ỨNG DỤNG THUẬT TOÁN NHÁNH CẬN ĐỂ GIẢI MỘT SỐ BÀI TOÁN TỐI ƯU LIÊN QUAN ĐẾN CHU TRÌNH HAMILTON DỰA TRÊN BÀI TOÁN TSP

Đỗ Như An[a*]

[a]*Khoa Công nghệ Thông tin, Trường Đại học Nha Trang, Khánh Hoà, Việt Nam*

Lịch sử bài báo
Nhận ngày 05 tháng 01 năm 2017 | Chỉnh sửa ngày 08 tháng 04 năm 2017
Chấp nhận đăng ngày 10 tháng 05 năm 2017

Tóm tắt

Bài toán người du lịch (Traveling Salesman Problem, viết tắt TSP) là một trong những bài toán tối ưu tổ hợp nổi bật thuộc lớp NP-khó. Thuật toán tốt nhất hiện nay để giải TSP là thuật toán nhánh-cận có độ phức tạp thời gian tính toán dạng hàm mũ. Bài báo này trình bày cách vận dụng thuật toán nhánh-cận để giải một số bài toán tối ưu liên quan đến chu trình Hamilton dựa trên bài toán TSP tương ứng.

Từ khóa: Bài toán người du lịch; Bài toán tối ưu tổ hợp; Chu trình Hamilton; Đường Hamilton; NP-đầy đủ; NP-khó; Thuật toán nhánh-cận.

---

1. GIỚI THIỆU

Phần này trình bày một số khái niệm và kiến thức cơ bản của lý thuyết đồ thị liên quan đến bài toán người du lịch và thuật toán nhánh-cận giải TSP. Nội dung của phần này chủ yếu được tham khảo trong tài liệu *Combinatorial Optimization: Theory and Algorithms* của Korter và Vygen (2002), và tài liệu *The Traveling Salesman Problem* của Lawler và Lenstra (1985).

Không giảm tính tổng quát, trong phần này chúng tôi không mô tả chi tiết quá trình tính toán của thuật toán nhánh-cận cho TSP, kết quả của thuật toán được xác định dễ dàng và chính xác thông qua các ví dụ minh họa đơn giản. Một số kết quả cơ bản của lý thuyết NP-đầy đủ được tham khảo tại tài liệu *Theory of computational complexity* của Ding (2001).

---

* Tác giả liên hệ: Email: andn@ntu.edu.vn

1.1. Đường Hamilton và chu trình Hamilton

Xét đồ thị có hướng $G = (V, E)$, trong đó V là tập các đỉnh, E là tập các cung. Một hành trình (*walk*) trong G là một dãy $W = v_0 e_1 v_1 e_2 ..., e_l v_l$, trong đó $v_0, v_1, ..., v_l$ là các đỉnh của V và $e_i = v_{i-1} v_i$, $1 \le i \le l$, là các cung của E. Để đơn giản, có thể biểu diễn một hành trình thông qua các đỉnh của nó. Chẳng hạn, với hành trình $W = v_0 e_1 v_1 e_2 ..., e_l v_l$ có thể viết $W = (v_0, v_1, ..., v_l)$ và gọi v_0 và v_l là đỉnh đầu và đỉnh cuối của W. Hành trình $W = (v_0, v_1, ..., v_l)$ được gọi là mở nếu $v_0 \ne v_l$ và được gọi là đóng (khép kín) nếu $v_0 = v_l$. Một hành trình $W = (v_0, v_1, ..., v_l)$ được gọi là một đường (*path*), hay đường đi, nếu các đỉnh $v_0, v_1, ..., v_l$ của nó là khác nhau. Một hành trình đóng $\mathrm{C} = (v_0, v_1, ..., v_l, v_0)$ mà trong đó $W = (v_0, v_1, ..., v_l)$ là một đường đi được gọi là chu trình (*cycle*). Một đường đi chứa tất cả các đỉnh của đồ thị được gọi là đường Hamilton (*Hamiltonian path*). Một chu trình chứa tất cả các đỉnh của đồ thị được gọi là chu trình Hamilton (*Hamiltonian cycle*).

Để đơn giản trong trình bày, chúng ta giả sử tập đỉnh của đồ thị có hướng $G = (V, E)$ là $V = \{1, 2, ..., n\}$. Với mỗi cặp đỉnh trong V, đỉnh i được gọi là kề với đỉnh j nếu (i, j) là một cung hay ký hiệu $(i, j) \in E$. Một đồ thị được gọi là đầy đủ nếu mọi cặp đỉnh của nó là kề nhau. Với một đồ thị có hướng đầy đủ n đỉnh luôn chứa $n!$ đường Hamilton hoặc chu trình Hamilton khác nhau (mỗi đường Hamilton hoặc chu trình Hamilton là một hoán vị của $V = \{1, 2, ..., n\}$). Tuy nhiên, với đồ thị G bất kỳ, bài toán xác định xem G có chứa một đường Hamilton hoặc một chu trình Hamilton hay không là bài toán NP-đầy đủ. Đồ thị có hướng có trọng số là đồ thị có hướng mà mỗi cung (i, j) được gán một số c_{ij} được gọi là trọng số của cung tương ứng ($i, j = \overline{1, n}$, $i \ne j$) và $C = (c_{ij})$ được gọi là ma trận trọng số của G.

