**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



**BÁO CÁO THỰC TẬP CƠ SỞ**

***Đề tài: Ant Colony Algorithm***

**Sinh viên thực hiện:** *Lê Trọng Linh*

***Mã sinh viên:*** *B19DCCN374*

***Giảng viên hướng dẫn:*** *Nguyễn Văn Thủy*

**HÀ NỘI – 23/4/2022**

**1.Chủ đề: Thuật toán Ant colony optimization (ACO) áp dụng giải bài toán TSP(Travelling Salesman Problem)**

*1.1 Giới thiệu chung về bài toán tổng chi phí*

Các phương pháp để giải quyết bài toán TCO(**Total Cost of Ownership- bài toán về tổng chi phí**) hiện tại có thể được chia thành ba nhóm: phương pháp tìm kiếm (heuristic methods), phương pháp quy hoạch toán học (mathematical programming models) và các thuật toán tối ưu dựa trên nền tảng của sự tiến hóa (evolutionary-based optimization algorithms\_EOAs). Phương pháp tìm kiếm là một kỹ thuật tìm kiếm dựa trên ý kiến chủ quan của của người ra quyết định. Các phương pháp tìm kiếm tiêu biểu dùng để giải quyết bài toán TCO gồm : phương pháp Fondahl (1963), phương pháp khung (Prager 1963), phương pháp độ dốc chi phí hiệu quả (Siemens 1971),… Phương pháp quy hoạch toán học sử dụng các chương trình toán học như quy hoạch tuyến tính (linear programming\_LP), được giới thiệu bởi Kelly (1961), Hendrickson and Au (1989) và Pagnoni (1990) để mô hình hóa mối quan hệ tuyến tính giữa thời gian – chi phí. Ngoài ra, quy hoạch số nguyên (integer programming\_IP) được giới thiệu bởi Meyer & Shaffer (1963) để giải quyết cả mối quan hệ tuyến tính và rời rạc giữa thời gian – chi phí. Gần đây, Burns cùng các cộng sự (1996) đã phát triển một mô hình lai ghép LP/IP nhằm thiết lập đáp án chính xác cho bất kỳ khoảng thời gian mong muốn nào.

Cả hai phương pháp tìm kiếm và quy hoạch toán học đều có những điểm mạnh cũng như nhược điểm riêng trong việc giải quyết bài toán TCO. Tuy nhiên, đối với các dự án lớn với sơ đồ mạng lớn, thì cả phương pháp tìm kiếm cũng như phương pháp quy hoạch toán học đều không thể đạt được lời giải tối ưu một cách hiệu quả. Với mục tiêu đạt được lời giải tối ưu cho bài toán TCO, nhiều nhà nghiên cứu đã bắt đầu khám phá khả năng sử dụng các phương pháp tiên tiến, như là EOAs. EOAs (evolutionary-based optimization algorithms) là phương pháp nghiên cứu dựa trên việc mô phỏng quá trình tiến hoá của thế giới tự nhiên hoặc hành vi xã hội của các loài.

Vào đầu thập niên 90, một thuật toán với tên gọi Tối ưu đàn kiến (Ant Colony Optimization\_ACO) được đề xuất như là một phương pháp mới trong việc tìm kiếm lời giải tối ưu cho những bài toán tối ưu đa mục tiêu. ACO lần tiên được ứng dụng để giải quyết bài toán người thương gia TSP (Traveling Salesmen Problem), và gần đây nó đã được mở rộng và cải tiến để áp dụng cho nhiều bài toán tối ưu khác nhau.

*1.2.Giới thiệu bài toán TSP*

Bài toán Travelling Salesman Problem (TSP) được nêu ra lần đầu tiên năm 1930 và đã trở thành một bài toán khó khi độ phức tạp của thuật toán gia tăng theo cấp số mũ. Nó là một trong những bài toán được nghiên cứu sâu nhất trong tối ưu hóa. Nó thường được dùng làm thước đo cho nhiều phương pháp tối ưu hóa.

Bài toán được phát biểu như sau:  Có một người giao hàng cần đi giao hàng tại n thành phố. Xuất phát từ một thành phố nào đó, đi qua các thành phố khác để giao hàng và trở về thành phố ban đầu. Mỗi thành phố chỉ đến một lần, khoảng cách từ một thành phố đến các thành phố khác là xác định được. Giả thiết rằng mỗi thành phố đều có đường đi đến các thành phố còn lại. Khoảng cách giữa hai thành phố có thể là khoảng cách địa lý, có thể là cước phí di chuyển hoặc thời gian di chuyển. Ta gọi chung là độ dài. Hãy tìm một chu trình (một đường đi khép kín thỏa mãn điều kiện trên) sao cho tổng độ dài các cạnh là nhỏ nhất. Hay còn nói là tìm một phương án có giá nhỏ nhất. Bài toán này cũng được gọi là bài toán người du lịch.

Một cách tổng quát, có thể không tồn tại một đường đi giữa hai thành phố a và b nào đó. Trong trường hợp đó ta cho một đường đi ảo giữa a và b với độ dài bằng ¥.

Bài toán có thể biểu diễn bởi một đồ thị vô hướng có trọng số G = (V,E), trong đó mỗi thành phố được biểu diễn bởi một đỉnh, cạnh nối hai đỉnh biểu diễn cho đường đi giữa hai thành phố và trọng số của cạnh là khoảng cách giữa hai thành phố. Một chu trình đi qua tất cả các đỉnh của G, mỗi đỉnh một lần duy nhất, được gọi là chu trình Hamilton. Vấn đề là tìm một chu trình Hamilton mà tổng độ dài các cạnh là nhỏ nhất.

