

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

LÊ NGỌC QUANG

**GIẢI BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT
BẰNG THUẬT TOÁN SONG SONG
META-HEURISTIC**

Chuyên ngành: Khoa học máy tính

Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng - Năm 2012

**Công trình được hoàn thành tại
ĐẠI HỌC ĐÀ NẴNG**

Người hướng dẫn khoa học: PGS.TSKH. Trần Quốc Chiến

Phản biện 1:

PGS.TS. Võ Trung Hùng

Phản biện 2:

TS. Hoàng Thị Lan Giao

Luận văn sẽ được bảo vệ tại Hội đồng chấm Luận văn tốt nghiệp Thạc sĩ kỹ thuật họp tại Đại học Đà Nẵng vào ngày 04 tháng 03 năm 2012.

** 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 tối ưu tổ hợp là dạng bài toán có độ phức tạp tính toán cao thuộc lớp NP khó. Sự ra đời của giải thuật Meta-Heuristic đã giải quyết các bài toán với hiệu quả cao cho kết quả lời giải gần tối ưu như họ giải thuật kiến (Ant Algorithm), giải thuật luyện thép SA (Simulated Annealing), giải thuật di truyền GA (Genetic Algorithm).

Với độ phức tạp tính toán cao của các bài toán tối ưu tổ hợp cũng như đòi hỏi về mặt thời gian, việc giải các bài toán này với tính chất tuần tự của giải thuật sẽ gặp phải những vấn đề về thời gian thực hiện chương trình, tốc độ xử lý, khả năng lưu trữ của bộ nhớ, xử lý dữ liệu với quy mô lớn... Kích thước bài toán tăng lên và không gian tìm kiếm càng lớn yêu cầu cần phải song song hóa các giải thuật để tăng tốc độ và hiệu quả của giải thuật.

Mục đích của đề tài là giải quyết bài toán tìm đường đi ngắn nhất bằng thuật toán kiến song song nhằm phát huy sức mạnh của bài toán. Trên cơ sở đó sẽ đưa ra kết quả đánh giá hiệu quả của thuật toán kiến trên các mô hình song song.

2. Mục đích nghiên cứu

Các mục tiêu cụ thể gồm:

- Nghiên cứu về giải thuật Meta-Heuristic đặc biệt là họ các giải thuật kiến
- Nghiên cứu về các vấn đề song song hóa và giải thuật đàn kiến song song.
- Áp dụng giải thuật kiến song song vào bài toán tìm đường đi ngắn nhất.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu

- Nghiên cứu các giải thuật kiến
- Mô hình tính toán song song Message Passing Interface
- Thuật toán kiến song song

Phạm vi nghiên cứu

- Tập trung nghiên cứu thuật toán song song áp dụng vào giải thuật kiến.

- Việc thử nghiệm đối với bài toán người du lịch - Travelling Salesman problem (TSP) được sử dụng là thư viện chuẩn TSPLIB

4. Phương pháp nghiên cứu

Phương pháp tài liệu:

Nghiên cứu lý thuyết về thuật toán kiến, các vấn đề song song hóa. Trên cơ sở lý thuyết nghiên cứu được sẽ vận dụng kết quả tìm kiếm tối ưu của thuật kiến song song vào bài toán người du lịch.

Phương pháp thực nghiệm

Xây dựng chương trình và đánh giá kết quả thử nghiệm với các mô hình song song.

5. Ý nghĩa khoa học và thực tiễn của đề tài

Nghiên cứu và giới thiệu thuật toán đàn kiến và thuật toán đàn kiến song song trong việc giải bài toán tìm đường đi ngắn nhất và ứng dụng thuật toán vào bài toán người du lịch.

6. Cấu trúc luận văn

Nội dung chính của luận văn này được chia thành ba chương với nội dung như sau:

Chương 1 – Cơ sở lý thuyết: Nội dung chính là tìm hiểu, nghiên cứu lý thuyết liên quan đến vấn đề nghiên cứu về lý thuyết đồ thị, các vấn đề lập trình song song, thuật toán đàn kiến .

Chương 2 – Thuật toán kiến song song: Từ thuật toán tối ưu đàn kiến tuần tự thực hiện chuyển sang thuật toán tối ưu đàn kiến song song trên mô hình truyền thông điệp.

Chương 3 – Phân tích, xây dựng và cài đặt chương trình: Phân tích chức năng và xây dựng chương trình ứng dụng vào bài toán người du lịch đồng thời tiến hành chạy thử nghiệm, đánh giá kết quả.

CHƯƠNG 1 CƠ SỞ LÝ THUYẾT

1.1. CÁC KHÁI NIỆM CƠ BẢN VỀ ĐỒ THỊ

1.1.1. Định nghĩa đồ thị

1.1.2. Tính liên thông của đồ thị

1.1.3. Đồ thị Euler và đồ thị Hamilton

Định nghĩa 1.13. Chu trình (tương ứng đường đi) đơn chứa tất cả các cạnh (hoặc cung) và các đỉnh của đồ thị (vô hướng hoặc có hướng) G được gọi là chu trình (tương ứng đường đi) Euler. Một đồ thị liên thông (liên thông yếu đối với đồ thị có hướng) có chứa một chu trình (tương ứng đường đi) Euler được gọi là đồ thị Euler (tương ứng nửa Euler).

