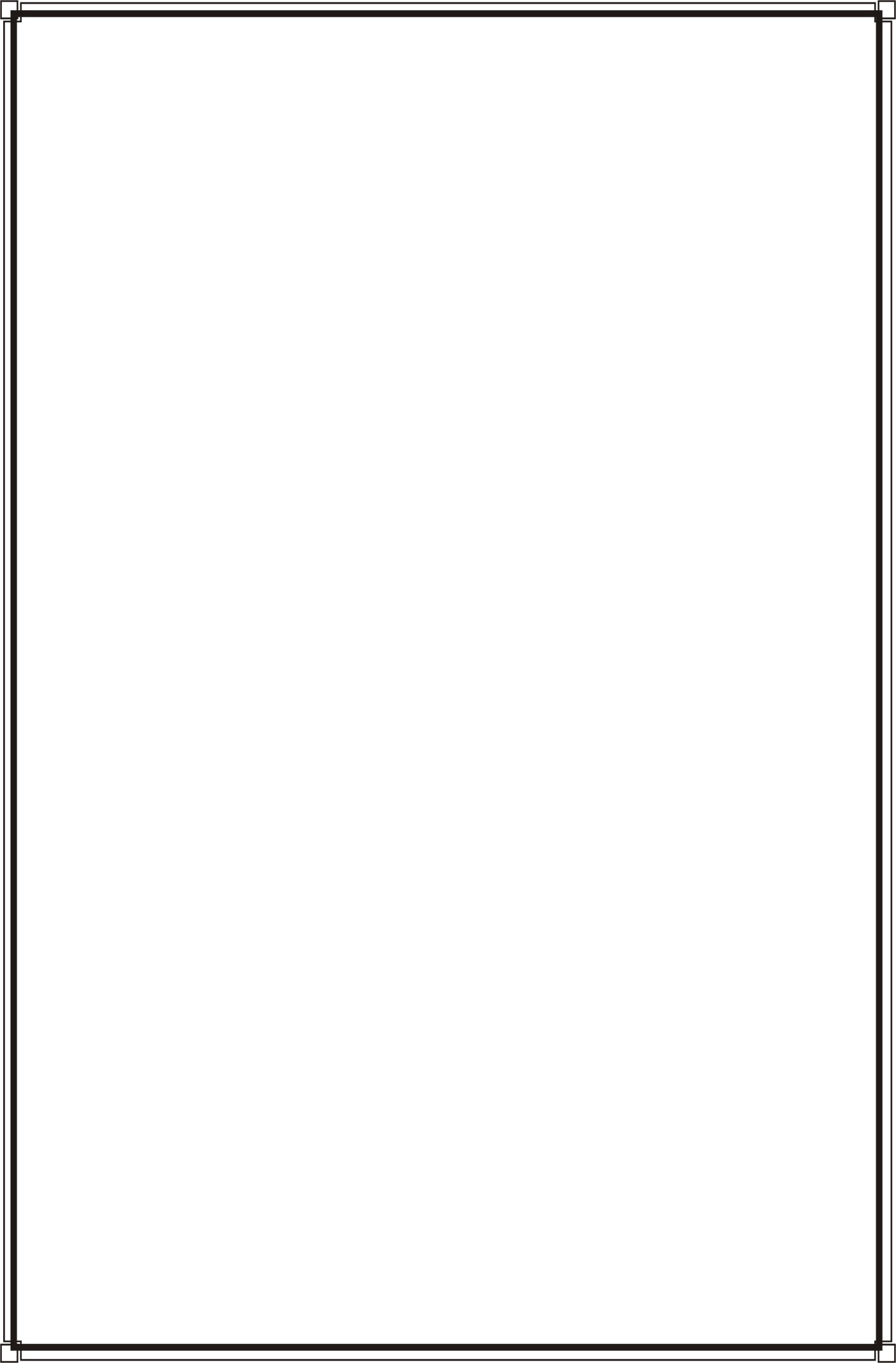
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA KHOA HỌC ỨNG DỤNG**

