

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

VŨ ĐỨC QUANG

**ÁP DỤNG THUẬT TOÁN TỐI ƯU HÓA
ĐÀN KIẾN ĐỂ GIẢI QUYẾT BÀI TOÁN
VỊ TRÍ CƠ SỞ**

Ngành : Công nghệ thông tin
Chuyên ngành : Hệ thống thông tin
Mã số : 60480104

TÓM TẮT LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội - 2016

MỤC LỤC

PHẦN MỞ ĐẦU	3
1.1. Độ phức tạp thuật toán.....	5
1.2. NP-đầy đủ.....	5
1.2.1. Bài toán quyết định	5
1.2.2. Bằng chứng ngắn gọn để kiểm tra.....	5
1.2.3. Lớp bài toán P, NP và co-NP	5
1.2.4. Lớp bài toán NP-khó và NP-đầy đủ.....	7
1.3. Bài toán vị trí cơ sở không hạn chế khả năng.....	7
1.4. Bài toán vị trí cơ sở có hạn chế khả năng.....	8
1.5. Bài toán vị trí cơ sở cạnh tranh	9
1.6. Bài toán bố trí vị trí xây dựng	10
1.7. Bài toán bố trí cơ sở theo hàng	11
1.8. Kết chương	12
CHƯƠNG 2. THUẬT TOÁN TỐI ƯU ĐÀN KIẾN.....	13
2.1. Từ kiến nhân tạo đến kiến thực.....	13
2.1.1. Kiến thực.....	13
2.1.2. Kiến nhân tạo.....	13
2.2. Phương pháp ACO cho bài toán TỰTH tổng quát.....	13
2.2.1. Đồ thị cấu trúc	13
2.2.2. Mô tả thuật toán ACO tổng quát.....	13
2.3. Phương pháp ACO giải bài toán TSP	14
2.3.1. Bài toán TSP và đồ thị cấu trúc	14
2.3.2. Các thuật toán ACO cho bài toán TSP	14

2.4. Một số vấn đề khác khi áp dụng ACO.....	15
2.4.1. Đặc tính hội tụ	15
2.4.2. Thực hiện song song.....	15
2.4.3. ACO kết hợp với tìm kiếm cục bộ	16
2.5. Kết luận chương.....	16
CHƯƠNG 3. CÀI ĐẶT THỬ NGHIỆM	18
3.1. Thuật toán r p-ACO giải bài toán r p trung tâm	18
3.1.1. Lược đồ tổng quát	18
3.1.3. Kết quả thử nghiệm.....	19
3.2. So sánh các thuật toán giải bài toán CSLP	19
3.3. Áp dụng thuật toán ACO-SRFL giải bài toán SRFL.....	20
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	21

PHẦN MỞ ĐẦU

Trong cuộc sống, việc đạt lợi nhuận cao hay thấp trong kinh doanh buôn bán, cung cấp dịch vụ phụ thuộc rất nhiều yếu tố. Trong đó, có một yếu tố quan trọng đầu tiên, đóng góp một phần rất lớn đó là xác định được địa điểm đặt dịch vụ thuật lợi – nơi cung cấp dịch vụ cho khách hàng. Có rất nhiều tiêu chí đặt ra khi chọn vị trí đặt cơ sở như: thuận tiện về giao thông, là nơi tập trung đông dân cư, ... để làm sao thu được lợi nhuận cao nhất. Đặc biệt, đối với các trường hợp khẩn cấp như cứu thương, cứu hỏa thì yêu cầu về khoảng cách nhỏ nhất là vô cùng quan trọng, có thể nói là quan trọng nhất trong các yếu tố. Bài toán đặt ra là: đặt các trạm dịch vụ ở đâu để thời gian di chuyển bệnh nhân từ nơi xa bệnh viện nhất (hoặc ngược lại, từ các trạm dịch vụ đến nơi bệnh nhân xa nhất) là nhỏ nhất có thể. Còn với dịch vụ phổ biến như trạm xăng, thùng phiếu, bất điện thoại, ... thì yêu cầu lại là tổng chi phí từ các khách hàng (hay người có nhu cầu) đến địa điểm phục vụ gần khách hàng nhất là nhỏ nhất.

Bài toán này thuộc dạng NP-khó, có rất nhiều các thuật giải khác nhau được đưa ra để có thể tìm lời giải tối ưu cho bài toán này như: thuật toán di truyền, thuật toán tham lam, thuật toán tối ưu hóa bầy đàn, tìm kiếm tabu... Tuy nhiên các giải thuật trên đều tốn chi phí về thời gian và/hoặc không gian lớn.

