LỜI CẢM ƠN

Để hoàn thành chương trình học và viết đồ án tốt nghiệp này, em đã nhận được sự hướng dẫn, giúp đỡ và góp ý nhiệt tình của thầy cô trường Đại học Thủy Lợi.

Trước hết, em xin gửi lời cảm ơn sâu sắc nhất đến quý thầy giáo, cô giáo trong trường **Khoa Công nghệ thông tin – Đại học Thủy Lợi,** đã tận tình giảng dạy, trang bị cho em vốn kiến thức, kinh nghiệm quý báu để em có thể đạt được kết quả tốt nhất trong học tập cũng như trong công việc sau khi ra trường.

Em cũng muốn gửi lời cảm ơn tới các thầy cô trong bộ môn “**Các hệ thống thông tin**”. Đặc biệt, em xin gửi lời biết ơn sâu sắc đến thầy giáo **Trần Mạnh Tuấn** - người đã trực tiếp hướng dẫn, giúp đỡ em suốt thời gian từ thực tập đến khi làm đồ án. Trong quá trình hướng dẫn, thầy luôn nhiệt tình chỉ bảo, truyền đạt cho em những kinh nghiệm của người thầy giáo, cô giáo đi trước giúp em luôn có phương hướng xây dựng, phát triển và hoàn thành đề tài đúng thời hạn.

Mặc dù đã có nhiều cố gắng hoàn thiện đồ án bằng tất cả sự nhiệt tình và năng lực của mình, tuy nhiên không thể tránh khỏi những thiếu sót, rất mong nhận được những đóng góp quý báu của quý thầy cô và các bạn.

*Em xin chân thành cảm ơn!*

Hà Nội, ngày 16 tháng 8 năm 2025

Học viên thực hiện

**Nguyễn Trọng Anh**

LỜI CAM ĐOAN

Để hoàn thành đồ án tốt nghiệp đúng thời gian quy định và đáp ứng được yêu cầu đề ra, em đã cố gắng tìm hiểu, học hỏi, tích lũy kiến thức đã học. Em có tham khảo một số tài liệu đã nêu trong phần ***“Tài liệu tham khảo”*** nhưng không sao chép nội dung từ bất kỳ đồ án nào khác.

Em xin cam đoan đồ án là công trình nghiên cứu của cá nhân em nghiên cứu, xây dựng dưới sự hướng dẫn của thầy giáo Trần Mạnh Tuấn. Nội dung lý thuyết trong đồ án có sự tham khảo và sử dụng của một số tài liệu, thông tin được đăng tải trên các tác phẩm, tạp chí và các trang web theo danh mục tài liệu của đồ án.

Các số liệu, chương trình phần mềm và những kết quả trong đồ án là trung thực và chưa được công bố trong bất kỳ một công trình nào khác.

Em xin cam đoan những lời khai trên là đúng, mọi thông tin sai lệch em xin hoàn toàn chịu trách nhiệm trước Hội đồng.

Học viên thực hiện

**Nguyễn Trọng Anh**

NHIỆM VỤ ĐỀ ÁN

Mục tiêu chính của đồ án này là khám phá các kỹ thuật tìm kiếm văn bản khác nhau và nghiên cứu cách áp dụng thuật toán di truyền để giải quyết vấn đề này. Dựa trên nền tảng này, luận văn hướng đến mục tiêu phát triển một công cụ phần mềm hiệu quả và thiết thực để truy xuất văn bản.

Đồ án tập trung nghiên cứu Giải thuật di truyền và bài toán tìm kiếm văn bản, từ đó tìm cách ứng dụng Giải thuật di truyền để giải quyết bài toán này. Trên cơ sở đó xây dựng phần mềm ứng dụng tìm kiếm văn bản một cách hiệu quả và thiết thực. Chương trình đã chạy đúng và đưa ra được kết quả chính xác như mong muốn, hỗ trợ cho việc tìm kiếm một cách nhanh gọn, chính xác.

DANH MỤC CÁC THUẬT NGỮ

Mutation : Đột biến

Crossover probability (Pc) : Xác suất lai ghép

Mutation probability (Pm) : Xác suất đột biến

One-point Crossover : Lai ghép một điểm

Multi-point Crossover : Lai ghép đa điểm

Uniform Crossover : Lai ghép đều

Use case : Trường hợp sử dụng

Knowledge-based systems : Hệ thống tri thức cơ sở

Split : Chia nhỏ

Genetic algorithm (GA) : Giải thuật di truyền

Time Table problem : Bài toán thời khóa biểu

Genome : Gen

Derivatives : Đạo hàm

Evaluation (eval) : Hàm lượng giá

Pop – size : Kích thước quần thể

Crossover : Lai ghép

DANH MỤC HÌNH VẼ

[Hình 1.1. Sơ đồ tổng quan của giải thuật di truyền 8](#_Toc206623699)

[Hình 1.2. Bánh xe Roulette 10](#_Toc206623700)

[Hình 1.3. Đồ thị của hàm f 17](#_Toc206623701)

[Hình 2.1. Mô hình cá thể trong lịch lớp 29](#_Toc206623702)

[Hình 2.2. Minh họa cho mô hình NST trong một cơ sở 32](#_Toc206623703)

[Hình 2.3. Mô hình cá thể trong lịch cơ sở. 33](#_Toc206623704)

[Hình 3.1. Giao diện thiết kế demo chính 36](#_Toc206623705)

[Hình 3.2. Văn bản tìm kiếm S 37](#_Toc206623706)

[Hình 3.3. Văn bản mẫu Sm 37](#_Toc206623707)

[Hình 3.4. Kết quả GA 38](#_Toc206623708)

[Hình 3.5. Đời 1của quá trình tiến hóa 38](#_Toc206623709)

[Hình 3.6. Đời 100 của quá trình tiến hóa 39](#_Toc206623710)

[Hình 3.7. Form về sinh viên thực hiện 39](#_Toc206623711)

[Hình 3.8. Form về tài liệu tham khảo 39](#_Toc206623712)

[Hình 3.9. Form thông tin về chương trình 39](#_Toc206623713)

[Hình 3.10. Form hướng dẫn cách chạy chương trình 39](#_Toc206623714)

[Hình 3.11. Form lập lịch 40](#_Toc206623715)

[Hình 3.12. Sơ đồ dữ liệu đầu vào 41](#_Toc206623716)

[Hình 3.13. Thời khóa biểu các lớp 41](#_Toc206623717)

[Hình 3.14. Thời khóa biểu của giảng viên 41](#_Toc206623718)

[Hình 3.15. Form tìm kiếm thời khóa biểu 41](#_Toc206623719)

DANH MỤC BẢNG

[Bảng 1.1. Kết quả giá trị thích nghi 21](#_Toc206623757)

[Bảng 1.2. Bảng mô tả bánh xe Roulette 22](#_Toc206623758)

[Bảng 1.3. Bảng mô tả chọn NST lai ghép 23](#_Toc206623759)

[Bảng 1.4. Bảng mô tả NST đột biến 24](#_Toc206623760)

MỤC LỤC

[CHƯƠNG 1.GIỚI THIỆU VỀ GIẢI THUẬT DI TRUYỀN 1](#_Toc206623922)

[1.1. Tổng quan về giải thuật di truyền 1](#_Toc206623923)

[1.1.1. Giới thiệu 1](#_Toc206623924)

[1.1.2. Mục tiêu nghiên cứu giải thuật di truyền 2](#_Toc206623925)

[1.1.3. Tính chất quan trọng của thuật toán di truyền 3](#_Toc206623926)

[1.1.4. Sự khác biệt của giải thuật di truyền so với các giải thuật khác 4](#_Toc206623927)

[1.2. Giải thuật di truyền cổ điển (GA) 4](#_Toc206623928)

[1.2.1. Giới thiệu 5](#_Toc206623929)

[1.2.2. Các toán tử di truyền 9](#_Toc206623930)

[1.2.3. Các bước giải một bài toán bằng giải thuật di truyền 14](#_Toc206623931)

[1.2.4. Các nguyên lý trong giải thuật di truyền 15](#_Toc206623932)

[1.2.5. Ví dụ GA 17](#_Toc206623933)

[CHƯƠNG 2.ỨNG DỤNG THUẬT TOÁN DI TRUYỀN TRONG SẮP XẾP THỜI KHÓA BIỂU 26](#_Toc206623934)

[2.1. Bài toán sắp xếp thời khóa biểu 26](#_Toc206623935)

[2.1.1. Phát biểu bài toán 26](#_Toc206623936)

[2.1.2. Phân tích bài toán 26](#_Toc206623937)

[2.2. Áp dụng giải thuật di truyền 28](#_Toc206623938)

[2.3. Điểm dừng của thuật toán 35](#_Toc206623939)

[CHƯƠNG 3.XÂY DỰNG CHƯƠNG TRÌNH VÀ KẾT QUẢ THỬ NGHIỆM 36](#_Toc206623940)

[3.1. Thiết kế giao diện chương trình 36](#_Toc206623941)

[3.1.1. Giao diện chính 36](#_Toc206623942)

[3.1.2. Giao diện nhập tham số cho bài toán 37](#_Toc206623943)

[3.1.3. Giao diện hiển thị kết quả khi thực hiện chương trình 37](#_Toc206623944)

[3.1.4. Dữ liệu đầu vào 37](#_Toc206623945)

[3.2. Một số kết quả thử nghiệm của chương trình 37](#_Toc206623946)

[3.2.1. Kết quả thử nghiệm tìm kiếm bằng giải thuật di truyền 37](#_Toc206623947)

[3.2.2. Một số hình ảnh của chương trình 39](#_Toc206623948)

[3.3. Ứng dụng bài toán sắp xếp thời khóa biểu 40](#_Toc206623949)

[3.3.1. Giao diện mô phỏng dữ liệu đầu vào 41](#_Toc206623950)

[3.3.2. Ứng dụng tìm kiếm thời khóa biểu 41](#_Toc206623951)

MỞ ĐẦU

1. **Đặt vấn đề**

Trong thế giới ngày nay, việc sắp xếp thời khóa biểu đã trở thành một yếu tố thiết yếu trong quản lý thời gian và nâng cao hiệu quả học tập cũng như làm việc. Sự phát triển của công nghệ thông tin đã dẫn đến việc sử dụng các ứng dụng và phần mềm hỗ trợ trong việc lập kế hoạch, giúp người dùng dễ dàng tổ chức công việc và cam kết của mình. Mỗi ngày, hàng triệu người dùng phải đối mặt với việc cân bằng giữa học tập, công việc và các hoạt động cá nhân. Việc có một thời khóa biểu hợp lý không chỉ giúp tiết kiệm thời gian mà còn giảm căng thẳng, nâng cao năng suất và tạo điều kiện cho sự phát triển cá nhân.

Điều này mang lại nhiều lợi ích, chẳng hạn như khả năng quản lý thời gian hiệu quả và tối ưu hóa các hoạt động hàng ngày. Tuy nhiên, nó cũng đặt ra những thách thức lớn. Một trong những vấn đề nổi bật là việc sắp xếp và điều chỉnh các lịch trình. Khi khối lượng công việc và hoạt động ngày càng tăng, việc duy trì một thời khóa biểu hợp lý trở nên phức tạp hơn. Người dùng thường phải đối mặt với việc xung đột lịch trình, dẫn đến tình trạng quá tải và căng thẳng. Việc không thể tìm ra thời gian phù hợp cho các nhiệm vụ quan trọng có thể gây ra sự chậm trễ và ảnh hưởng đến hiệu suất làm việc và học tập.