**-----🙠☯🙢-----**

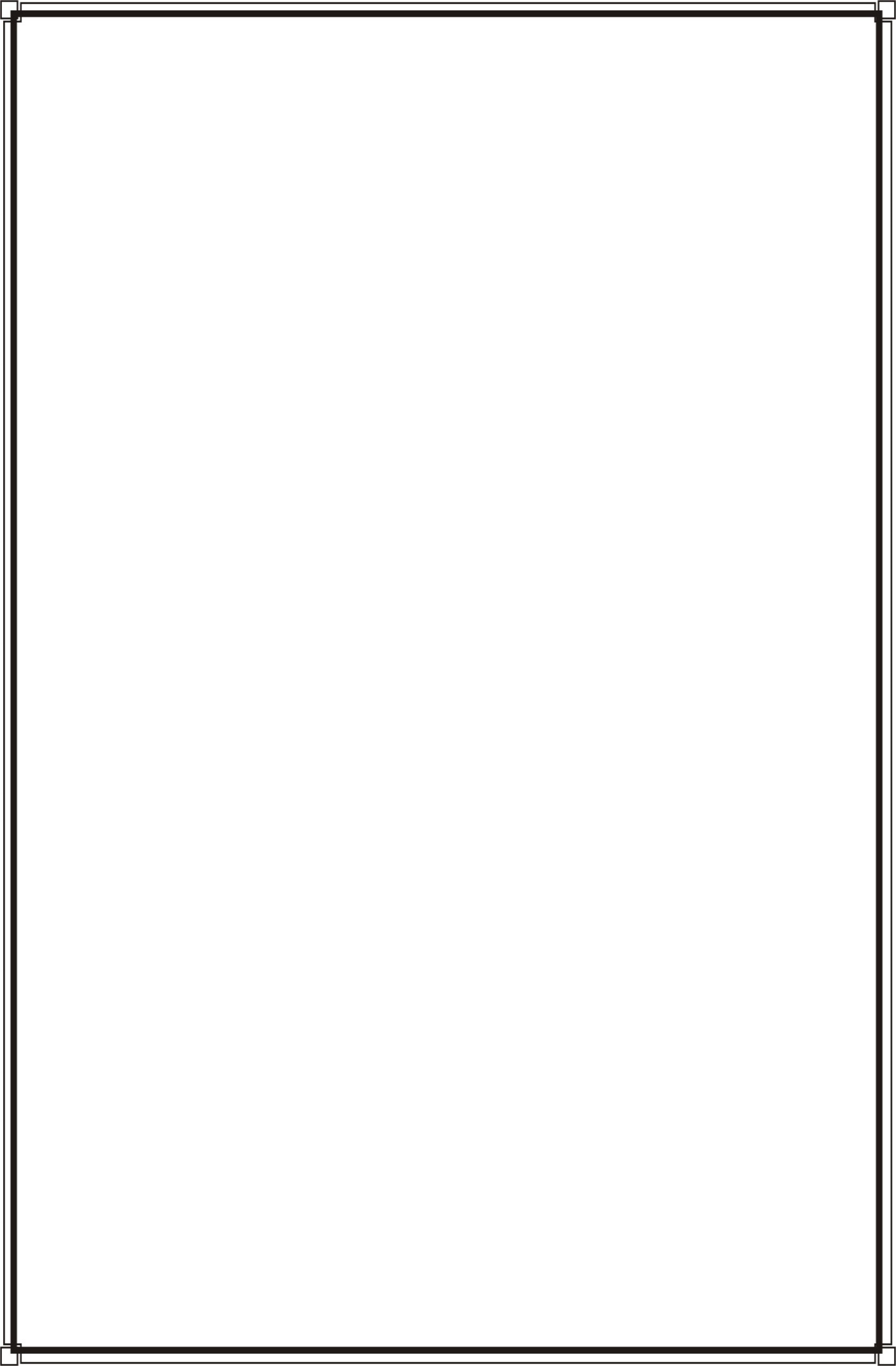


**NHÓM 7**

**TÌM HIỂU THUẬT GTS VÀ TÔ MÀU**

**BÀI TẬP NHÓM MÔN:TRÍ TUỆ NHÂN TẠO**

**TP. HCM, NĂM 2022**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA KHOA HỌC ỨNG DỤNG**

**-----🙠☯🙢-----**



**NHÓM 7**

**TÌM HIỂU THUẬT GTS VÀ TÔ MÀU**

**BÀI TẬP NHÓM MÔN:TRÍ TUỆ NHÂN TẠO**

**GVHD:Trần Đình Toàn**

**NHÓM THỰC HIỆN:**

1.Vũ Minh Nghĩa - 2001206981

**TP.HCM, NĂM 2022**

# BẢNG ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN CÔNG VIỆC NHÓM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Công việc đảm nhận** | **Nhóm đánh giá kết quả** | **Ghi chú** |
| 1 | Vũ Minh Nghĩa | Tìm Hiểu Thuật Toán, Nguyên lý thuật toán | 100% |  |
| 2 | Vũ Minh Nghĩa | Viết Chương Trình Minh Hoạ Python | 100% |  |
| 3 | Vũ Minh Nghĩa | Viết Báo Cáo Và Thuyết Trình | 100% |  |

# Lời cam đoan

Chúng em xin cam đoan đề tài *“Tìm hiểu tìm hiểu giải thuật gts và tô màu, Và viết chương trình minh hoạ”* do nhóm 7 nghiên cứu và thực hiện, không sao chép của bất kỳ nhóm hay của ai khác.

Chúng em đã kiểm tra dữ liệu theo quy định hiện hành.

Kết quả bài làm đề tài “*GTS và tô màu, Và viết chương trình ứng dụng của thuật toán*” do nhóm 7 nghiên cứu và thực hiện là trung thực và không sao chép từ bất kỳ bài tập của nhóm khác.

Các tài liệu được sử dụng trong tiểu luận có nguồn gốc, xuất xứ rõ ràng.

**(Ký và ghi rõ họ tên)**

Nghĩa

Vũ Minh Nghĩa

# LỜI CẢM ƠN

Trong quá trình học tập và thực hiện bài tiểu luận chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin,Trường Đại Học Công Nghiệp Thực Phẩm Tp.Hồ Chí Minh.