Tối ưu hóa đàn kiến (*Ant Colony Optimization - ACO*) là cách tiếp cận metaheuristic tương đối mới, do Dorigo giới thiệu vào năm 1991 và liên tục được phát triển cho đến nay. Thành công đầu tiên của các thuật toán ACO là giải quyết bài toán Người chào hàng nổi tiếng với số đỉnh lên tới hơn 2000 với kết quả thu được là tốt, hiệu quả của nó được chứng minh bằng thực nghiệm.

Đầu tiên, luận văn đã hệ thống hóa các kiến thức cơ sở về lý thuyết độ phức tạp thuật toán, lớp các bài toán P, NP, NP-khó và NP-đầy đủ. Sau đó, luận văn trình bày các bài toán điển hình trong lớp các bài toán vị trí cơ sở cùng các nghiên cứu đã được công bố gần đây. Tiếp theo, tác giả đề xuất thuật toán dựa trên giải thuật tối ưu đàn kiến giải một số bài toán vị trí cơ sở hiện nay và so sánh kết quả thu được với một số công trình đã được công bố gần đây nhằm rút ra được các ưu nhược điểm của thuật toán. Kết quả này đã được tác giả công bố trong 2 công trình nghiên cứu khoa học.

Nội dung chính của luận văn được chia thành 4 chương như sau:

Chương 1: Tìm hiểu tổng quan về các kiến thức cơ sở về độ phức tạp thuật toán, lớp các bài toán P, NP và NP-khó và các bài toán thuộc lớp bài toán vị trí cơ sở cũng như các công bố gần đây.

Chương 2: Trình bày chi tiết về thuật toán tối ưu hóa đàn kiến.

Chương 3: Trình bày về cài đặt chương trình, thử nghiệm và so sánh kết quả với một số công trình đã công bố gần đây.

Kết luận

Tài liệu tham khảo

CHƯƠNG 1

MỘT SỐ KIẾN THỨC TỔNG QUAN VÀ BÀI TOÁN VỊ TRÍ CƠ SỞ

Trong cuộc sống, việc đạt lợi nhuận cao hay thấp trong kinh doanh buôn bán, cung cấp dịch vụ phụ thuộc rất nhiều yếu tố. Trong đó, có một yếu tố quan trọng đầu tiên, đóng góp một phần rất lớn đó là xác định được địa điểm đặt dịch vụ thuận lợi – nơi cung cấp dịch vụ. Có rất nhiều tiêu chí đặt ra khi chọn địa điểm: thuận tiện về giao thông, là nơi tập trung đông dân cư... để làm sao thu được lợi nhuận cao nhất. Đặc biệt, đối với các trường hợp khẩn cấp như cứu thương, cứu hỏa thì yêu cầu về khoảng cách nhỏ nhất là vô cùng quan trọng, có thể nói là quan trọng nhất trong các yếu tố.

Yêu cầu của bài toán vị trí cơ sở là tìm phương án đặt các trạm dịch vụ ở đâu để thời gian di chuyển bệnh nhân từ nơi xa bệnh viện nhất (hoặc ngược lại, từ các trạm dịch vụ đến nơi bệnh nhân xa nhất) là nhỏ nhất có thể. Còn với các dịch vụ phổ biến như trạm xăng, thùng phiếu, bộ điện thoại,... thì yêu cầu lại là tổng chi phí từ khách hàng (hay người có nhu cầu) đến địa điểm phục vụ gần khách hàng nhất là nhỏ nhất.

1.1. Độ phức tạp thuật toán

1.2. NP-dầy đủ

1.2.1. Bài toán quyết định

1.2.2. Bằng chứng ngắn gọn để kiểm tra

1.2.3. Lớp bài toán P , NP và $co-NP$

Định nghĩa: Ta gọi P là lớp các bài toán có thể giải được sau thời gian đa thức.

Ví dụ 5:

Bài toán về tính liên thông của đồ thị có thể giải được nhờ thuật toán với thời gian tính là $O(n^2)$, vì vậy, nó là bài toán thuộc lớp P. Bài toán cây khung nhỏ nhất giải được nhờ thuật toán Prim với thời gian $O(n^2)$, cũng thuộc vào lớp P.

Định nghĩa: Ta gọi NP là lớp các bài toán quyết định mà để xác nhận câu trả lời 'yes' của nó ta có thể đưa ra bằng chứng ngắn gọn dễ kiểm tra.

Ví dụ 6:

Các bài toán trình bày trong ví dụ 2 đều thuộc lớp NP.

