

**ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

NGHIÊM QUANG KHẢI

**THUẬT TOÁN DIJKSTRA FIBONACCI HEAP, THUẬT
TOÁN ACO TÌM ĐƯỜNG ĐI TỐI ƯU VÀ ỨNG DỤNG**

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

THÁI NGUYÊN - 2015

LỜI CAM ĐOAN

Tôi xin cam đoan luận văn này là kết quả nghiên cứu của riêng tôi. Các thông tin trích dẫn trong luận văn lấy từ các nguồn đã được công khai hoặc đã được sự đồng ý của tác giả. Các kết quả nêu trong luận văn là kết quả nghiên cứu riêng của tác giả luận văn, chưa có ai công bố trong các công trình khác.

Thái Nguyên, ngày 10 tháng 4 năm 2015

Học viên

Nghiêm Quang Khải

LỜI CẢM ƠN

Được sự phân công của trường Đại Học Công Nghệ Thông Tin Và Truyền Thông - Đại Học Thái Nguyên và sự đồng ý của thầy giáo hướng dẫn PGS - TS Đoàn Văn Ban, tôi đã thực hiện đề tài “Thuật toán Dijkstra Fibonacci heap, thuật toán ACO tìm đường đi tối ưu và ứng dụng”.

Để hoàn thành được đề tài này, tôi đã nhận được sự hướng dẫn tận tình chu đáo của thầy hướng dẫn PGS – TS Đoàn Văn Ban, qua đây cho phép tôi được bày tỏ lòng biết ơn chân thành tới Thầy và gia đình Thầy.

Tôi cũng xin được tỏ lòng cảm ơn đối với các thầy các cô đã tận tình hướng dẫn, giảng dạy lớp cao học 12G trong suốt hai năm qua, cảm ơn những tri thức các thầy cô đã truyền thụ, cảm ơn những tình cảm chân thành các thầy cô đã dành cho lớp.

Xin chân thành cảm ơn những ý kiến đóng góp quý báu của các thầy cô giáo và các bạn đồng nghiệp đối với đề tài này.

Chắc chắn đề tài này sẽ không tránh khỏi những thiếu sót, rất mong nhận được các ý kiến đóng góp của các thầy cô, các bạn đồng nghiệp và các bạn độc giả, tôi xin chân thành cảm ơn.

Thái Nguyên, ngày 10 tháng 4 năm 2015

Học viên

Nghiêm Quang Khải

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN	iii
MỞ ĐẦU.....	1
CHƯƠNG 1	4
CÁC THUẬT TOÁN TÌM ĐƯỜNG ĐI TỐI ƯU TRÊN ĐỒ THỊ.....	4
1.1. Các khái niệm cơ bản của lý thuyết đồ thị	4
1.1.1. Định nghĩa đồ thị	4
1.1.2. Các thuật ngữ cơ bản	5
1.1.3. Đường đi, chu trình, đồ thị liên thông	6
1.1.4. Đồ thị có trọng số	7
1.2. Cây.....	8
1.3. Bài toán đường đi tối ưu trên đồ thị.....	8
1.4. Thuật toán Dijkstra.....	10
1.4.1. Phát biểu bài toán.....	10
1.4.2. Mô tả thuật toán	10
1.5. Thuật toán Dijkstra kết hợp với Fibonacci heap.....	11
1.5.1. Hàng đợi ưu tiên	11
1.5.2. Fibonacci heap	14
1.5.3. Sơ đồ thuật toán Dijkstra kết hợp với Fibonacci Heap.....	30
1.6. Kết luận chương	32
CHƯƠNG 2	33
THUẬT TOÁN ĐÀN KIẾN GIẢI BÀI TOÁN TÌM ĐƯỜNG ĐI TỐI ƯU	33
2.1. Từ kiến tự nhiên đến kiến nhân tạo	33
2.1.1. Kiến tự nhiên	33
2.1.2. Kiến nhân tạo	36
2.2. Thuật toán ACO tổng quát giải bài toán người chào hàng.....	37
2.2.2. Thuật toán ACO tổng quát giải bài toán TSP	38
2.3. Các thuật toán ACO giải bài toán TSP	39
2.3.1. Thuật toán AS	40
2.3.2. Thuật toán ACS	42
2.3.3. Thuật toán Max-Min (MMAS).....	44
2.4. Một số vấn đề trong việc áp dụng ACO tìm đường đi tối ưu	46