Trước hết nhóm em cảm ơn thầy Trần Đình Toàn là giảng viên đứng lớp môn Trí Tuệ Nhân Tạo của bọn em, dù dịch bệnh khó khăn vẫn tạo điều kiện cho bọn em làm bài của mình một cách tốt nhất, luôn nhiệt tình với bọn em dù trong hoàn cảnh nào đi nữa.

Mặc dù nhóm em đã có nhiều cố gắng thực hiện đề tài, có thể còn nhũng hạn chế và thiếu sót.Chúng em rất mong nhận được sự thông cảm,đóng góp ý kiến và giúp đỡ từ quý thầy cô và các bạn.

Chúng em xin cảm ơn!

Nhóm 7 thực hiện.

# MỤC LỤC

[BẢNG ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN CÔNG VIỆC NHÓM i](#_Toc118665533)

[Lời cam đoan ii](#_Toc118665534)

[LỜI CẢM ƠN iii](#_Toc118665535)

[MỤC LỤC iv](#_Toc118665536)

[PHẦN MỞ ĐẦU 1](#_Toc118665537)

[1. Thuật toán GTS 1](#_Toc118665538)

[2. Mã giả 1](#_Toc118665539)

[3. Các bước giải thuật toán 4](#_Toc118665540)

[4. Ví dụ 5](#_Toc118665541)

[5. Ứng dụng 6](#_Toc118665542)

[6. Thuật toán tô màu 6](#_Toc118665543)

[KẾT LUẬN 9](#_Toc118665544)

[TÀI LIỆU THAM KHẢO 10](#_Toc118665545)

# PHẦN MỞ ĐẦU

1. **Thuật toán GTS**

Chiến lược này bảo đảm tìm được đường đi (tối ưu) nhưng phải duyệt nhiều trạng thái, đặc biệt khi bài toán có độ sâu lời giải lớn. Các bài toán dưới đây áp dụng các chiến lược tìm kiếm heuristic (cố gắng đưa ra lời giải tốt tại mỗi bước thực hiện) và không quay lui. Do đó thuật toán này không phải vét cạn không gian tìm kiếm và chỉ tìm được những lời giải ‘đủ tốt’ ở trạng thái nào đó.