Định nghĩa 1.14. Chu trình (tương ứng đường đi) sơ cấp chứa tất cả các đỉnh của đồ thị (vô hướng hoặc có hướng) G được gọi là chu trình (tương ứng đường đi) Hamilton. Một đồ thị có chứa một chu trình (tương ứng đường đi) Hamilton được gọi là đồ thị Hamilton (tương ứng nửa Hamilton).

1.1.4. Biểu diễn đồ thị trên máy tính

1.1.4.1. Ma trận kề, ma trận trọng số

1.1.4.2. Danh sách cạnh (cung)

1.1.4.3. Danh sách kề

1.2. TÍNH TOÁN SONG SONG VÀ CÁC VẤN ĐỀ SONG SONG HÓA

1.2.1. Mô hình máy tính song song

Một hệ thống máy tính song song là một máy tính với nhiều hơn một bộ xử lý cho phép xử lý song song. Dựa vào sự phân biệt ở kết nối giữa các bộ xử lý (hay thành phần xử lý), giữa

bộ xử lý và bộ nhớ mà có rất nhiều loại kiến trúc máy tính song song khác nhau. Nhưng theo nguyên tắc phân loại của Flynn thì có hai kiến trúc máy tính song song song song thông dụng sau:

1.2.1.1. Mô hình Đơn dòng lệnh đa dữ liệu – SIMD

1.2.1.2. Mô hình đa lệnh đa dữ liệu - MIMD

1.2.2. Mô hình lập trình song song

Một mô hình lập trình song song là sử dụng một tập các kỹ thuật phần mềm để thể hiện các giải thuật song song và đưa ứng dụng vào thực hiện trong hệ thống song song. Mô hình bao gồm các ứng dụng, ngôn ngữ, bộ biên dịch, thư viện, hệ thống truyền thông và vào/ra song song. Hiện nay có rất nhiều mô hình lập trình song song: Đa luồng (Threads), Truyền thông điệp (Message Passing), Song song dữ liệu (Data Parallel), Lai (Hybird).

1.2.3. Các chiến lược song song hóa thuật toán

1.2.3.1. Song song hóa kết quả

1.2.3.2. Song song hóa đại diện

1.2.3.3. Song song hóa chuyên biệt

1.3. MÔ HÌNH TRUYỀN THÔNG ĐIỆP - MPI

MPI - Message Passing Interface - là thư viện truyền thông điệp tiêu chuẩn dựa trên sự đồng thuận của diễn đàn MPI với hơn 40 tổ chức tham gia bao gồm cả các nhà cung cấp, các nhà nghiên cứu, các nhà phát triển thư viện phần mềm và người sử dụng. Đó là một thư viện các hàm (trong C) hoặc tiến trình con (trong Fortran) cho phép bạn chèn vào trong code để thực hiện trao đổi dữ liệu giữa các tiến trình. Mục tiêu của MPI là cung cấp một tiêu chuẩn được sử dụng rộng rãi để viết các chương trình chuyển gói tin đảm bảo tính di động, hiệu quả và linh hoạt.

1.3.1. Các khái niệm cơ bản

1.3.2. Các đặc tính cơ bản của một chương trình MPI

1.3.3. Trao đổi thông tin điểm – điểm

Kỹ thuật truyền thông cơ bản của MPI là sự chuyển giao dữ liệu giữa hai xử lý, một bên gửi và một bên nhận, chúng ta gọi hình thức này là Point to Point (điểm - điểm). Hầu hết các cấu trúc xử lý của chuẩn MPI đều dựa trên truyền thông Point to Point.

1.3.3.1. Các thông tin của thông điệp

1.3.3.2. Các hình thức truyền thông

1.3.4. Các kiểu dữ liệu đã được định nghĩa của MPI

1.4. TỐI ƯU TỔ HỢP VÀ BÀI TOÁN NGƯỜI DU LỊCH

1.4.1. Bài toán Tối ưu tổ hợp

Bài toán tối ưu hóa tổ hợp liên quan tới việc tìm giá trị cho các biến số rời rạc như lời giải tối ưu mà có lưu ý tới hàm mục tiêu (objective function) cho trước. Bài toán có thể là bài toán tìm cực đại hoặc tìm cực tiểu. Một cách thông thường, bài toán tối ưu hoá tổ hợp được cho dưới dạng bộ 3 (S, f, Ω) . Trong đó S là tập các lời giải ứng cử viên, f là hàm mục tiêu (hàm này gán giá trị $f(s)$ cho mỗi lời giải ứng cử viên $s \in S$), và Ω là tập các ràng buộc của bài toán. Các lời giải thuộc tập $S^* \subseteq S$ thỏa mãn tập các ràng buộc Ω gọi là lời giải khả thi. Mục tiêu bài toán là tìm ra một lời giải khả thi tối ưu toàn cục s^* . Với các bài toán tối ưu hóa cực tiểu là tìm lời giải s^* với giá nhỏ nhất, nghĩa là $f(s^*) \leq f(s)$ với mọi lời giải $s \in S$. Ngược lại bài toán tối ưu hóa cực đại là tìm lời giải s^* với giá lớn nhất, nghĩa là $f(s^*) \geq f(s)$ với mọi lời giải $s \in S$. Bài toán tối ưu hóa tổ hợp có thể chia 2 loại: Bài toán tĩnh và bài toán động.