2.4.1. ACO kết hợp với tìm kiếm cục bộ.....	46
2.4.2. Thông tin heuristic	47
2.4.3. Số lượng kiến.....	47
2.4.4. Tham số bay hơi	48
2.4.5. Một số đề xuất cải tiến.....	48
2.5. <i>Kết luận chương</i>	49
CHƯƠNG 3	50
ỨNG DỤNG THUẬT TOÁN DIJKSTRA FIBONACCI HEAP, THUẬT TOÁN ACO GIẢI CÁC BÀI TOÁN TÌM ĐƯỜNG ĐI TRÊN MẠNG GIAO THÔNG....	50
3.1. <i>Ứng dụng Dijkstra Fibonacci heap</i>	50
13.1.1. <i>Phát biểu bài toán 1</i>	50
3.1.2. Mô hình hoá bài toán	50
3.1.3. Mô tả input, output	50
3.1.4. Một số kiểu dữ liệu và các biến trong chương trình.....	51
3.1.5. Một số hàm và thủ tục trong chương trình	52
3.1.6. Sơ đồ thuật toán	55
3.1.7. Các kết quả thực nghiệm giải bài toán 1.....	56
3.2. <i>Ứng dụng Dijkstra Fibonacci heap, ACO giải bài toán TSP mở rộng</i>	58
3.2.1. Phát biểu bài toán 2.....	58
3.2.2. Mô hình hoá bài toán	58
3.2.3. Mô tả input, output	58
3.2.4. Thuật toán tổng quát giải bài toán 2	59
3.2.5. Một số hàm và thủ tục trong chương trình	59
3.2.6. Sơ đồ tổng quát của thuật toán giải bài toán 2.....	62
3.2.7. Các kết quả thực nghiệm giải bài toán 2.....	63
3.3. <i>Kết luận chương</i>	65
KẾT LUẬN.....	67
TÀI LIỆU THAM KHẢO.....	69

MỞ ĐẦU

Thuật toán tìm đường đi tối ưu có nhiều ứng dụng trong thực tế, nếu xây dựng được các thuật toán tốt sẽ giúp tiết kiệm được rất nhiều tiền bạc, thời gian, công sức của con người. Một số bài toán thực tế điển hình cần phải sử dụng thuật toán tìm đường đi tối ưu như:

- Tìm đường đi từ địa điểm A đến địa điểm B sao cho độ dài đường đi là tối ưu hoặc nhanh nhất hoặc giá cước là nhỏ nhất.

- Tìm đường đi ngắn nhất xuất từ một điểm cho trước, đi qua một số địa điểm cố định cho trước rồi quay trở về điểm xuất phát.

- Tương tự ta cũng có bài toán tìm đường đi cho gói tin được gửi từ nút A đến nút B trên mạng máy tính sao cho giá cước là nhỏ nhất hoặc nhanh nhất...

- Tìm đường đi tối ưu cho robot, cho tên lửa hành trình, máy bay, phi thuyền v.v cũng là những bài toán đang được quan tâm.

Đã có nhiều công trình nghiên cứu về lĩnh vực này và có nhiều thuật toán nổi tiếng đã được phát minh như: Thuật toán Bellman – Ford, thuật toán Dijkstra, thuật toán Floyd, thuật toán Johnson...