1. **Mã giả**

Trong các thập niên 1950s và 1960s, một số các nhà khoa học máy tính, trong quá nghiên cứu các hệ thống tiến hóa, đã đề xuất ý tưởng trong tất cả các hệ thống này là để phát triển một số các giải pháp ứng cử viên cho một vấn đề nào đó, bằng cách sử dụng toán tử lấy cảm hứng từ sự biến đổi di truyền tự nhiên và chọn lọc tự nhiên lấy từ thuyết tiến hóa của nhà sinh vật học Darwin.

Năm 1965, Rechenberg giới thiệu cuốn sách " Evolution Strategies "[15]-[16], trong đó ông đề xuất một phương pháp mà ông sử dụng để tối ưu hóa các thông số giá trị thực cho các thiết bị như Airfoils. Ý tưởng này đã được phát triển thêm bởi Schwefel [18]. Trong các lĩnh vực chiến lược tiến hóa chủ yếu là phát triển một cách độc lập từ các lĩnh vực của các thuật toán di truyền.

Năm 1966, Fogel, Owens, và Walsh đã phát triển "lập trình tiến hóa (evolutionary programming)", một kỹ thuật trong đó các giải pháp ứng cử viên với nhiệm vụ được biểu diễn như là máy trạng thái hữu hạn (finite−state machine), được phát triển bởi đột biến ngẫu nhiên sơ đồ trạng thái của họ và lựa chọn thích hợp nhất. Cùng với các chiến lược tiến hóa, lập trình tiến hóa, và các thuật toán di truyền tạo thành xương sống của lĩnh vực tính toán tiến hóa.

Một số nhà khoa học khác trong thập niên 1950 và thập niên 1960 đã phát triển các thuật toán tiến hóa lấy cảm hứng để tối ưu hóa và máy học như Box (1957), Friedman (1959), Bledsoe (1961), Bremermann (1962), Reed, Toombs và Baricelli (1967).

Thuật toán di truyền (Genetic algorithms - GAs) đã được phát minh bởi John Holland trong những năm 1960; sau đó được tiếp tục phát triển bởi sinh viên và đồng nghiệp của ông tại Đại học Michigan trong thập niên 1960 và thập niên 1970[9]. Ngược lại với các chiến lược phát triển và lập trình tiến hóa, mục tiêu ban đầu của Holland không để thiết kế các thuật toán để giải quyết các vấn đề cụ thể, mà là để chính thức nghiên cứu các hiện tượng của sự thích nghi khi nó xảy ra trong tự nhiên và phát triển những cách thức mà các cơ chế thích ứng tự nhiên có thể được đem vào vào hệ thống máy tính.

1975 cuốn sách Adaptation in Natural and Artificial Systems của Holland [10] đã trình bày các thuật toán di truyền như là một sự trừu tượng của sự tiến hóa sinh học và đã đưa ra một khuôn khổ lý thuyết cho thích ứng theo GA. GA của Holland là một phương pháp để di chuyển từ một dân số "nhiễm sắc thể" (ví dụ, chuỗi của những số 1 và số 0, hoặc "bits") cho một dân số mới bằng cách sử dụng một loại "chọn lọc tự nhiên" cùng với các hoạt động di truyền như lai chéo, đột biến, và đảo ngược. Mỗi nhiễm sắc thể bao gồm "Genetics" (ví dụ, bits), mỗi gen là một thể hiện của một "allele" đặc biệt (ví dụ, 0 hoặc 1). Thuật toán sẽ lựa chọn chọn những nhiễm sắc thể trong dân số sẽ được cho phép để tái sản xuất và sinh sản nhiều hơn so với những người ít phù hợp. Trao đổi chéo (Crossover) các thành phần con của hai nhiễm sắc thể bắt chước tái tổ hợp sinh học giữa hai đơn nhiễm sắc thể ("đơn bội") sinh vật. Đột biến thay đổi ngẫu nhiên (mutation randomly) các giá trị alen của một số địa điểm trong các nhiễm sắc thể; và đảo ngược thứ tự (Inversion Reverses) của một phần tiếp giáp của các nhiễm sắc thể, do đó sắp xếp lại thứ tự các gen.