Thêm vào đó, các phương pháp sắp xếp thời khóa biểu truyền thống thường không đáp ứng được nhu cầu của người dùng. Chẳng hạn, nhiều ứng dụng lập kế hoạch chỉ cho phép người dùng nhập các sự kiện hoặc nhiệm vụ một cách đơn giản, mà không cung cấp các tùy chọn linh hoạt để điều chỉnh thời gian hoặc thứ tự ưu tiên. Điều này có thể khiến người dùng cảm thấy bối rối, đặc biệt khi họ cần điều chỉnh lịch trình vào phút chót hoặc khi phải cân nhắc giữa nhiều hoạt động khác nhau. Việc không có một hệ thống hỗ trợ thông minh có thể dẫn đến việc lãng phí thời gian và giảm hiệu quả trong việc quản lý thời gian.

Bên cạnh đó, khi khối lượng công việc và hoạt động tăng lên, vấn đề về chất lượng thời gian và hiệu quả sắp xếp cũng trở nên nghiêm trọng hơn. Không phải tất cả các nhiệm vụ đều có mức độ ưu tiên giống nhau, và việc xác định các hoạt động quan trọng cần hoàn thành trước trở nên khó khăn hơn. Điều này khiến cho việc lập thời khóa biểu không chỉ là một nhiệm vụ đơn giản mà còn yêu cầu người dùng phải có kỹ năng phân tích và đánh giá mức độ quan trọng của từng nhiệm vụ để đảm bảo rằng thời gian được sử dụng một cách tối ưu.

Mục tiêu nghiên cứu là phát triển một phương pháp sắp xếp thời khóa biểu mới mẻ và hiệu quả hơn. Việc áp dụng các thuật toán tối ưu hóa hiện đại, như thuật toán di truyền, đại diện cho một hướng nghiên cứu tương đối mới nhằm giải quyết vấn đề lập kế hoạch thời gian, với hy vọng rằng nó sẽ mang lại kết quả tốt hơn so với các phương pháp truyền thống hiện có. Qua đó, nghiên cứu này không chỉ nhằm cải thiện khả năng quản lý thời gian mà còn hướng tới việc tăng cường trải nghiệm người dùng trong việc tổ chức và sắp xếp các hoạt động hàng ngày, giúp họ đạt được sự cân bằng tốt hơn giữa công việc và cuộc sống.

Với mục tiêu này, chủ đề nghiên cứu được chọn là "Sắp xếp thời khóa biểu bằng thuật toán di truyền". Phương pháp này đại diện cho một hướng tương đối riêng để giải quyết vấn đề, với hy vọng rằng nó sẽ mang lại kết quả hiệu quả hơn so với các phương pháp sắp xếp khác.

Với mục tiêu đó, em lựa chọn đề tài nghiên cứu của đồ án là ***“Ứng dụng giải thuật di truyền trong sắp xếp thời khóa biểu”.*** Đây là hướng tiếp cận riêng đối với bài toán này, hy vọng rằng kết quả đạt được sẽ có hiệu quả đáng kể so với các phương pháp sắp xếp thời khóa biểu khác.

1. **Mục đích của đề án**

Mục đích của đề án là: nghiên cứu giải thuật di truyền và tìm cách ứng dụng giải thuật di truyền để sắp xếp thời khóa biểu, trên cơ sở đó xây dựng phần mềm ứng dụng sắp xếp thời khóa biểu một cách hiệu quả và thiết thực.

1. **Nội dung của đề án**

Đề án tập trung vào bài toán sắp xếp thời khóa biểu theo hướng tiếp cận sau: Tìm các khoảng thời gian phù hợp trong lịch trình có thể chứa các nhiệm vụ hoặc hoạt động giống hoặc gần giống với các yêu cầu lập kế hoạch (xuất hiện gần giống trong trường hợp người dùng không thể xác định thời gian cụ thể cho các nhiệm vụ). Mục tiêu là tối ưu hóa việc sử dụng thời gian và đảm bảo rằng các hoạt động quan trọng được thực hiện đúng lúc, từ đó nâng cao hiệu suất và chất lượng cuộc sống. Trên cơ sở đó, nội dung của đề tài gồm ba chương:

* *Chương 1:* Nghiên cứu khái quát về giải thuật di truyền.
* *Chương 2:* Xây dựng và phát biểu bài toán, đề xuất phương pháp sử dụng giải thuật di truyền trong sắp xếp thời khóa biểu.
* *Chương 3:* Kết quả thử nghiệm và phát triển phần mềm ứng dụng.

1. **Phương pháp nghiên cứu**

Nghiên cứu tài liệu, đề xuất giải pháp và lập trình thử nghiệm. Đề án đã bước đầu đề xuất phương pháp ứng dụng giải thuật di truyền vào việc giải quyết bài toán sắp xếp thời khóa biểu, các chương trình thử nghiệm đã minh chứng rằng hướng tiếp cận là đúng đắn và có hiệu quả. Đặc biệt, chương trình đã chỉ ra được các khoảng thời gian phù hợp để thực hiện các nhiệm vụ giống hoặc gần giống với yêu cầu lập kế hoạch (trong trường hợp người dùng không thể xác định rõ ràng thời gian cụ thể) cần tìm trong thời gian cho phép, từ đó giúp nâng cao hiệu suất quản lý thời gian.

# 

# GIỚI THIỆU VỀ GIẢI THUẬT DI TRUYỀN

*Giải thuật di truyền (Genetic Algorithm - GA) là một phương pháp tối ưu hóa và lập kế hoạch dựa trên nguyên lý của tiến hóa tự nhiên. GA mô phỏng quá trình chọn lọc tự nhiên, nơi các lịch trình hiệu quả hơn có khả năng được tối ưu hóa và cải thiện trong quá trình lặp lại. Bằng cách áp dụng các nguyên tắc này, GA giúp tìm ra các cấu trúc thời khóa biểu tối ưu, đảm bảo rằng các nhiệm vụ được sắp xếp một cách hợp lý để tối đa hóa hiệu suất và giảm thiểu xung đột lịch trình. Phần này sẽ tìm hiểu cơ bản về giải thuật di truyền, những khái niệm cơ bản, nguyên lý hoạt động và một ví dụ áp dụng giải thuật di truyền.*

## Tổng quan về giải thuật di truyền

### Giới thiệu

Giải thuật di truyền được lập dựa trên cơ sở lý thuyết Darwin và được giới thiệu lần đầu tiên bởi Holland (1975), sau đó là Goldberg (1989). Sau đó đến năm 1992 Michalewicz đã phát triển và hoàn thành phương pháp này. Từ đó thuật giải di truyền đã được áp dụng trong các lĩnh vực khác nhau. Thuật giải di truyền được ứng dụng đầu tiên trong hai lĩnh vực chính đó là: *tối ưu hóa và học tập của máy*. Trong lĩnh vức tối ưu hóa thuật toán di truyền được phát triển nhanh chóng và ứng dụng trong nhiều lĩnh vực khác nhau như: tối ưu hàm, xử lý ảnh, bài toán hành trình của người bán hàng, nhận dạng hệ thống và điều khiển…

Giải thuật di truyền áp dụng nguyên lý tiến hóa tự nhiên để giải quyết các bài toán tối ưu trong thực tế. Quá trình này bắt đầu từ một tập hợp các lời giải khả thi, qua nhiều bước tiến hóa, hình thành các tập hợp mới với các lời giải tốt hơn. Cuối cùng, phương pháp này sẽ tìm ra lời giải gần tối ưu cho bài toán.

Giải thuật di truyền, giống như các thuật toán tiến hóa khác, được hình thành dựa trên quan niệm rằng tiến hóa tự nhiên là quá trình hoàn hảo và hợp lý nhất, tự nó mang tính tối ưu. Quan niệm này có thể coi là một tiên đề không thể chứng minh nhưng phù hợp với thực tế khách quan. Tính tối ưu trong tiến hóa thể hiện ở chỗ, thế hệ sau thường tốt hơn (phát triển và hoàn thiện hơn) so với thế hệ trước nhờ vào tính kế thừa và sự đấu tranh sinh tồn. Tiến hóa tự nhiên được duy trì thông qua hai quá trình cơ bản: sinh sản và chọn lọc tự nhiên. Trong suốt quá trình này, các thế hệ mới liên tục được sinh ra để thay thế thế hệ cũ. Những cá thể phát triển tốt hơn và thích ứng tốt với môi trường sẽ tồn tại, trong khi những cá thể không thích ứng sẽ bị loại bỏ. Sự thay đổi trong môi trường là động lực thúc đẩy tiến hóa, và ngược lại, quá trình tiến hóa cũng góp phần làm thay đổi môi trường.

Giải thuật di truyền hoạt động trên các quần thể bao gồm nhiều cá thể khác nhau. Trong quá trình tiến hóa, các cá thể mới được tạo ra thông qua sự lai ghép giữa các thế hệ cha mẹ. Một cá thể mới có thể kế thừa các đặc điểm di truyền từ cha mẹ, nhưng cũng có thể xuất hiện những đặc điểm hoàn toàn mới do quá trình đột biến. Cả di truyền và đột biến đều đóng vai trò quan trọng trong tiến trình tiến hóa, mặc dù xác suất xảy ra của đột biến thường rất thấp so với hiện tượng di truyền. Di truyền đảm bảo rằng những đặc điểm có lợi sẽ được truyền lại cho thế hệ sau, trong khi đột biến tạo ra sự đa dạng di truyền, cung cấp nguyên liệu cho quá trình chọn lọc tự nhiên. Hai cơ chế này phối hợp chặt chẽ, giúp quần thể phát triển và thích nghi với môi trường một cách hiệu quả hơn theo thời gian.

Trong giải thuật di truyền, mỗi cá thể được mã hóa thông qua một cấu trúc dữ liệu đặc biệt, mô tả cấu trúc genome của cá thể đó, thường được gọi là nhiễm sắc thể. Mỗi nhiễm sắc thể được cấu tạo từ các đơn vị cơ bản được gọi là genome. Các genome này có thể coi là các gen, mỗi gen đại diện cho một đặc điểm hoặc thuộc tính cụ thể của cá thể. Mỗi nhiễm sắc thể sẽ biểu diễn một lời giải khả thi cho bài toán mà chúng ta đang cố gắng giải quyết. Cấu trúc của nhiễm sắc thể có thể khác nhau tùy thuộc vào cách mà bài toán được định nghĩa, nhưng mục tiêu chung là tạo ra một tập hợp các nhiễm sắc thể đủ đa dạng để cho phép quá trình tiến hóa tìm ra giải pháp tối ưu thông qua các phép lai ghép, đột biến và chọn lọc tự nhiên. Sự kết hợp này giúp tối ưu hóa dần dần các lời giải, với hy vọng rằng một hoặc nhiều cá thể trong quần thể sẽ dẫn đến lời giải gần tối ưu cho bài toán.

### Mục tiêu nghiên cứu giải thuật di truyền

Trừu tượng hóa và diễn đạt chính xác về các quá trình thích nghi trong thế giới tự nhiên.

Thiết kế những phần mềm về hệ thống nhân tạo nhằm duy trì các cơ chế quan trọng của hệ thống tự nhiên.

Những mục tiêu này đã dẫn đến những khám phá quan trọng trong hệ thống khoa học tự nhiên lẫn nhân tạo.

GA ra đời và phát triển dựa trên quá trình tiến hóa trong tự nhiên và đã được ứng dụng thành công trong nhiều lĩnh vực nhất là tối ưu hóa và máy học.

### Tính chất quan trọng của thuật toán di truyền

Giải thuật di truyền sử dụng yếu tố ngẫu nhiên để tìm kiếm giải pháp tối ưu cho các vấn đề phức tạp. Tuy nhiên, yếu tố ngẫu nhiên này không hoàn toàn ngẫu nhiên mà được hướng dẫn bởi giá trị của hàm thích nghi. Chính hàm thích nghi đóng vai trò như một ánh sáng dẫn đường cho GA, giúp nó xác định và tìm ra lời giải tối ưu trong vô vàn các lời giải khả thi. Nhờ vào hàm thích nghi, GA có thể tập trung vào những hướng đi tiềm năng, tăng cường khả năng đạt được kết quả tốt nhất trong quá trình tối ưu hóa.