1.2. Bài toán người du lịch và thuật toán nhánh-cận

Bài toán tối ưu liên quan đến chu trình Hamilton là bài toán người du lịch (TSP) được phát biểu như sau: Người du lịch muốn viếng thăm *n* thành phố và trở về lại nơi đã xuất phát. Biết chi phí đi lại giữa các thành phố, hãy tìm cách đi cho người du lịch sao cho có thể đến thăm mỗi thành phố đúng một lần và tổng chi phí đi lại là bé nhất.

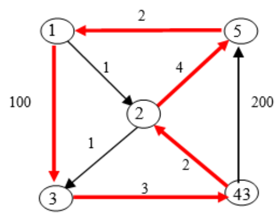
TSP được biểu diễn bằng một đồ thị có hướng $G=(V,E)$, trong đó tập hợp của các đỉnh $V=\{1,2,...,n\}$ biểu thị các thành phố, E là tập hợp của các cung là con đường nối các thành phố. Trọng số c_{ij} biểu thị chi phí đi lại từ thành phố i đến thành phố j với $(i, j=\overline{1,n})$. Cần tìm một chu trình Hamilton trên G có tổng trọng số nhỏ nhất. TSP là bài toán thuộc lớp NP-khó.

Bài toán TSP tìm một chu trình Hamilton có tổng trọng số nhỏ nhất, hay còn được gọi là hành trình du lịch tối ưu, được đưa ra lần đầu tiên bởi Whitney năm 1934. Sau đó, ba thành viên của Trường Đại học Rand là Dantzig, Fulkerson, và Johnson (1954), trong cuốn *Solution of lagre-scale traveling salesman problem* đã công bố lời giải cho bài toán tìm hành trình tối ưu của 49 thành phố gồm Washington D.C. và thủ phủ của 48 bang khác. Thuật toán của họ là sự kết hợp giữa quy hoạch tuyến tính (*linear programming*) và lý thuyết đồ thị (*graph theory*) mà ngày nay chúng ta sử dụng như là công cụ chuẩn trong quy hoạch nguyên tuyến tính, đó chính là thuật toán nhánh-cận.

Thuật toán nhánh-cận xác định phương án tối ưu cho TSP được mô tả tóm tắt như sau: Thuật toán nhánh-cận thực chất là thuật toán quay lui (*backtracking algorithm*). Thuật toán từng bước xây dựng các phương án cho bài toán với tất cả các khả năng có thể xảy ra, trong đó mỗi nhánh của phương án đang được xây dựng bởi thuật toán sẽ chấm dứt khi biết được tổng trọng số của phương án này vượt quá *giá trị cận dưới* (giá trị hàm mục tiêu của phương án đã được xác định trước đó tính đến thời điểm hiện tại là tốt nhất). Ưu điểm của thuật toán nhánh-cận là hạn chế được nhiều tính toán trong quá trình xây dựng phương án và được xem là thuật toán tốt nhất hiện nay cho TSP. Tuy nhiên, độ phức tạp tính toán của thuật toán trong trường hợp tổng quát vẫn là $O(n!)$.

Hình 1 là ví dụ minh họa TSP bằng đồ thị có hướng G gồm $n=5$ đỉnh và ma trận chi phí tương ứng $M=(c_{ij})$. Lưu ý, trong ma trận này, gán $c_{ij}=0$ nếu $i=j$,

$c_{ij} = +\infty$ nếu đỉnh i không kề với j. Trong cài đặt thuật toán cho bài toán, có thể thay giá trị $+\infty$ bởi một số dương đủ lớn Θ, chẳng hạn $\Theta = n.\max\{c_{ij}, 1 \le i, j \le n\}$.



$$M = \begin{bmatrix} 0 & 1 & 100 & +\infty & +\infty \\ +\infty & 0 & 1 & 0 & 4 \\ +\infty & +\infty & 0 & 3 & +\infty \\ +\infty & 2 & +\infty & 0 & 200 \\ 2 & +\infty & +\infty & +\infty & 0 \end{bmatrix}$$

Hình 1. Đồ thị của TSP và ma trận chi phí tương ứng

Một phương án tối ưu của G được xác định bởi thuật toán nhánh-cận là chu trình $C_{\min}$ = (1, 3, 4, 2, 5, 1) gồm các cung được tô đậm trên đồ thị của Hình 1 và giá trị tối ưu tương ứng là $w(C_{\min}) = c_{13} + c_{34} + c_{42} + c_{25} + c_{51} = 100 + 3 + 2 + 4 + 2 = 111$.

