

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG**

ĐẶNG QUÝ LINH

**NGHIÊN CỨU ỨNG DỤNG GIẢI THUẬT ĐÀN KIẾN
ĐỂ GIẢI QUYẾT BÀI TOÁN NGƯỜI DU LỊCH**

Chuyên ngành: Khoa học máy tính

Mã số: 60 48 01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng – Năm 2013

Công trình được hoàn thành tại

ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: **TS. Nguyễn Tấn Khôi**

Phản biện 1: Nguyễn Văn Hiệu

Phản biện 2: Nguyễn Mậu Hân

Luận văn sẽ được bảo vệ trước Hội đồng chấm Luận văn
tốt nghiệp thạc sĩ khoa học máy tính họp tại Đại học Đà Nẵng vào
ngày 16 tháng 11 năm 2013

* Có thể tìm hiểu luận văn tại:

- Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng
- Trung tâm Học liệu, Đại học Đà Nẵng

MỞ ĐẦU

1. Lý do chọn đề tài

Bài toán Người du lịch (Travelling Salesman Problem - TSP) là một trong những bài toán kinh điển và khó trong tin học [1][2][3][4][5][6]. Bài toán có phát biểu rất đơn giản nhưng rất khó giải trong trường hợp tổng quát với không gian tìm kiếm rộng lớn, khó bởi các thuật toán hiệu quả nhất đã được biết đến có thời gian giải quyết bài toán này tăng dần theo cấp số nhân của n , hay độ phức tạp thuật toán tăng theo hàm số mũ [14]. Có rất nhiều cách tiếp cận giải bài toán này ngay từ khi nó mới ra đời, như sử dụng quy hoạch tuyến tính, thuật toán vét cạn, thuật toán người láng giềng gần nhất, kỹ thuật nhánh và cận, nhưng mới chỉ dừng lại ở các bộ dữ liệu nhỏ. Gần đây có nhiều thuật toán ra đời theo hướng tiếp cận về tiến hóa như thuật toán di truyền Genetic Algorithm hay cách mô phỏng hành vi của đàn kiến như thuật toán đàn kiến được áp dụng cho kết quả tốt hơn rất nhiều.

Thuật toán đàn kiến do Thomas Stutzle và Marco Dorigo đề xuất là một thuật toán độc đáo và có thể áp dụng cho nhiều bài toán tối ưu tổ hợp với một bộ dữ liệu lớn. Thuật toán kiến mô phỏng hành vi của đàn kiến trong tự nhiên nhằm tìm kiếm đường đi ngắn nhất giữa tổ kiến và nguồn thức ăn dựa trên mật độ mùi (pheromone) mà các con kiến để lại trên đường đi [1][3][4][5]. Người ta đã áp dụng rất thành công các thuật toán đàn kiến trong các bài toán tối ưu như bài toán người đưa thư, bài toán gán, bài toán tô màu đồ thị, bài toán lập lịch và bài toán nổi tiếng nhất là bài toán người du lịch. Từ bài toán người du lịch này có thể áp dụng cho nhiều tình huống trong

thực tế như: lập lịch tối ưu cho dự án, sắp xếp các hành trình du lịch, định tuyến trong các mạng viễn thông... [2][5][10]

Hiệu quả của thuật toán đàn kiến đã được thể hiện khi so sánh với các thuật toán nổi tiếng khác như thuật toán di truyền (Genetic Algorithm), thuật toán mô phỏng luyện kim (Simulated Annealing), thuật toán tìm kiếm Tabu (Tabu-Search) [2][29]

Xuất phát từ nhu cầu tìm đường đi ngắn nhất với một giải thuật tốt cho không gian tìm kiếm rộng lớn, áp dụng được cho nhiều bài toán tối ưu tổ hợp trong thực tế nên tôi chọn đề tài: *“Nghiên cứu ứng dụng thuật toán đàn kiến để giải bài toán người du lịch”* nhằm tìm hiểu thuật toán đàn kiến, xem xét hiệu quả của thuật toán đàn kiến áp dụng vào bài toán tối ưu tổ hợp và so sánh tính hiệu quả của thuật toán đàn kiến với thuật toán di truyền.

2. Mục tiêu và nhiệm vụ nghiên cứu

2.1. Mục tiêu

- Áp dụng thuật toán tối ưu đàn kiến ACO để tìm đường đi ngắn nhất trong bài toán Người du lịch
- Xây dựng ứng dụng mô phỏng bài toán người du lịch giải bằng thuật toán tối ưu đàn kiến ACO
- Đánh giá hiệu quả của thuật toán tối ưu đàn kiến ACO so với thuật toán di truyền trong bài toán người du lịch

2.2. Nhiệm vụ chính của đề tài

- Tìm hiểu về bài toán người du lịch
- Tìm hiểu các thuật toán truyền thống và thuật toán di truyền cho bài toán người du lịch
- Tìm hiểu thuật toán tối ưu đàn kiến ACO
- Áp dụng thuật toán ACO vào bài toán người du lịch

- Đánh giá hiệu quả của thuật toán tối ưu đàn kiến ACO so với thuật toán di truyền trong việc giải bài toán người du lịch
- Xây dựng chương trình giải quyết bài toán người du lịch với số lượng dữ liệu lớn.