Tuy nhiên việc nghiên cứu cải tiến nâng cao hiệu quả của các thuật toán tìm đường đi tối ưu luôn nhận được sự quan tâm của nhiều người, nhiều tổ chức, cơ quan. Vì lý do nói trên và được sự gợi ý của PGS – TS Đoàn Văn Ban, tác giả đã chọn đề tài này để nghiên cứu trong luận văn tốt nghiệp thạc sĩ của mình.

- **Phạm vi nghiên cứu của đề tài**

Các khái niệm cơ bản về đồ thị, các thuật toán tìm đường đi tối ưu trên đồ thị, cấu trúc dữ liệu Fibonacci heap, ứng dụng cấu trúc dữ liệu này vào việc cải tiến nâng cao hiệu quả của thuật toán tìm đường đi tối ưu trên đồ thị.

Ứng dụng các thuật toán tìm đường đi tối ưu trên đồ thị đã nghiên cứu để giải quyết một số bài toán tìm đường đi tối ưu trong mạng giao thông.

- **Hướng nghiên cứu của đề tài**

- Nghiên cứu thuật toán Dijkstra tìm đường đi tối ưu trên đồ thị, nghiên cứu về Fibonacci heap và ứng dụng cấu trúc dữ liệu này để cải tiến thuật toán Dijkstra.

- Nghiên cứu về thuật toán tối ưu đàn kiến, ứng dụng thuật toán này để giải quyết bài toán tìm đường đi tối ưu trên đồ thị.

- Ứng dụng hai thuật toán trên giải quyết một số bài toán tìm đường đi tối ưu trên mạng giao thông.

- **Đề tài gồm có 3 chương:**

Chương 1: Trình bày một số khái niệm cơ bản về đồ thị, một số dạng bài toán tìm đường đi tối ưu trên đồ thị, phần chủ yếu của chương này là trình bày về Fibonacci heap và dùng cấu trúc dữ liệu này để cải tiến nâng cao hiệu quả thuật toán Dijkstra.

Chương 2: Trình bày về thuật toán tối ưu đàn kiến và thuật toán ACO giải bài toán tìm đường đi tối ưu. Thuật toán đàn kiến là một thuật toán tương đối mới và khả năng ứng dụng thực tế cao.

Chương 3: Ứng dụng thuật toán Dijkstra đã cải tiến và thuật toán đàn kiến vào việc giải một số bài toán tìm đường đi tối ưu trên mạng giao thông.

- **Ý nghĩa khoa học của đề tài:**

- Thuật toán Dijkstra Fibonacci heap là thuật toán mạnh, nó có thể được ứng dụng để giải quyết các bài toán cả trong nghiên cứu lý thuyết và trong thực tiễn. Hiện tại thuật toán này chưa phổ biến ở Việt Nam, vì thế đề tài này có thể sẽ có ích cho những người quan tâm đến lĩnh vực này. Đề tài cũng có

thể giúp cho các em học sinh Chuyên Tin có thêm một công cụ mạnh để giải quyết một số bài toán có liên quan trong lập trình.

- Thuật toán ACO là thuật toán gần đúng, tuy nhiên nó rất hiệu quả trong việc giải quyết các bài toán thực tiễn . Đề tài đã ứng dụng thành công hai thuật toán nói trên vào việc giải quyết một số bài toán mà thực tiễn đang đặt ra.