Vấn đề phù hợp nhất cho Giải thuật di truyền là tìm kiếm các điều kiện tối ưu. Khái niệm tối ưu ở đây không nhất thiết phải là tuyệt đối; nó có thể chỉ là tương đối, tùy thuộc vào hoàn cảnh và thời gian cho phép. GA tập trung vào việc xác định các giải pháp tốt nhất có thể trong các điều kiện cụ thể, giúp đạt được hiệu quả tối đa trong bối cảnh đã đặt ra.

Một trong những bước quan trọng và thách thức nhất trong giải thuật di truyền là xác định hàm số thích nghi. Hàm số này cần phải có mối liên hệ trực tiếp với vấn đề mà chúng ta đang cố gắng giải quyết. Việc xây dựng một hàm số thích nghi phù hợp sẽ giúp GA đánh giá và phân loại các giải pháp, từ đó dẫn đến những lựa chọn hiệu quả hơn trong quá trình tối ưu hóa.

Giải thuật di truyền và mạng nơron nhân tạo đều thuộc lĩnh vực khoa học trí tuệ nhân tạo, nhưng chúng có cách tiếp cận khác nhau. GA dựa vào nguyên lý tiến hóa và xem xét vấn đề ở cấp độ của gen và nhiễm sắc thể, trong khi mạng nơron nhân tạo dựa trên kinh nghiệm và phương pháp giải quyết vấn đề mà bộ não con người thường sử dụng. Sự khác biệt này khiến hai phương pháp này có những ứng dụng và ưu điểm riêng trong việc giải quyết các bài toán phức tạp.

### Sự khác biệt của giải thuật di truyền so với các giải thuật khác

Giải thuật di truyền khác biệt so với các phương pháp tối ưu hóa thông thường và các giải thuật tìm kiếm khác nhờ vào những đặc điểm sau:

* GA làm việc với việc mã hóa một bộ các thông số, thay vì xử lý trực tiếp các thông số đó.
* GA mang tính chất ngẫu nhiên trong quá trình tìm kiếm.
* GA thực hiện tìm kiếm từ nhiều điểm trong quần thể, chứ không chỉ từ một điểm duy nhất.
* GA tận dụng thông tin về hàm mục tiêu, không phụ thuộc vào các đạo hàm hay những tri thức bổ sung khác.
* GA áp dụng các quy tắc chuyển đổi dựa trên xác suất, thay vì các quy tắc chuyển đổi cố định.
* GA yêu cầu một tập hợp các thông số tự nhiên của bài toán tối ưu để mã hóa thành các chuỗi có chiều dài hữu hạn, dựa trên một số ký tự giới hạn.

## Giải thuật di truyền cổ điển (GA)

Trong tự nhiên, để tồn tại và phát triển, mỗi cá thể cần phải thích nghi với môi trường xung quanh. Những cá thể có khả năng thích nghi tốt hơn sẽ sống sót, trong khi những cá thể kém thích nghi sẽ bị loại bỏ. Bên trong mỗi cá thể, các gen được tổ chức thành chuỗi, gọi là nhiễm sắc thể (NST). Mỗi NST đặc trưng cho một loài và ảnh hưởng đến khả năng sống sót của cá thể đó. Vì môi trường tự nhiên luôn thay đổi, cấu trúc của NST cũng cần điều chỉnh để phù hợp hơn, khiến cho thế hệ sau thường có khả năng thích nghi tốt hơn thế hệ trước. Việc thay đổi này diễn ra thông qua sự trao đổi thông tin ngẫu nhiên, cả với môi trường bên ngoài lẫn giữa các NST với nhau.

Dựa trên ý tưởng đó, các nhà khoa học đã phát triển giải thuật di truyền, lấy cảm hứng từ cơ sở chọn lọc tự nhiên và quy luật tiến hóa. Giải thuật này sử dụng các thuật ngữ từ di truyền học như: lai ghép, đột biến, nhiễm sắc thể (NST), và cá thể. Trong mô hình đơn giản mà chúng ta xem xét, mỗi cá thể được đặc trưng bởi một NST duy nhất. Các NST được phân chia thành các gen, sắp xếp theo một chuỗi tuyến tính. Mỗi cá thể (hay NST) đại diện cho một lời giải khả thi cho bài toán. Quá trình tiến hóa trên tập hợp các NST tương đương với việc tìm kiếm lời giải trong không gian giải pháp của bài toán. Mục tiêu của quá trình tìm kiếm này bao gồm:

* Khai thác những lời giải tốt nhất.
* Khảo sát toàn bộ không gian tìm kiếm.

### Giới thiệu

Giải thuật di truyền cổ điển là những kỹ thuật tìm kiếm và tối ưu hóa giải pháp cho các vấn đề, được lấy cảm hứng từ quá trình thích nghi và tiến hóa của các quần thể sinh học theo học thuyết của Darwin. Mục tiêu của GA không phải là tìm ra lời giải chính xác tối ưu, mà là cung cấp các lời giải tương đối tối ưu.

* Các thành phần của GA

Một thuật giải di truyền, giải một bài toán được cho phải có năm thành phần sau:

* Một cấu trúc dữ liệu **I** biểu diễn không gian lời giải của bài toán.
* Phương pháp khởi tạo quần thể ban đầu **P(0).**
* Hàm định nghĩa độ thích nghi **eval()** đóng vai trò môi trường.
* Các phép toán di truyền như đã mô phỏng ở trên.
* Các tham số thuật giải di truyền sử dụng (kích thước quần thể, xác suất lai, đột biến …)

Ban đầu, chúng ta sẽ tạo ra một số lượng lớn cá thể với gen ngẫu nhiên, tức là phát sinh một tập hợp các chuỗi bit ngẫu nhiên. Tập hợp này được gọi là quần thể ban đầu (initial population). Sau đó, dựa vào một hàm nhất định, chúng ta sẽ xác định một giá trị gọi là độ thích nghi (fitness). Để dễ hiểu, giá trị này có thể được coi là độ "tốt" của lời giải, giống như độ cao trong quá trình tìm kiếm theo kiểu leo đồi. Do được phát sinh ngẫu nhiên, độ "tốt" của lời giải và tính thích nghi của các cá thể trong quần thể ban đầu là không xác định.

Để nâng cao 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 giúp đảm bảo rằng độ thích nghi của thế hệ sau luôn được duy trì ở mức hợp lý. Các cá thể được chọn 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 tác 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 (tương tự như trường hợp leo đồi, các người leo đồi tập trung dồn vào một góc trên vùng đất). 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.

* Cấu trúc của GA

Trong GA các cá thể (hay còn gọi là các NST) được mã hóa bởi các chuỗi nhị phân, mỗi vị trí trên chuỗi nhị phân chỉ nhận một trong hai giá trị “0” hoặc “1”. Một NST trong GA có dạng như sau:

1 0 1 1 0 0 1 0 0 1

GA cổ điển được J. H Holland giới thiệu để giải bài toán tối ưu:

max {f (x) /x A},

Trong đó A là một miền trong không gian n-chiều, f (x) >0 với mọi x  A.

Cấu trúc của GA cổ điển như sau:

Procedure Genetic\_Algorithm;

Begin

t  0;

Khởi tạo P(t);

Tính độ thích nghi cho các cá thể thuộc P(t);

While (điều kiện dừng chưa thỏa) do

t  t+1;

Tái sinh P’(t) từ P(t-1);

Lai Q(t) từ P(t-1);

Đột biến R(t) từ P(t-1);

Chọn lọc P(t) từ P(t-1)  Q(t)  R(t)  P’(t);

End;

End;

*Trong đó:*

t: vị trí của thế hệ.

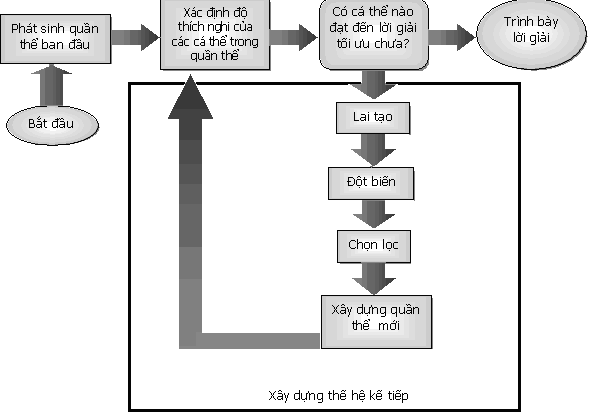
P(t): Quần thể ban đầu/ quần thể sau khi chọn lọc.

P’(t): Quần thể được tái sinh từ thế hệ trước.

Q(t): Quần thể được lai ghép từ thế hệ trước.

R(t): Quần thể được đột biến từ thế hệ trước.

* **Sơ đồ hoạt động của giải thuật di truyền:**



Hình 1.1. Sơ đồ tổng quan của giải thuật di truyền

***Giải thích:***

Bắt đầu di truyền ngẫu nhiên của quần thể của n nhiễm sắc thể (thay thế các lời giải cho bài toán). Ở bước lặp t, thuật giải di truyền duy trì một quần thể các lời giải (các NST, các vectơ) P(t) = {}. Mỗi lời giải  được đánh giá để biết được “*độ thích nghi*” của nó. Rồi một quần thể mới (ở lần lặp thứ t+1) được hình thành bằng cách chọn lại những cá thể thích nghi nhất. Một số cá thể của quần thể này trải qua những phép biến đổi nhờ *lai tạo* (phép lai) và *đột biến* (phép đột biến) hình thành nên những lơi giải mới.

Tính độ thích nghi cho các cá thể thuộc quần thể ta dùng hàm đánh giá để đánh giá xem một cá thể có tốt hay không. Một cá thể càng tốt tức là độ thích nghi của nó càng cao và tiến tới trở thành một lời giải đúng của bài toán. Việc thiết kế một hàm đánh giá tốt là rất quan trọng trong thuật toán di truyền. Một hàm đánh giá không chính xác có thể làm mất đi các cá thể tốt trong quần thể.

* **Điều kiện kết thúc**

Thoát ra quá trình tiến hóa quần thể, dựa vào bài toán mà có các cách kết thúc vấn đề khác nhau một khi đạt đến mức yêu cầu. Điều kiện kết thúc vòng lặp có thể là một số thế hệ đủ lớn nào đó, hoặc độ thích nghi của các cá thể tốt nhất trong các thế hệ kế tiếp nhau khác nhau không đáng kể. Khi thuật toán dừng, cá thể tốt nhất trong thế hệ cuối cùng được chọn làm nghiệm cần tìm. 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.

### Các toán tử di truyền

Trong thuật giải di truyền, các cá thể mới liên tục được sinh ra trong quá trình tiến hoá nhờ sự lai ghép ở thế hệ cha-mẹ. Một cá thể mới có thể mang những tính trạng của cha-mẹ (di truyền), cũng có thể mang những tính trạng hoàn toàn mới (đột biến). Di truyền và đột biến là hai cơ chế có vai trò quan trọng như nhau trong tiến trình tiến hoá, dù rằng đột biến xảy ra với xác xuất nhỏ hơn rất nhiều so với hiện tượng di truyền. Các thuật toán tiến hoá, tuy có những điểm khác biệt, nhưng đều mô phỏng bố toán tử cơ bản: ***Tái sinh****,* ***Chọn lọc****,* ***lai ghép****,* ***đột biến***.Nhưng trong thực tế thì không có sinh vật nào có thể tái sinh nên khi ta thực hiện quá trình tiến hóa thì chúng ta có thể bỏ qua toán tử tái sinh này.