*1.3. Giới thiệu cơ chế hoạt động của thuật toán ACO*

ACO (Ant Colony Optimization – Tối ưu đàn kiến) là một phương pháp nghiên cứu lấy cảm hứng từ việc mô phỏng hành vi của đàn kiến trong tự nhiên nhằm mục tiêu giải quyết các bài toán tối ưu phức tạp.

Được giới thiệu lần đầu tiên vào năm 1991 bởi A. Colorni và M. Dorigo, Giải thuật kiến đã nhận được sự chú ý rộng lớn nhờ vào khả năng tối ưu của nó trong nhiều lĩnh vực khác nhau. Khái niệm ACO lấy cảm hứng từ việc quan sát hành vi của đàn kiến trong quá trình chúng tìm kiếm nguồn thức ăn. Người ta đã khám phá ra rằng, đàn kiến luôn tìm được đường đi ngắn nhất từ tổ của chúng đến nguồn thức ăn. Phương tiện truyền đạt tín hiệu được kiến sử dụng để thông báo cho những con khác trong việc tìm đường đi hiệu quả nhất chính là mùi của chúng (pheromone). Kiến để lại vệt mùi trên mặt đất khi chúng di chuyển với mục đích đánh dấu đường đi cho các con theo sau. Vệt mùi này sẽ bay hơi dần và mất đi theo thời gian, nhưng nó cũng có thể được củng cố nếu những con kiến khác tiếp tục đi trên con đường đó lần nữa. Dần dần, các con kiến theo sau sẽ lựa chọn đường đi với lượng mùi dày đặc hơn, và chúng sẽ làm gia tăng hơn nữa nồng độ mùi trên những đường đi được yêu thích hơn. Các đường đi với nồng độ mùi ít hơn rốt cuộc sẽ bị loại bỏ và cuối cùng, tất cả đàn kiến sẽ cùng kéo về một đường đi mà có khuynh hướng trở thành đường đi ngắn nhất từ tổ đến nguồn thức ăn của chúng (Dorigo và Gambardella 1996).

**2.Cách giải**

*2.1 Cơ sở lý thuyết*

Trong tự nhiên loại kiến hoạt động theo quy luật sau: Khi tìm kiếm thức ăn, trong quá trình di chuyển kiến thường để lại mùi riêng của nó để các con kiến trong đàn có thể nhận diện ra lối di chuyển của chúng, mùi của kiến để lại trên đường đi đó được gọi là pheromone. Đây là yếu tố giúp đàn kiến đánh dấu và trao đổi thông tin với nhau khi chúng đi tìm nguồn thức ăn. Khi tìm thấy nguồn thức ăn, mỗi con kiến sẽ di chuyển từ tổ tới nơi có nguồn thức ăn theo đường đi đã tìm thấy. Từ tổ kiến đến nơi có thức ăn sẽ có nhiều đường đi. Vì thế ban đầu các chú kiến sẽ đi ngẫu nhiên theo các con đường từ điểm xuất phát đến đích. Tuy nhiên, sau một thời gian di chuyển nhất định, các chú kiến trong đàn sẽ tìm ra được đường đi ngắn nhất từ tổ tới nguồn thức ăn. Sau khi quan sát và nghiên cứu hoạt động của đàn kiến trong tự nhiên, chúng ta nhận thấy rằng quá trình tìm đường đi ngắn nhất từ tổ tới nguồn thức ăn dựa trên các quy luật sau: Khi di chuyển kiến sẽ dùng phoremone để đánh dấu và trao đổi thông tin với nhau nhằm tìm ra đường đi ngắn nhất từ tổ đến nguồn thức ăn, đồng thời trong quá trình di chuyển đó chúng để lại một lượng mùi (phoremone) trên đường chúng di chuyển qua. Các con kiến đi sau sẽ dựa vào lượng mùi mà các con kiến đi trước để lại để tiến hành tìm đường đi cho mình, khi trên đường đi chưa có mùi để lại thì chúng di chuyển một cách ngẫu nhiên, lượng mùi để lại trên đường đi của các con kiến di chuyển trước cũng bị bay hơi theo thời gian. Đường đi nào có lượng thông tin mùi để lại càng nhiều thì xác suất được các con kiến chọn càng cao, ngược lại đường đi có lượng mùi thấp thì xác suất chọn càng thấp. Qua một quá trình di chuyển đàn kiến sẽ chọn được cho mình một đường đi ngắn nhất từ tổ đến nguồn thức ăn. Dựa vào cơ chế hoạt động của đàn kiến trong tự nhiên mà thuật toán Ant Colony đã ra đời. Thuật toán Ant Colony mô phỏng hoạt động của đàn kiến nhân tạo như sau: Thuật toán sử dụng một đàn kiến nhân tạo (Artificial Ants) để mô phỏng lại các hoạt động trong tự nhiên của đàn kiến. Đàn kiến nhân tạo có một số điều chỉnh về mặt hoạt động so với đàn kiến tự nhiên nhằm tăng tính hiệu quả của thuật toán. Giải thuật này chủ yếu mô tả hoạt động tìm đường đi ngắn nhất của các con kiến nhân tạo dựa vào lượng mùi để lại trong quá trình di chuyển của đàn kiến. Cụ thể về hoạt động của đàn kiến nhân tạo được trình bày như sau: Cho một đồ thị đầy đủ gồm các đỉnh và cạnh trong đó đỉnh là nơi kiến sẽ đi qua và các cạnh là các đoạn đường. Nút ban đầu cho đường đi của mỗi con kiến được chọn một cách ngẫu nhiên và thông tin mùi (pheromone) được tính toán và đặt trên mỗi đoạn đường. Mỗi con kiến sẽ chọn đường đi theo các nguyên tắc:

+ Dựa vào thông tin mùi để lại có trên các đoạn đường để tính xác suất chọn đoạn đường tiếp theo và làm đường đi cho con kiến.