1.4.2. Bài toán Người du lịch

1.4.2.1. Phát biểu bài toán

Có một người du lịch hay một người giao hàng cần đi giao hàng tại n thành phố. Anh ta 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, và khoảng cách từ một thành phố đến các thành phố khác đã được biết trước. 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.

1.4.2.2. Phân loại bài toán

1.4.2.3. Các phương pháp tiếp cận giải bài toán TSP

Có nhiều hướng để tiếp cận bài toán TSP như thiết kế thuật toán tìm lời giải chính xác, thuật toán xấp xỉ, thuật toán Heuristic và giải quyết các trường hợp đặc biệt.

1.5. TỔNG QUAN VỀ THUẬT TOÁN KIẾN

1.5.1. Giới thiệu chung

Tối ưu hóa thuật toán đàn kiến (Ant Colony Optimization - ACO) là một trong các thuật toán MetaHeuristic được thiết kế để giải quyết các bài toán tối ưu tổ hợp, sử dụng phương pháp tính xác suất để tìm đường đi ngắn nhất của đồ thị. Hệ thống ACO lần đầu tiên được Marco Dorigo giới thiệu trong luận văn của mình vào năm 1992, và được gọi là Hệ thống kiến (Ant System, hay AS).

1.5.2. Đàn kiến tự nhiên

Kiến là loại cá thể sống bầy đàn. Chúng giao tiếp với nhau thông qua mùi mà chúng để lại trên hành trình mà chúng đi qua. Mỗi kiến khi đi qua một đoạn đường sẽ để lại trên đoạn đó một chất mà chúng ta gọi là mùi. Số lượng mùi sẽ tăng lên khi có nhiều kiến cùng đi qua. Các con kiến khác sẽ tìm đường dựa vào mật độ mùi trên

đường, mật độ mùi càng lớn thì chúng càng có xu hướng chọn. Dựa vào hành vi tìm kiếm này mà đàn kiến tìm được đường đi ngắn nhất từ tổ đến nguồn thức ăn và sau đó quay trở về tổ của mình.

1.5.3. Đàn kiến nhân tạo

Đàn kiến nhân tạo (Artificial Ants) mô phỏng các hoạt động của đàn kiến tự nhiên và có một số thay đổi, điều chỉnh so với đàn kiến tự nhiên để tăng tính hiệu quả của thuật toán. Các tính chất của đàn kiến nhân tạo như sau:

- Ngoài thông tin pheromone thì đàn kiến nhân tạo còn sử dụng thông tin heuristic trong xây dựng luật di chuyển của chúng.
- Kiến nhân tạo có bộ nhớ để lưu thông tin của kiến nhằm mục đích xác định hành trình đã đi qua và để tính toán độ dài của hành trình đó.
- Lượng thông tin mùi pheromone được thêm vào bởi kiến nhân tạo là hàm của chất lượng lời giải mà chúng tìm được. Kiến nhân tạo thường chỉ thực hiện tăng lượng thông tin mùi sau khi đã hoàn thành lời giải.
- Kiến nhân tạo sử dụng cơ chế bay hơi thông tin pheromone để tránh bế tắc trong bài toán tối ưu cục bộ.

1.5.4. Các nguyên tắc của thuật toán kiến

CHƯƠNG 2

TỐI ƯU ĐÀN KIẾN VÀ THUẬT TOÁN KIẾN SONG SONG

2.1. TỐI ƯU ĐÀN KIẾN –ACO

2.1.1. Thuật toán Ant System (AS)

2.1.1.1. Quy tắc di chuyển của kiến

Trong thuật toán AS, kiến xây dựng một đường đi bắt đầu tại một đỉnh được chọn ngẫu nhiên.

Tại đỉnh i , một con kiến k sẽ chọn đỉnh j chưa được đi qua trong tập láng giềng của i theo công thức sau:

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

Trong đó:

p_{ij}^k : xác suất con kiến k lựa chọn cạnh (i,j)

τ_{ij} : nồng độ vết mùi trên cạnh (i,j)

α : hệ số điều chỉnh ảnh hưởng của τ_{ij}

η_{ij} : thông tin heuristic giúp đánh giá chính xác

sự lựa chọn của con kiến khi quyết định đi từ đỉnh i qua đỉnh j và được tính theo công thức:

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (2.2)$$

d_{ij} : khoảng cách giữa đỉnh i và đỉnh j

β : hệ số điều chỉnh ảnh hưởng của η_{ij}

N_i^k : tập các đỉnh láng giềng của i mà con kiến k

chưa đi qua

2.1.1.2. Quy tắc cập nhật thông tin mùi

Trong quá trình di chuyển tìm đường đi của đàn kiến, chúng thực hiện việc cập nhật thông tin mùi trên những đoạn đường mà

chúng đã đi qua. Gắn với mỗi cạnh (i,j) nồng độ vết mùi τ_{ij} và thông số heuristic η_{ij} trên cạnh đó.