*Toán tử tái sinh (Reproduction)*

Tái sinh là quá trình chọn quần thể mới thỏa phân bố xác suất dựa trên độ thích nghi. Độ thích nghi là một hàm gán một giá trị thực cho cá thể trong quần thể. Các cá thể có độ thích nghi lớn sẽ có nhiều bản sao trong thế hệ mới. Hàm thích nghi có thể không tuyến tính, không đạo hàm, không liên tục bởi vì thuật giải di truyền chỉ cần kên kết hàm thích nghi với các chuỗi số.

Quá trình này được thực hiện dựa trên bánh xe quay Roulette (bánh xe xổ số) với các rãnh được định kích thước theo độ thích nghi.

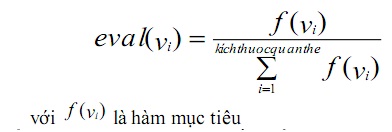
Một bánh xe Roulette minh họa:



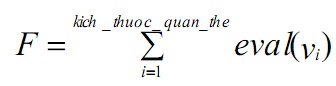
Hình 1.2. Bánh xe Roulette

Bánh xe Roulette dược xây dưng như sau (giả định rằng các độ thích nghi đều dương, trong trường hợp còn lại thì ta có thể dùng một vài phép biến đổi tương ứng để định lại tỷ lệ sao cho các độ thích nghi đều dương):

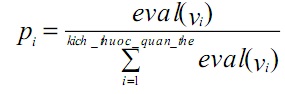
* Tính độ thích nghi **eval(vi)** của mỗi NST vi (i=1..kích thước quần thể).



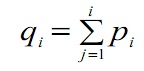
* Tìm tổng giá trị thích nghi của quần thể:



* Tính xác suất chọn Pi cho mỗi NST:



* Tính vị trí xác suất qi của mỗi NST:



*Toán tử chọn lọc*

Toán tử chọn lọc là một quá trình loại bỏ các NST kém thích nghi trong quần thể. Có các toán tử chọn lọc sau:

* **Toán tử chọn lọc tỷ lệ:** Được sử dụng thường xuyên nhất trong GA. Xác suất lựa chọn của mỗi cá thể tỷ lệ thuận với giá trị độ thích hợp của nó, được tính theo công thức:

Pi = f (vi) /F

(i = 1..pop-size – kích cỡ của quần thể) gọi là xác suất chọn cho mỗi nhiễm sắc thể vi.

Trong đó: f (vi) là hàm thích nghi của mỗi cá thể vi

F là tổng của các giá trị thích nghi của quần thể.

Việc chọn lọc cá thể nào phụ thuộc vào vị trí xác suất qi của mỗi nhiễm sắc thể vi được tính như sau: 

Tiến trình chọn lọc được thực hiện bằng cách quay bánh xe ru lét pop- size lần; mỗi lần chọn một nhiễm sắc thể từ quần thể hiện hành vào quần thể mới theo cách sau:

- Phát sinh ngẫu nhiên một số r trong khoảng [0..1].

- Nếu r < qi thì chọn nhiễm sắc thể đầu tiên (vi ); ngược lại thì chọn nhiễm sắc thể thứ i , vi ( 2 ) sao cho  .

Hiển nhiên, có thể sẽ có một só nhiễm sắc thể được chọn nhiều lần. Điều này phù hợp với lý thuyết sơ đồ (Nguyễn Đình Thúc, [6]): các nhiễm sắc thể tốt nhất có nhiều bản sao hơn, các nhiễm sắc thể trung bình không thay đổi, các nhiễm sắc thể kém nhất thì chết đi.

* **Toán tử chọn lọc cạnh tranh:** Mỗi lần chọn lọc ta tiến hành chọn ngẫu nhiên t cá thể từ quần thể hiện tại. Bản sao của cá thể tốt nhất trong t cá thể kể trên được sao chép vào quần thể bố mẹ. Tiến hành N lần chọn như vậy ta thu được quần thể bố mẹ. Giá trị t được gọi là kích cỡ cạnh tranh.
* **Toán tử chọn lọc xếp hạng:** Các cá thể của quần thể hiện tại được sắp xếp theo thứ tự giảm dần của giá trị độ thích nghi. Cá thể tốt nhất được xếp thứ nhất và cá thể tồi nhất xếp cuối cùng.

*Toán tử lai ghép*

Toán tử lai ghép là quá trình tạo NST mới trên cơ sở các NST cha - mẹ bằng cách ghép một đoạn trên NST cha mẹ với nhau. Toán tử lai ghép được gán với một xác suất pc. Quá trình được mô tả như sau:

Chọn ngẫu nhiên một cặp NST (để làm cha mẹ) trong quần thể. Giả sử, NST cha mẹ có cùng độ dài m.

Tạo một số ngẫu nhiên trong khoảng từ 1 đến m-1 (gọi là điểm lai ghép). Điểm lai ghép chia NST cha mẹ thành hai chuỗi con có độ dài m1, m2.

***Ví dụ:***

Cha: **101101100**

Mẹ: 000011100

Thì việc trao đổi chéo các NST sau gen thứ 5 sẽ tạo ra hai con:

Con 1: **1011**11100

Con 2: 0000**01100**

Có một số dạng toán tử lai ghép như:

* Lai ghép một điểm (One-point Crossover)

Lai ghép một điểm là loại lai ghép đơn giản nhất, được sử dụng cả trong GA mã hoá nhị phân lẫn GA mã hoá số thực. Với cặp cha mẹ X, Y là các vectơ m chiều như ký hiệu trên, toán tử lai ghép 1 điểm chọn ngẫu nhiên một vị trí k (1  k  m) rồi sinh ra 2 cá thể con theo công thức

X’ = (x1,..., xk, yk+1,..., ym )

Y’ = (y1,..., yk, xk+1,..., xm )

* Lai ghép đa điểm (Multi-point Crossover)

Lai ghép nhiều điểm được thực hiện tương tự như lai ghép một điểm. Toán tử lai ghép đa điểm được mô tả như sau:

Chọn ngẫu nhiên k điểm j1,..., jk (1 <= j1 < j2 <... < jk < m), lai ghép đa điểm tạo ra cặp con (X', Y') bằng cách đánh số các đoạn [jt, jt+1] từ 0 trở đi, sau đó :

* x'i lấy bằng xi tại những đoạn có số hiệu chẵn và bằng yi tại những đoạn có số hiệu lẻ.
* y'i lấy bằng xi tại những đoạn có số hiệu lẻ và bằng yi tại những đoạn có số hiệu chẵn.
* Lai ghép đều hay lai ghép mặt nạ (Uniform Crossover)

Với hai cá thể cha mẹ đã chọn P1, P2 trước hết phát sinh một chuỗi nhị phân ngẫu nhiên cũng có độ dài L gọi là chuỗi mặt nạ. Sau đó các con được tạo ra dựa trên chuỗi mặt nạ này để quyết định lấy thành phần của cá thể cha hay mẹ. Chẳng hạn gen thứ i của cá thể con C1 được lấy là gen thứ i của P1 nếu bit mặt nạ tương ứng là 1 và lấy gen thứ i của P2 nếu bit mặt nạ là 0. Cá thể con C2 được tạo ngược lại.

Trong lai ghép đều, ta chọn ngẫu nhiên k vị trí 1 < i1 < i2 <... < ik < m.

Các cá thể con X', Y' được lập như sau:





*Toán tử đột biến*

Đột biến là hiện tượng NST con mang một số đặc tính không có trong mã di truyền của cha- mẹ. Toán tử đột biến được gán xác suất pm (nhỏ hơn nhiều so với xuất suất lai ghép pc). Điều này được suy diễn bởi trong tự nhiên, đột biến gen thường rất ít xảy ra.

Theo nghiên cứu của Keneth De Jong thì:

* Tỷ lệ lại ghép trung bình: 0.6
* Tỷ lệ đột biến: 0.001
* Tỷ lệ tái sinh: 0.399

Mỗi gen trong tất cả các NST có cơ hội đột biến như nhau, nghĩa là đối với mỗi NST trong quần thể hiện hành (sau khi lai) và đối với mỗi gen trong NST, quá trình đột biến được thực hiện như sau:

* Chọn ngẫu nhiên một NST trong quần thể.
* Tạo một số ngẫu nhiên k trong khoảng từ 1 tới m, 1 ≤ k ≤ m.
* Thay đổi bít thứ k. Đưa nhiễm sắc thể này vào quần thể để tham gia quá trình tiến hóa ở thế hệ tiếp theo.

Đột biến làm tăng khả năng tìm được lời giải gần tối ưu của thuật toán di truyền. Đột biến không được dùng thường xuyên vì nó là phép toán tìm kiếm ngẫu nhiên. Với tỷ lệ đột biến cao thì thuật toán di truyền còn cho kết quả xấu hơn là phương pháp tìm kiếm ngẫu nhiên.

Ví dụ:

V1: 1011**0**1010

V2: 1011**1**1010

NST V1 được chọn để đột biến tại vị trí gen thứ năm, gen này hiện tại là 0, sau khi đột biến sẽ trở thành 1. Khi đó NST v1 trở thành v2.

Sau khi *tái sinh, lai và đột biến,* quần thể mới được tiếp tục tính toán giá trị thích nghi, sự tính toán này được dùng để xây dựng phân bố xác suất cho quá trình tái sinh tiếp theo, nghĩa là để xây dựng lại bánh xe Roulette với các rãnh được định kích thước theo các giá trị thích nghi hiện hành. Phần còn lại của thuật toán di truyền chỉ là sự lặp lại chu trình của các bước trên.

### Các bước giải một bài toán bằng giải thuật di truyền

Để giải quyết vấn đề bài toán bằng giải thuật di truyền, chúng ta cần thực hiện 7 bước quan trọng sau:

Bước 1: Chọn mô hình cho giải pháp của vấn đề, chọn một số đặc trưng cho toàn bộ các giải pháp (quần thể) có thể có cho vấn đề.

Bước 2: Chỉ định cho mỗi giải pháp (cá thể) một ký hiệu. Ký hiệu có thể là một dãy các số 0, 1 thuộc hệ nhị phân, hay dãy các số thập phân, dãy các chữ hay hỗn hợp của số và chữ. Ký hiệu đơn giản nhất và thường dùng nhất là số nhị phân.

Bước 3: Tìm hàm số thích nghi cho vấn đề và tính hệ số thích nghi cho từng giải pháp (lời giải).

Bước 4: Dựa trên hệ số thích nghi của các giải pháp để thực hiện sự tạo sinh (reproduction) và biến hóa các giải pháp. Các phương thức biến hóa bao gồm: lai ghép (crossover), đột biến (mutation).

Bước 5: Tính các hệ số thích nghi cho các giải pháp mới và loại bỏ những giải pháp kém nhất để chỉ còn giữ lại một số nhất định của giải pháp.

Bước 6: Nếu chưa tìm được giải pháp tối ưu hay tương đối khá nhất hay chưa hết kỳ hạn ấn định, trở lại bước 4 để tìm giải pháp mới.

Bước 7: Tìm được giải pháp tối ưu hoặc nếu thời gian cho phép đã chấm dứt thì kết thúc giải thuật và báo cáo kết quả tìm được.

### Các nguyên lý trong giải thuật di truyền

*Nguyên lý về xác định cấu trúc dữ liệu*

Việc mã hóa lời giải cho bài toán gồm hai phần cơ bản:

* Xây dựng cấu trúc gen cho mỗi lời giải của bài toán để từ mỗi lời giải ta có thể mã hoá thành một NST (chuỗi các gen).
* Giải mã các NST để nhận được lời giải.

Đây là vấn đề cần giải quyết trước khi giải bài toán với GA. Tùy thuộc vào nội dung của mỗi bài toán mà ta có cách mã hoá khác nhau.

Để có thể giải bài toán bằng GA cần “gen hóa” cấu trúc dữ liệu của bài toán. Có nhiều phương pháp mã hoá nhưng ta thường sử dụng một trong ba loại cấu trúc dữ liệu (CTDL) sau: *chuỗi nhị phân, chuỗi số thực* và *cấu trúc cây.* Trong đó, chuỗi nhị phân và chuỗi số thực thường được sử dụng hơn.