+ Đường đi có lượng mùi (pheromone) nhiều hơn thì sẽ được gán một xác suất lớn hơn. Ngược lại các đoạn đường đi có lượng thông tin mùi thấp hơn sẽ có xác suất được chọn thấp hơn. Con kiến sẽ tìm đường đi cho tới khi hoàn thành một đường đi của nó (thỏa mãn điều kiện dừng của con kiến).

+ Một đường đi đầy đủ và hoàn chỉnh được gọi là một lời giải đúng cho bài toán đặt ra. Các đường đi tìm được sẽ được phân tích, so sánh và đánh giá để tìm phương án tối ưu nhất có thể. Đó là lời giải tối ưu của bài toán.

+ Sau khi tất cả kiến trong đàn hoàn thành việc tìm đường đi của nó, ta sẽ tiến hành cập nhật thông tin mùi cho các cung. Số lượng của mùi sẽ được tính toán và điều chỉnh để tìm được phương án tối ưu tốt hơn:

• Các lời giải có đường đi ngắn hơn sẽ có khối lượng pheromone lớn hơn để đặt trên các cung đã được đi qua. Ngược lại, các lời giải cho đường đi dài hơn sẽ có khối lượng pheromone bé hơn.

• Con kiến sẽ chọn đường đi có lượng mùi lớn hơn và xác suất chọn đường đi đó cũng cao hơn.

**Thuật toán ACO**

|  |
| --- |
| **Input:** Số kiến, khởi tạo số thành phố, vị trí tọa độ các thành phố, điểm xuất phát, khởi tạo mùi.  **Output:** Chu trình đường đi ngắn nhất từ thành phố xuất phát qua các thành phố còn lại một lần và quay về lại thành phố ban đầu  **Method:**  **Begin:**  B1:Nhập dữ liệu đầu vào: số kiến, vị trí tọa độ các thành phố, khoảng cách giữa các thành phố, điểm xuất phát, khởi tạo mùi.  B2: Xây dựng phương án lựa chọn đường đi  B3: Cập nhật mùi  B4: Nếu chu trình đường đi là ngắn nhất thì qua  B5 ngược lại thì quay lại B2  **End** |

*2.2. Xây dựng giải pháp*

- Xây dựng công thức xác suất lựa chọn đường đi cho con kiến k theo thuật toán Ant Colony: Chọn con kiến k trong số m con đang ở thành phố i, chọn ngẫu nhiên thành phố j với xác suất pij. Xác suất chọn ngẫu nhiên thành phố i được tính theo công thức sau:



:Xác xuất con kiến k chọn cạnh(i,j)

:Nồng độ vết mùi trên cạnh(i,j)

:Thông tin hueristic

:Tập các đỉnh láng giềng của i mà con kiến k chưa đi qua

α :Hệ số điều chỉnh ảnh hưởng

β :Hệ số điều chỉnh ảnh hưởng

- Xây dựng công thức cập nhật mùi theo thuật toán Ant Colony

+ Bay hơi mùi các thành phố không nằm trong hành trình tốt nhất:

+ Gia tăng mùi cho các thành phố nằm trong hành trình tốt nhất. 

P: Tốc độ bay hơi

nếu kiến đi qua cạnh (i,j)

=

0 nếu ngược lại

:là quãng đường mà con kiến k đã đi qua

*2.3 Lưu đồ thuật toán:*

Bắt đầu

Khởi tạo điều kiện ban đầu cho đàn kiến

Xây dựng phương án

Cập nhật mùi

Các hoạt động khác

No

Kết thúc ?

yes

Trở lại phương án tốt nhất

*2.4* *Cách tạo dữ liệu để chạy thử với thuật toán:*

-Tạo các bộ test trong file .txt chứa các tọa độ điểm

- Dùng phần mềm MatLap để cài đặt thuật toán (chạy code tối ưu đưa ra lộ trình, đồng thời vẽ tọa độ điểm và đường đi nhờ công cụ trên Matlap)

2.5 Nội dung code và kết quả demo

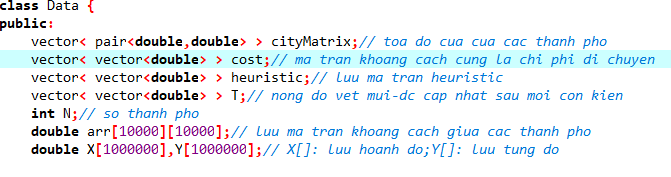
2.5.1 Nội dung code

* **Code C++:**

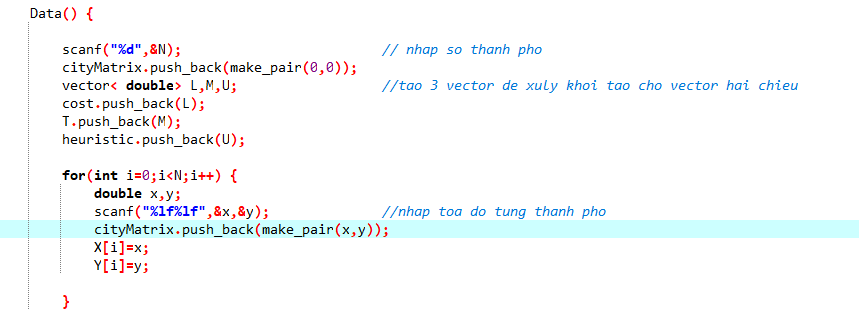
-Sử dụng 3 class và 1 hàm main:class Data,class Ant,class ACO,hàm main để run chương trình