Thuật toán dựa trên dân số của Holland với các với hoạt động lai ghép, đảo ngược, và đột biến là một sự đổi mới lớn và là nguồn cảm hứng cho việc giải quyết các vấn đề về tính toán.

Chẳng hạn, việc tìm kiếm tìm kiếm các chuỗi thứ tự các axit amin có thể với một protein cho trước hoặc tìm kiếm một bộ quy tắc hoặc các phương trình dự đoán những thăng trầm của thị trường tài chính. Hoặc trong điều khiển robot, làm thế nào robot có thể thực hiện một nhiệm vụ trong một điều kiện môi trường thay đổi v.v...

Thông thường, các vấn đề này đòi hỏi tìm kiếm một tập liên tục thay đổi các khả năng với số lượng lớn các khả năng để thích nghi cho điều kiện môi trường thay đổi. Các giải pháp. tìm kiếm như vậy thường yêu cầu sử dụng hiệu quả của xử lý song song, trong đó có nhiều khả năng khác nhau được khám phá cùng một lúc một cách hiệu quả để tạo ra thay đổi hàng triệu loài song song. Có thể nói, nhìn từ một mức độ cao các "nguyên tắc" của tiến hóa là khá đơn giản: các loài tiến hóa bằng phương tiện của biến ngẫu nhiên (thông qua đột biến, tái tổ hợp, và khai khác), tiếp theo là chọn lọc tự nhiên trong đó có xu hướng các thích nghi (fitness) để tồn tại và tái sản xuất; do đó truyền cho các thế hệ tương lai tạo ra sự đa dạng bất thường và phức tạp như chúng ta thấy trong sinh quyển ngày nay.

## **3. Các bước giải thuật toán**

***Thuật toán GTS1***

**Bước 1**: *Khởi đầu*

COST = 0, TOUR = , v = u( u là thành phố xuất phát)

**Bước 2**: *Thăm tất cả các thành phố*

Cho k chạy từ 1 đến n – 1

**Bước 3**: *Tìm cạnh có chi phí thấp nhất*

Tìm (v,w) là cạnh có chi phí thất nhất từ v đến các đỉnh chưa đi qua w

COST = COST + C[v,w]

TOUR = TOUR + (v,w)

v = w

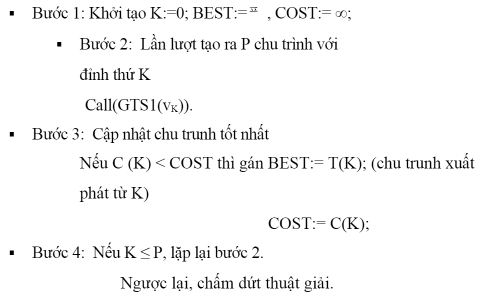
Bước 4: Quay về thành phố xuất phát

COST = COST + C[v,u]

TOUR = TOUR + (v,u)

***Thuật toán GTS2***

Giải thuật này sẽ tạo ra các lịch trình từ P thành phố xuất phát riêng biệc cho bài toán tìm chu trình đi qua N thành phố(GTS1) như đã nói ở trên, với 1 < P < N. Khi đó P chu trình đc tạo ra 1 cách tuần tự và chỉ có chu trình tốt nhất đã tìm thấy được giữ lại.



