sau đó ghép với nhau được nhiễm sắc thể mới.



#### Mutation

Thay đổi toán tử,số(Changing operator, number )- chọn note sau đó thay đổi nó. Một số toán tử lai ghép trong GA mã hóa số thực. Xét bài toán tối ưu max f(x1,x2,…,xn) trên miền D thuộc không gian Rm

Trong bài toán dạng này, một cách tự nhiên thường sử dụng GA mã hóa số thực, mỗi cá thể được biểu thị bởi một vectơ Rn (tất nhiên phải thuộc miền xác định D). Mỗi quần thể kích cỡ m (có m cá thể) có thể biểu diễn như một ma trận thực cấp m×n . Với GA mã hóa số thực, có nhiều dạng toán tử lai ghép khác nhau . Sau đây là một số phương pháp lai ghép thường dùng.

* + Lai số học

Với hai cá thể cha mẹ p1 = (x1,x2,…,xn) và p2 = (y1,y2,…,yn) các cá thể con được sinh ra như sau.

c1= ap1 + (1-a)p2

c2=(1-a)p1 + ap2

a: là số lấy ngẫu nhiên trong khoảng [0,1]

* Lai đơn giản

Phép lai này tương tự như lai một điểm của GA kinh điển. Với một vị trí k chọn ngẫu nhiên (1< k < n), các cá thể con được sinh ra như sau:

c 1 = (x1,x2,..,xk , yk+1,…,yn)

c2 = (y1,y2,…,yk, xk+1,…,xn)

* Lai ghép mặt lạ

Phép lai này khởi tạo một vectơ ngẫu nhiên r= (r1, r2 ,…, rn) trong đó các ri chỉ là 0 hay 1. Sau đó cá thể con được sinh ra như sau:

c1 = (z1, z2,…,zn) trong đó zi = xi nếu ri =1 và zi = yi nếu ri = 0

c2 = (u1, u2 ,…, un ) trong đố ui = xi nếu ri =0 và ui = yi nếu ri = 1

* Lai ghép BLX-α

Phép lai ghép này chỉ tạo một cá thể con từ hai cá thể cha mẹ. Mỗi thành phần zi của cá thể con được chọn theo phân phối ngẫu nhiên đều trong khoảng [min(xi,yi) – I.α, max(xi,yi) + I.α], trong đó

I= max(xi, yi) – min(xi, yi)

Tham số α thường được chọn là 0.5 . Khi đó toán tử này thường được gọi là BLX-0.5 

## Các toán tử trong giải thuật di truyền

### Toán tử Chọn lọc

Việc chọn lọc các cá thể từ một quần thể dựa vào độ thích nghi của mỗi cá thể. Các cá thể có độ thích nghi cao có nhiều khả năng được chọn lựa (những cá thể khỏe mạnh có nhiều khả năng được phối giống). Hàm thích nghi chỉ cần là một hàm thực dương, nó có thể không tuyến tính, không liên tục, không khả vi. Giả sử thế hệ hiện thời là P(t) gồm n cá thể {x[1], x[2], ..., x[n]}. Số n được gọi là cỡ của quần thể. Với cá thể x[\*], ta tính độ thích nghi f(x[\*]). Tính tổng độ thích nghi của toàn bộ quần thể: F = f(x[1]) + f(x[2]) + ... + f(x[n]);

Mỗi lần chọn lọc, ta thực hiện hai bước sau:

Bước 1: Sinh một giá trị thích nghi ngẫu nhiên là một số thực q trong khoảng (0, F);

Bước 2: x[k] là cá thể được chọn nếu k là số nhỏ nhất sao cho tổng độ thích nghi của k cá thể đầu tiên không nhỏ hơn q, tức là f(x[1])+f(x[2])+... +f(x[k]) >= q.  
Rõ ràng với cách chọn này, các cá thể có độ thích nghi càng cao (gây ra tổng lớn hơn q) thì càng được chọn. Các cá thể có độ thích nghi cao có thể có một hay nhiều bản sao, các cá thể có độ thích nghi thấp có thể không có mặt trong thế hệ sau (nó bị chết đi).

### Toán tử lai ghép

Trên các cá thể được chọn lọc (sau khi thực hiện xong toán tử chọn lọc), ta tiến hành toán tử lai ghép. Với cỡ của quần thể là n, ta đưa ra một xác suất lai ghép là pc. Xác suất này đưa ra hy vọng là có n.pc cá thể được lai ghép.

Với mỗi cá thể, ta thực hiện hai bước sau đây:

Bước 1: Sinh ra một xác suất lai ghép là số thực r nào đó trong đoạn [0, 1]

Bước 2: Nếu r < pc thì cá thể đó được chọn để lai ghép

Từ các cá thể được chọn để lại ghép, ta cặp đôi chúng một cách ngẫu nhiên. Trong trường hợp nhiễm sắc thể là các chuỗi nhị phân có độ dài cố định, giả sử là m, ta có thể thực hiện phép lai ghép như sau:

Với mỗi cặp, sinh ra một vị trí ngẫu nhiên làm điểm bắt đầu ghép là một số nguyên p trong đoạn [0, m-1].