* Biểu diễn gen bằng chuỗi nhị phân

Về nguyên tắc, mọi cấu trúc dữ liệu trên máy tính, cuối cùng được chuyển về các chuỗi nhị phân.

Ở đây, chúng ta sử dụng chuỗi nhị phân một cách tường minh để thể hiện cấu trúc “gen” của một cá thể và để thực hiện các thao tác lai ghép, đột biến trên cấu trúc này.

Quy tắc biểu diễn gen qua chuỗi nhị phân:chọn chuỗi nhị phân ngắn nhất nhưng đủ để thực hiện được tất cả kiểu gen.

* Biểu diễn gen bằng chuỗi số thực

Đối với những bài toán có nhiều tham số, việc biểu diễn gen bằng chuỗi nhị phân đôi lúc làm kiểu gen của cá thể trở lên phức tạp. Khi đó, người ta sẽ chọn kiểu gen dưới dạng chuỗi số thực.

Quy tắc biểu diễn kiểu gen bằng chuỗi số thực : biểu diễn kiểu gen bằng chuỗi số thực phải đảm bảo tiết kiệm không gian đối với từng thành phần gen.

* Cấu trúc cây

Cấu trúc cây thường được dùng trong trường hợp bản thân cấu trúc dữ liệu của bài toán cũng có dạng cây.

Đây là trường hợp phức tạp nên hiếm khi sử dụng. Một loại cây thường được sử dụng trong GA là dạng cây hai nhánh.

*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 đó.

*Nguyên lý chọn lọc các cá thể có độ thích nghi tốt*

Khả năng được chọn vào quá trình sinh sản thế hệ kế tiếp đồng biến với tính thích nghi của cá thể.

Độ thích nghi càng cao thì xác suất được chọn lọc trong quá trình sinh sản thế hệ kế tiếp cao hơn.

Một số quy tắc chọn lọc cá thể dựa trên độ thích nghi.

* Quy tắc chọn lọc xén

Sắp xếp thứ tự quần thể theo độ thích nghi. Các cá thể có độ thích nghi cao nhất sẽ nằm ở đầu danh sách. Xác định một ngưỡng xén Trunc là tỷ lệ phần trăm. Giá trị này sẽ xác định bao nhiêu phần trăm cá thể tốt nhất trong quần thể sẽ được chọn lọc để tham gia vào quá trình sinh sản thế hệ mới. Những cá thể nằm ngoài đường xén này sẽ không được chọn.

* Quy tắc chọn lọc Roulete

Dựa vào hoạt động của bàn Roulete. Mỗi cá thể được xem là một cung trên bàn, cá thể có độ thích nghi cao tương ứng với góc càng lớn.

Để tiến hành chọn lọc, cho bàn Roulete quay, khi mũi tên dừng lại ở vị trí nào thì cá thể tương ứng được chọn lọc. Như vậy, mỗi lần hoạt động của bản roulete ta sẽ chọn được một cá thể.

*Nguyên lý kết hợp các cá thể*

Bản chất của việc tạo sinh các lời giải mới trong GA là sự kết hợp các lời giải cha mẹ bằng các toán tử mô phỏng thao tác lai ghép trong sinh học.

*Nguyên lý đột biến lời giải*

Để mở rộng kết quả tìm kiếm được quy định bởi thế hệ cá thể trước, cần áp dụng các toán tử thay đổi giá trị cá thể mô phỏng hiện tượng đột biến trong sinh học.

Quy tắc chọn xác suất đột biến:chọn xác suất đột biến tỷ lệ nghịch với kích thước gen và không phụ thuộc kích thước quần thể.

### Ví dụ GA

Ta sẽ đi vào một ví dụ cụ thể: Ứng dụng giải thuật di truyền giải bài toán tối ưu.

*Bài toán:**Tính giá trị cực đại của hàm sau:*

***f(x1 , x2 )=21.5+x1.sin (4Πx1)+ x2.sin (20Πx2)***

*với -3.0 ≤ x1 ≤ 12.1 và 4.1 ≤ x2 ≤ 5.8*

Ta cần cực đại hóa hàm f (x1, x2):

**

Hình 1.3. Đồ thị của hàm f

Ta sẽ lần lượt trình bày về năm thành phần chính của giải thuật di truyền để giải bài toán này.

* Biểu diễn NST

Ta sử dụng một véc tơ nhị phân làm NST để biểu diễn các giá trị thực của biến x1, x2. Chiều dài của vectơ này phụ thuộc vào độ chính xác cần có, trong ví dụ này độ chính xác cần 4 số lẻ.

* Khởi tạo quần thể ban đầu

Tiến trình khởi tạo quần thể rất đơn giản: ta tạo một quần thể các NST, trong đó mỗi NST là một vectơ nhị phân 33 bít. Tất cả 33 bít của mỗi NST đều được khởi tạo ngẫu nhiên.

* Hàm lượng giá

Hàm lượng giá eval của các vectơ nhị phân v chính là hàm f:

eval (v) = f(x)

Trong đó, NST v biểu diễn giá trị thực x như đã nói ở trên, hàm lượng giá đóng vai trò môi trường, đánh giá từng lời giải theo độ thích nghi của chúng.

Giả sử kích cỡ quần thể là 20, các xác suất di truyền tương ứng là pc=0.25 và pm=0.01. Tính chính xác đến 4 chữ số thập phân?

*Giải*

* Bước 1:
* Sử dụng hệ nhị phân để xây dựng mô hình bài toán.
* Số khoảng mà mỗi biến cần dùng là:

Biến x1 cần: (12.1-(-3)).104 =151000 khoảng.

Mặt khác, do 217<151000<218 nên ta cần dùng 18 bit nhị phân để biểu diễn biến x1.

Biến x2 cần: (5.8-4.1).104 =17000 khoảng

Do 214<17000<215 nên ta dùng 15 bít nhị phân để biểu diễn biến x2.

* Chiều dài của NST là m=18+15=33 bít.

Kí hiệu NST thứ i là vi = (x1,x2) với i=1..20.

* Khởi tạo quần thể ban đầu một cách ngẫu nhiên như sau:

V1=100110100000001111 **111010011011111**

V2=111000100100110111 **001010100011010**

V3=000010000011001000 **001010111011101**

V4=100011000101101001 **111000001110010**

V5=000111011001010011 **010111111000101**

V6=000101000010010101 **001010111111011**

V7=001000100000110101 **111011011111011**

V8=100001100001110100 **010110101100111**

V9=100000000101100010 **110000001111100**

V10=000001111000110000 **011010000111011**

V11=011001111110110101 **100001101111000**

V12=110100010111101101 **000101010000000**

V13=111011111010001000 **110000001000110**

V14=010010011000001010 **100011111011110**

V15=111011101101110000 **100011111011110**

V16=110011110000011111 **100001101001011**

V17=011010111111001111 **010001101111101**

V18=011101000000001110 **100111110101101**

V19=000101010011111111 **110000110001100**

V20=101110010110011110 **011000101111110**

* Bước 2: Xác định giá trị thích nghi cho từng NST trong quần thể ban đầu.

Xét NST thứ nhất (V1)2 =100110100000001111 **111010011011111**

Theo như cách mã hóa bằng chuỗi nhị phân (đã trình bày ở trên) thì ta xác định được giá trị thập phân của các biến x1, x2 ứng với NST V1 như sau:

x1 = -3.0 + dec(100110100000001111) x  = 6.0845

x2 = 4.1 + dec(**111010011011111**) x  = 5.6522

* (V1)10 =(x1, x2) = (6.0845, 5.6522)
* *Giá trị hàm thích nghi là:*

f1(x1, x2) = f(6.0845, 5.6522) = 26.0196

* Tương tự tính giá trị hàm thích nghi cho các NST còn lại: f2, f3,….., f20

Giá trị tổng hàm thích nghi của toàn quần thể là:

F= f1­+f2+……+f20 = 387.7768

Xác suất chọn p1 là: P1 =  = 0.067

Vị trí xác suất q1 là: q1 = p1 =0.067

Cũng tương tự ta tính cho các NST còn lại. Kết quả sau khi tính toán được đưa bởi bảng sau:

* Bảng kết quả thu được:

| **NST** | **X1** | **X2** | **Gía trị thích nghi** | **Xác suất chọn** | **Vị trị xác suất** |
| --- | --- | --- | --- | --- | --- |
| **1** | 6.0845 | 5.6522 | 26.0196 | 0.067 | 0.067 |
| **2** | 10.3484 | 4.3802 | 7.5800 | 0.019 | 0.086 |
| **3** | -2.5166 | 4.3904 | 19.6263 | 0.050 | 0.137 |
| **4** | 5.2786 | 5.5934 | 17.4067 | 0.044 | 0.181 |
| **5** | -1.2551 | 4.7345 | 25.3411 | 0.065 | 0.247 |
| **6** | -1.8117 | 4.3919 | 18.1004 | 0.046 | 0.293 |
| **7** | -0.9915 | 5.6803 | 16.0208 | 0.041 | 0.335 |
| **8** | 4.9106 | 4.7030 | 17.9597 | 0.046 | 0.381 |
| **9** | 0.7954 | 5.3814 | 16.1278 | 0.041 | 0.423 |
| **10** | -2.5548 | 4.7937 | 21.2784 | 0.054 | 0.478 |
| **11** | 3.1300 | 4.9961 | 23.4107 | 0.060 | 0.538 |
| **12** | 9.3561 | 4.2395 | 15.0116 | 0.038 | 0.577 |
| **13** | 11.1346 | 5.3786 | 27.3167 | 0.070 | 0.647 |
| **14** | 1.3359 | 5.1513 | 19.8762 | 0.051 | 0.698 |
| **15** | 11.0890 | 5.0545 | 30.0602 | 0.077 | 0.776 |
| **16** | 9.2116 | 4.9937 | 23.9672 | 0.061 | 0.837 |
| **17** | 3.3675 | 4.5713 | 13.6961 | 0.035 | 0.873 |
| **18** | 3.8430 | 5.1582 | 15.4141 | 0.039 | 0.812 |
| **19** | -1.7466 | 5.3955 | 20.0959 | 0.051 | 0.964 |
| **20** | 7.9359 | 4.7573 | 13.6669 | 0.035 | 1.000 |

Bảng 1.1. Kết quả giá trị thích nghi

* **Bước 3:**
* ***Tái sinh:*** Xây dựng hệ thống bánh xe Roulette cho tiến trình chọn lọc. Ta quay bánh xe 20 lần, phát sinh ngẫu nhiên 20 số r trong khoảng [0..1]. Mỗi lần quay chọn được 1 NST mới.

Ví dụ, ở lần quay đầu tiên, ta chọn ngẫu nhiên số **r =0.514**. Dễ thấy, q10 < r <q11

* **NST 11** được chọn thay **cho NST 1** trong quần thể mới.