CHƯƠNG 1

CÁC THUẬT TOÁN TÌM ĐƯỜNG ĐI TỐI ƯU TRÊN ĐỒ THỊ

1.1. Các khái niệm cơ bản của lý thuyết đồ thị

1.1.1. Định nghĩa đồ thị

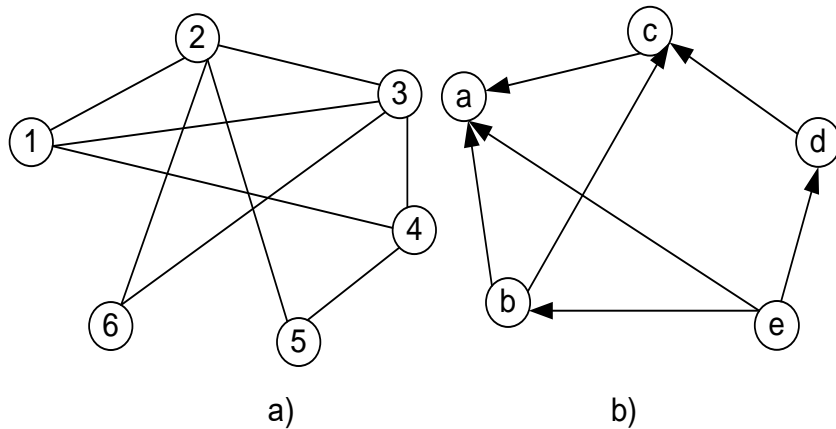
Đồ thị là một cấu trúc rời rạc bao gồm các đỉnh và các cạnh nối các đỉnh này. Chúng ta phân biệt các loại đồ thị khác nhau bởi kiểu và số lượng cạnh nối hai đỉnh nào đó của đồ thị.

Định nghĩa 1.1. Đơn đồ thị vô hướng $G = (V, E)$ bao gồm V là tập các đỉnh, và E là tập các cặp không có thứ tự gồm hai phần tử khác nhau của V gọi là các cạnh [3].

Định nghĩa 1.2. Đa đồ thị vô hướng $G = (V, E)$ bao gồm V là tập các đỉnh, và E là họ các cặp không có thứ tự gồm hai phần tử khác nhau của V gọi là các cạnh. Hai cạnh e_1 và e_2 được gọi là cạnh lặp nếu chúng cùng tương ứng với một cặp đỉnh [3].

Định nghĩa 1.3. Đơn đồ thị có hướng $G = (V, E)$ bao gồm V là tập các đỉnh, và E là tập các cặp có thứ tự gồm hai phần tử khác nhau của V gọi là các cung [3].

Định nghĩa 1.4. Đa đồ thị có hướng $G = (V, E)$ bao gồm V là tập các đỉnh, và E là họ các cặp có thứ tự gồm hai phần tử khác nhau của V gọi là các cung. Hai cung e_1 và e_2 được gọi là cung lặp nếu chúng cùng tương ứng với một cặp đỉnh [3].



Hình 1.1. Hai loại đồ thị cơ bản:

a) Đồ thị vô hướng (6 đỉnh, 9 cạnh). b) Đồ thị có hướng (5 đỉnh, 7 cung).

1.1.2. Các thuật ngữ cơ bản

Định nghĩa 1.5. Hai đỉnh u và v của đồ thị vô hướng G được gọi là kề nhau nếu (u,v) là cạnh của đồ thị G . Nếu $e = (u,v)$ là cạnh của đồ thị thì chúng ta nói cạnh này là liên thuộc với hai đỉnh u và v , hoặc cũng nói là cạnh e là nối đỉnh u và đỉnh v , đồng thời các đỉnh u và v sẽ được gọi là các đỉnh đầu của cạnh (u,v) [3].

Để có thể biết được bao nhiêu cạnh liên thuộc với một đỉnh, chúng ta đưa vào định nghĩa sau:

Định nghĩa 1.6. Gọi bậc của đỉnh v trong đồ thị vô hướng là số cạnh liên thuộc với nó và sẽ kí hiệu là $\deg(v)$.

Định lý 1.1. Giả sử $G = (V, E)$ là đồ thị vô hướng với m cạnh. Khi đó $2m = \sum_{v \in V} \deg(v)$ [3].

Hệ quả 1.1. Trong đồ thị vô hướng, số đỉnh bậc lẻ (nghĩa là có bậc là số lẻ) là một số chẵn [3].

Định nghĩa 1.7. Nếu $e = (u,v)$ là cung của đồ thị có hướng G thì chúng ta nói hai đỉnh u và v là kề nhau, và nói cung (u,v) nối đỉnh u và đỉnh v hoặc