Tổng quát, giả sử có hai cặp nhiễm sắc thể của hai cá thể được chọn lai ghép:

a = (a[1], ..., a[p], a[p+1], ..., a[m])

b = (b[1], ..., b[p], b[p+1], ..., b[m])

Cặp này được thay thế bởi hai đoạn con của nhau từ vị trí thứ p+1:

a = (a[1], ..., a[p], b[p+1], ..., b[m])

b = (b[1], ..., b[p], a[p+1], ..., a[m])

### **Toán tử đột biến**

Ta thực hiện đột biến trên các cá thể sau khi đã lai ghép. Đột biến là thay đổi trạng thái của một số gen nào đó trong nhiễm sắc thể. Một gen chịu một xác suất đột biến là pm. Xác suất đột biến pm do ta xác định và là xác suất thấp.

Tổng quát với nhiễm sắc thể là chuỗi nhị phân. Với mỗi vị trí i trong nhiễm sắc thể:

a = (a[1], ..., a[p], a[p+1], ..., a[m])

ta sinh ra một số thực ngẫu nhiên pi trong đoạn [0,1]. Đột biến a được biến thành á như sau:

a = (a[1], ..., a[\*], ..., a[m]), trong đó:

a[\*] = a[\*] nếu pi >= pm và a[\*] = 1  a[\*] nếu pi < pm.

Sau quá trình chọn lọc, lai ghép, đột biến, một thế hệ mới được sinh ra. Công việc còn lại của thuật toán là chỉ việc lặp lại các bước trên.

### Toán tử sinh sản

* Sinh sản hữu tính: sử dụng phép lai ghép từ cha me để tạo ra các cá thể mới mang một phần đặc điểm cha và một phần đặc điểm của mẹ.
* Sinh sản vô tính: sử dụng phép đột biến để sinh ra cá tể mới khác cha mẹ

## Các tham số cần sử dụng trong giải thuật di truyền.

Kích thước quần thể: PopSize, là số cá thể duy trì qua mỗi thế hệ tiến hóa của thuật giải di truyền. Xác xuất đột biến: Pm là xác suất đột biến của gen. Xác suất lai: Pc là xác suất một cá thể được chọn cho phép lai ghép.

## Điều kiện kết thúc thuật giải di truyền.

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 quy định trước, không cần biết kết quả thế nào.
* Tính theo thời gian: Không cần biết đã bao nhiêu thế hệ hay kết quả thế nào, chỉ dựa vào số giờ quy định mà kết thúc.
* Tổ hợp: Dùng nhiều phương án khác nhau cho vấn đề.

## Nguyên lý hoạt động của giải thuật di truyền.

Thuật gải di truyền dựa trên nền tảng lý thuyết biểu diễn chuỗi nhị phân và lý thuyết sơ đồ.

## Ứng dụng của thuật giải di truyền.

Trên thực tế, giải thuật di truyền đã được ứng dụng thành công trong nhiều lĩnh vực như kế hoạch hóa và tìm đường, quan sát máy(machine learning), thiết kế chế tạo máy, các hệ thống điều khiển ống dẫn dầu, học máy(machine learning)… cùng những nghiên cứu sâu hơn về sử dụng giải thuật di truyền trong các bài toán mang tính tổng chất quát hơn như phân tích biệt số tuyến tính, mô phỏng dự báo…

# CHƯƠNG II. BÀI TOÁN SẮP XẾP THỜI KHÓA BIỂU

## Tìm hiểu chung

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

### 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

### Bài Toán sắp xếp Thời Khoá Biểu ở trường Trung học phổ thông

#### Giới thiệu bài toán

Bài toán sắp xếp thời khóa biểu ở trường học nói chung và sắp xếp thời khóa biểu ở trường trung học phổ thông nói riêng là một bài toán khó, sự phức tạp của bài toán không chỉ ở vấn đề tìm ra môt thời khóa biểu cho học sinh thỏa mãm các ràng buộc về thời gian, ràng buộc chuyên môn, quy định của bộ giáo dục và nhà trường mà còn một vấn đề khó khăn hơn là ta phải tìm được một thời khóa biểu tốt thích hợp cho tất cả các giáo viên, phải thỏa mãn các điều kiện, yêu cầu về thời gian, hạn chế số tiết trống trong một ngày và số ngày lên lớp của giáo viên trong thời khóa biểu.

Việc sắp xếp thời khóa biểu của các trường phổ thông luôn luôn phải thực hiên trước khi học kỳ mới bắt đầu. Trước khi học kỳ mới bắt đầu, nội dung các môn học và giáo viên phụ trách môn học của từng lớp phải được xác định thông qua cuộc họp chuyên môn, kết quả của cuộc họp này được gửi cho ban giám hiệu nhà trường và việc lên lịch cho toàn bộ trường do hai phó hiệu trường đảm nhiệm. Hiện nay, việc sắp lịch này ở hầu hết các trường phổ thông đều được thực hiện một cách thủ công, và hầu như phải dựa vào kinh nghiệm thực tế mới có thể làm được. Thông thường việc sắp xếp thời khóa biểu này phải mất trung bình một tuần.