Làm tương tự với các lần quay tiếp theo với các số ngẫu nhiên r ta được các kết quả sau:

| **Lần quay** | **Số ngẫu nhiên r** | **NST được chọn** |
| --- | --- | --- |
| 1 | 0.514 | V11 |
| 2 | 0.175 | V4 |
| 3 | 0.308 | V7 |
| 4 | 0.534 | V11 |
| 5 | 0.947 | V19 |
| 6 | 0.171 | V4 |
| 7 | 0.072 | V15 |
| 8 | 0.226 | V5 |
| 9 | 0.494 | V11 |
| 10 | 0.424 | V3 |
| 11 | 0.704 | V15 |
| 12 | 0.389 | V9 |
| 13 | 0.277 | V6 |
| 14 | 0.368 | V8 |
| 15 | 0.983 | V20 |
| 16 | 0.005 | V1 |
| 17 | 0.765 | V10 |
| 18 | 0.646 | V13 |
| 19 | 0.767 | V15 |
| 20 | 0.780 | V16 |

Bảng 1.2. Bảng mô tả bánh xe Roulette

Như vậy, ta sẽ có quần thể mới là:

V’1=V11=011001111110110101 **100001101111000**

V’2= V4=100011000101101001 **111000001110010**

V’3= V7=001000100000110101 **111011011111011**

V’4=V11=011001111110110101 **100001101111000**

V’5= V19=000101010011111111 **110000110001100**

V’6= V4=100011000101101001 **111000001110010**

V’7= V15=111011101101110000 **100011111011110**

V’8= V5=000111011001010011 **010111111000101**

V’9= V11=011001111110110101 **100001101111000**

V’10= V3=000010000011001000 **001010111011101**

V’11= V15=111011101101110000 **100011111011110**

V’12= V9=100000000101100010 **110000001111100**

V’13= V6=000101000010010101 **001010111111011**

V’14= V8=100001100001110100 **010110101100111**

V’15= V20=101110010110011110 **011000101111110**

V’16= V1=100110100000001111 **111010011011111**

V’17= V10=000001111000110000 **011010000111011**

V’18= V13=111011111010001000 **110000001000110**

V’19= V15=111011101101110000 **100011111011110**

V’20= V16=110011110000011111 **100001101001011**

* *Quá trình lai ghép:* xác suất lai pc =0.25

Đối với mỗi NST trong quần thể mới phát sinh ngẫu nhiên 1 số r trong khoảng [0..1]. Nếu r ứng với NST nào < 0.25 thì ta chọn NST đó để lai tạo.

Giả sử thứ tự của các số ngẫu nhiên là:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.822 | **0.1512** | 0.625 | 0.314 | 0.346 |
| 0.917 | 0.519 | 0.401 | 0.606 | 0.785 |
| **0.03111** | 0.869 | **0.16613** | 0.574 | 0.758 |
| 0.581 | 0.389 | **0.20018** | 0.355 | 0.86 |

Bảng 1.3. Bảng mô tả chọn NST lai ghép

Dễ thấy, các NST V’2, V’11, V’13, V’18 sẽ được chọn để lai tạo. Ở đây, chúng ta có số NST cần được lại tạo là một số chẵn nên ta có thể ghép cặp dễ dàng. Nếu là số lẻ thì ta có thể công thêm một NST ngoài hoặc bớt đi 1 NST ngẫu nhiên.

Ta sẽ ghép cặp (V’2, V’11) và (V’13, V’18)

* Cặp NST (V’2, V’11): giả sử vị trí lai là 9

V’2= V4= 100011000 101101001 **111000001110010**

V’11= V15= 111011101 101110000 **100011111011110**

* Các con: V”2 = 100011000 1 01110000 **100011111011110**

V”11 = 111011101 1 01101001 **111000001110010**

* Cặp NST (V’13, V’18): giả sử vị trí lai là 20

V’13= V6= 000101000010010101 **00 1010111111011**

V’18= V13=111011111010001000 **11 0000001000110**

* Các con là: V”13 = 000101000010010101 **00 0000001000110**

V”18 = 111011111010001000 **11 1010111111011**

* *Quá trình đột biến:* xác suất đột biến pm =0.01

Toàn bộ quần thể có 33x20=660 bit. Như vậy, trung bình có 6.6 đột biến trông mỗi thế hệ. Vì mỗi bít đều có cơ hội đột biến như nhau nên ta phát sinh ngẫu nhiên số r trong khoảng [0..1]. Nếu bit nào có r < 0.01 thì xảy ra đột biến ở bit đó.

Giả sử có 5 vị trí đột biến như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Vị trí bit** | **Số ngẫu nhiên** | **NST thứ** | **Vị trí bit trong NST** |
| 112 | 0.0002 | 4 | 13 |
| 349 | 0.0009 | 11 | 19 |
| 418 | 0.0088 | 13 | 22 |
| 429 | 0.0054 | 13 | 33 |
| 603 | 0.0028 | 19 | 9 |

Bảng 1.4. Bảng mô tả NST đột biến

Ta thấy có 4 NST bị đột biến. Trong đó, NST thứ 13 bị đột biến ở 2 vị trí. Sau khi đột biến ta có quần thể hiện hành là:

V1=011001111110110101 **100001101111000**

V2=100011000101101001 **111000001110010**

V3=001000100000110101 **111011011111011**

V4=011001111110010101 **100001101111000**

V5= 000101010011111111 **110000110001100**

V6=100011000101101001 **111000001110010**

V7=111011101101110000 **100011111011110**

V8=000111011001010011 **010111111000101**

V9=011001111110110101 **100001101111000**

V10=000010000011001000 **001010111011101**

V11=111011101101110000 **000011111011110**

V12=100000000101100010 **110000001111100**

V13=000101000010010101 **001000111111010**

V14=100001100001110100 **010110101100111**

V15=101110010110011110 **011000101111110**

V16=100110100000001111 **111010011011111**

V17=000001111000110000 **011010000111011**

V18=111011111010001000 **110000001000110**

V19=111011100101110000 **100011111011110**

V20=110011110000011111 **100001101001011**

* Bước 4:

Ta đi tính lại độ thích nghi cho quần thể mới. Tính tương tự như trong bước 2 ta sẽ thu được tổng độ thích nghi của quần thể mới sau khi thưc hiện 1 vòng lặp giải thuật di truyền là: F= 447.049.

* Bước 5: Dễ thấy 447.049 > 387.77 nên lời giải trên thỏa mãn.

*Tóm tắt:* Trong chương 1, chúng ta đã cùng nhau tìm hiểu một số vấn đề cơ bản của Giải thuật di truyền và cách giải quyết vấn đề - bài toán bằng GA. Hoạt động của GA dựa trên sự mô phỏng quá trình thích nghi và tiến hóa của tự nhiên. Để giải bài toán bằng GA ta cần phải tìm kiểu gen thích hợp - chuỗi nhị phân, chuỗi số thực hoặc cấu trúc cây. Bước tiếp theo là xây dựng hàm tính giá trị thích nghi của cá thể, sau đó xác định một trong các phương pháp chọn lọc như: xén, Roulete… Một bước quan trọng không thể thiếu là phải xác định các thao tác lai ghép, đột biến và mô phỏng trong sinh học trên kiểu gen đã chọn.

Quá trình tiến hoá thể hiện tính tối ưu ở chỗ, thế hệ sau bao giờ cũng tốt hơn, phát triển hơn, hoàn thiện hơn thế hệ trước. Tiến hoá tự nhiên được duy trì nhờ hai quá trình cơ bản: sinh sản và chọn lọc tự nhiên. Xuyên suốt quá trình tiến hoá tự nhiên, các thế hệ mới luôn được sinh ra để bổ xung thay thế thế hệ cũ. Cá thể nào phát triển hơn, thích ứng hơn với môi trường sẽ tồn tại, cá thể nào không thích ứng với môi trường sẽ bị đào thải. Sự thay đổi môi trường là động lực thúc đẩy quá trình tiến hoá. Ngược lại, tiến hoá cũng tác động trở lại góp phần làm thay đổi môi trường.

Sau cùng, chúng ta khảo sát một ứng dụng của thuật giải di truyền: tối ưu hàm đa biến. Từ đây, chúng ta thấy rằng tiềm năng của thuật giải di truyền trong việc giải quyết vấn đề - bài toán thật to lớn.

# ỨNG DỤNG THUẬT TOÁN DI TRUYỀN TRONG SẮP XẾP THỜI KHÓA BIỂU

Trong phần này sẽ trình bày các nội dung nghiên cứu chính của đề tài, từ yêu cầu đặt ra cho bài toán tìm kiếm văn bản ta đi xây dựng hàm mục tiêu tìm kiếm. Trên cơ sở đó phát biểu bài toán dưới dạng tối ưu hàm một biến và dùng phương pháp giải thuật di truyền để giải quyết bài toán.

Ngoài ra, nghiên cứu về bài toán lập thời khóa biểu để hiểu rõ hơn về giải thuật di truyền vào giải quyết bài toán thực tế.

## Bài toán sắp xếp thời khóa biểu

### Phát biểu bài toán

Trong mỗi học kỳ, các lớp đều có thời khóa biểu của mình trong học kỳ đó gồm lịch học các môn lý thuyết và lịch thực hành. Mục tiêu của chương trình là giúp Giáo Vụ khoa dễ dàng trong việc xếp thời khoá biểu lý thuyết phù hợp cho cả giảng viên, các lớp thuộc khoa và các khoa khác của trường dựa trên một số điều kiện thực tế.

Bài toán thời khóa biểu kết hợp nhiều ràng buộc không tầm thường thuộc nhiều loại. Trên thực tế có nhiều phiên bản của bài toán thời khóa biểu, một trong những bài toán này có thể được mô tả như sau:

Có một danh sách các giáo viên {GV1, GV2, …., GVm}.

Có một danh sách các lớp {L1,L2, …, Ln}.

Có một danh sách các môn { M1, M2, …, Mk}.

Có một danh sách các phòng học {P1, P2, …, Pl}.

Bài toán cần tìm thời khóa biểu tối ưu giữa giáo viên – môn – lớp - phòng, hàm mục tiêu phải thỏa mãn các ràng buộc của bài toán.

### Phân tích bài toán

*Dữ liệu đầu vào*

Như đã nói ở trên, thông tin sẽ phát sinh từ các đối tượng chính trong bài toán. Do đó, các dữ liệu luôn có mối liên hệ với nhau, phần lớn vì nhu cầu nghiệp vụ mà dữ liệu xuất hiện tương đối nhiều. Trong bài toán xếp thời khóa biểu chế độ niên chế, cụ thể sẽ đòi hỏi các thông tin sau:

* Danh sách lớp học.
* Danh sách giáo viên.
* Danh sách môn học.
* Số ngày học cho một thời khóa biểu.
* Số tiết tối đa trong một ngày.
* Số tiết mỗi môn trong một tuần.
* Bảng phân công giáo viên giảng dạy tại các lớp.
* Bảng phân công giáo viên dạy các môn.
* Bảng phân công giữa lớp và môn học.
* Lịch bận của giáo viên.

*Dữ liệu đầu ra*

* Thời khóa biểu dạy của các giáo viên.
* Thời khóa biểu học của học sinh

*Các ràng buộc của bài toán*

Bài toán này có 2 ràng buộc:

* Ràng buộc cứng:
* Tại cùng một thời điểm một giáo viên không thể dạy hai lớp.
* Một lớp không thể học trong cùng một thời điểm hai môn học.
* Số giờ học của một môn/ngày nhỏ hơn hoặc bằng số giờ học lớn nhất theo quy định dành cho một môn/một ngày.
* Tổng số giờ phải học trong một ngày nhỏ hơn hoặc bằng số giờ học lớn nhất theo quy định trong một buổi.
* Tổng số giờ phải học cho từng môn bằng số giờ phải học của môn đó/tuần.
* Phải học đủ các môn theo quy định.
* Giáo viên đăng ký dạy lớp nào thì dạy lớp đó.
* Giáo viên đăng ký dạy môn nào thì dạy môn đó.
* Đảm bảo được lịch bận của giáo viên, giáo viên đăng ký bận ngày nào thì không phải dạy vào ngày đó.
* Ràng buộc mềm:
* Thứ tự các môn học cần được đảm bảo. Tránh tình trạng các môn học được giảng dạy lộn xộn gây ra chất lượng không tốt cho các lớp học.
* Khoảng thời gian nghỉ giữa hai lần đi dạy của các giáo viên phải đảm bảo có độ giãn cách hợp lý. Không có giáo viên nào dạy hai lần giống nhau trong một buổi.

## Áp dụng giải thuật di truyền

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: Xếp lịch học các lớp. 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: Xếp lịch học cho toàn bộ cơ sở. 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.