Ban đầu nồng độ mùi trên mỗi cạnh (i,j) được khởi tạo bằng một hằng số c , hoặc được xác định theo công thức:

$$\tau_{ij} = \tau_0 = \frac{m}{C^m}, \forall (i, j) \quad (2.3)$$

Việc cập nhật pheromone được tiến hành như sau:

- Đầu tiên tất cả pheromone trên các cung sẽ được giảm đi bởi một lượng:

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} \quad (2.4)$$

Với ρ trong khoảng $(0,1)$ là tốc độ bay hơi của pheromone.

- Tiếp theo mỗi con kiến trong đàn sẽ đặt thêm một lượng thông tin pheromone trên những cung mà chúng đã đi qua trong hành trình của chúng.

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (2.5)$$

Trong đó: $\Delta \tau_{ij}^k$ là lượng pheromone mà con kiến k đặt lên cạnh mà nó đã đi qua và được tính như sau:

$$\Delta \tau_{ij}^k = \begin{cases} 1 / C^k, & \text{nếu kiến } k \text{ qua cung } (i,j) \\ 0, & \text{ngược lại} \end{cases} \quad (2.6)$$

Với: C^k là độ dài đường đi của con kiến thứ k sau khi hoàn thành đường đi, tức là bằng tổng các cung thuộc đường đi mà kiến đã đi qua.

2.1.2. Thuật toán Ant Colony System (ACS)

2.1.2.1. Quy tắc di chuyển của kiến

Trong thuật toán ACS, con kiến k đang ở đỉnh i, việc kiến chọn đỉnh j để di chuyển đến được xác định bằng qui luật như sau:

- Cho q_0 là một hằng số cho trước ($0 \leq q_0 \leq 1$)
- Chọn ngẫu nhiên một giá trị q trong khoảng $[0, 1]$
- Nếu $q \leq q_0$ kiến k chọn điểm j di chuyển tiếp theo dựa trên

giá trị lớn nhất của thông tin mùi và thông tin heuristic có trên cạnh tương ứng với công thức:

$$j = \arg_{l \in N_i^k} \max(\tau_{il} (\eta_{il})^\beta) \quad (2.7)$$

- Nếu $q > q_0$ kiến k sẽ chọn đỉnh j chưa được đi qua trong tập láng giềng của i theo một qui luật phân bố xác suất được xác định theo công thức sau:

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

2.1.2.2. Quy tắc cập nhật thông tin mùi

Cập nhật thông tin mùi toàn cục:

Một con kiến có đường đi tốt nhất sau mỗi lần lặp thì được phép cập nhật thông tin pheromone. Việc cập nhật được thực hiện theo công thức sau:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \quad (2.8)$$

Với $\Delta\tau_{ij}^{bs} = \frac{1}{C_{bs}}$ là lượng pheromone đặt lên cạnh (i,j)

mà kiến đi qua.

C_{bs} là độ dài đường đi tốt nhất của con kiến thứ k sau khi hoàn thành đường đi, tức là bằng tổng các cung thuộc đường đi tốt nhất mà kiến đã đi qua.

Cập nhật thông tin mùi cục bộ:

Công thức như sau:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\tau_0 \quad (2.9)$$

Với:

- ρ : là tham số bay hơi nằm trong khoảng $(0,1)$

- $\tau_0 = \frac{1}{nC_{mn}}$

- n : là số đỉnh hay số thành phố

- C_{mn} : chiều dài hành trình cho bởi phương pháp tìm kiếm gần nhất (nearest neighbor – nn).

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

Luật di chuyển của kiến được thực hiện tương tự như trong thuật toán ACS dựa trên công thức (2.7).

2.1.3.1. Quy tắc cập nhật thông tin mùi

Thuật toán MMAS thực hiện việc cập nhật thông tin mùi khi toàn bộ kiến trong đàn hoàn thành lời giải và lượng thông tin mùi chỉ cập nhật trên các cạnh thuộc lời giải tối ưu nhất. Ban đầu cũng thực hiện bay hơi thông tin mùi trên các cạnh thuộc lời giải tối ưu với một lượng được xác định tại công thức (2.4).

Lượng pheromone trên một cạnh được xác định như sau :

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{best}$$

với $\Delta\tau_{ij}^{best} = \begin{cases} 1/C_{best} & \text{nếu kiến qua cạnh (i,j)} \\ 0 & \text{ngược lại} \end{cases}$

C_{best} là độ dài đường đi ngắn nhất của con kiến thứ k sau khi cả đàn hoàn thành đường đi.