* Class Data

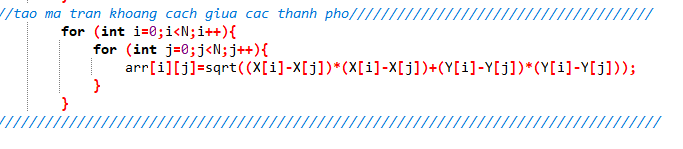
-Khởi tạo các biến cơ sở:



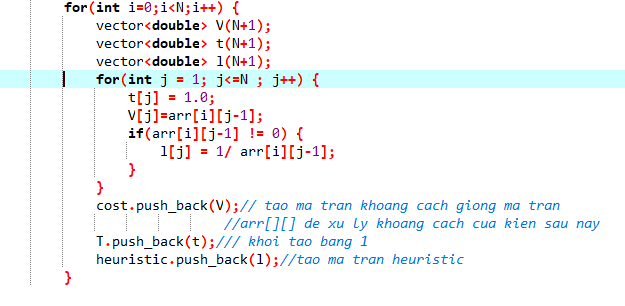
-Nhập vào số thành phố và tọa độ từng thành phố



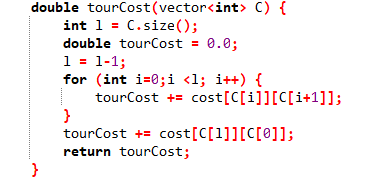
-Thiết lập ma trận khoảng cách(chi phí di chuyển) giữa các thành phố



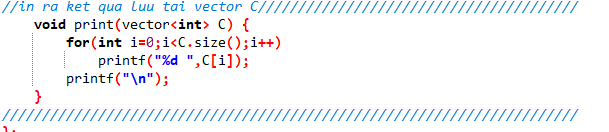
-Thiết lập ma trận heuristic, vector lưu nồng độ mùi



-Tạo hàm xử lý chi phí của chu trình cho trước(chi phí di chuyển của mỗi con kiến)

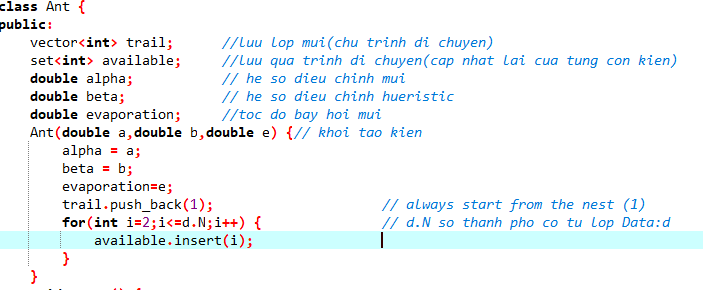


-Tạo hàm in ra chu trình(với chu trình được lưu vào vector)

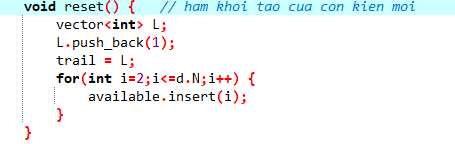


* Class Ant

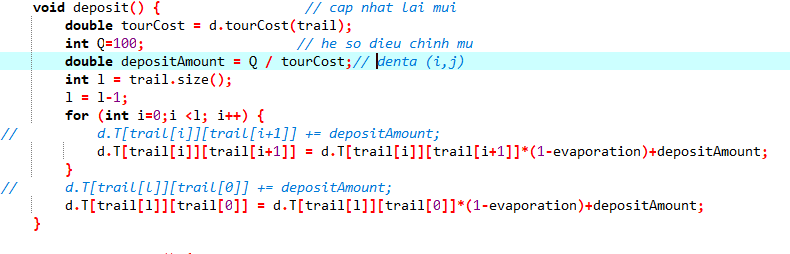
-Các biến khởi tạo



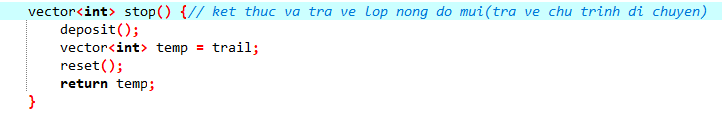
-Hàm thiết lập lại sau khi một con kiến kết thúc chu trình



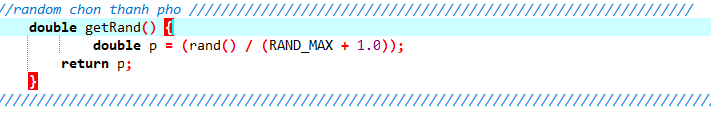
-Hàm cập nhật lại mùi khi một con kiến kết thúc chu trình