*Mô hình của bài toán có thể được mô phỏng theo mô hình sau:*

**Quần thể**

🡨 là một tập nhiều NST (thời khóa biểu) toàn khóa.

🡨 đại diện cho một cá thể (một tập thời khóa biểu cho toàn khóa)

NST

🡨 đại diện cho cấu tạo lên NST (một TKB cho một lớp)

Ngày

Gen

🡨là một ngày học của một TKB (đơn vị cấu tạo lên gen)

Tiết

*Giai đoạn 1 – Biểu diễn NST*

Lịch học của một lớp gồm có số ngày học trong tuần, môn học, các giờ học và các giảng viên. Việc đặt ngẫu nhiên các môn học với các giờ học sẽ tạo thành một lịch học cho từng lớp. Như vậy một lớp học tương ứng sẽ có nhiều lịch học khác nhau, do đó ta chọn mỗi lịch học làm NST trong thuật giải di truyền.

Mỗi NST là một thời khóa biểu cho một lớp, mỗi thời khóa biểu là một tập các ngày ( một tuần), mỗi gen hình thành lên NST là sự kết hợp của nhiều ngày. Mỗi ngày gồm nhiều tiết học, mỗi tiết gồm : Tên môn học, giảng viên dạy, số giờ phải học và lớp phải học.

****

Hình 2.1. Mô hình cá thể trong lịch lớp

Dựa vào các dữ liệu và ràng buộc của bài toán ta sẽ chọn ra được các môn cần phải học và số giờ tương ứng phải học trong một tuần của một lớp.

Ví dụ: Lớp A cần phải học:

Môn Toán: 6 giờ

Môn Anh: 5 giờ

Môn Triết: 4 giờ

Từ dữ liệu đó thông qua hàm Split() ta sẽ chia các môn và số giờ phải học trong tuần thành các ca học sao cho số giờ phải học không lớn hơn số giờ tối đa được dạy trong một ca.

Ví dụ: Áp dụng vd trên ta có:

Môn Toán chia thành: T3 + T3

Môn Anh chia thành: A3 + A2

Môn Triết chia thành: Tr2+ Tr2

Sau khi chia ta được một mảng các ca phải học trong tuần tương ứng với số giờ học: T3 T3 A3 A2 Tr2 Tr2. Hiển nhiên, từ mảng các ca học trên ta có thể tạo một thứ tự ngẫu nhiên các môn phải học.

Vd : T3 A2 Tr2 T3 A3 Tr2

Bước quan trọng nhất trong giai đoạn này chính là khởi tạo thời khóa biểu cho một lớp học. Đưa các môn được xếp theo thứ tự ngẫu nhiên ở trên lần lượt vào từng ngày trong thời khóa biểu sao cho phải thỏa mãn các ràng buộc của bài toán.

Như ta đã nói phần trên, 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.

*Giai đoạn 1 - Tạo quần thể ban đầu*

Tạo quần thể đầu tiên trong giải thuật, là nơi xuất phát quá trình tiến hóa, bao gồm tất cả các giá trị thô ban đầu. Tùy theo vấn đề của bài toán mà có cách khởi tạo khác nhau.

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 và số tiết học của từng môn.
* 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.

*Giai đoạn 1 - Độ thích nghi*

Chắc chắn rằng việc chọn cá thể sẽ thông qua kết quả, hay mục đích của vấn đề. Các cá thể tốt được chọn lọc để đưa vào thế hệ sau. Sự lựa chọn này được thực hiện dựa vào độ thích nghi với môi trường của mỗi cá thể.

Có nhiều phương pháp để chọn các nhiễm sắc thể tốt nhất, ví dụ: chọn lọc roulette wheel, chọn lọc xếp hàng, chọn lọc cạnh tranh, v.v…

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.

* Trước hết ta nói về giáo viên: Tại cùng một thời điểm một giáo viên không thể dạy hai lớp. 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.
* Cuối cùng kiểm tra lại sự chồng chéo giờ lẫn nhau của các môn học. Một lớp không thể học trong cùng một thời điểm hai 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 tạo, 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.

*Giai đoạn 2 – Biểu diễn NST*

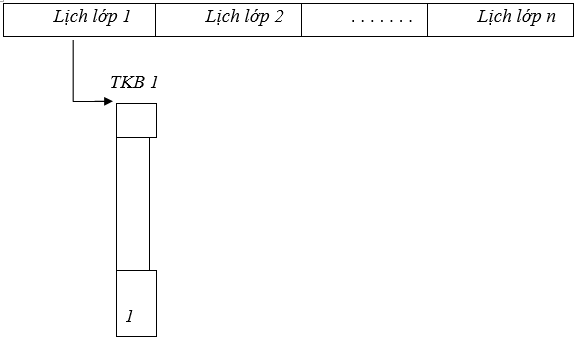
Lịch học tại cơ sở bao gồm tất cả các lịch học của các lớp hiện có trong cơ sở, nếu mỗi lớp đều có một lịch học rõ ràng thì đó có nghĩa là có lịch cơ sở.

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 các ngày học trong tuần, cho nên ta chọn lớp học làm đơn vị của nhiễm sắc thể trong mô hình thuật toán di truyền trong xếp lịch cơ sở. Và tương tự, ta chọn lịch cơ sở làm cá thể.



Hình 2.2. Minh họa cho mô hình NST trong một cơ sở

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).

**

Hình 2.3. Mô hình cá thể trong lịch cơ sở.

*Giai đoạn 2 - Tạo quần thể ban đầu*

Quần thể khởi đầu gồm những cá thể được tạo ra như mô hình trên. 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 ngày học trong tuần của lớp.

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

*Giai đoạn 2 - Độ thích nghi*

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.

Cuối cùng kiểm tra lại sự chồng chéo giờ lẫn nhau của các môn học. Một lớp không thể học trong cùng một thời điểm hai môn học. Ta sẽ đưa ra được tổng số lần chồng chéo nhau của các lớp trong một cơ sở.

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.

*Giai đoạn 2 – Toán tử di truyền*

Hai phần này, có lẽ đã được nói rõ trong chương trước. Sử dụng lại của phần xếp lịch lớp, chọn cá thể theo độ thích nghi, lai ghép ngẫu nhiên đoạn và đột biến hoán vị điểm.

1. **Phép lai**

Giả sử lai hai NST sau

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A1 | B1 | C1 | D1 | E1 | F1 | G1 | H1 | I1 | K1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A2 | B2 | C2 | D2 | E2 | F2 | G2 | H2 | I2 | K2 |

Trong đó: A, B, C…: là thời khóa biểu của từng lớp trong một cơ sở.

D, E, F, G: là đoạn chọn để lai ghép.

Sau khi thực hiện phép lai ta được hai NST con sau:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A1 | B1 | C1 | D2 | E2 | F2 | G2 | H1 | I1 | K1 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A2 | B2 | C2 | D1 | E1 | F1 | G1 | H2 | I2 | K2 |

1. **Phép đột biến**

Được thực hiện bằng cách chọn ngẫu nhiên vị trí đột biến sau đó khởi tạo lại vị trí này trong thời khóa biểu của lớp này ta sẽ được một NST mới sau khi thực hiện toán tử đột biến.

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.

## Điểm dừng của 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. Với giải thuật di truyền cổ điển thong thường có 4 điều kiện dừng:

1. GA lặp cho đến khi nào có được kết quả tối ưu thì thuật toán sẽ dừng. Khi đó, ta sẽ thu được kết quả là tốt nhất.
2. Ta biết trước được số thế hệ mà GA sẽ thực hiện. GA lặp cho đến khi bằng số đời mà người dùng thiết lập thì thuật toán sẽ dừng. Khi đó, ta sẽ thu được kết quả dù là tốt hay chưa tối ưu thì ta cũng sẽ phải chấp nhận.
3. GA lặp khi biết trước thời gian. Khi đó, ta sẽ thu được kết quả dù là tốt hay chưa tối ưu thì ta cũng sẽ phải chấp nhận.
4. Sự kết hợp của các điều kiện trên.

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, mặc 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.

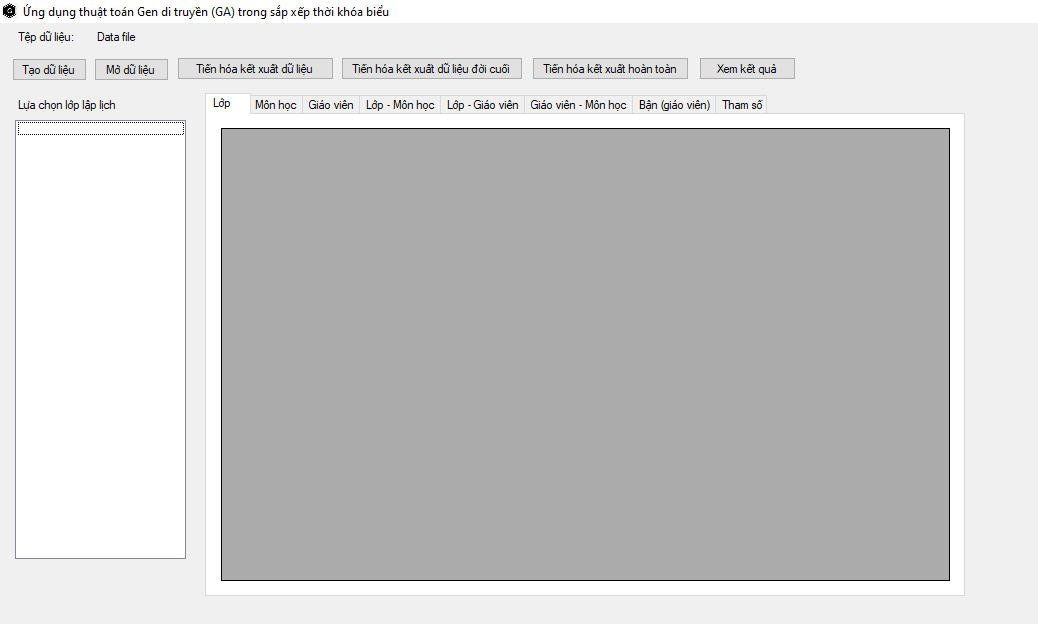
*Tóm tắt:* Chương 3, chúng ta đã được xem xét một ví dụ cụ thể và có ứng dụng cài đặt áp dụng GA. Đây là một sáng kiến nhỏ nhằm ứng dụng cải tiến việc tìm kiếm văn bản. Hiện nay, thuật toán di truyền cùng với logic mờ được ứng dụng rất rộng rãi trong các lĩnh vực phức tạp. Thuật toán di truyền kết hợp với logic mờ chứng tỏ được hiệu quả của nó trong các vấn đề khó có thể giải quyết bằng các phương pháp thông thường hay các phương pháp cổ điển. Nhất là trong các bài toán cần có sự lượng giá, đánh giá sự tối ưu của kết quả thu được.

# XÂY DỰNG CHƯƠNG TRÌNH VÀ KẾT QUẢ THỬ NGHIỆM

Các kết quả thử nghiệm thu được từ việc lập trình cài đặt trên C# với file thời khóa biểu, nội dung file có các môn học, danh sách giáo viên, thời gian bận. Dựa vào thông tin đầu vào, người dùng có thể tạo trực tiếp hai thời khóa biểu ngay trên giao diện demo. Kết quả thu được như sau:

## Thiết kế giao diện chương trình

### Giao diện chính



Hình 3.1. Giao diện thiết kế demo chính

Ở đây, em đã xây dựng giao diện hỗ trợ cho việc sắp xếp thời khóa biểu, nhằm giúp người dùng quản lý và tổ chức các tiết học một cách hiệu quả. Mỗi tab sẽ đảm nhận nhiệm vụ hiển thị dữ liệu liên quan riêng, giúp người dùng dễ dàng theo dõi và điều chỉnh thời khóa biểu của mình.