3. Đối tượng và phạm vi nghiên cứu

3.1. Đối tượng nghiên cứu

- Bài toán người du lịch
- Lý thuyết về thuật toán truyền thống và thuật toán di truyền giải bài toán người du lịch
- Lý thuyết về thuật toán đàn kiến

3.2. Phạm vi nghiên cứu

- Nghiên cứu thuật toán đàn kiến để xây dựng ứng dụng giải bài toán người du lịch, qua đó đánh giá hiệu quả của thuật toán đàn kiến so với thuật toán di truyền

4. Phương pháp nghiên cứu

4.1. Phương pháp lý thuyết

- Nghiên cứu tìm hiểu các phương pháp giải bài toán người du lịch
- Nghiên cứu về thuật toán di truyền GA áp dụng cho bài toán người du lịch
- Cơ sở lý thuyết về thuật toán đàn kiến
- Cơ sở lý thuyết về thuật toán đàn kiến áp dụng cho bài toán người du lịch

4.2. Phương pháp thực nghiệm

- Lựa chọn ngôn ngữ lập trình để cài đặt thuật toán
- Thực nghiệm thuật toán trên bộ dữ liệu thử nghiệm
- Đánh giá, kiểm tra kết quả

5. Dự kiến kết quả

5.1. Kết quả lý thuyết

- Hiểu được thuật toán giải bài toán người du lịch truyền thống
- Nắm vững được thuật toán đàn kiến

5.2. Kết quả thực tiễn

- Xây dựng chương trình ứng dụng thuật toán đàn kiến ACO để giải bài toán người du lịch
- Đánh giá hiệu quả thuật toán đàn kiến so với thuật toán di truyền

6. Ý nghĩa khoa học và thực tiễn

6.1. Ý nghĩa khoa học

- Áp dụng lý thuyết của thuật toán đàn kiến ACO để áp dụng trong các bài toán tối ưu tổ hợp
- So sánh và đánh giá hiệu quả của thuật toán di truyền và thuật toán đàn kiến ACO trong giải bài toán người du lịch

6.2. Ý nghĩa thực tiễn

- Thuật toán đàn kiến có thể áp dụng trong các bài toán thực tế: lập lịch đi hành trình với chi phí tối thiểu, định tuyến trên mạng,...

7. Cấu trúc luận văn

Bố cục của luận văn gồm 3 chương với các nội dung như sau:

Chương 1: Tổng quan đề tài. Tìm hiểu nghiên cứu lý thuyết liên quan về đồ thị; lý thuyết về bài toán tối ưu tổ hợp; tìm hiểu nội dung bài toán người du lịch và các phương pháp giải bài toán người

du lịch: thuật toán vét cạn, thuật toán láng giềng gần nhất, thuật toán tìm kiếm cục bộ, thuật toán nhánh cận và thuật toán di truyền.

Chương 2: Thuật toán tối ưu đàn kiến ACO. Tìm hiểu về nội dung thuật toán đàn kiến; thuật toán đàn kiến giải bài toán người du lịch; thuật toán tối ưu đàn kiến ACO bao gồm các thuật toán Ant System, Max-Min Ant System và Ant Colony System; cách thức nâng cao hiệu quả của thuật toán ACO và các ứng dụng của thuật toán ACO.

Chương 3: Ứng dụng thuật toán ACO vào bài toán người du lịch. Phân tích yêu cầu của bài toán, từ đó phân tích các chức năng, xây dựng chương trình ứng dụng vào bài toán người du lịch đồng thời tiến hành chạy thử, đánh giá kết quả; và so sánh tính hiệu quả của thuật toán tối ưu đàn kiến ACO với thuật toán di truyền.

Cuối cùng là kết luận và hướng phát triển của đề tài.

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

Chương này tìm hiểu nghiên cứu lý thuyết liên quan về đồ thị; lý thuyết về bài toán tối ưu tổ hợp; tìm hiểu nội dung bài toán người du lịch và các phương pháp giải bài toán người du lịch: thuật toán vét cạn, thuật toán láng giềng gần nhất, thuật toán tìm kiếm cục bộ, thuật toán nhánh cận và thuật toán di truyền.

1.1. CÁC KHÁI NIỆM CƠ BẢN VỀ ĐỒ THỊ

1.1.1. Định nghĩa đồ thị

1.1.2. Bậc của đỉnh

1.1.3. Tính liên thông của đồ thị

1.1.4. Biểu diễn đồ thị trên máy tính

1.1.5. Đồ thị Euler và đồ thị Hamilton

1.2. BÀI TOÁN TỐI ƯU TỔ HỢP

1.3. BÀI TOÁN NGƯỜI DU LỊCH

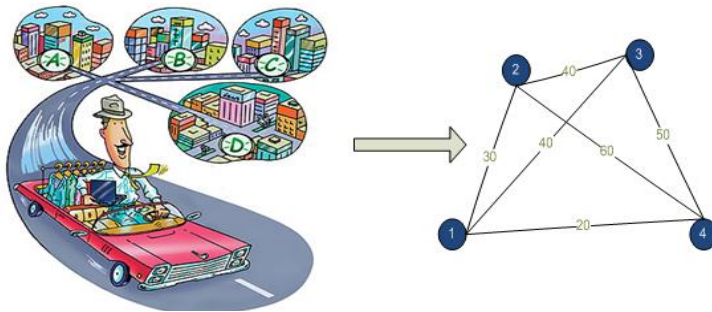
1.3.1. Giới thiệu bài toán

Bài toán người du lịch hay còn được gọi là bài toán TSP [1][2] là một bài toán khá nổi tiếng trong lĩnh vực tối ưu tổ hợp được nghiên cứu trong lý thuyết khoa học máy tính có nội dung như sau: Một người bán hàng xuất phát từ thành phố của anh ta, anh ta muốn tìm một đường đi ngắn nhất đi qua tất cả các thành phố của khách hàng mỗi thành phố đúng một lần và sau đó trở về thành phố ban đầu. Nó nhanh chóng trở thành bài toán khó thách thức toàn thế giới bởi độ phức tạp thuật toán tăng theo hàm số mũ (trong chuyên ngành thuật toán người ta còn gọi chúng là những bài toán NP-khó).