Vậy bài toán đặt ra vấn đề cần sắp xếp thời khoá biếu cho một trường THPT. Cần có sự sắp xếp lịch học cho các lớp sao cho vừa phù hợp lại vừa tiện dụng nhất.

#### Dữ liệu bài toán

* Danh sách khoá học
* Danh sách môn học và lớp học trong học kỳ
* Danh sách lớp học
* Danh sách giáo viên
* Danh sách môn học và số tiết
* Bảng phân công giáo viên giáo dạy tại các lớp
* Bảng yêu cầu ràng buộc của giáo viên với lịch dạy
* Bảng yêu cầu ràng buộc của lớp với lịch học

#### Các ràng buộc của bài toán

Ràng buộc cứng:

* + Một giáo viên chỉ dạy được một lớp trong cùng một tiết
  + Các lớp chỉ có một môn học trong cùng một tiết
  + Một giáo viên không thể dạy quá 20 giờ mỗi tuần.

Ràng buộc mềm:

* + Một giáo viên không dạyquá 5 tiết/ngày
  + Tất cả các bài học của một môn nào đó dạy tại một lớp phải được dạy bởi cùng một giáo viên.
  + Mỗi lớp chỉ học 1 môn tại 1 thời điểm
  + Một lớp có thể có các giờ trống
  + Một lớp có thể có
  + cùng một môn nhiều lần trong một ngày
  + Tất cả các giáo viên có cùng số lượng giờ dạy như nhau.

### Một số bước cơ bản để giải quyết bài toán lập lịch thời khoá biếu

Bước 1. Khởi tạo dữ liệu thời khóa biểu mới

Bước 2. Nhập, điều chỉnh dữ liệu gốc thời khóa biểu

Bước 3. Nhập, sửa, điều chỉnh các ràng buộc chính của thời khóa biểu

Các ràng buộc chính của thời khóa biểu là nhóm các dữ liệu có nhiệm vụ định hình và khuôn dạng của thời khóa biểu. Đây là nhóm các lệnh rất quan trọng của bài toán và phần mềm thời khóa biểu.

Bước 4. Nhập bảng Phân công giảng dạy (PCGD)

Bảng phân công giảng dạy (hay còn gọi là Phân công chuyên môn) là phần dữ liệu quan trọng nhất và phức tạp nhất của mọi thời khóa biểu. Bảng này chỉ ra các phân công cụ thể của thời khóa biểu: giáo viên nào dạy lớp nào, môn học nào và một tuần dạy bao nhiêu tiết

Bước 5. Chuẩn bị xếp thời khóa biểu

Bước 6. Xếp tự động TKB

Bước 7. Điều chỉnh, tinh chỉnh dữ liệu thời khóa biểu

Bước 8. Hoàn thiện thời khóa biểu

Bước 9. In ấn TKB

Bước 10. Tổng hợp, thống kê và truy vấn thông tin thời khóa biểu

## Áp dụng giải thuật di truyền vào bài toán sắp xếp thời khóa biểu

Vấn đề của bài toán khá phức tạp về mặt ràng buộc, nhưng phương pháp chia để trị vẫn là biện pháp hữu hiệu trong mọi vấn đề phức tạp. Ở đây cũng vậy, theo phân cấp các ràng buộc mà ta giải quyết bài toán xếp thời khóa biểu này thành hai giai đoạn khác nhau:

* Giai đoạn 1: nhằm giải quyết thành phần ràng buộc ở mức lớp học, với các vấn đề cơ bản phức tạp của những đối tượng liên quan tới việc học của lớp. Khi đã có được kết quả cuối cùng là lịch học cho từng lớp một cách hoàn chỉnh, chúng sẽ được dùng làm thông tin cho giai đoạn sau.
* Giai đoạn 2 : tổng hợp lại các ràng buộc còn lại và đã được đơn giản hóa trong giai đoạn trước. Kết quả của giai đoạn này chính là mục tiêu cuối cùng của bài toán. Đó là lịch học của các lớp trong một cơ sở.

Cả hai giai đoạn tuy có mục tiêu và dữ liệu khác nhau, nhưng về cách giải quyết có tính tương tự nhau, nên không khác gì nhiều khi áp dụng vào mô hình thuật giải di truyền.

### Giai đoạn 1: Xếp lịch học các lớp

#### Chọn mô hình cá thể.

Lịch học của một lớp gồm 3 thành phần chính, bao gồm: giáo viên, môn học, tiết học trong tuần. Việc đặt ngẫu nhiên các môn học, giáo viên giảng dạy vào các tiết họcsẽ tạo thành thời khóa biểu của từng lớp. Như vậy một lớp học tương ứng sẽ có nhiều lịch học khác nhau, do đó ta chọn mỗi lịch học làm cá thể trong thuật giải di truyền.