1. **Ví dụ**

Tìm hành trình tốt nhất và chi phí tương ưng theo thuật giải GTS1 với thành phố xuất phát là A.

**Giải**

**Bước** 1: COST = 0, TOUR = , v = A

**Bước 2**: k = 1 .... 5

**Bước 3**:

***K = 1***

- Từ A có thể đi đến B,C,D,E,F : {AB = 100, AC = 50, AD = 30, AE = 200, AF = 150}

→ w= D, COST = 0 + 30 = 30, TOUR = {(AD)}, v = D

***K = 2***

- Từ D có thể đi đến B,C, E,F : {DB = 70, DC = 130, DE = 50, DF = 120}

→ w= E, COST = 30 + 50 = 80, TOUR = {(AD),(DE)}, v = E

K = 3

- Từ E có thể đi đến B,C, F : {EB = 150, EC = 130, EF = 200}

→ w= C, COST = 80 + 130 = 210, TOUR = {(AD),(DE),(EC)}, v = C

***K = 4***

-Từ C có thể đi đến B, F : {CB = 20, CF = 30}

→ w= B, COST = 210 + 20 = 230, TOUR = {(AD),(DE),(EC),(CB)}, v = B

***K = 5***

*-*Từ B có thể đi đến F : {BF = 50}

→ w= F, COST = 230 + 50 = 280, TOUR = {(AD),(DE),(EC),(CB), (BF)}, v = F

**Bước 4**: Quay về A {FA = 200}

COST = 480, TOUR = {(AD),(DE),(EC),(CB), (BF),(FA)}

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| A | 0 | 100 | 50 | 30 | 200 | 150 |
| B | 120 | 0 | 30 | 80 | 120 | 50 |
| C | 80 | 20 | 0 | 120 | 100 | 30 |
| D | 40 | 70 | 130 | 0 | 50 | 120 |
| E | 180 | 150 | 130 | 70 | 0 | 200 |
| F | 200 | 80 | 60 | 100 | 150 | 0 |

1. **Ứng dụng**

***Trong lĩnh vực tối ưu hóa chi phí***

***Trong lĩnh vực trò chơi tìm đường ra mê cung***

1. **Thuật toán tô màu**
2. Khái niệm tô màu đồ thị

* Tô màu đồ thị (Graph coloring) là trường hợp đặc biệt của gán nhãn đồ thị, mà trong đó mỗi đỉnh hay mỗi cạnh hay mỗi miền của đồ thị có thể được gán bởi một màu hay một tập hợp các màu nào đó thoả mãn các ràng buộc được đặt ra trong đồ thị đó.

1. Nguồn gốc

* Thuật toán tô màu đồ thị lần đầu xuất hiện dưới dạng định lý Bốn màu trong việc tô màu bản đồ nước Anh năm 1852.
* Định lí Bốn màu được giới thiệu một cách chính thức ra công chúng bởi Arthur Caley năm 1879 tại Hội Toán học London.
* Có rất nhiều chứng minh sai về thuật toán này
* Năm 1976, Appel và Haken đã chứng minh được tính đúng đắn bằng cách sử dụng máy tính.

1. Các dạng thường gặp

* Tô màu đồ thị theo các đỉnh (vertex coloring) là gán cho mỗi đỉnh của đồ thị một màu nào đó sao cho không có hai đỉnh nào [liền kề](https://vi.wikipedia.org/w/index.php?title=Li%E1%BB%81n_k%E1%BB%81&action=edit&redlink=1) lại trùng màu nhau (dạng cơ bản nhất).
* Tô màu đồ thị theo các cạnh (edge coloring) là gán cho mỗi cạnh của đồ thị một màu nào đó sao cho không có 2 cạnh nào trùng màu.
* Tô màu đồ thị theo các miền (face coloring) là gán cho mỗi miền của [đồ thị phẳng](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%93_th%E1%BB%8B_ph%E1%BA%B3ng) một màu sao cho không có 2 miền có chung đường biên lại cùng màu.