Định nghĩa: Ta gọi co-NP là lớp các bài toán quyết định mà để xác nhận câu trả lời 'no' của nó ta có thể đưa ra bằng chứng ngắn gọn dễ kiểm tra.

Ví dụ 7:

Các bài toán trình bày trong ví dụ 3 đều thuộc lớp co-NP.

Bài toán trong ví dụ 4 còn chưa biết có thuộc vào lớp nào trong hai lớp NP và co-NP hay không.

Rõ ràng, nếu một bài toán thuộc lớp P, thì ta có thể tìm được lời giải của nó sau thời gian đa thức, và vì thế ta cũng có thể xác nhận được câu trả lời 'yes' của nó (bằng việc giải nó) sau thời gian đa thức.

Vì vậy:

P NP

Tương tự như vậy ta có:

P co-NP

Một trong những vấn đề trung tâm của lý thuyết tính toán, đó là chứng minh hoặc bác bỏ đẳng thức:

$P = NP$

Cho đến hiện nay vấn đề này vẫn là vấn đề mở.

1.2.4. Lớp bài toán NP-khó và NP-đầy đủ

Ta sẽ đưa ra định nghĩa về những bài toán khó nhất trong lớp NP: bài toán NP-đầy đủ (NP-complete).

Định nghĩa:

Một bài toán quyết định A được gọi là NP-đầy đủ nếu như:

- i. A là bài toán trong NP;
- ii. Mọi bài toán trong NP đều có thể qui dẫn về A.

Như vậy, có thể nói khái niệm về "bài toán khó nhất" trong lớp NP được xây dựng trên cơ sở phép qui dẫn. Nếu tất cả các bài toán trong NP có thể qui dẫn về một bài toán A thì A khó không kém bất cứ bài toán nào trong số chúng. Điều đáng ngạc nhiên là sự tồn tại của những bài toán có tính chất như vậy.

Khó khăn nhất là việc tìm ra được một bài toán như vậy. Bởi vì hễ chúng ta đã có một bài toán NP-đầy đủ thì để ta có thể dễ dàng chứng minh nhiều bài toán khác là NP-đầy đủ nhờ sử dụng kết quả sau đây.

1.3. Bài toán vị trí cơ sở không hạn chế khả năng

Bài toán vị trí cơ sở không hạn chế khả năng được phát biểu như sau: Xét một tập $I = \{1, 2, 3, \dots, N\}$ các cơ sở tiềm năng cung cấp sản phẩm hoặc dịch vụ. Một cơ sở $i \in I$ có chi phí xây dựng là C_i ($C_i > 0$). Mỗi cơ sở mở có thể cung cấp một số lượng không giới hạn hàng hóa cho mỗi khách hàng. Và một tập $J = \{1, 2, \dots, M\}$ là tập các khách hàng sử dụng dịch vụ. Giá trị g_{ij} (với $i \in I$ và $j \in J$) là chi phí vận chuyển từ cơ sở i đến khách hàng j . Mục tiêu là xác định một tập hợp con S của tập hợp các địa điểm cơ sở tiềm năng I ($S \subseteq I, S \neq \emptyset$) để cung cấp cho tất cả các khách hàng sao cho tổng chi phí xây dựng và chi phí vận chuyển là nhỏ nhất.

$$F(S) = \sum_{i \in S} C_i + \sum_{j \in J} \min_{i \in S} g_{ij} \rightarrow \min_{S \subseteq I}$$

1.4. Bài toán vị trí cơ sở có hạn chế khả năng

Bài toán vị trí cơ sở có hạn chế khả năng được phát biểu như sau:

- $I = \{1, 2 \dots n\}$ các khách hàng
 - $J = \{1, 2 \dots m\}$ các cơ sở tiềm năng
 - F_j chi phí xây dựng cơ sở j ($j \in J$)
 - H_i nhu cầu của mỗi khách hàng i ($i \in I$)
 - S_j khả năng cung cấp của cơ sở j ($j \in J$)
 - C_{ij} chi phí di chuyển (vận chuyển) từ khách hàng i đến cơ sở j .
 - a là số lượng ít nhất các cơ sở có thể được chọn để mở
 - b là số lượng nhiều nhất các cơ sở có thể được chọn để mở
- $$x_{ij} = \begin{cases} 1 & \text{nếu khách hàng } i \text{ chọn cơ sở } j, \text{ và} \\ 0 & \text{nếu ngược lại;} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{nếu cơ sở } j \text{ được mở} \\ 0 & \text{nếu ngược lại} \end{cases}$$