1.3.2. Lịch sử bài toán TSP

1.3.3. Mô tả bài toán TSP

TSP có thể được mô hình như một đồ thị (hình 1.5), các đỉnh của đồ thị tương ứng với các thành phố và các cạnh thì tương ứng với đường nối giữa các thành phố, chiều dài của một cạnh tương ứng với khoảng cách giữa 2 thành phố. Một đường đi trong bài toán TSP là một chu trình Hamilton trên đồ thị và một lời giải tối ưu của bài toán là chu trình Hamilton ngắn nhất.



Hình 1.5. Mô tả bài toán TSP

Đồ thị thường là đồ thị đầy đủ, vì vậy mọi cặp cạnh đều được nối bởi các cạnh. Đây là bước đơn giản hóa bài toán vì việc tìm chu trình Hamilton trong một đồ thị đầy đủ là dễ. Các bài toán mà không phải hai thành phố nào cũng được nối với nhau có thể được chuyển đổi thành đồ thị đầy đủ bằng cách thêm những cạnh có độ dài lớn giữa các thành phố này, những cạnh này sẽ không xuất hiện trong chu trình tối ưu.

1.3.4. Phân loại bài toán TSP

- Đối xứng và bất đối xứng*
- Khoảng cách là đơn vị độ dài hay không phải đơn vị độ dài*

1.4. CÁC THUẬT TOÁN GIẢI BÀI TOÁN TSP

1.4.1. Thuật giải chính xác

Trong các thuật giải chính xác cho bài toán người du lịch, đầu tiên phải kể đến thuật toán vét cạn. Thuật toán này tìm tất cả các chu trình hamilton trong đồ thị, sau đó chọn một chu trình nhỏ nhất làm đáp án.

1.4.2. Thuật giải gần đúng heuristic

Khi bài toán có kích thước n đỉnh nhỏ thì các thuật giải chính xác được áp dụng cho kết quả nhanh chóng và duy nhất. Nhưng khi số đỉnh của bài toán tăng lên đáng kể thì độ phức tạp của thuật toán do đó cũng tăng lên. Trong trường hợp này, chất lượng của giải pháp không phải là vấn đề quan tâm nhất mà hiệu suất tính toán và sự đơn giản về khái niệm được ưu tiên hơn, khi đó thuật toán heuristic được sử dụng để đưa ra một giải pháp không phải là tối ưu nhất nhưng chấp nhận được do sai số so với giải pháp tối ưu nhất không nhiều. Trong luận văn này giới thiệu bốn thuật toán nổi tiếng nhất là: thuật toán láng giềng gần nhất, thuật toán tìm kiếm cục bộ, thuật toán nhánh cận và thuật toán di truyền. Trong đó thuật toán cục bộ thường được sử dụng kết hợp với thuật toán đàn kiến ACO để tăng hiệu suất tìm kiếm giải pháp.

1.4.2.1. Thuật toán láng giềng gần nhất

Thuật giải vét cạn ở trên cho ta một đáp án tối ưu, tuy nhiên độ phức tạp của nó là quá cao ($O(n!)$). Do đó trong thực tế, người ta chấp nhận các thuật giải cho kết quả tốt (nhưng không phải lúc nào cũng tốt) bởi sự đơn giản, nhanh chóng và cài đặt dễ dàng. Một trong các

thuật giải đó là thuật toán láng giềng gần nhất hay còn được gọi là thuật toán tham lam [9][19][20].

1.4.2.2. Thuật toán tìm kiếm cục bộ

Thuật toán tìm kiếm cục bộ [13][22][26] là giải pháp metaheuristic cho việc giải các bài toán tính toán tối ưu khó trong máy tính. Thuật toán này có thể được áp dụng cho các bài toán tìm kiếm lời giải gần đúng tối ưu trong một loạt các lời giải ứng viên. Phương pháp tìm kiếm sẽ duyệt qua các lời giải trong không gian tìm kiếm cho đến khi tìm ra lời giải được cho là tối ưu hoặc vượt quá thời gian tìm kiếm cho phép. Thuật toán tìm kiếm cục bộ sẽ bắt đầu từ một ứng viên lời giải (chưa tối ưu), kiểm tra và cải thiện dần bằng cách chỉ quan tâm tới giải pháp hiện thời rồi xem xét chuyển sang ứng viên lời giải láng giềng của lời giải hiện thời đến khi dừng thuật toán. Tuy nhiên mỗi ứng viên lời giải đều có thể có hơn một lời giải láng giềng, nên mỗi cách lựa chọn lời giải láng giềng trong danh sách láng giềng để thành bước duyệt kế tiếp có thể trở thành một thuật toán khác.