2.1.3.2. Khởi tạo và khởi tạo lại thông tin mùi

Thuật toán MMAS đã thêm vào giá trị cận trên và giá trị cận dưới cho thông tin pheromone gọi là τ_{\min} và τ_{\max} .

Sau mỗi lần cập nhật giá trị thông tin mùi τ_{ij} , nếu $\tau_{ij} < \tau_{\min}$ thì sẽ gán $\tau_{ij} = \tau_{\min}$ và nếu $\tau_{ij} > \tau_{\max}$ thì gán $\tau_{ij} = \tau_{\max}$.

Giá trị cận trên τ_{\max} thường được thiết lập với công thức sau:

$$\tau_{\max} = \frac{1}{\rho C_{best}}$$

Giá trị cận dưới τ_{\min} được xác định bằng công thức

$$\tau_{\min} = \tau_{\max} / 2n.$$

2.2. THUẬT TOÁN KIẾN CHO BÀI TOÁN NGƯỜI DU LỊCH

Áp dụng thuật toán kiến vào bài toán người du lịch với các thông số của kiến như sau:

- m: số lượng kiến
- τ_{ij} : nồng độ vết mùi trên cạnh (i,j)
- ρ : là tham số bay hơi nằm trong khoảng (0,1)
- p_{ij}^k : xác suất con kiến k lựa chọn cạnh (i,j)
- α : hệ số điều chỉnh ảnh hưởng của τ_{ij}
- η_{ij} : thông tin heuristic giúp đánh giá chính xác sự lựa

chọn của kiến đi từ đỉnh i tới đỉnh j

- β : hệ số điều chỉnh ảnh hưởng của η_{ij}
- N_i^k : tập các đỉnh láng giềng của i mà con kiến k chưa đi

qua

- $\Delta\tau_{ij}^k$ là lượng pheromone mà con kiến k đặt lên cạnh mà nó

đã đi qua

- NC: là số bước lặp của thuật toán
- S_k : đường đi của kiến k (tập các đỉnh mà kiến k đi qua)
- q: là giá trị ngẫu nhiên trong khoảng [0, 1]

- a_k^i : nhận giá trị True, False tương ứng với con kiến k đã thăm hoặc chưa đi qua đỉnh i

Các bước xây dựng thuật toán như sau:

➤ **Đầu vào:**

Đồ thị $G=(V,E)$, với tập đỉnh V và tập cạnh E, ma trận trọng số $D = d[i,j]$, với $i, j \in V$.

Số lượng kiến m.

➤ **Đầu ra:** đường đi S với khoảng cách ngắn nhất C_s

➤ **Các bước**

Bước 1: Khởi tạo

Khởi tạo các tham số NC, β , α , ρ , số lượng kiến m và số đỉnh n

Khởi tạo độ dài đường đi ngắn nhất C_{best} là hằng số

Tính độ dài đường đi ngắn nhất C_m

Tính giá trị $\tau_{\max} = \frac{1}{\rho C_m}$ và $\tau_{\min} = \tau_{\max} / 2n$

Khởi tạo giá trị q_0 với $(0 \leq q_0 \leq 1)$

Khởi gán điều kiện kết thúc stop := 1

Thiết lập ma trận pheromone trên tất cả các cạnh

for i:=1 to n do

for j:=1 to n do $\tau_{ij} = \tau_{\max}$

Bước 2: Xây dựng đường đi của kiến

Trường hợp hoàn thành xong tất cả các bước lặp: stop > NC thì xuất ra đường đi ngắn nhất và kết thúc

2.1 Thiết lập các đỉnh khi kiến chưa đi qua nhận giá trị false

for k:=1 to m do

for i:=1 to n do $a_k^i := false$

Thiết lập đường đi của kiến $S_k = \phi$

2.2 Chọn ngẫu nhiên vị trí xuất phát của kiến

for k:=1 to m do
 for i:=1 to n do
 $r \leftarrow \text{random}\{1..n\}$
 Bổ sung r vào đường đi $S_k := \{r\}$, $a_k^r := \text{true}$;
 Gán độ dài đường đi $C_k := 0$

2.3 Xác định đỉnh đến tiếp theo của kiến k

- Chọn ngẫu nhiên một giá trị q : $q \leftarrow \text{random}\{0..1\}$
- Nếu $q \leq q_0$ kiến k chọn điểm u di chuyển tiếp theo với

$$u = \arg \max_{l \in N_r^k} (\tau_{rl} (\eta_{rl})^\beta)$$

- Nếu $q > q_0$ kiến k sẽ chọn đỉnh u chưa được đi qua trong tập láng giềng của r theo công thức sau:

$$p_{ru}^k = \frac{(\tau_{ru})^\alpha (\eta_{ru})^\beta}{\sum_{l \in N_r^k} (\tau_{rl})^\alpha (\eta_{rl})^\beta}, u \in N_r^k$$

- Chọn đỉnh u là đỉnh tiếp theo, bổ sung đỉnh u vào S_k
 $S_k := \{r, u\}$
- Tăng độ dài đường đi $C_k := C_k + d_{ru}$
- Gán $a_k^u := \text{true}$

Bước 3: Xác định đường đi ngắn nhất

Ta có C_k là độ dài đường đi của con kiến k với $k=[1..m]$ thu được từ bước 2.