Hàm mục tiêu có thể được biểu diễn như sau:

$$\text{Min} \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (1)$$

sao cho

$$\sum_{j \in J} x_{ij} = 1, \quad \forall i \in I,$$

$$\sum_{i \in I} h_i x_{ij} \leq S_j y_j, \quad \forall j \in J,$$

$$a \leq \sum_{i \in I} y_j \leq b \quad \forall j \in J,$$

$$x_{ij} \in \{0,1\}, \forall i \in I, \forall j \in J,$$

$$y_j \in \{0,1\}, \forall j \in J$$

1.5. Bài toán vị trí cơ sở cạnh tranh

Bài toán $(r|p)$ -trung tâm có thể phát biểu dưới dạng bài toán tìm minimax trong bài toán quy hoạch hai mức. Ký hiệu:

$$x_i = \begin{cases} 1 & \text{nếu Trước mở cơ sở } i \\ 0 & \text{nếu ngược lại} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{nếu Sau mở cơ sở } i \\ 0 & \text{nếu ngược lại} \end{cases}$$

$$z_j = \begin{cases} 1 & \text{nếu khách hàng } j \text{ được Trước phục vụ} \\ 0 & \text{nếu khách hàng } j \text{ được Sau phục vụ} \end{cases}$$

Khi đó

$$X = \{i \in I \mid x_i = 1\}, Y = \{i \in I \mid y_i = 1\}.$$

Với mỗi khách hàng j , chúng ta định nghĩa tập cơ sở $I_j(X)$ cho phép Sau “thu hút” khách hàng j .

$$I_j(X) = \left\{ i \in I \mid d_{ij} < \min_{l \in I} \{d_{lj} \mid x_l = 1\} \right\}$$

Với các ký hiệu như trên, bài toán $(r|p)$ -trung tâm được biểu diễn như sau:

$$\max_x \sum_{j \in J} w_j z_j^*(X),$$

$$\sum_{i \in I} x_i = p,$$

$$x_i \in \{0,1\}, i \in I,$$

Với $z_j^*(X), y_i^*(X)$ là phương án tối ưu thì bài toán Sau sẽ là:

$$\begin{aligned} \max_{y,z} \quad & \sum_{j \in J} w_j (1 - z_j), \\ \sum_{i \in I} \quad & y_i = r, \\ 1 - z_j \leq \quad & \sum_{i \in I_j(X)} y_i, \quad i \in I, \\ y_i, z_j \in \quad & \{0, 1\}, \quad i \in I, j \in J. \end{aligned}$$

Lưu ý rằng, hàm mục tiêu $W^*(X) = \sum_{j \in J} w_j z_j^*(X)$ là tổng lợi nhuận của Trước khi nó mở đúng p cơ sở, giá trị này phụ thuộc vào lời giải tối ưu của Sau.

1.6. Bài toán bố trí vị trí xây dựng

1.6.1. Hàm mục tiêu thứ nhất

Hàm mục tiêu thứ nhất được chia nhỏ ra thành ba trường hợp ứng với ba loại điều kiện khác nhau trong thực tế, ta kí hiệu các trường hợp này lần lượt là TH1, TH2 và TH3.

$$\text{Min} F = \sum_{i=1}^n \sum_{x=1}^n \sum_{j=1}^n \delta_{xi} f_{ij} d_{ij}$$

sao cho

$$\sum_{x=1}^n \delta_{xi} = 1 \quad \{i = 1, 2, \dots, n\}$$

Với n là số lượng cơ sở, δ_{xi} là ma trận hoán vị. Hệ số f_{ij} là tần suất di chuyển được thực hiện bởi nhân viên giữa cơ sở i và cơ sở j . Từ đó, có thể thấy f_{ij} bằng f_{ji} . Tần số này biểu thị bằng số lượng di chuyển trong một khoảng thời gian nhất định, thông thường là một ngày. Hệ số d_{ij} là khoảng cách giữa vị trí i và vị trí j . Do đó, hàm mục tiêu F là tổng khoảng cách di chuyển được thực hiện bởi nhân viên.

1.6.2. Hàm mục tiêu thứ hai

Hàm mục tiêu thứ hai được đưa ra với hai loại trường hợp ứng với các bộ dữ liệu khác nhau, chúng ta sẽ kí hiệu các trường hợp đó lần lượt là TH4 và TH5.