1.4.2.3. Thuật toán nhánh cận

Thuật toán nhánh cận [19][20][21] là phương pháp chủ yếu để giải các bài toán tối ưu tổ hợp. Tư tưởng cơ bản của thuật toán là trong quá trình tìm kiếm lời giải, sẽ phân hoạch tập các phương án của bài toán thành hai hay nhiều tập con biểu diễn như một nút của một cây tìm kiếm và cố gắng bằng cách đánh giá cận các nút, tìm cách loại bỏ những nhánh cây (những tập con

các phương án của bài toán) mà biết chắc chắn không phải phương án tối ưu. Mặc dù trong trường hợp tồi nhất, thuật toán sẽ trở thành duyệt toàn bộ, nhưng trong những trường hợp cụ thể nó có thể rút ngắn đáng kể thời gian tìm kiếm.

1.4.2.4. Thuật toán di truyền

Thuật toán di truyền [23] là thuật toán metaheuristic, metaheuristic là một cách gọi chung cho các thuật toán heuristic trong việc giải quyết các bài toán tổ hợp khó. Hầu hết các thuật toán metaheuristic đều lấy cảm hứng từ tự nhiên như: thuật toán luyện thép (SA), thuật toán di truyền (GA), thuật toán đàn kiến (ACO), ... Thuật toán đàn kiến là metaheuristic dùng chiến lược của kiến trong thế giới thực để giải bài toán tối ưu. SA xuất phát từ phương thức xác suất và kỹ thuật luyện bao gồm việc nung và điều khiển âm nguội các kim loại để đạt được trạng thái năng lượng nhỏ nhất. Trong khi đó thuật toán di truyền dựa trên ý tưởng từ cơ chế di truyền trong sinh học và tiến trình tiến hóa trong cộng đồng các cá thể của một loài.

1.5. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

1.6. KẾT CHƯƠng

Chương này trình bày tổng quan các lý thuyết liên quan về đồ thị, bài toán người du lịch, các phương pháp giải bài toán người du lịch. Trong số các phương pháp đã giới thiệu ở chương 1, cách giải bài toán người du lịch bằng thuật toán đàn kiến được lựa chọn làm thuật toán chính để trình bày trong luận văn này. Nội dung chi tiết thuật toán đàn kiến sẽ được trình bày ở chương 2.

CHƯƠNG 2. THUẬT TOÁN TỐI ƯU ĐÀN KIẾN ACO

Chương này tìm hiểu về nội dung thuật toán đàn kiến; thuật toán đàn kiến giải bài toán người du lịch; thuật toán tối ưu đàn kiến ACO bao gồm các thuật toán Ant System, Max-Min Ant System và Ant Colony System; cách thức nâng cao hiệu quả của thuật toán ACO và các ứng dụng của thuật toán ACO.

2.1. GIỚI THIỆU

Thuật toán kiến được phát triển trong những năm gần đây là một cách tiếp cận phổ biến đã giải quyết được những bài toán tối ưu tổ hợp NP khó [6][12]. Như tên gọi của nó, thuật toán kiến bắt nguồn từ hành vi của đàn kiến thực trong tự nhiên, cụ thể hơn là hành vi tìm kiếm thức ăn của chúng. Một trong những ý tưởng chính của thuật toán kiến là việc giao tiếp gián tiếp của đàn kiến nhân tạo dựa trên tuyến đường vết mùi (vết mùi cũng là chất sử dụng trong giao tiếp của kiến thực). Các tuyến đường vết mùi nhân tạo là một dạng phân phối thông tin số được sửa đổi bởi kiến phản ánh kinh nghiệm của chúng khi giải quyết một vấn đề cụ thể. Gần đây, thuật toán meta heuristics Tối ưu hóa kiến (ACO) được đề xuất đã cung cấp một khuôn khổ thống nhất cho hầu hết các ứng dụng của các thuật toán kiến để tối ưu hóa các vấn đề tổ hợp. Tất cả các thuật toán kiến áp dụng cho bài toán TSP được gọi chung là thuật toán ACO. [3][4][5]

2.1.1. Hoàn cảnh ra đời và lịch sử phát triển của thuật toán ACO

2.1.2. Tư tưởng thuật toán

Thuật toán đàn kiến được ra đời và phát triển xuất phát từ quan sát thực tế hành vi kiến trong tự nhiên và đó là một nguồn cảm hứng cho việc thiết kế các thuật toán mới cho các giải pháp tối ưu hóa và các vấn đề điều khiển phân tán. [7][8]

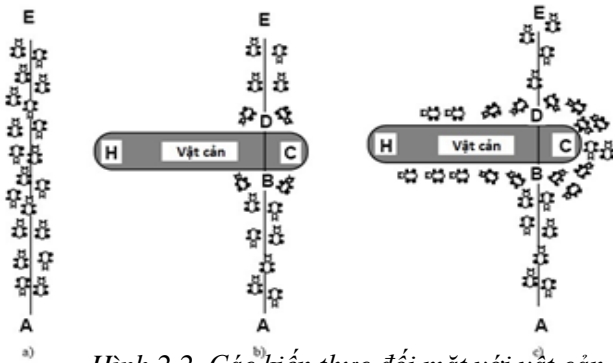
Đàn kiến tự nhiên (hình 2.1): Là một loài có tổ chức cao, mỗi con kiến khi di chuyển sẽ để lại một lượng thông tin pheromone trên mặt đất. Đây là phương tiện để đánh dấu và để đàn kiến trao đổi thông tin khi tìm kiếm thức ăn. Khi đi tìm kiếm thức ăn, sau khi tìm thấy nguồn thức ăn, thì mỗi con kiến sẽ tìm ra đường đi của nó để đi từ tổ tới nguồn thức ăn. Chúng sẽ giao tiếp trao đổi thông tin với nhau, sau một thời gian cả đàn kiến gần như tìm ra và đi theo con đường ngắn nhất từ tổ kiến tới nguồn thức ăn.