Và trong ba thành phần đó, các tiết học là thành phần ổn định hơn về số lượng cũng như về giá trị của chúng, cho nên ta chọn giáo viên, môn học làm đơn vị nhiễm sắc thể trong cá thể. Vì đối với môn học việc làm nhiễm sắc thể là phù hợp với tính không ổn định của nó : với số lượng các môn phụ thuộc từng lớp học, cũng giống như số lượng nhiễm sắc thể trong cá thể, có chiều dài không nhất thiết phải cố định hay bằng nhau. Ngoài ra chưa kể đến tính phức tạp của môn học về số tiết phải học luôn bị thay đổi, trong khi giá trị các giờ học thì ngược lại, có thể xác định một cách rõ ràng và nhanh chóng. Tương ứng mỗi cá thể là một lịch học thực thụ của lớp. Vì vậy, khi tạo cá thể, chúng ta vẫn phải đảm bảo sự đúng đắn về tính chất trong lịch học : phải đủ số tiết học, số môn học, không có sự chồng chéo lên nhau tại cùng thời điểm trong các môn... Để giải quyết việc này, chúng ta sử dụng một tham biến đánh dấu các tiết học đã lên lịch, để môn học sau sẽ không bị sắp trùng vào những vị trí này, mà môn học này sẽ được đưa vào vị trí khác. Tất nhiên, với mỗi lịch học sẽ có sự sắp xếp khác nhau.

#### Tạo quần thể ban đầu

Trước khi tạo quần thể ban đầu trong phần này, chúng ta phải chuẩn bị sẵn về dữ liệu cho quá trình thực thi, từ lúc khởi tạo đến khi cho ra kết quả, bao gồm đầy đủ thông tin của một lớp đang được chọn. Tất cả như sau :

* Các ràng buộc lớp, giáo viên được phân công dạy.
* Các môn học
* Tính toán số tiết học tương ứng các môn.
* Chọn qui định đọc và ghi nhận nhiễm sắc thể.
* …

Giống như cá thể được mô tả ở trên, hàng loạt các cá thể được tạo ra và được xem như quần thể ban đầu trong mô hình thuật giải di truyền của phần xếp lịch lớp. Sau khi quần thể có đủ số lượng, bước tiếp theo là đánh giá quần thể, kiểm tra xem độ thích nghi tốt nhất hiện đang tồn tại của quần thể.

#### Độ thích nghi - chọn cá thể

Đây là phần giải quyết các yêu cầu đưa ra cho bài toán, chủ yếu vẫn xem xét trên các thành phần ràng buộc. Tương ứng với mỗi loại ràng buộc, chúng ta sẽ gán cho chúng một giá trị thích nghi nào đó, mà một khi cá thể đi qua, các ràng buộc được lắp đặt vào, và sẽ cho ra giá trị thích nghi cụ thể cho cá thể đó, kết thúc công việc tính độ thích nghi. Nghe rất đơn giản nhưng thực chất đây là vấn đề khó nhất, quan trọng nhất của bài toán. Chi tiết cụ thể như sau:

* Trước hết ta nói về giáo viên. Khi chọn phân công giảng dạy, chúng ta phải biết chắc rằng giáo viên đó sẽ trống vào giờ đó, môn đó, buổi đó của lớp học. Hay nói cách khác, chúng ta cần kiểm tra ràng buộc tiết học, mà đã tương ứng với mỗi môn trong lịch học, xem xét lại các môn có thể học giờ đó hay không. Kế tiếp là xét giờ học của lớp. Do một qui định nào đó mà lớp có thể học giờ này hay giờ kia, chẳng hạn như không học ba tiết đầu trong ngày thứ hai,...
* Cuối cùng kiểm tra lại sự chồng chéo giờ lẫn nhau của các môn học. Việc kiểm tra này nhất thiết phải làm, vì trong lúc lai ghép, đột biến, có thể gây ra sai lệch. Cho nên tốt nhất ta phải kiểm tra chúng. Giống như lúc khởi động, ta dùng một biến chứa tất cả giờ học ở các môn để giúp cho việc đánh giá. Tương tự các ràng buộc giáo viên và lớp. Mỗi vấn đề sẽ có một biến lưu trữ giờ làm việc, để tránh các tiết học theo qui định mà ta đã ghi nhận cho một giáo viên hay lớp học tương ứng.

Có nhiều cách để chọn một cá thể tốt. Chọn cách tính theo độ thích nghi cao nhất hoặc thấp nhất. Thông thường, người ta chọn cách tính tốt nhất. ở đây, chúng ta cũng chọn cách tính tốt nhất tức là xếp theo giá trị giảm dần của giá trị bị phạt theo độ thích nghi.

#### Lai ghép và đột biến

Về thuật toán lai ghép, ta dùng lai ghép tại 1 điểm: lấy ngẫu nhiên một đoạn nhiễm sắc thể bên nhiễm sắc thể cha, số còn lại sẽ lấy ở bên nhiễm sắc thể mẹ. Còn thuật toán đột biến : chỉ việc hoán vị hai nhiễm sắc thể một cách ngẫu nhiên trong cá thể. Ta có thể sửa thông số xác xuất về đột biến, lai ghép của chương trình trong lúc chạy thực thi.