Mục tiêu bố trí cơ sở ở phần này là chi phí khoảng cách giữa các đối tượng liền kề. Bài toán được mô phỏng bởi Yeh và Mawdesley cùng các cộng sự. Bài toán CSLP còn có thể được xây dựng như sau:

$$\text{Min } F = \sum_{x=1}^n \sum_{i=1}^n \delta_{xi} C_{xi} + \sum_{x=1}^n \sum_{i=1}^n \sum_{y=1}^n \sum_{j=1}^n \delta_{xi} \delta_{yj} A_{ij} D_{xy}$$

sau cho

$$\begin{aligned} \delta_{yj} &= 0 \text{ if } \delta_{xi} = 1 \text{ and } y \neq x \\ \delta_{xj} &= 1 \text{ if } \delta_{xi} = 1 \text{ and } i \neq j \end{aligned}$$

Với hàm mục tiêu F được tính bằng tổng các tích của khoảng cách và chi phí. δ_{xi} là giá trị của ma trận hoán vị (=1 nếu cơ sở x được đặt vào vị trí i), C_{xi} là chi phí xây dựng cơ sở x ở vị trí i. $A_{ij} = 1$ nếu vị trí i là hàng xóm của vị trí j, D_{xy} là chi phí tương tác giữa cơ sở x với cơ sở y.

1.7. Bài toán bố trí cơ sở theo hàng

Bài toán bố trí cơ sở theo hàng (SRFL) là một bài toán đặc biệt của bài toán vị trí cơ sở, giả sử có cơ sở được sắp xếp trên một hàng dài, mỗi cơ sở i có độ dài L_i ($L_i > 0$). Một ma trận kích thước $n \times n$ được ký hiệu là $C = (C_{ij})$ với C_{ij} là chi phí vận chuyển từ cơ sở i đến cơ sở j. Một phương án Π được gọi là một cách sắp xếp các cơ sở thành hàng theo thứ tự $\Pi = \{\pi_1, \dots, \pi_n\}$ trong đó Π_i là vị trí i đặt cơ sở Π_i . Khi đó, chi phí được tính như sau:

$$\text{Minimize } \sum_i^{n-1} \sum_j^n c_{\pi_i \pi_j} d_{\pi_i \pi_j}$$

với

$$d_{\pi_i \pi_j} = l_{\pi_i} / 2 + \sum_{i < k < j} l_{\pi_k} + l_{\pi_j} / 2$$

là khoảng cách giữa trung tâm cơ sở đến trong phương án sắp xếp. Mục tiêu của bài toán là tìm ra phương án sắp xếp sao cho hàm mục tiêu thu được là nhỏ nhất.

1.8. Kết chương

Trên đây chúng ta đã tìm hiểu các kiến thức tổng quan về độ phức tạp thuật toán, lớp các bài toán P, NP, NP-khó. Việc đánh giá một bài toán thuộc lớp nào là công đoạn đầu tiên và vô cùng quan trọng, nó góp phần giúp người lập trình định hướng và lựa chọn các thuật toán giải phù hợp cho bài toán.

Các bài toán điển hình trong lớp các bài toán vị trí cơ sở cùng các thuật toán đã được đề xuất giải các bài toán đó. Chi tiết việc đánh giá, so sánh hiệu năng của các thuật toán sẽ được trình bày trong chương 3.

CHƯƠNG 2.

THUẬT TOÁN TỐI ƯU ĐÀN KIẾN

Tối ưu đàn kiến (ACO) là một phương pháp metaheuristic dựa trên ý tưởng mô phỏng cách tìm đường đi từ tổ tới nguồn thức ăn của các con kiến tự nhiên. Đến nay nó được cải tiến đa dạng và có nhiều ứng dụng. Trước khi giới thiệu phương pháp ACO, cần giới thiệu phương thức trao đổi thông tin gián tiếp của các con kiến thực và mô hình kiến nhân tạo.

2.1. Từ kiến nhân tạo đến kiến thực

2.1.1. Kiến thực

2.1.2. Kiến nhân tạo

Nhờ các con kiến nhân tạo này (về sau cũng gọi đơn giản là kiến) Dorigo đã xây dựng hệ kiến (AS) giải bài toán người chèo hàng, hiệu quả của nó so với các phương pháp mô phỏng tự nhiên khác như SA và GA đã được kiểm chứng bằng thực nghiệm và được phát triển và ứng dụng phong phú với tên gọi chung là phương pháp ACO.