Cách tìm đường đi của kiến thực được minh họa ở hình 2.2 với ví dụ sau: Các con kiến đang đi trên một con đường thẳng từ tổ kiến A đến nguồn thức ăn E. Một lúc sau, đột nhiên có một chướng ngại vật cản trên đường đi, lúc này đường đi từ A đến E bị chia làm 2 hướng. Vậy các con kiến sẽ đi theo hướng nào?

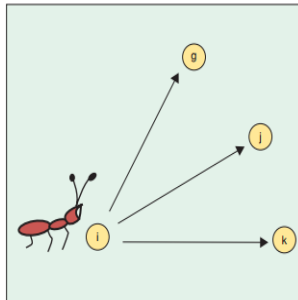
Đàn kiến nhân tạo (hình 2.3): Để bắt chước hành vi của các con kiến thực, Dorigo xây dựng các con kiến nhân tạo cũng có đặc trưng sản sinh ra vết mùi để lại trên đường đi và khả năng lần vết theo nồng độ mùi để lựa chọn con đường có nồng độ mùi cao hơn để đi. Với bài toán người du lịch trên đồ thị trong không gian hai chiều với trọng số là khoảng cách giữa hai đỉnh bất kỳ, Dorigo gán với mỗi cạnh (i, j) ngoài trọng số d_{ij} trên là nồng độ vết mùi trên cạnh đó, đặt là $\tau_{\max} = 1/pC^{bs} \tau_{ij}$.



Hình 2.1. Kiến tự nhiên



Hình 2.2. Các kiến thực đối mặt với vật cản



Hình 2.3. Một con kiến ở thành phố *i* chọn lựa thành phố *j* kế tiếp để đi dựa vào xác suất tỉ lệ với vệt mùi để lại trên cạnh

Phương pháp tìm đường đi mô phỏng hành vi con kiến

Các con kiến sẽ tiến hành tìm đường đi từ đỉnh xuất phát qua một loạt các đỉnh và quay trở về đỉnh ban đầu, tại đỉnh i một con kiến sẽ chọn đỉnh j chưa được đi qua trong tập láng giềng của i theo xác suất như ở công thức (2.1):

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l \in N_i^k} \tau_{il}^\alpha \eta_{il}^\beta}, \text{ với } j \in N_i^k \quad (2.1)$$

Công thức (2.1) có ý nghĩa như sau: quyết định lựa chọn đỉnh tiếp theo để đi của con kiến được lựa chọn ngẫu nhiên theo xác suất (tức là đỉnh nào có xác suất cao hơn sẽ có khả năng được chọn cao hơn, nhưng không có nghĩa là các đỉnh có xác suất thấp hơn không được chọn mà nó được chọn với cơ hội thấp hơn mà thôi). Và xác suất này (hay khả năng chọn đỉnh tiếp theo của con kiến) tỷ lệ thuận với nồng độ vết mùi trên cạnh được chọn (theo đặc tính của con kiến tự nhiên) và tỷ lệ nghịch với độ dài cạnh, là những hệ số điều khiển việc lựa chọn của con kiến nghiêng về phía nào.

2.2. THUẬT TOÁN TỐI ƯU ĐÀN KIẾN ACO

2.2.1. Thuật toán Ant System (AS)

- a. Quy tắc di chuyển của kiến
- b. Quy tắc cập nhật thông tin mùi

2.2.2. Thuật toán Max-Min Ant System (MMAS)

MMAS và một số thuật toán khác như Elitist AS, Rank-Based AS là các thuật toán có được hiệu suất cao hơn nhiều so với thuật toán AS nhờ vào những thay đổi nhỏ trong thuật toán AS, đây được coi là các thuật toán kế thừa trực tiếp từ thuật toán AS vì chúng về cơ bản là không khác gì nhiều so với AS.

- a. Quy tắc cập nhật mùi
- b. Giới hạn thông tin mùi

2.2.3. Thuật toán Ant Colony System (ACS)

Trong khi MMAS là thuật toán chỉ thay đổi phần nhỏ từ thuật toán AS, thì các thuật toán khác như ACS, Ant-Q,.. đạt được hiệu suất cao bằng cách đưa hẳn các kỹ thuật hoàn toàn mới mà ý tưởng của nó không có trong thuật toán AS cơ bản. Đây là những thuật toán mở rộng của AS.

a. Quy tắc di chuyển của kiến

b. Quy tắc cập nhật thông tin mùi

b.1. Cập nhật thông tin mùi toàn cục

b.2. Cập nhật mùi cục bộ

2.3. THUẬT TOÁN ĐÀN KIẾN GIẢI BÀI TOÁN TSP

Thuật toán đàn kiến về sau được gọi là thuật toán ACO, ACO có thể áp dụng cho bài toán TSP dưới dạng biểu diễn bằng đồ thị $G=(C,L)$, trong đó L là tập các cạnh kết nối đầy đủ tất cả các đỉnh của tập C . Tập giải pháp của vấn đề chính là tập các hành trình khả dụng bắt đầu từ thành phố xuất phát đến thành phố đích.[1]

Trong tất cả các thuật toán ACO áp dụng cho bài toán TSP, các tuyến đường vết mùi đều có liên quan đến các cung, vì vậy τ_{ij} biểu thị cho khả năng (mong muốn) thăm thành phố j ngay sau khi vừa thăm thành phố i . Thông tin heuristic được chọn ở đây là: $\eta_{ij} = 1/d_{ij}$. η_{ij} là mong muốn phỏng đoán (kinh nghiệm) của việc đi từ thành phố i trực tiếp đến thành phố j tỉ lệ nghịch với d_{ij} - khoảng cách giữa 2 thành phố i, j . Trong trường hợp có vài cung có $d_{ij} = 0$ thì η_{ij} được đặt cho một giá trị rất nhỏ. Mục đích thực hiện các đường đi vết mùi là thu thập thành một ma trận vết mùi mà các phần tử của nó là các τ_{ij} . Việc này cũng giống như thông tin heuristic.