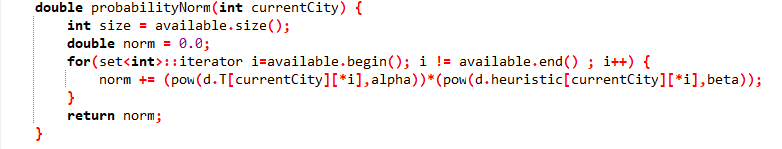
-Hàm trả về chu trình đã di chuyển



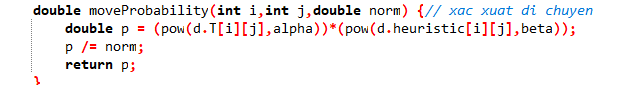
-Hàm random xác xuất để chọn thành phố kế tiếp



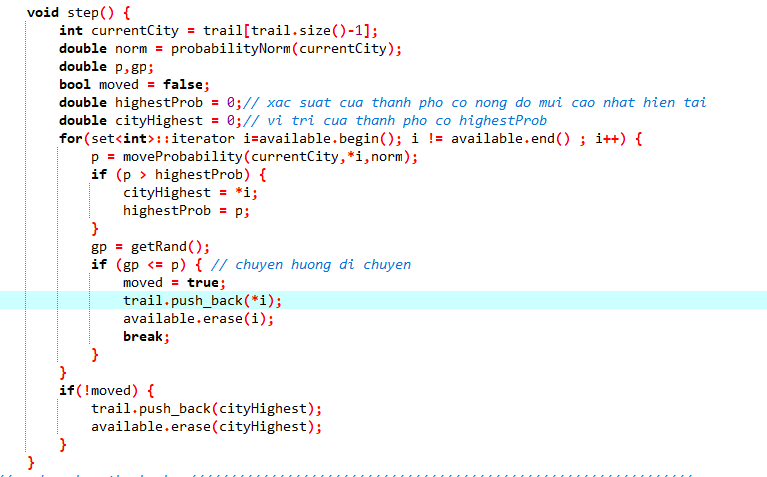
-Hàm để tính mẫu của công thức tính xác suất di chuyển



-Hàm tính xác suất di chuyển ( giá trị biến norm là kết quả của hàm tính mẫu bên trên)

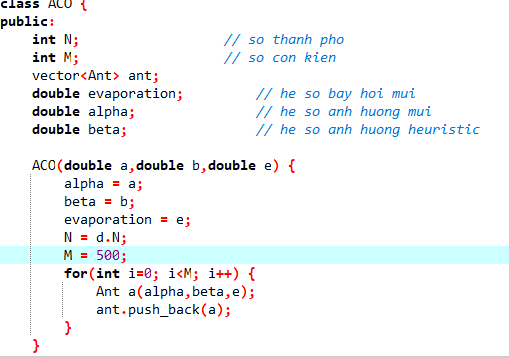


-Hàm xử lý một lần di chuyển của kiến( ví dụ như từ thành phố A sang thành phố B theo nồng độ mùi)

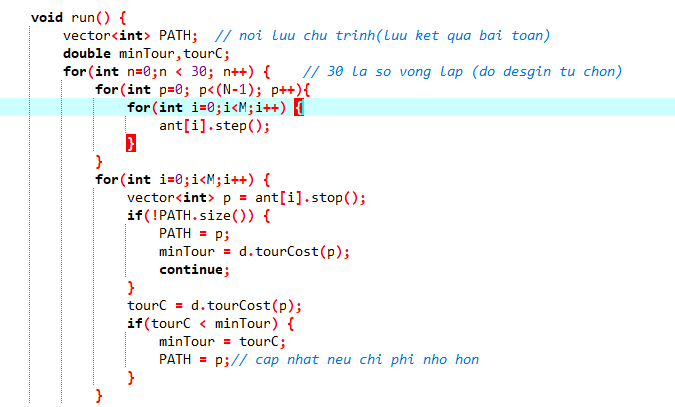


* Class ACO

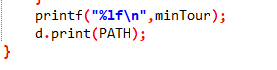
-Các biến khởi tạo



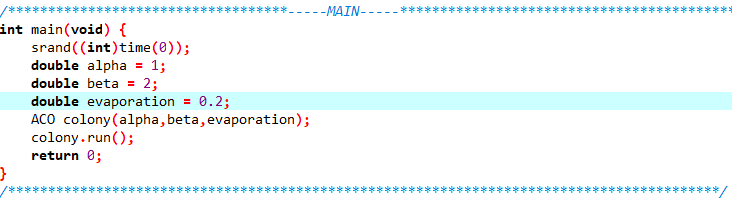
-Hàm xử lý vòng lặp,chu trình của từng con kiến để đưa ra kết quả



-Hàm in ra kết quả với hàm print từ class Data



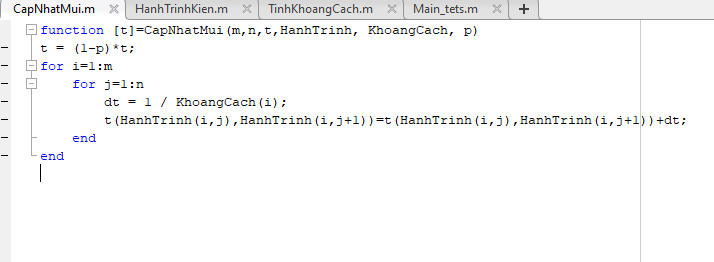
* Hàm main



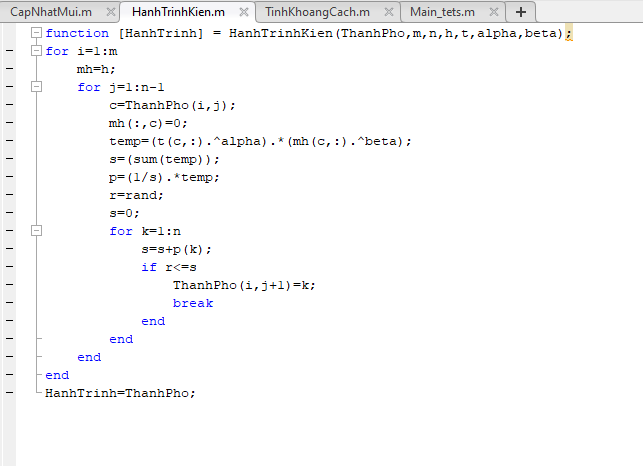
* Code MatLap( cơ bản giống code C++, có thêm phần vẽ tọa độ điểm)