2.2. Phương pháp ACO cho bài toán TỰTH tổng quát

2.2.1. Đồ thị cấu trúc

Mỗi bài toán TỰTH được xem như một bài toán tìm kiếm sâu độ dài không quá h trên đồ thị đầy có các đỉnh có nhãn trong tập C . Để tìm các lời giải chấp nhận được, ta xây dựng đồ thị đầy với tập đỉnh V mà mỗi đỉnh của nó tương ứng với mỗi thành phần của C . Các lời giải chấp nhận được sẽ là các xâu được tìm theo thủ tục tuần tự hay là bước ngẫu nhiên.

2.2.2. Mô tả thuật toán ACO tổng quát.

Procedure of ACO algorithms;

Begin

Initialize; // khởi tạo ma trận mùi, khởi tạo m con kiến

Repeat

Construct solutions; // mỗi con kiến xây dựng lời giải

Update trail; // cập nhật mùi

Until End condition; //điều kiện kết thúc

End;

2.3. Phương pháp ACO giải bài toán TSP

2.3.1. Bài toán TSP và đồ thị cấu trúc

Bài toán TSP xuất phát từ thực tế, một người giới thiệu sản phẩm muốn tìm một hành trình ngắn nhất xuất phát từ thành phố của mình đi qua tất cả các thành phố mà khách hàng cần giới thiệu sản phẩm và sau đó trở về thành phố xuất phát với điều kiện các thành phố của khách hàng chỉ đi qua đúng một lần.

Khi đó đồ thị cấu trúc của bài toán là đồ thị đầy $G = (N, E, H, \tau)$. Nếu $x_k = \langle u_0, \dots, u_k \rangle$ là đường đi mở rộng được, tức là các đỉnh u_i đều khác nhau và $k < n$ thì $J(x_k) = N_{u_k}$ là các đỉnh mà đường x_k chưa đi đến. Các thuật toán ACO cho bài toán TSP đã có đều thực hiện trên đồ thị cấu trúc này.

2.3.2. Các thuật toán ACO cho bài toán TSP

Quá trình mỗi con kiến xây dựng lời giải theo thủ tục bước ngẫu nhiên như sau:

- Lựa chọn thành phố xuất phát cho kiến (có thể theo một số tiêu chí nào đó).
- Thực hiện lặp thủ tục mở rộng bằng cách lặp đi lặp lại việc thêm một thành phố mà kiến chưa đi qua (xem hình 3.5) cho

đến khi tất cả các thành phố đều được thăm: tính xác suất lựa chọn đỉnh mới nhờ giá trị thông tin mùi và thông tin heuristic rồi chọn ngẫu nhiên đỉnh mới thêm vào theo phân bố ngẫu nhiên này.

- Quay trở lại thành phố xuất phát.

Procedure Thuật toán ACOTSP

Begin

Initialize: Khởi tạo vết mùi

while Khi điều kiện dừng chưa thỏa mãn **do**

for $i=1$ to n_ants *do*

Xây dựng lời giải;

Cải tiến lời giải do kiến xây dựng bằng tìm kiếm cục

bộ;

Cập nhật lời giải tốt

Cập nhật mùi

end while

End

2.4. Một số vấn đề khác khi áp dụng ACO

2.4.1. Đặc tính hội tụ

2.4.2. Thực hiện song song

Đặc tính tự nhiên của các thuật toán ACO giúp cho chúng có thể thực hiện song song theo dữ liệu hoặc theo quần thể. Trên thực tế, có nhiều mô hình song song được sử dụng cho các thuật toán dựa trên quần thể có thể dễ dàng tương thích với ACO. Hầu hết các chiến lược song song trực tiếp có thể chia thành chiến lược mịn (fine-grained) và thô (coarse-grained). Đặc tính của fine-grained là rất ít bộ xử lý được chỉ định để xử lý đơn và việc trao đổi thông tin giữa các

bộ xử lý thường xuyên. Ngược lại, với coarse-grained thì một lượng lớn, thậm chí tất cả bộ xử lý được chỉ định để xử lý đơn và thông tin trao đổi là rất ít.

2.4.3. ACO kết hợp với tìm kiếm cục bộ

Mô hình ACO có thể bao gồm cả tìm kiếm cục bộ. Sau khi kiến xây dựng xong lời giải, có thể áp dụng tìm kiếm cục bộ để nhận được lời giải tối ưu địa phương. Việc cập nhật mùi được thực hiện trên các cạnh thuộc lời giải tối ưu địa phương. Việc kết hợp xây dựng lời giải với tìm kiếm cục bộ là một cách tiếp cận đầu hứa hẹn. Trên thực tế, bởi vì cách xây dựng lời giải của ACO sử dụng lân cận khác với tìm kiếm cục bộ. Thực nghiệm cho thấy khả năng kết hợp tìm kiếm cục bộ cải tiến được lời giải là khá cao.