1. Ý tưởng

* Bắt đầu từ một màu, cố gắng tô nhiều đỉnh nhất có thể.
* Tiếp tục dùng một màu khác, lại cố gắng tô các đỉnh chưa tô sao cho được càng nhiều đỉnh càng tốt.
* Quá trình này được lặp lại với những màu khác cho đến khi mọi đỉnh đều được tô màu.

1. Các bước của thuật toán

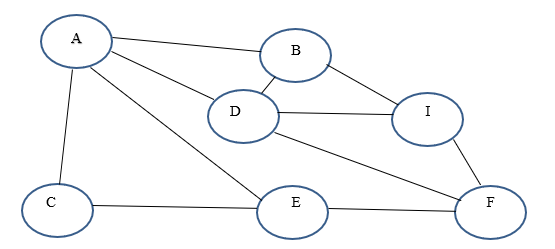
* Bước 1: Tính giá trị bậc của các đỉnh trong V. Lập danh sách V’:=[v1,v2, ...,vn] là các đỉnh của đồ thị được sắp xếp theo thứ tự bậc giảm dần: d(v1) > d(v2) > ... > d(vn). Ban đầu tất cả các đỉnh trong V (V’) đều chưa được tô màu.  
  Gán i := 1;
* Bước 2: Tô màu i cho đỉnh đầu tiên trong danh sách V’. Duyệt lần lượt các đỉnh khác trong V’(nếu có) và chỉ tô màu i cho các đỉnh không kề đỉnh đã có màu i.
* Bước 3: Kiểm tra nếu tất cả các đỉnh trong V đã được tô màu thì thuật toán kết thúc, đồ thị đã sử dụng  i màu để tô. Ngược lại, nếu vẫn còn đỉnh chưa được tô thì chuyển sang bước 4.
* Bước 4: Loại khỏi danh sách V’ các đỉnh đã tô màu. Sắp xếp lại các đỉnh trong V’ theo thứ tự bậc giảm dần. Gán i := i + 1 và quay lại bước 2.

1. Ứng dụng

* Thuật toán tô màu đồ thị thường được ứng dụng để giải quyết vấn đề tô màu các nước trên bản đồ, giải quyết bài toán Sudoku.
* Ứng dụng vào thực tiễn liên quan đến phân chia thời gian biểu làm việc, dạy học, ... Sắp xếp các thanh ghi trong quá trình biên dịch, phân chia tần số của các kênh truyền thông, bố trí các con vật trong sở thú.

1. Ví dụ

* Tô màu cho các đỉnh của đồ thị (vertex coloring) : Gán màu cho mỗi đỉnh của đồ thị, sao cho hai đỉnh kề nhau không cùng một màu, và số màu được sử dụng là ít nhất



* Để tô màu các đỉnh trong đồ thị với một màu mới, ta làm như sau:
* Bước 1: Sắp xếp các đỉnh theo thứ tự giảm dần của bậc
* Bước 2: Xét đỉnh đầu tiên (bậc cao nhất) để gán màu
* Gán màu cho các đỉnh không kề với đỉnh đang xét, đảm bảo hai đỉnh kề không cùng màu
* Lặp lại bước 2 đến hết danh sách

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Đỉnh | A | D | B | E | F | I | C |
| Bậc | 4 | 4 | 3 | 3 | 3 | 3 | 2 |
| Màu | Màu đỏ | Màu vàng | Màu xanh | Màu vàng | Màu đỏ | Màu hồng | Màu xanh |

# KẾT LUẬN

Qua đề tài nghiên cứu này, chúng em đã rút ra được nhiều bài học về ngôn ngữ lập trình Python, các giải thuật gts và tô màu. Tìm hiểu về cấu tạo của giải thuật gts và tô màu trên ngôn ngữ Python. Tuy còn nhiều mặt hạn chế về số lượng thành viên nhưng em đã cố gắng hoàn thành tốt bài tập tuần lần này. Từ đó rút ra được nhiều bài học quan trọng để phát triển tốt hơn trong học tập.

# TÀI LIỆU THAM KHẢO