Phần này áp dụng thực thi cho tất cả các lớp, tương ứng với mỗi lớp sẽ có một file lưu trữ tất cả các lịch lớp mà có thể sử dụng, dưới hình thức các nhiễm sắc thể trong quẩn thể. Ngoài mục đích xem xét kiểm tra, chúng còn được dùng làm thông tin để chạy lịch sau này

**2.2.1.5. Chọn lọc**

Trong bài toán này ta dùng phương pháp chọn lọc tự nhiên dựa vào độ thích nghi và vòng quay bánh xe Roulette

Thuật giải chọn lọc roulette(Davis, [1991,8]) như sau:

Tính tổng giá trị thích nghi của tất cả thành viên quần thể và gọi nó là tổng thích nghi (total fitness).

Phát sinh một số n là số ngẫu nhiên trong khoảng từ 0 đến tổng thích nghi.

Trả lại thành viên quần thể đầu tiên mà độ thích nghi của nó cộng với độ thích nghi của các thành viên quần thể trước đó lớn hơn hay bằng n…

Ví dụ:

o Với pop\_size = 3

o Fit[0] = 0,0032

o Fit[1] = 0,0576

o Fit[2] = 0,0264

Sum = 0,0872

Giả sử: random01() = 0,5

rand = 0,5 \* 0,0872 = 0,0436

j = 0; partsum = fit[0] = 0,0032

j = 1; partsum = 0,0032 + fit[1] = 0,0608

partsum > rand 🡪 i = j = 1; select() = 1;

### Giai đoạn 2: Xếp lịch học cho toàn trường.

#### Chọn mô hình cá thể

Lịch học toàn trường bao gồm tất cả các lịch học của các lớp hiện có trong cơ sở, nếu mỗi lớp đều có một lịch học rõ ràng thì đó có nghĩa là có lịch toàn trường. Dựa vào giai đoạn đầu, trên mỗi lớp đã cho ra hàng loạt các lịch học, việc chọn ngẫu nhiên lịch học của một lớp thì không có gì khó khăn. Nhìn mô hình cá thể trong lịch lớp ta thấy lớp học trong cơ sở có tính chất như môn học trong lớp, cho nên ta chọn lịch lớp học làm đơn vị của nhiễm sắc thể trong mô hình thuật toán di truyền trong xếp lịch toàn trường. Và tương tự, ta chọn 1 thời khóa biểu toàn trường làm cá thể.

Ở mỗi nhiễm sắc thể là một con số mang tính chất như một trong những chỉ số trong file lưu trữ thông tin cá thể của lịch lớp ( chỉ số một lịch học của lớp ). Như vậy phạm vi giá trị các nhiễm sắc thể sẽ khác nhau, nhưng ta luôn xác định được phạm vi đó một cách rõ ràng, chỉ cần đọc giá trị kích thước của file tương ứng của lớp mà thôi.

Giống như trong lịch lớp, cá thể lịch toàn trường cũng phải qua một giai đoạn kiểm tra ban đầu, để có thể ở mức đạt được dạng đúng của một thời khóa biểu.

#### Tạo quần thể ban đầu

Quần thể khởi đầu gồm những cá thể được tạo ra như mô hình trên, nhưng thông tin các lớp học phải được chọn cùng trong một buổi học thuộc cơ sở, và có file lịch lớp đầy đủ. Ở đây kích thước cá thể là số lớp hiện có, cho nên dài hay ngắn tùy theo cơ sở, cũng giống như lịch lớp chiều dài được tính theo số môn hiện có của lớp.

Công việc này sẽ tốn rất nhiều thời gian, vì công việc đọc file để kiểm tra, nhưng quần thể sẽ cho ra kết quả đúng nhất về mặt áp dụng bài toán vào thuật giải di truyền.

#### Độ thích nghi - chọn cá thể

Giai đoạn hội tụ cá thể trong quần thể, trên cơ bản việc đánh giá cơ sở tùy theo số lớp, số giờ học và số phòng học. Phòng học phải hợp lý về sức chứa của nó đối với lớp học, thông thường người ta chọn phòng lớn nhất cho lớp tương ứng lớn nhất. Nhưng ở đây ta chọn phương pháp khác, lớp và phòng sẽ được xếp theo thứ tự lớn dần. Cho nên khi xếp lịch, lớp sẽ được đặt ở một phòng vừa nhất mà cơ sở đang có. Điều này tuy mất thời gian hơn nhưng thực tế thì nó sẽ hợp lý hơn.

Cũng đánh giá lại ràng buộc lịch giáo viên, nhưng lần này chỉ xét về mặt trùng giờ dạy ở các lớp cùng một thời điểm. Tương tự, ta sẽ sử dụng một tham biến lịch dạy cho mỗi giáo viên, để ghi nhận và tránh trường hợp trùng giờ này.

Với các lần kiểm tra tương ứng với một giá trị thích nghi, cuối cùng tổng các giá trị này chính là độ thích nghi của cá thể. Công việc không khác gì trong lịch lớp, cá thể được chọn là cá thể tốt nhất, giá trị thích nghi đạt ở mức đỉnh là 0.

#### Thuật toán chọn lọc, lai ghép và đột biến

Sử dụng lại của phần xếp lịch lớp, chọn cá thể theo độ thích nghi, lai ghép ngẫu nhiên tại một đoạn và đột biến hoán vị điểm. Do giống nhau về mặt dữ liệu, yêu cầu và cấu trúc thuật toán cũng không khác nhau nhiều, việc dùng lại này, sẽ không gây ảnh hưởng gì trong quá trình thực hiện xếp lịch.