2.5. Kết luận chương

Phương pháp ACO là một phương pháp metaheuristic đang được sử dụng rộng rãi để giải các bài toán TỰTH khó và hiệu quả nổi trội của chúng đã được chứng tỏ bằng thực nghiệm. Phương pháp này mô phỏng cách tìm đường đi của kiến tự nhiên. Trong đó, lời giải chấp nhận được của bài toán được các con kiến nhân tạo xây dựng nhờ thủ tục bước ngẫu nhiên trên đồ thị cấu trúc. Việc tìm kiếm đỉnh mới của đường đi dựa trên sự kết hợp thông tin heuristic và thông tin học tăng cường biểu thị bởi vết mùi.

Khi áp dụng phương pháp này, có ba yếu tố quan trọng:

- 1) Xây dựng đồ thị cấu trúc
- 2) Xác định thông tin heuristic
- 3) Chọn quy tắc cập nhật mùi

Trong đó hai yếu tố đầu phụ thuộc vào từng bài toán cụ thể, còn yếu tố thứ ba có nhiều đề xuất và nghiên cứu cải tiến nhưng vẫn còn có thể nghiên cứu sâu hơn để có các cải tiến hiệu quả.

CHƯƠNG 3.

CÀI ĐẶT THỬ NGHIỆM

3.1. Thuật toán r|p-ACO giải bài toán r|p trung tâm

3.1.1. Lược đồ tổng quát

Bước 1. Khởi tạo ma trận vết mùi cho *Trước* và $n_{ant}T$, $n_{ant}S$, N_{total} ;

Bước 2. Thực hiện lặp:

2.1. ACO- *Trước*; // Với mỗi kiến k tìm lời giải X_k cho *Trước*

2.2. ACO- *Sau*; // Tìm lời giải Y_k cho *Sau* với lời giải của *Trước*

là X_k ;

2.3. Chọn ra X^* là lời giải tốt nhất trong số các lời giải X_k ;

2.4. LS(X^*); // Tìm kiếm địa phương cho lời giải X^* tốt nhất

2.5. Cập nhật X^* và trở lại 2.1 nếu chưa kết thúc.

Bước 3. Trích lời giải cho trước và sau.

3.1.2. Thủ tục ACO

Procedure ACO- *Trước*

Begin

for mỗi kiến $k \in n_{ant}T$ do

 Xây dựng lời giải X_k cho kiến thứ k ;

Return X ;

End;

Procedure ACO- *Sau*(X)

Begin

 Khởi tạo ma trận vết mùi cho *Sau* và $n_{ant}S$ kiến;

repeat

 for mỗi kiến $k \in n_{ant}S$ do

 Xây dựng lời giải Y_k cho kiến thứ k ;

```

    Chọn ra  $Y^*$  là lời giải tốt nhất trong các  $Y_k$ ;
    LS( $Y^*$ ); //Tìm kiếm địa phương cho phương án  $Y^*$ 
    Cập nhật  $Y^*$ ;
    until gặp điều kiện dừng
    return  $Y^*$ ;
End;

```

Procedure LS(X)

Begin

$U = I - X$;

for mỗi $x \in X$ do

 for mỗi $u \in U$ do

 Thay thế x bằng u ;

 If (lợi nhuận thu được là tốt hơn) then Cập nhật X ;

Return X ;

End;

3.1.3. Kết quả thử nghiệm

Thuật toán r|p-ACO cho kết quả chính xác hơn thuật toán IM do Alekseeva và các cộng sự đề xuất và VNS do Davydov cùng các cộng sự đề xuất nhưng vẫn chậm hơn so với thuật toán STS do Davydov cùng các cộng sự đề xuất.

3.2. So sánh các thuật toán giải bài toán CSLP

Nhìn bảng 4.4. ta có thể thấy trong TH1 và TH3 thì kết quả của lopt-aiNet và ACO-PA là tương đương nhau (bằng 12150 ở TH1 và 12606 ở TH2). Tuy nhiên, trong TH2 thuật toán ACO-PA tỏ ra kém hiệu quả hơn so với thuật toán lopt-aiNet cụ thể là thuật toán lopt-aiNet cho kết quả tốt hơn 0.25% so với thuật toán ACO-PA. Về mặt thời gian, để tìm ra lời giải cho mỗi lần chạy thì thuật toán ACO-

PA mất 1.15 giây với máy tính Intel Core 2 Duo 2.66GHz và 4GB RAM. Trong khi đó, lopt-aiNet chỉ mất 0.15 giây trên máy tính cấu có cấu hình thấp hơn là CPU Pentium P6200 2.13GHz, RAM 2GB.