Chương trình cho phép ta hiển thị được kết quả của quá trình tiến hóa như cấu trúc của GA, tính toán được giá trị hàm thích nghi, xác xuất sống của mỗi cá thể trong quần thể, vị trí xác xuất.

Chương trình được thiết kế dựa trên lớp cơ bản: C*lassCaThe, ClassQuanThe,* *ProcessString.cs, class GA, class Guide.*

Giải thích

* *Form GiaoDien*: hiển thị giao diện chính của hệ thống.
* *class sinhvien:* Mở ra thông tin về sinh viên thực hiện.
* *class huongdan*: Lớp này dùng để mở ra thông tin hướng dẫn người thực hiện chương trình chạy bài demo.

### Giao diện nhập tham số cho bài toán

|  |  |
| --- | --- |
|  |  |
|  |  |

### Giao diện hiển thị kết quả khi thực hiện chương trình

* txtChuKy: hiển thị số đời của quá trình tiến hóa.
* txtSoKyTuVanBan: độ dài văn bản tìm kiếm.
* txtViTri: những vị trí mà văn bản mẫu xuất hiện trong văn bản tìm kiếm.
* txtSoLanXuathHien: tổng số lần Sm xuất hiện trong S.

### Dữ liệu đầu vào

Hình 3.2. Văn bản tìm kiếm S

Hình 3.3. Văn bản mẫu Sm

Trong đó:

Hình 4.4 (a) hiển thị nội dung văn bản mà chúng ta cần tìm kiếm.

Hình 4.4 (b) hiển thị nội dung văn bản mẫu để so sánh tìm kiếm trên S.

Để hiển thị nội dung hai văn bản này ta có thể đánh trực tiếp nội dung nếu chiều dài văn bản ngắn. Đối với văn bản có số ký tự rất lớn ta có thể mở như windown.

## Một số kết quả thử nghiệm của chương trình

### Kết quả thử nghiệm tìm kiếm bằng giải thuật di truyền

*Dữ liệu đầu vào*

* File văn bản để tìm kiếm *S=”test.txt”* có chiểu dài 3141 ký tự.
* Văn bản mẫu là *Sm =”* *genetic”.*

*Các tham số*

Đối với bài toán này, ta sử dụng các tham số sau đây:

* Kích thước quần thề pop-size = 40.
* Chu kỳ quần thể = 100.
* Xác suất lai tạo pc = 0.8.
* Xác suất đột biến pm = 0.01 (nhỏ hơn nhiều so với xác suất lai).

Xác suất lai pc = 0.8 nghĩa là cá thể v trong quần thể có 80% cơ hội được chọn để thực hiện phép lai; còn xác suất đột biến pm = 0.01 lại là 1% 1 bít bất kỳ của 1 cá thể bất kỳ trong quần thể bị đột biến.

*Kết quả tiến hóa*

Dưới đây là các kết quả thực nghiệm sau khi chạy chương trình cài đặt bằng giải thuật di truyền với bài toán tìm kiếm trên. Mỗi lần lặp ta cho tiến hoá 100 thế hệ.

Hình 3.4. Kết quả GA

Ta thấy có 9 vị trí Sm xuất hiện trong văn bản S và giải thuật di truyền đã thực hiện 100 thế hệ với các tham số thiết lập ban đầu.

1. Kết quả quần thể khởi tạo và quần thể cuối cùng của thế hệ thứ 1.

Hình 3.5. Đời 1của quá trình tiến hóa

Với giải thuật di truyền thì sau 100 lần lặp cũng cho ta kết quả là tất cả các vị trí xuất hiện mẫu trong văn bản với thời gian của mỗi lần thực hiện là rất nhỏ (khoảng 30 - 40% giây). Quan sát bảng trên ta thấy có 9 lần đạt cực đại với sự xuất hiện cả 9 vị trí (tìm được tối đa các vị trí xuất hiện mẫu).

Như vậy ta hoàn toàn có thể dùng thuật giải di truyền để tìm kiếm chính xác tất cả các vị trí của mẫu trong văn bản. Trong trường hợp không có đoạn văn bản nào trùng với mẫu thì thuật toán sẽ phát huy được hiệu quả là đưa ra các vị trí tốt nhất (các đoạn văn bản gần giống với văn bản mẫu).

1. Kết quả quần thể khởi tạo và quần thể cuối cùng (thế hệ thứ 100):

Hình 3.6. Đời 100 của quá trình tiến hóa

Hàm F có thể đạt giá trị vượt ngưỡng k tại nhiều vị trí, giá trị lớn nhất của hàm F là M. Để thuận lợi cho việc đánh giá ta định lại giá trị F:= F/M. Khi đó hàm F sẽ đạt giá trị lớn nhất = 1 và F có miền giá trị [0, 1]. Những vị trí có hàm mục tiêu F(x) =1, chính là bằng độ dài của văn bản mẫu là những vị trí chính xác xuất hiện mẫu.

Với cách tiếp cận tìm kiếm này sẽ giúp chúng ta tối ưu về mặt thời gian rất nhiều. Bởi vì, thuật toán này tìm kiếm theo một cách ngẫu nhiên các vị trí, nên nếu vị trí ngẫu nhiên mà gần với vị trí trùng khớp của văn bản mẫu và văn bản tìm kiếm thì kết quả tìm kiếm là nhanh nhất. Tuy nhiên, vẫn có trường hợp vị trí đó cách xa vị trí xuất hiện.

Như vậy ta hoàn toàn có thể dùng thuật giải di truyền để tìm kiếm chính xác tất cả các vị trí của mẫu trong văn bản. Trong trường hợp không có đoạn văn bản nào trùng với mẫu thì thuật toán sẽ phát huy được hiệu quả là đưa ra các vị trí tốt nhất (các đoạn văn bản gần giống với văn bản mẫu).

### Một số hình ảnh của chương trình

Hình 3.7. Form về sinh viên thực hiện

Hình 3.8. Form về tài liệu tham khảo

Hình 3.9. Form thông tin về chương trình

Hình 3.10. Form hướng dẫn cách chạy chương trình

## Ứng dụng bài toán sắp xếp thời khóa biểu

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 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), hệ chuyên gia, mạng Nơron…

Một trong các phương pháp đó không thể không kể đến Giải thuật di truyền. Và kết quả sau khi áp dụng giải thuật di truyền như sau:

Hình 3.11. Form lập lịch

Kết quả hiển thị trên Form main gồm các RibbonTab: Control, Information.

* Các chức năng trên tab Control gồm:

Create Data: Tạo một dữ liệu mới cho bài toán.

Open Data: Mở một dữ liệu đã có.

Edit Data: Chỉnh sửa một file dữ liệu.

Run All: Chạy và hiển thị từng chu kỳ tiến hóa theo số đời do người dùng nhập vào.

Run End: Chạy và hiển thị chu kỳ cuối cùng.

Run Perfect: Chạy và hiển thị chu kỳ tốt nhất.

Show result: Hiện thị kết quả dạng thời khóa biểu.

Updata img: Cập nhật các thông số của một sơ đồ dữ liệu dạng ảnh.

Open img: Mở một file ảnh.

Save img: Lưu sơ đồ dữ liệu bài toán dưới dạng một file ảnh.

* Chức năng trên tab Information gồm:

TKB: Giới thiệu về chương trình phần mềm.

Guide: Hướng dẫn sử dụng.

### Giao diện mô phỏng dữ liệu đầu vào

Hình 3.12. Sơ đồ dữ liệu đầu vào

Sơ đồ thể hiện các rang buộc của bài toán: Cột Class thể hiện tên lớp, cột Teachers thể hiện tên giáo viên, cột Subjects thể hiện tên môn học…

### Ứng dụng tìm kiếm thời khóa biểu

Sau khi có được kết quả của bài lập lịch là một thời khóa biểu của các lớp và các giảng viên.

Hình 3.13. Thời khóa biểu các lớp

Hình 3.14. Thời khóa biểu của giảng viên

Từ đó, em ứng dụng vào bài tìm kiếm của mình như sau: tìm kiếm theo tên giảng viên hoặc theo lớp trên thời khóa biểu vừa xuất ra, để đưa ra một thời khóa biểu của lớp hoặc giảng viên đó.

Hình 3.15. Form tìm kiếm thời khóa biểu

Các chức năng trên tab Điều khiển gồm:

Mở Excel: Mở một dữ liệu file excel cho bài toán.

Thực hiện: Chạy và hiển thị kết quả dạng thời khóa biểu.

txtTimKiem: Hiển thị tên cần tìm kiếm.

dgvThoiKhoaBieu: Hiển thị file execl cần tìm kiếm do người dùng nhập vào.

dgvTimKiem: Hiển thị thời khóa biểu của lớp hoặc giảng viên muốn tìm.

KẾT LUẬN

* + - 1. **Đánh giá kết quả nghiên cứu**

Tóm lại, đề tài đã giải quyết được những vấn đề sau đây:

* Đề tài đã đưa ra được tính cấp thiết và tình trạng của bài toán tìm kiếm văn bản hiện nay.
* Đề tài đã bước đầu đề xuất được một số thuật toán áp dụng giải bài toán tìm kiếm văn bản, từ đó đề xuất phương pháp ứng dụng giải thuật di truyền vào giải quyết bài toán.
* Tìm hiểu được các thuật toán tìm kiếm văn bản theo cách tuyến tính, qua đó làm cơ sở để so sánh với các kết quả nghiên cứu của đề tài.
* Xây dựng được các hàm tính toán cho bài toán và phát biểu bài toán tìm kiếm văn bản để có thể áp dụng giải thuật di truyền.
* Chương trình và kết quả thử nghiệm đã minh chứng hướng tiếp cận giải thuật di truyền giải quyết bài toán tìm kiếm văn bản là đúng đắn và có hiệu quả. Đặc biệt chương trình cài đặt đã chỉ ra được các vị trí xuất hiện đoạn văn bản giống văn bản mẫu hoặc gần giống với văn bản mẫu (trong trường hợp văn bản không chứa văn bản mẫu) trong thời gian cho phép.
  + - 1. **Những vấn đề chưa đạt được**

Kết quả thử nghiệm của chương trình mới chỉ áp dụng trên tập dữ liệu tương đối, với tập dữ liệu lớn chương trình có thể báo lỗi.

* + - 1. **Kiến nghị hướng phát triển**
* Đưa phần mềm ứng dụng này vào ứng dụng thử nghiệm. Do thời gian hạn chế nên phần mềm chưa phát triển được đáng kể.
* Sau khi phát triển thành công phần mềm ứng dụng, hướng nghiên cứu tiếp theo của là tìm hiểu ứng dụng giải thuật di truyền cho nhiều dạng bài toán tìm kiếm, chẳng hạn bài toán tìm kiếm trên các file dữ liệu có cấu trúc đặc biệt.

**TÀI LIỆU THAM KHẢO**

1. Nguyễn Quang Thạch, Bùi Chí Bửu, *Kỹ thuật di truyền nguyên lý và ứng dụng*, NXB Học viện Nông nghiệp Việt Nam, 2021
2. Nguyễn Bá Nghiễn, Vương Quốc Hùng, Trần Hùng Cường, *Một số phương pháp tính toán mềm*, NXB Khoa học và kỹ thuật, 2019
3. Nguyễn Đình Thúc, *Lập trình tiến hóa. Cấu trúc dữ liệu + Thuật toán di truyền = Chương trình tiến hóa*, NXB Giáo dục, 2021.
4. An Văn Minh, Trần Hùng Cường, *Cấu trúc dữ liệu và giải thuật*, NXB Khoa học và kỹ thuật, 2023
5. Goldberg, David; Sastry, Kumara; Goldberg; *Genetic Algorithms*; Springer; ISBN-9780387353746.

**PHỤ LỤC**

Danh mục và nội dung đĩa CD kèm theo:

1. Tệp văn bản Đề Án Tốt Nghiệp
2. Chương trình nguồn