Nếu $C_k < C_{\text{best}}$ thì hiệu chỉnh $C_{\text{best}} := C_k$

Bước 4: Cập nhật thông tin mùi

Tại bước này, chỉ cập nhật thông tin mùi trên đường đi của con kiến k có giá trị C_k nhỏ nhất thu được từ bước 3, tức là giá trị C_{best}

4.1 Bay hơi thông tin mùi trên các cạnh

for i:=1 to n do

for j:=i to n do $\tau_{ij} := (1 - \rho)\tau_{ij}$

4.2 Thêm thông tin mùi trên các cạnh thuộc đường đi tốt nhất

$S_{k,\text{best}}$

Tính giá trị pheromone $\Delta\tau = \frac{1}{C_{k,\text{best}}}$

Nếu cạnh $(i, j) \in S_{k,\text{best}}$ thì $\tau_{ij} := \tau_{ij}^{\text{best}} + \Delta\tau$

Kiểm tra thông tin pheromone với cận trên và cận dưới

Nếu $\tau_{ij} < \tau_{\min}$ thì $\tau_{ij} = \tau_{\min}$

Nếu $\tau_{ij} > \tau_{\max}$ thì $\tau_{ij} = \tau_{\max}$

Tăng giá trị stop:=stop+1.

2.3. SONG SONG HÓA THUẬT TOÁN KIẾN

2.3.1. Tổng quan thuật toán kiến song song

Trong luận văn này trình bày chi tiết hai chiến lược song song hóa thuật toán kiến của B. Bullnheimer, G. Kotsis, C. Strauss là **song song đồng bộ** và **song song bất đồng bộ một phần**.

2.3.2. Thuật toán song song đồng bộ

2.3.2.1. Ý tưởng thuật toán

Thuật toán sử dụng mô hình Master/Slave trên kiến trúc bộ nhớ phân tán gồm một vi xử lý đảm nhận vai trò Master và các vi xử lý còn lại là Slave. Mỗi một Slave được gán cho một tác tử kiến và thực hiện nhiệm vụ xây dựng một đường đi. Master có nhiệm vụ khởi tạo các thông tin ban đầu, cập nhật thông tin pheromone toàn cục nhận từ các Slave và đưa ra kết quả tối ưu.

2.3.2.2. Các bước thuật toán

Đối với Master

Bước 1: Khởi tạo

Gửi các tham số đến mỗi Slave

Gửi ma trận d và pheromone τ đến mỗi Slave

Bước 2: Xác định đường đi ngắn nhất của kiến

Bước 3: Cập nhật thông tin Pheromone

Đối với Slave

Bước 1: Nhận các tham số từ Master

Bước 2: Xây dựng đường đi

Bước 3: Gửi đường đi S_k và độ dài đường đi C_k về Master

2.3.2.3. Sơ đồ thuật toán

2.3.3. Song song bất đồng bộ một phần

2.3.3.1. Ý tưởng thuật toán

Trong chiến lược này số lượng kiến m nhiều hơn số vi xử lý P, ($P < m$). Như vậy, mỗi Slave đảm nhận thực hiện xây dựng đường đi cho một đàn kiến với số lượng là m/P . Việc phân chia theo chiến lược này là giảm đáng kể chi phí truyền thông giữa Master và Slave.

2.3.3.2. Các bước thuật toán

Đối với Master

Bước 1: Khởi tạo

Bước 2: Xác định đường đi ngắn nhất

Trường hợp $stop > NC$ thì xuất ra đường đi ngắn nhất và kết thúc

2.1 Gửi các tham số đến mỗi Slave

Gửi ma trận d và pheromone τ đến mỗi Slave

2.2 Nhận đường đi S_k độ dài đường đi C_k từ các Slave

2.3 Xác định đường đi ngắn nhất

Nếu $C_k < C_{best}$ thì $C_{best} := C_k$

Bước 3: Cập nhật thông tin Pheromone

3.1 Bay hơi thông tin mùi trên các cạnh

3.2 Thêm thông tin mùi trên các cạnh thuộc đường đi ngắn nhất $S_{k,best}$

3.3 Gửi ma trận Pheromone τ mới cho Slave

Đối với Slave

Bước 1: Nhận các tham số từ Master

Bước 2: Xây dựng đường đi cho đàn kiến

Nếu $step > T$ thì dừng và chuyển sang bước 3

2.1 Khởi tạo thành phố xuất phát của mỗi kiến tại mỗi Slave

2.2 Xây dựng đường đi cho từng kiến

2.3 Cập nhật ma trận pheromone cục bộ

Bước 4: Gửi đường đi S_k và độ dài đường đi C_{best} về

Master

2.3.3.3. Sơ đồ thuật toán

2.3.4. Hiệu quả của thuật giải song song

Việc xem xét hiệu quả của thuật giải song song thường căn cứ vào các yếu tố như: thời gian thi hành, tốc độ (Speedup), hiệu suất (efficiently), chi phí (Cost) và qui mô (Scaling).