Sau khi khởi tạo các thông số và những con đường mòn vết mùi, các thuật toán ACO lặp thông qua một vòng lặp chính: đầu tiên là xây dựng tất cả các hành trình có thể của kiến, sau đó là cải thiện kết quả bằng cách kết hợp thuật toán tìm kiếm cục bộ, và cuối cùng là cập nhật lại vết mùi cho các con đường. Bước cuối cùng này bao gồm sự bay hơi vết mùi và các kiến sẽ cập nhật lại vết mùi trên các con đường mà kiến đã đi qua (gọi là con đường vết mùi) để phản ánh kinh nghiệm tìm kiếm của chúng.

2.4. NÂNG CAO HIỆU QUẢ CỦA THUẬT TOÁN ACO

2.5. CÁC ỨNG DỤNG CỦA ACO

2.6. KẾT CHƯƠng

Thuật toán ACO hiện nay ngày càng phong phú, không chỉ có 3 thuật toán ACO như đã trình bày trong luận văn này mà còn được phát triển thành nhiều phiên bản với nhiều tên gọi khác nhau, tuy nhiên vẫn mang những tư tưởng cốt lõi nhất của thuật toán AS. Chương 3 sẽ trình bày một thực thi thuật toán ACO cho bài toán người du lịch

CHƯƠNG 3. ỨNG DỤNG THUẬT TOÁN ACO VÀO BÀI TOÁN NGƯỜI DU LỊCH

Chương này phân tích yêu cầu của bài toán, từ đó phân tích các chức năng, xây dựng chương trình ứng dụng vào bài toán người du lịch đồng thời tiến hành chạy thử, đánh giá kết quả; và so sánh tính hiệu quả của thuật toán tối ưu đàn kiến ACO với thuật toán di truyền.

3.1. PHÂN TÍCH YÊU CẦU

Bài toán đặt ra là xây dựng một chương trình minh họa thuật toán tối ưu hóa đàn kiến ACO cho bài toán người du lịch đối xứng trên một giao diện đồ họa với dữ liệu thử nghiệm được lấy từ các nguồn dữ liệu sau:

- Dữ liệu tọa độ các điểm trong thư viện TSPLib
- Từ tập tin ma trận khoảng cách giữa các thành phố
- Dữ liệu được phát ra ngẫu nhiên

Dự kiến kết quả của chương trình sẽ là:

- Xuất ra đường đi ngắn nhất xuất phát từ một đỉnh bất kỳ đi qua tất cả các thành phố mỗi thành phố một lần
- Lưu kết quả chạy chương trình vào một tập tin văn bản
- So sánh kết quả của chương trình chạy bằng thuật toán tối ưu đàn kiến ACO với kết quả của chương trình chạy bằng thuật toán di truyền

3.2. ĐẶC TẢ CẤU TRÚC DỮ LIỆU

3.2.1. Biểu diễn thông tin các thành phố

3.2.2. Biểu diễn thông tin vết mùi trên các con đường

3.2.3. Biểu diễn sự kết hợp thông tin vết mùi và thông tin heuristic.

3.2.4. Biểu diễn thông tin kiến

3.3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.3.1. Phân tích chức năng

3.3.2. Xây dựng thuật toán

3.4. KẾT QUẢ THỰC NGHIỆM CHƯƠNG TRÌNH

3.4.1. Chức năng 1: Thực hiện thuật toán ACO từ tập dữ liệu

3.4.2. Chức năng 2: Thực hiện thuật toán ACO từ các dữ liệu được phát sinh ngẫu nhiên

3.4.3. Chức năng 3: Thực hiện thuật toán ACO từ tập tin thư viện TSPLIB.

3.5. ĐÁNH GIÁ KẾT QUẢ

3.5.1. Kết quả thực hiện của thuật toán ACO

Kết quả thực nghiệm thuật toán ACO trên 3 bộ dữ liệu thử nghiệm Eil51, Eil76, Eil101 về số vòng lặp khi thuật toán ACO cho kết quả tối ưu nhất được chỉ ra trong bảng 3.1.

Kết quả thực nghiệm thuật toán ACO trên 3 bộ dữ liệu thử nghiệm Eil51, Eil76, Eil101 cho 10 lần thực hiện và giá trị trung bình của 10 lần thực hiện được chỉ ra trong bảng 3.2.

Kết quả thực nghiệm thuật toán ACO trên 3 bộ dữ liệu thử nghiệm Eil51, Eil76, Eil101 về kết quả tối ưu nhất của thuật toán ACO được chỉ ra trong bảng 3.3.

Bảng 3.1. Bảng so sánh số vòng lặp mà thuật toán ACO thực hiện khi tìm thấy giá trị tối ưu nhất.