Bảng 4.5. thể hiện hiệu suất của các thuật toán GA, PSO, ACO, opt-aiNet và lopt-aiNet trong 5 lần chạy. Thuật toán tốt nhất cho kết quả tối ưu (= 90) là thuật toán lopt-aiNet do Quang cùng các cộng sự đề xuất, sau đó đến thuật toán ACO với kết quả trung bình là 90.8 trong TH1 và 91.0 với TH2. Về mặt thời gian, thuật toán opt-aiNet và thuật toán lopt-aiNet chạy trong thời gian ngắn nhất chỉ mất 0.19 – 0.2 giây trong TH4 và 0.19 – 0.21 giây trong TH5, thuật toán ACO mất 0.33 giây trong TH4 và 0.35 giây trong TH5, thuật toán GA sử dụng 0.54 giây trong cả hai loại TH4 và TH5, thuật toán PSO được đánh giá là thuật toán chậm nhất với 1.96 giây trong TH4 và 1.87 giây trong TH5.

3.3. Áp dụng thuật toán ACO-SRFL giải bài toán SRFL

Nhìn bảng 4.7 ta có thể thấy, thuật toán ACO cho kết quả chính xác ở các bộ dữ liệu LW5, S8H, S10, LW11, H20 và cho kết quả tốt hơn [29] [54]. Khi so sánh tốc độ thực hiện của thuật toán ACO với thời gian chạy của các thuật toán khác thì chúng ta có thể thấy rằng thuật toán ACO chạy nhanh hơn tất cả các thuật toán khác. Tuy nhiên, trong các thuật toán đã công bố chỉ có thuật toán đàn dơi [54] được thực hiện trên máy tính tốt hơn của tác giả. Do vậy, để đảm bảo tính khách quan, bảng 4.8 chỉ so sánh thời gian chạy thuật toán ACO với thuật toán đàn dơi [54].

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận

Phương pháp tối ưu đàn kiến là phương pháp tương đối mới mẻ và tỏ ra đặc biệt hiệu quả, điều này đã được chứng minh thông qua thực nghiệm. Phương pháp tối ưu đàn kiến luôn được quan tâm, phát triển kể từ khi giới thiệu cho đến nay thể hiện qua sự phong phú, đa dạng của các thuật toán. Các thuật toán trực tiếp đưa ra hướng tiếp cận mới giải các bài toán tối ưu tổ hợp, qua đó có nhiều ứng dụng trong thực tiễn trên các lĩnh vực như: sản xuất, truyền thông, sinh học, hoạt động xã hội...

Bài toán vị trí cơ sở là một bài toán lớn bao hàm nhiều bài toán con có ứng dụng thực tế cao, nó giúp chúng ta lựa chọn các vị trí cơ sở để đặt các trạm dịch vụ một cách tối ưu nhất.

Đối với bài toán rlp-trung tâm, bài toán CSLP và bài toán SRFL, chúng tôi đã đề xuất thuật toán dựa trên thuật toán ACO, đồng thời có so sánh đánh giá thuật toán với một số thuật toán khác để thấy được ưu, nhược điểm của thuật toán.

Hướng phát triển

Cải thiện tốc độ thực hiện của thuật toán thông qua cải tiến tìm kiếm địa phương và/hoặc kết hợp với phần mềm CPLEX.

Tiếp cận với các bài toán tương tự về mạng, khi khách hàng nằm ở các đỉnh của đồ thị còn các cơ sở có thể mở tại các điểm tùy ý trên các cạnh của nó.

DANH MỤC CÁC CÔNG TRÌNH ĐÃ CÔNG BỐ CỦA TÁC GIẢ

- [1]. Vũ Đức Quang, Hoàng Xuân Huân, Đỗ Thanh Mai, (2016), “Một thuật toán hiệu quả dựa trên giải thuật tối ưu đàn kiến giải bài toán r|p trung tâm”, trong Kỷ yếu Hội nghị Quốc gia lần thứ 9 về Nghiên cứu cơ bản và ứng dụng Công Nghệ thông tin (FAIR) tại Đại học Cần Thơ. tr 488 – 494.
- [2]. Duc Quang Vu, Van Truong Nguyen, Xuan Huan Hoang, (2016), “An Improved Artificial Immune Network For Solving Construction Site Layout Optimization”, in Proceeding of the 12th IEEE-RIVF International Conference on Computing and Communication Technologies, pp 37 – 42.