2.3.4.1. Chi phí

Ký hiệu T_0 là hàm overhead, ta có:

$$T_0 = pT_P - T_S$$

2.3.4.2. Tốc độ

Tốc độ ký hiệu $S = T_S/T_P$.

2.3.4.3. Hiệu suất

Hiệu suất là độ đo thời gian mà phần tử xử lý (PE) sử dụng hữu ích, là tỷ số giữa tốc độ (Speedup) và số phần tử xử lý PE ($E = S/p$).

CHƯƠNG 3

PHÂN TÍCH, XÂY DỰNG VÀ CÀI ĐẶT CHƯƠNG TRÌNH

3.1. ĐẶC TẢ CẤU TRÚC DỮ LIỆU VÀ CÁC CHỨC NĂNG CHÍNH CỦA CHƯƠNG TRÌNH

3.1.1. Cấu trúc dữ liệu

3.1.1.1. Các thông số chính

Trong phần này mô tả cấu trúc dữ liệu của các thông số chính của chương trình. Chương trình sử dụng mã nguồn thuật toán kiến trúc tuần tự ACOTSP.V1.01 tham khảo trong [15] và phát triển thành chương trình song song. Chương trình sử dụng các tập tin gr24.tsp, st70.tsp, kroA100.tsp trong thư viện TSPLIB [13] để đánh giá kết quả thử nghiệm.

3.1.1.2. Cấu trúc dữ liệu của kiến

3.1.1.3. Kiểu dữ liệu của MPI

3.1.2. Các chức năng chính

3.1.2.1. Khởi tạo dữ liệu

Chức năng khởi tạo dữ liệu bao gồm các hàm khởi tạo cho cấu trúc dữ liệu đã nêu trong phần 3.1.1 Đầu tiên sẽ thực hiện đọc các dữ liệu từ tập tin trong TSP, sau đó tạo lập ma trận trọng số, danh sách các láng giềng, khởi tạo thông tin Heuristic, khởi tạo cấu trúc kiến và khởi tạo một số biến bổ sung để theo dõi thực hiện thuật toán.

3.1.2.2. Xây dựng đường đi

Chức năng xây dựng đường đi cho kiến được thực hiện thông qua 3 giai đoạn riêng biệt. Đầu tiên mỗi kiến sẽ được phân bố ngẫu nhiên tại một thành phố xuất phát, sau đó kiến xây dựng một đường đi qua tất cả các thành phố bằng cách xác định vị trí thành phố di chuyển tiếp theo và cuối cùng là trở về lại vị trí xuất phát và tính độ dài đường đi hoàn thành.

3.1.2.3. Cập nhật Pheromone

Có hai giai đoạn chính trong chức năng cập nhật thông tin pheromone đó là thực hiện bay hơi trên các cung và cập nhật thông tin pheromone mới trên các cung thuộc đường đi tốt nhất của kiến. Trong thuật toán MMAS có bổ sung thêm giá trị cận trên và giá trị cận dưới của thông tin pheromone nên có chức năng kiểm tra các giá trị này.

3.1.2.4. Thực hiện song song

Việc thực hiện song song trong luận văn này là cho p tác tử kiến hoặc đàn kiến tiến hành tìm đường đi hay xây dựng giải pháp cho mình trên p bộ vi xử lý.

Bài toán được xây dựng chạy trên kiến trúc bộ nhớ phân tán với mô hình song song truyền thông điệp, sử dụng thư viện MPI.

3.2. NGÔN NGỮ LẬP TRÌNH VÀ MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG

Ngôn ngữ lập trình được sử dụng là ngôn ngữ C, trình biên dịch là GCC (GNU C compiler), phần mềm hỗ trợ phát triển ứng dụng là CodeBlocks. Thư viện chuẩn đặc tả truyền thông điệp sử dụng trong chương trình là bản MPICH2.

3.3. KẾT QUẢ THỰC NGHIỆM CHƯƠNG TRÌNH

3.3.1. Thực nghiệm chương trình

3.3.2. Đánh giá kết quả

3.3.2.1. Đánh giá chương trình song song

- Chương trình song song không phá vỡ cấu trúc của thuật toán tuần tự cho nên chất lượng lời giải của nó tương tự như chương trình tuần tự.

- Chương trình chỉ thực hiện song song ở bước xây dựng giải pháp đường đi của kiến.

- Chương trình không thực hiện cấu trúc liên kết mạng trao đổi thông tin giữa các đàn kiến, chi phí truyền thông giữa Master/Slave lớn, thời gian thực hiện phụ thuộc vào cấu hình máy tính chạy chương trình.

3.3.2.2. Đánh giá kết quả thực hiện

- Kết quả của chương trình tuần tự và song song là tương tự nhau căn cứ trên các tập tin kết quả xuất ra của chương trình.