Một lần nữa nói về thời gian thực thi, sẽ mất nhiều thời gian hơn công việc xếp lịch lớp, do số lượng và phạm vi ràng buộc khá lớn và phải đọc dữ liệu trên các file. Nhưng về mặt hoạt động không khác nhau.

#### Chọn điểm dừng thuật toán

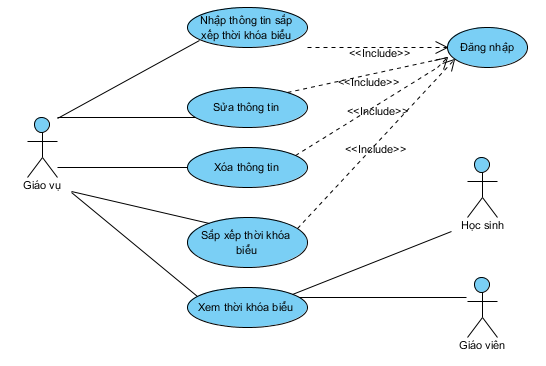
Đã được nói ở trong từng giai đoạn của các phần áp dụng thuật giải di truyền vào bài toán, điểm dừng thuật toán dựa trên độ thích nghi của nó. Một số bài toán chọn điểm dừng theo số thế hệ, hoặc dựa trên tính tương đối của kết quả, nhưng với bài toán này cần có một kết thúc tuyệt đối tốt nhất, mặt dù số thế hệ vẫn phải được chọn trước ngay từ đầu. Vì tính chất yêu cầu trong bài toán này là không bị sai lệch.

Nếu trong quá trình thực thi qua các giai đoạn, chỉ cần một kết quả không đạt đến điểm dừng, xem như bài toán sẽ không có kết quả.

# CHƯƠNG III. PHÂN TÍCH THIẾT KẾ BÀI TOÁN

## Phân tích thiết kế với UML

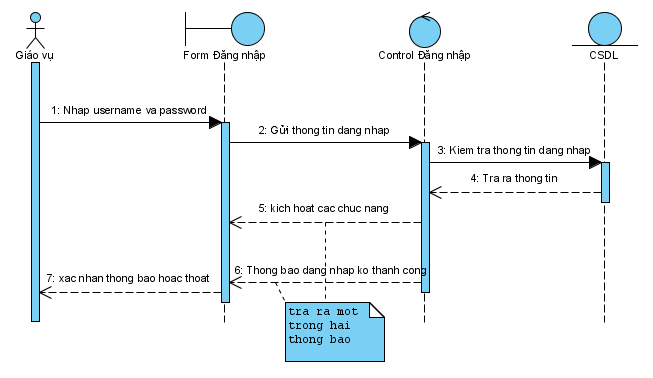
### Biểu đồ Use Case



Hình 2. Biểu đồ Use Case mức tổng quát

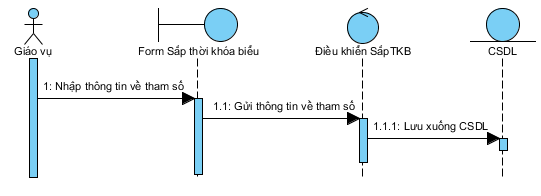
### Biểu đồ Tuần tự

#### Biểu đồ Tuần tự chức năng Đăng nhập



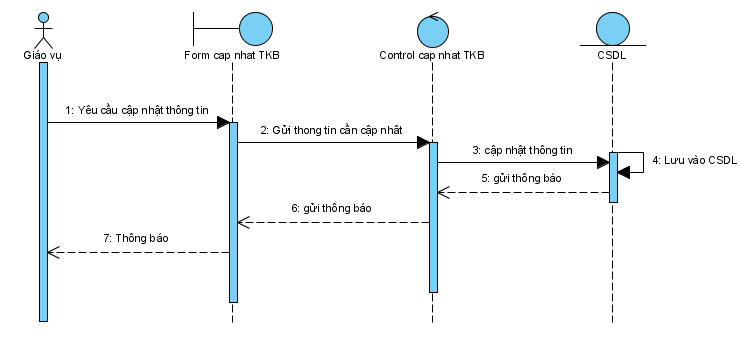
Hình 3: Biểu đồ tuần tự chức năng “Đăng nhập”

#### Biểu đồ tuần tự chức năng “sắp xếp TKB”



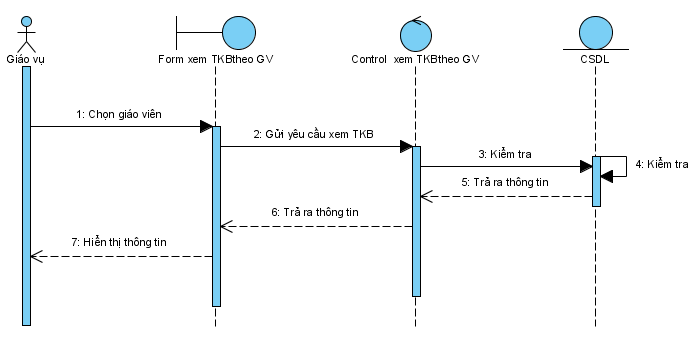
Hình 4:Biểu đồ tuần tự chức năng “sắp xếp TKB”