$\alpha=1, \beta=2, Q=2$									
Rho	0.5	0.2	0.1	0.5	0.2	0.1	0.5	0.2	0.1
Số kiến	M=10			M=20			M=30		
Eil51	570	252	783	556	686	224	264	798	567
Eil76	966	857	900	780	625	705	953	977	717
Eil101	835	795	917	884	990	902	732	920	976

Bảng 3.2. Bảng kết quả thực hiện thuật toán với các tham số đầu vào: $\alpha=1, \beta=2, \rho=0.2, \text{số kiến}=30, Q=2$.

Dữ liệu	Lần thực hiện										Trung bình
	1	2	3	4	5	6	7	8	9	10	
Eil51	468.58	466.53	467.22	466.05	465.06	463.39	463.89	461.31	459.70	463.15	464.49
Eil76	609.00	603.74	640.85	628.19	611.58	620.97	591.59	612.70	601.68	581.74	610.20
Eil101	714.12	718.77	715.47	723.72	734.51	709.66	722.29	718.83	734.35	712.73	720.45

Bảng 3. 3. Bảng so sánh giá trị tối ưu nhất mà thuật toán ACO đã thực hiện

$\alpha=1, \beta=2, Q=2$									
Rho	0.5	0.2	0.1	0.5	0.2	0.1	0.5	0.2	0.1
Số kiến	M=10			M=25			M=30		
Eil51	478.31	460.05	455.72	464.89	455.68	459.68	443.77	465.10	445.59
Eil76	607.20	627.07	611.19	613.71	581.74	601.49	643.64	644.15	608.99
Eil101	784.11	779.47	746.95	719.53	727.63	747.59	730.70	706.09	739.98

Bảng 3.4. Bảng kết quả thực hiện thuật toán với các tham số đầu vào: $\alpha=1, \beta=2, \rho=0.2$, số kiến=30, $Q=2$.

Dữ liệu	Lần thực hiện										Trung bình
	1	2	3	4	5	6	7	8	9	10	
Eil51	472.76	465.97	471.70	478.68	474.24	469.50	462.88	462.05	472.76	476.11	470.67
Eil76	609.56	611.61	630.93	638.89	614.18	601.95	598.44	614.45	606.13	626.51	615.27
Eil101	731.09	708.60	715.47	729.72	712.53	734.33	704.60	746.27	724.50	717.66	722.48

3.5.2. Đánh giá hiệu quả thuật toán ACO

3.5.3. So sánh thuật toán ACO với thuật toán di truyền GA

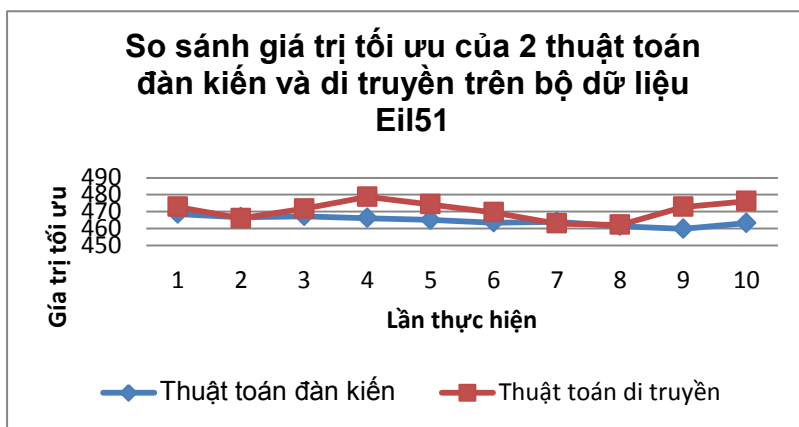
Thực nghiệm 10 lần cho mỗi thuật toán ACO và thuật toán di truyền trên 3 bộ dữ liệu thử nghiệm Eil51, Eil76 và Eil101 (kết quả thử nghiệm chỉ ra trong bảng 3.4), sau đó so sánh các giá trị tối ưu nhất mà mỗi thuật toán tìm được.

Kết quả so sánh áp dụng cho bộ dữ liệu thử nghiệm Eil51, Eil76, Eil101 lần lượt được chỉ ra trong biểu đồ 3.1, biểu đồ 3.2 và biểu đồ 3.3.

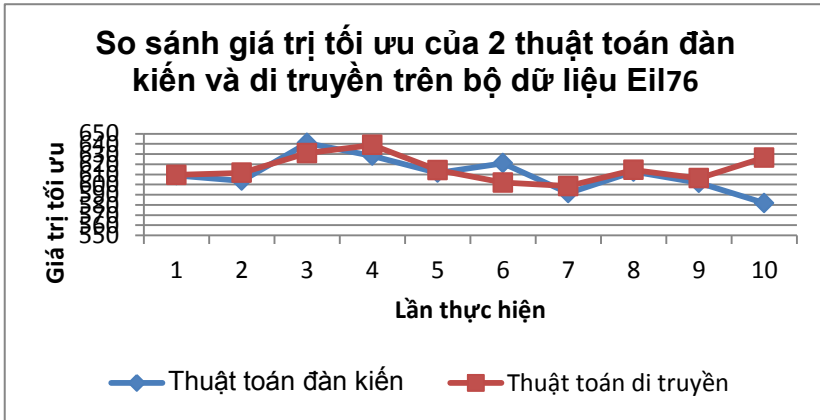
3.6. KẾT CHƯƠng

Chương 3 đã hoàn tất một thực thi của thuật toán đàn kiến cho bài toán người du lịch, từ đặc tả cấu trúc dữ liệu, xây dựng các chức năng, các kịch bản thực nghiệm chương trình rồi từ đó đánh giá hiệu quả thuật toán đàn kiến, thực hiện so sánh thuật toán ACO với thuật toán di truyền GA trên tiêu chí kết quả của giải pháp và thời gian thực hiện thuật toán. Cuối cùng là kết luận và hướng phát triển sau này của đề tài.