- Qua thực nghiệm, thời gian thực hiện thuật toán song song tìm kiếm kết quả tối ưu nhanh hơn so với tìm kiếm bằng thuật toán tuần tự và tỉ lệ phụ thuộc vào số lượng vi xử lý sử dụng.

- Khi số lượng bộ vi xử lý tăng lên thì thời gian trung bình để tìm ra giải pháp tối ưu sẽ giảm.

- Đối với cùng số lượng các bộ vi xử lý thì thời gian trung bình để tìm ra giải pháp tối ưu tại mỗi vòng lặp khác nhau cũng khác nhau, được mô tả ở bảng 3.1.

Bảng 3.1 Tốc độ thực hiện tương ứng với các dữ liệu của bài toán TSP

Bài toán TSP	Tốc độ thực hiện		
	2	4	6
Gr24.tsp	0.8	0.7	0.7
St70.tsp	1.27	1.23	1.23
kroA100.tsp	1.41	1.37	1.37

Bảng 3.2 Thời gian trung bình tìm ra giải pháp tối ưu trên số vi xử lý

Số vi xử lý	2	4	6
Thời gian cho mỗi vòng lặp (giây)	15	5	10
	12	9	7
	21	15	11
	29	33	13
	19	10	8
	10	7	6
	17	5	7
Trung bình	17.57	12	8.85

Bảng 3.3 Hiệu suất tương ứng với các dữ liệu của bài toán TSP

Bài toán TSP	Hiệu suất thực hiện		
	2	4	6
Gr24.tsp	0.4	0.175	0.116
St70.tsp	0.63	0.307	0.205
kroA100.tsp	0.705	0.342	0.228

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Những kết quả nghiên cứu của luận văn cho phép rút ra những kết luận:

Về lý thuyết

- Tìm hiểu được thuật toán đàn kiến trong việc xác định đường đi ngắn nhất bằng phương pháp tính xác suất giữa các điểm di chuyển dựa trên các thông tin về độ dài đường đi, thông tin heuristic và thông tin mùi pheromone mà các con kiến để lại trên đường đi trong quá trình di chuyển.

- Hiểu được các phương pháp tiếp cận tối ưu thuật toán để nâng cao độ chính xác cho thuật toán kiến, cụ thể là cải tiến quy luật di chuyển của kiến và qui luật cập nhật thông tin mùi pheromone thể hiện ở thuật toán Ant Colony System và Max-Min Ant System.

- Hiểu được các chiến lược song song hóa thuật toán và thực hiện song song hóa thuật toán kiến trên kiến trúc bộ nhớ phân tán và mô hình song song Master/Slave. Qua đó đánh giá được thuật toán dựa trên các tiêu chí về tốc độ, hiệu suất và chi phí.

Về thực nghiệm

Phân tích và mô tả được các nhóm chức năng của chương trình ứng dụng. Đây là cơ sở để tiếp tục phát triển đề tài trong tương lai.

Cài đặt thành công chương trình thực nghiệm tìm đường đi ngắn nhất của thuật toán kiến áp dụng vào bài toán người du lịch trên môi trường song song. Chương trình được kiểm tra với dữ liệu đầu vào được khai thác từ thư viện TSPLIB.

Đánh giá ưu điểm và hạn chế của đề tài

Về lý thuyết

- Luận văn chỉ tập trung nghiên cứu thuật toán đàn kiến trong họ P-Meta heuristic, chưa nghiên cứu các thuật toán khác (như thuật toán đàn ong) để có những so sánh đánh giá.

- Trong việc song song hóa thuật toán đàn kiến, chưa tìm hiểu về quá trình trao đổi thông tin mùi và giải pháp giữa các đàn kiến trong việc xây dựng đường đi.

Về thực nghiệm

- Mặc dù đã phân tích được các nhóm chức năng nhưng còn nhiều chức năng chưa cài đặt hoàn chỉnh hoặc đã cài đặt nhưng chưa kiểm tra thử nghiệm trong thực tế.

- Chương trình chỉ mới song song hóa được một phần dựa theo ý tưởng đã nêu, chưa thực nghiệm trên kiến trúc song song bộ nhớ chia sẻ để có đánh giá hiệu quả của thuật toán trên các loại kiến trúc khác nhau.

- Chương trình còn tính chất mô phỏng chưa phải là một ứng dụng cụ thể.

2. Hướng phát triển

Về lý thuyết

- Tiếp tục nghiên cứu sâu thêm về thuật toán đàn kiến với những phương pháp tối ưu hóa được cải tiến áp dụng trong các bài toán tối ưu hóa tổ hợp phức tạp.

- Nghiên cứu một số thuật toán thuộc họ P-Meta heuristic và song song hóa để so sánh, đánh giá tính chính xác và mức độ hiệu quả giữa các thuật toán meta heuristic.

Về thực nghiệm

- Tiếp tục cài đặt các chức năng còn thiếu, đặc biệt là bổ sung chức năng trao đổi thông tin giữa các đàn kiến trên các mô hình liên kết mạng khác nhau.