-Chia làm nhiều Function để hàm Main dễ gọi vào để xử lý các công thức xác xuất và tính tổng chi phí di chuyển

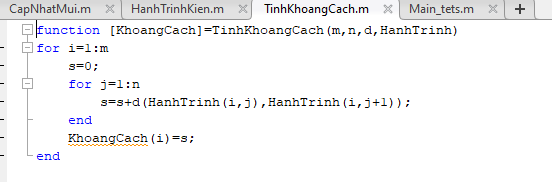
-Hàm Cập nhật mùi



-Hàm tính chu trình của một con kiến

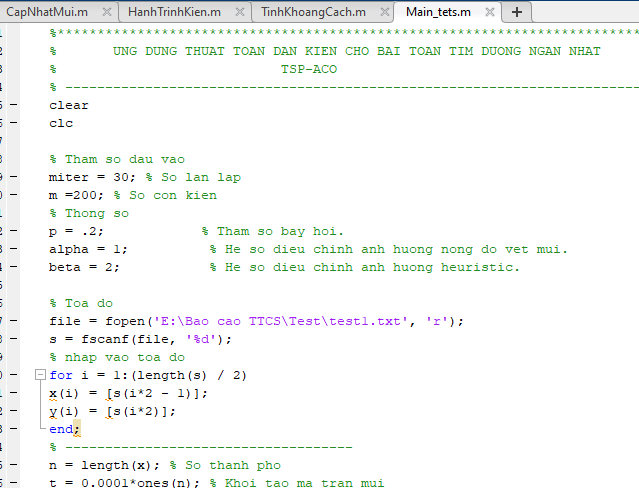


-Hàm tính chi phí

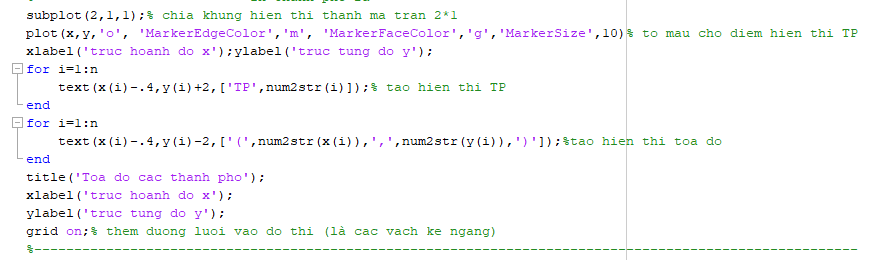


-Hàm Main

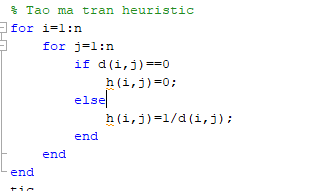
+ Các biến khởi tạo



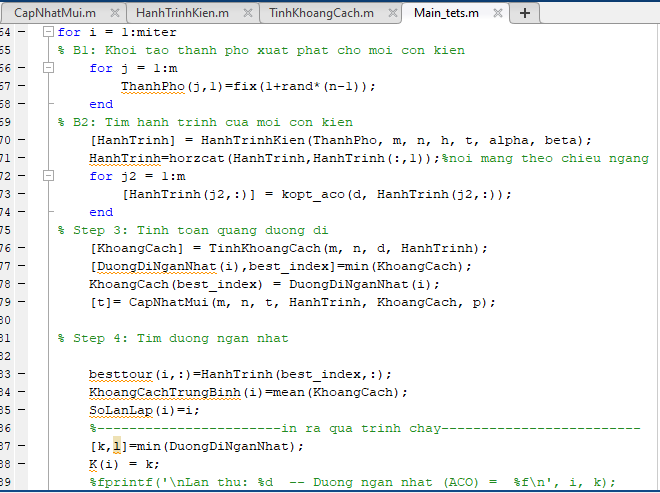
+Xử lý hiển thị



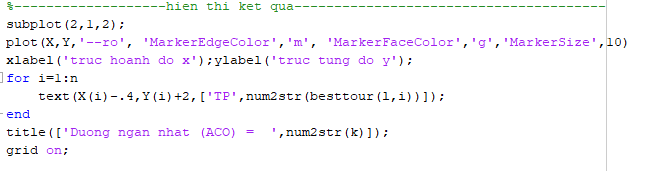
+Tạo ma trận Heuristic