#### Biểu đồ tuần tự chức năng “cập nhật thông tin TKB” bao gồm Nhập, Sửa, Xóa.



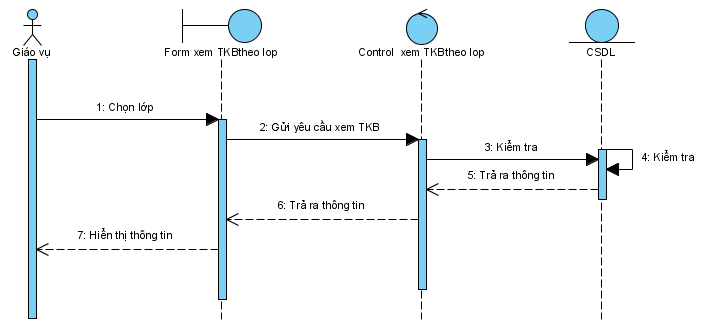
Hình 5: Biểu đồ tuần tự chức năng “cập nhật thông tin TKB”

#### Biểu đồ tuần tự chức năng “xem thời khóa biểu theo giáo viên”



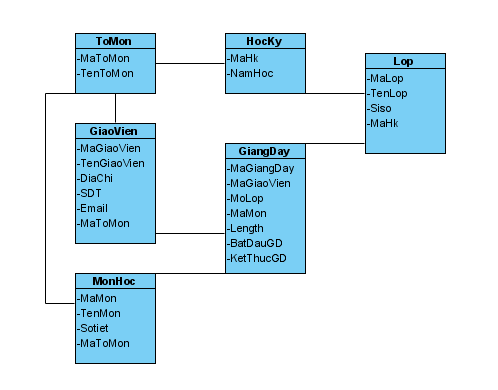
Hình 6: Biểu đồ tuần tự chức năng “xem thời khóa biểu theo giáo viên”

#### Biểu đồ tuần tự chức năng “xem thời khóa biểu theo lớp”



Hình 7: Biểu đồ tuần tự chức năng “xem thời khóa biểu theo lớp”

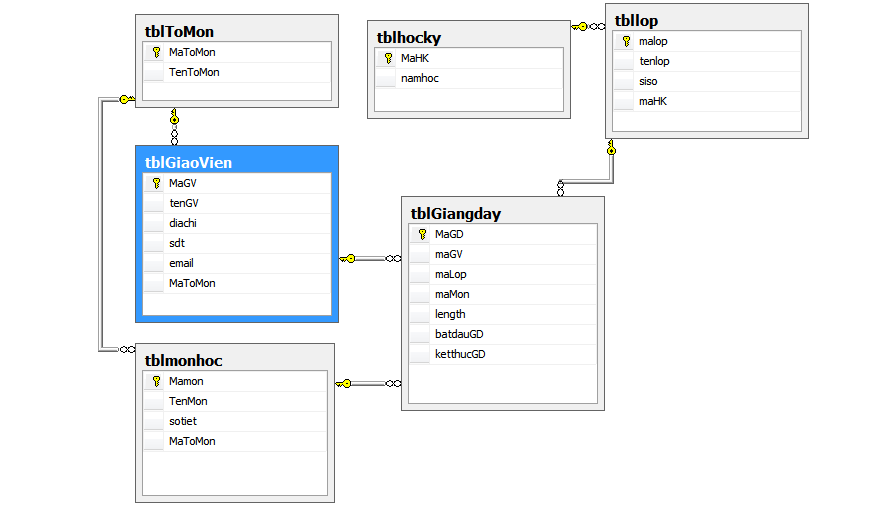
### Biểu đồ Lớp



Hình 8: Biểu đồ lớp mức Tổng quát

## Thiết kế cơ sở dữ liệu bài toán

### Cơ sở dữ liệu



Hình 9: Cơ sở dữ liệu

### Mô tả bảng trong Cơ sở dữ liệu

#### Bảng tblToMon

Bảng dữ liệu này chứa thông tin về các tổ môn trong trường. Mỗi tổ môn có mã và tên riêng.

Bảng thuộc tính:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaToMon | Nvarchar(50) | Primary key | Mã để phân biệt giữa các tổ môn |
| 2 | TenToMon | Nvarchar(50) | Not null | Tên tổ môn |

#### Bảng tblGiaoVien

Bảng dữ liệu này chứa thông tin về giáo viên, mỗi giáo viên có một mã riêng và các thông tin cá nhân của giáo viên.