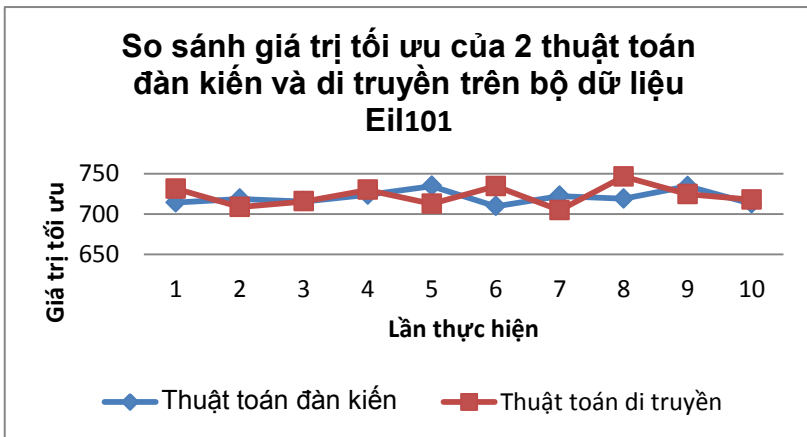
Biểu đồ 3.1. So sánh giá trị tối ưu của 2 thuật toán đàn kiến và di truyền trên bộ dữ liệu Eil51



Biểu đồ 3.2. So sánh giá trị tối ưu của 2 thuật toán đàn kiến và di truyền trên bộ dữ liệu Eil76



Biểu đồ 3.3. So sánh giá trị tối ưu của 2 thuật toán đàn kiến và di truyền trên bộ dữ liệu Eil101



KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. KẾT QUẢ ĐẠT ĐƯỢC

Trong thời gian nghiên cứu thuật toán đàn kiến giải bài toán người du lịch, luận văn đã đạt được các kết quả sau:

Kết quả về mặt lý thuyết:

- Lý thuyết đồ thị, nghiên cứu tìm hiểu nội dung, lịch sử, mô tả về bài toán người du lịch.
- Nghiên cứu tìm hiểu các phương pháp giải bài toán người du lịch: thuật toán vét cạn, thuật toán người láng giềng gần nhất, thuật toán tìm kiếm cục bộ, thuật toán nhánh cận, thuật toán di truyền.
- Nghiên cứu tìm hiểu thuật toán đàn kiến, các phiên bản thuật toán đàn kiến trong tập thuật toán tối ưu đàn kiến ACO, cách nâng cao hiệu quả thuật toán đàn kiến, các ứng dụng của ACO.

Kết quả thực nghiệm:

- Luận văn đã áp dụng thuật toán tối ưu đàn kiến ACO để giải quyết bài toán người du lịch và so sánh kết quả thực hiện so với thuật toán di truyền.
- Mô hình giải quyết bài toán đơn giản, dễ cài đặt và thích hợp, không cần đòi hỏi quá nhiều về phần cứng. Lập trình đơn giản, ngắn gọn, kết quả chính xác, áp dụng được cho nhiều bộ dữ liệu lớn.
- Thực nghiệm tìm đường đi tối ưu nhất của bài toán người du lịch có thể áp dụng cho nhiều nguồn dữ liệu khác nhau: dữ liệu ngẫu nhiên, dữ liệu từ tập tin khoảng cách giữa các điểm, dữ liệu thử nghiệm chuẩn TSPLIB. Kết quả thử

nghiệm giải bài toán người du lịch bằng thuật toán đàn kiến cho thấy kết quả khá gần so với kết quả tối ưu nhất được tìm thấy cho đến thời điểm hiện tại.

Tuy nhiên chương trình vẫn còn tồn tại một ít hạn chế như thời gian thực hiện thuật toán đàn kiến còn chậm, trong thuật toán chưa áp dụng một số kỹ thuật mới được giới thiệu gần đây để cải tiến thuật toán kiến: kết hợp thuật toán di truyền vào thuật toán đàn kiến, áp dụng các kỹ thuật tìm kiếm nâng cao vào thuật toán chương trình để cải thiện thời gian tìm kiếm...

2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Từ kết quả nghiên cứu lý thuyết và thực tiễn, luận văn đề ra các vấn đề cần tiếp tục hoàn thiện, phát triển và nghiên cứu như sau:

- Tiếp tục nghiên cứu thêm các thuật toán đàn kiến khác trong họ thuật toán ACO và các thuật toán đàn kiến cải tiến mới khác như kết hợp thuật toán ACO và di truyền áp dụng cho các bài toán tối ưu tổ hợp phức tạp.
- Nghiên cứu và cài đặt các thuật toán tìm kiếm cục bộ khác áp dụng vào thuật toán đàn kiến giải bài toán người du lịch. Từ đó rút ra nhận xét và đánh giá về tính hiệu quả của việc kết hợp các thuật toán tìm kiếm cục bộ vào thuật toán đàn kiến.
- Nghiên cứu và cài đặt các phương pháp song song hóa thuật toán và phương pháp thực hiện song song hóa thuật toán đàn kiến trên kiến trúc bộ nhớ phân tán và mô hình song song Master/Slave. Từ đó đánh giá tính hiệu quả của thuật toán đàn kiến trên các mô hình song song và so sánh hiệu quả của thuật toán đàn kiến trên hệ phân tán.