+Xử lý gọi các Function vào thực thi

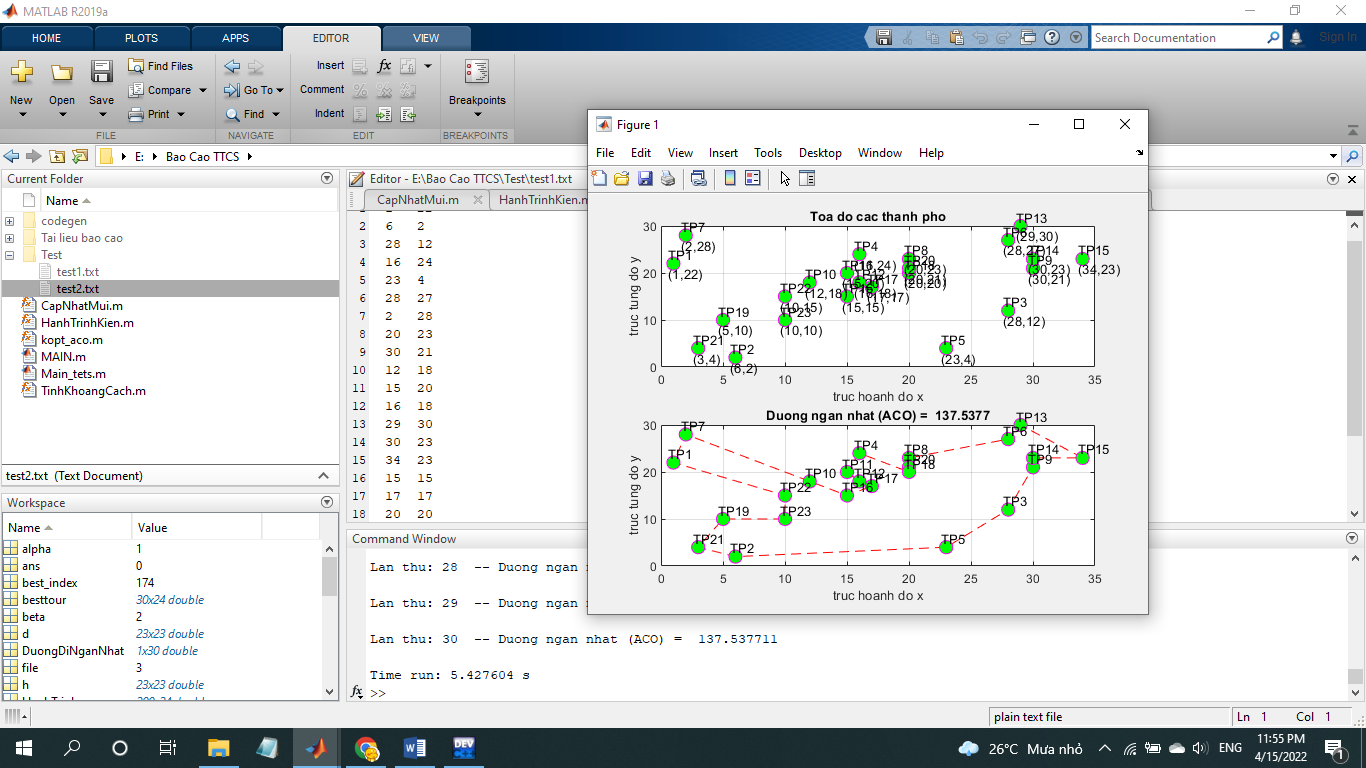


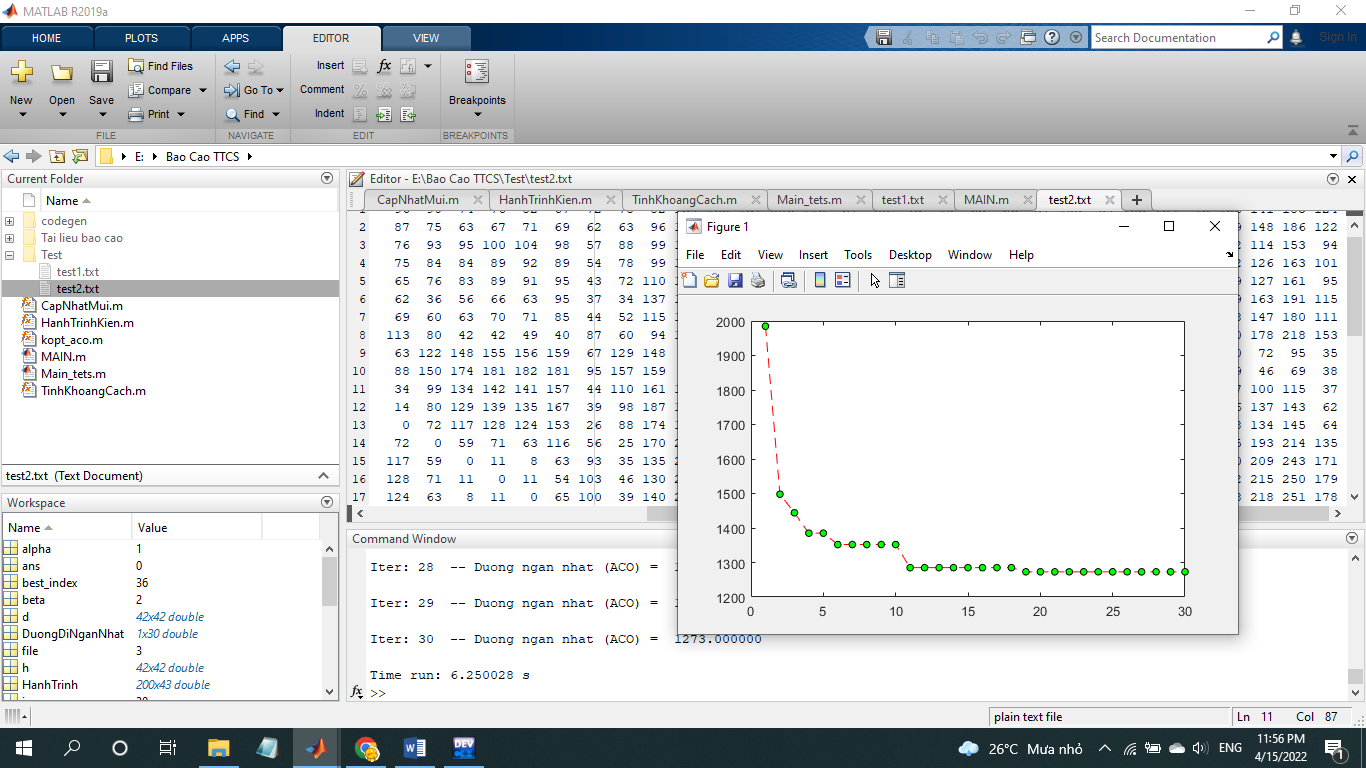
+Xử lý hiển thị kết quả



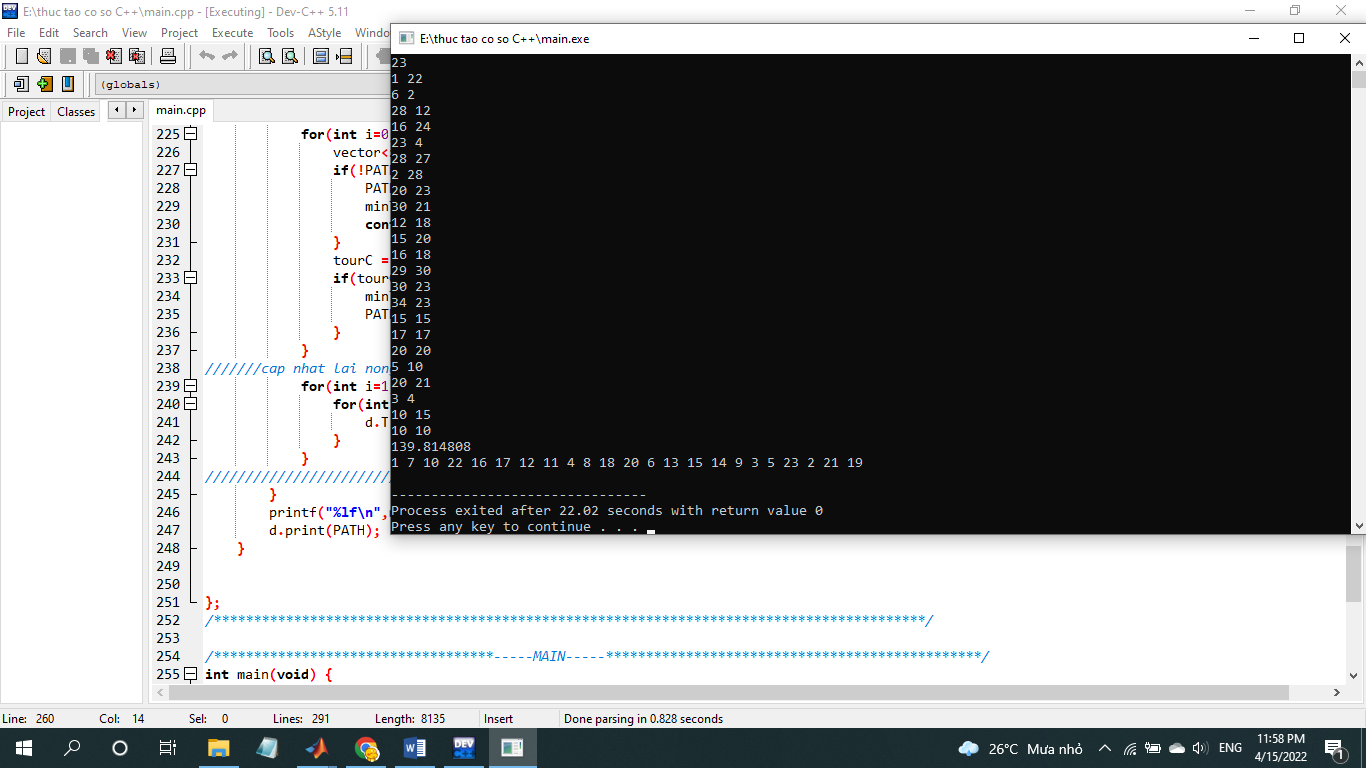
**2.5.2 Kết quả demo**

-Sử dụng MatLap





-Sử dụng code C++



**3.Tài liệu tham khảo**

[1] Phạm Hồng Luân, Dương Thành Nhân, “Nghiên cứu ứng dụng thuật toán aco (ant colony optimization) tối ưu thời gian và chi phí cho dự án xây dựng”, Tạp chí Phát triển Khoa học và Công nghệ, tập 13, số q1– 2010.

[2] Đỗ Đức Đông và Hoàng Xuân Huấn, ”Multi-level Ant System: A new approach through the new pheromone update of Ant Colony Optimization”, Kỷ yếu Hội nghị quốc tế Khoa học máy tính RIVF lần thứ 4, Tp.Hồ Chí Minh, Tháng 2/2006.

[3] Marco Dorigo, Luca Maria Gambardella, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”, Accepted for publication in the IEEE Transactions on Evolutionary Computation, Vol.1, No.1, 1997

[4] kênh youtube : https://www.youtube.com/channel/UCvT69hssHU42\_ZJCAxXhNsg/search?query=ACO