Bảng thuộc tính:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaGV | Nvarchar(50) | Primary key | Mã của mỗi giáo viên để phân biệt giáo viên này với GV khác |
| 2 | TenGV | Nvarchar(50) | Not null | Tên giáo viên |
| 3 | DiaChi | Nvarchar(100) |  | Địa chỉ liên lạc của giáo viên |
| 4 | SDT | Int |  | Số điện thoại liên lạc của giáo viên |
| 5 | Email | Nvarchar(50) |  | Địa chỉ email liên lạc của giáo viên |
| 6 | MaToMon | Nvarchar(50) | foreign key | Khóa ngoại liên kết bảng tblGiaovien và tblToMon |

#### Bảng tblMonhoc

Bảng dữ liệu này chứa thông tin về môn học, các thông tin bao gồm:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaMonHoc | Nvarchar(50) | Primary key | Mã của môn học, để phân biệt môn học này với môn học khác |
| 2 | TenMonHoc | Nvarchar(50) |  | Tên mỗi môn học |
| 3 | SoTiet | Int |  | Số tiết học của mỗi môn |
| 4 | MaToMon | Nvarchar(50) | foreign key | Khóa ngoại liên kết giữa bảng tblMonHoc và tblToMon. |

#### Bảng tblHocKy

Bảng dữ liệu bao gồm các thông tin sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiễu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaHocKy | Nvarchar(50) | Primary key | Mã học kỳ |
| 2 | NamHoc | Nvarchar(50) |  | Năm học |

#### Bảng tblLop

Bảng này chứa các thông tin về tất cả các lớp. Bao gồm các thông tin sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaLop | Nvarchar(50) | Primary key | Mã của mỗi lớp |
| 2 | TenLop | Nvarchar(50) |  | Tên lớp |
| 3 | Siso | Int |  | Sĩ số của mỗi lớp |
| 4 | MaHocKy | Nvarchar(50) | foreign key | Khóa ngoại liên kết bảng tblLop và bảng tblHocKy |

#### Bảng tblGiangDay

Bảng này chứa các thông tin chung nhất của các bảng, nó hầu như liên kết các bảng lại thành một cấu trúc chặt chẽ.

Bảng bao gồm các thông tin sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Ràng buộc | Mô tả |
| 1 | MaGiangDay | Nvarchar(50) | Primary key | Mã thông tin giảng dạy |
| 2 | MaLop | Nvarchar(50) | foreign key | Khóa ngoại liên kết tới bảng tblLop |
| 3 | MaMon | Nvarchar(50) | foreign key | Khóa ngoại liên kết tới bảng tblMaMon |
| 4 | MaGV | Nvarchar(50) | foreign key | Khóa ngoại liên kết tới bảng tblGiaoVien |
| 5 | Length | Nvarchar(50) |  | Tổng thời gian |
| 6 | batDauGD | Nvarchar(50) |  | Thời gian bắt đầu giảng dạy |
| 7 | ketthucGD | Nvarchar(50) |  | Thời gian kết thúc giảng dạy |

# CHƯƠNG IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

* Nghiên cứu giải thuật di truyền áp dụng phân tích bài toán sắp xếp thời khóa biểu
* Xây dựng một demo mô phỏng giải thuật di truyền trên ngôn ngữ lập trình C#

## Hạn chế, hướng phát triển

### Hạn chế

* Chưa xây dựng thành công chương trình sắp xếp thời khóa biểu

### Hướng phát triển

* Tiếp tục xây dựng chương trình sắp xếp thời khóa biểu cho trường trung học phổ thông

# TÀI LIỆU THAM KHẢO

[1]. Nguyễn Đình Thúc, *Lập trình tiến hóa,* Nhà xuất bản Giáo dục (http://www.box.net/public/pbuo21pcrp)

[2]. Nguyễn Đình Thái, *Trí tuệ nhân tạo - Lập trình tiến hóa* (http://ngocchistar.gocom.vn/ebook/Tin-Hoc-Lap-Trinh-p34704c54641/tri-tue-nhan-taolap-trinh-tien-hoa-i37988)

[3]. Hoàn Kiếm, Lê Hoàng Thái, *Giải thuật di truyền, cách giải tự nhiên các bài toán trên máy tính*, NXB Giáo dục, 2000

[4]. PTS. Nguyễn Thanh Thủy , *Trí tuệ nhân tạo – Các phương pháp giải quyết vấn đề và kỹ thuật xử lý tri thức*, NXBGiáo dục, 1995.

[5]. Nguyễn Quang Hoan**,** *Nhập môn TTNT*, Tài liệu nội bộ

[6]. Stuart J. Russell and Peter Norvig, *Artifical Intelligence: A Modern Approach*, Prentice Hall. Second edition, 2002

….

Một số bài báo-báo cáo

[1]. THUẬT GIẢI DI TRUYỀN VÀ ỨNG DỤNG

[2]. A PSEUDO GENETIC ALGORITHM FOR SOLVINGBEST PATH PROBLEM của S. Behzadi, Ali A.Alesheikh

[3]. Genetic Algorithm Finding the ShortestPath in Networks c?a Bilal Gonen

[4]. Genetic Algorithms in Search, Optimizationand Machine Learning

[5]. A Genetic Algorithm Approach to Solve theShortest Path Problem for Road Maps c?a Sachith Abeysundara, BaladasanGiritharan, Saluka Kodithuwakku.

[6]. Introduction to Genetic Algorithms. Assaf Zaritsky . Ben-Gurion University, Israel www.cs.bgu.ac.il/~assafza

[7]. Introduction to Genetic Algorithms and Evolutionary Computation. Andrew L. Nelson http://s.pudn.com/search\_hot\_en.asp?k=genetic+algorithm#