2. KẾT QUẢ

Phần này trình bày phương pháp giải một số bài toán tối ưu tổ hợp liên quan đến đường và chu trình Hamilton bằng cách đưa bài toán cần giải về TSP tương ứng, sau đó xác định lời giải của bài toán đã cho từ lời giải của TSP tương ứng được xác định bởi thuật toán nhánh-cận.

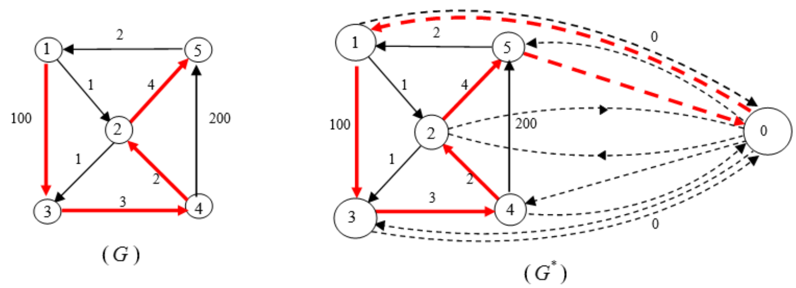
Bài toán 1: Giả sử G là đồ thị có hướng có trọng số của các cung. Tìm một đường Hamilton trên G sao cho tổng trọng số là nhỏ nhất.

Đưa Bài toán 1 về bài toán TSP tương ứng như sau: Xây dựng đồ thị có hướng G^* từ đồ thị G bằng cách thêm một đỉnh mới 0 và các cung có trọng số tương ứng là $c_{0j} = c_{j0} = 0$, $j = 1,2,...,n$. Khi đó, một chu trình Hamilton tối ưu trong G^* sẽ là một đường Hamilton tối ưu tương ứng trên G sau khi loại bỏ đỉnh 0.

Thật vậy, giả sử $C_{\min} = (0, v_1, v_2, ..., v_n, 0)$ là một chu trình Hamilton có tổng trọng số bé nhất trên G^* thu được bởi thuật toán nhánh-cận. Khi đó, theo cách xây dựng G^*, $P_{\min} = (v_1, v_2, ..., v_n)$ là một đường Hamilton trên G thu được từ $C_{\min}$ bằng cách bỏ đi hai cung (*0*, v_1) và (v_n, *0*). Hơn nữa, $P_{\min}$ có tổng trọng số là bé nhất $w(P_{\min}) = w(C_{\min}) - (c_{0v_1} + c_{v_n0}) = w(C_{\min})$.

Trường hợp G^* không tồn tại chu trình Hamilton, khi đó G cũng không có đường Hamilton. Hơn nữa, dễ dàng khẳng định được việc xây dựng G^* từ G có thời gian tính toán là $O(n)$.

Ví dụ 1: Hình 2 là đồ thị G của Bài toán 1 và đồ thị G^* của TSP tương ứng.



Hình 2. Đồ thị G của bài toán 1 và G^* của TSP tương ứng

Giải TSP trên G^* bằng thuật toán nhánh-cận, chu trình Hamilton tối ưu thu được là $C_{\min}$ = (1, 3, 4, 2, 5, 1), gồm các cung được tô đậm trên G^*, với tổng trọng số $w(C_{\min})$ = 100+3+2+4+0+0= 109. Đường Hamilton tối ưu trên G cần tìm là $P_{\min}$ = (1, 3, 4, 2, 5) với các cung được tô đậm trên G, tổng trọng số là $w(P_{\min}) = w(C_{\min}) = 109$.

Nhận xét, có thể giải trực tiếp Bài toán 1 bằng cách áp dụng thuật toán nhánh-cận một cách thích hợp mà không cần chuyển đổi về TSP. Tuy nhiên, việc đưa Bài toán 1 về bài toán TSP nêu ở trên là một trong những kỹ thuật thường được vận dụng trong lý thuyết đồ thị để chứng minh sự tồn tại của một đường Hamilton thông qua một chu trình Hamilton trong đồ thị. Việc làm này có ý nghĩa về mặt lý thuyết của thuật toán nhằm khẳng định thêm rằng Bài toán 1 và TSP là hai vấn đề có độ phức tạp thời gian tính toán tương đương (cùng đa thức hoặc không đa thức) và cùng thuộc lớp NP-khó.

Bài toán 2: Giả sử G là đồ thị có hướng có trọng số của các cung. Tìm đường Hamilton có tổng trọng số nhỏ nhất từ đỉnh xác định s đến đỉnh xác định t trong G.

Đường Hamilton tối ưu cần tìm của Bài toán 2 được hiểu như sau: Xuất phát từ đỉnh *S*, đường lần lượt đi qua tất cả các đỉnh, mỗi đỉnh chính xác một lần, và kết thúc tại đỉnh t của đồ thị G sao cho tổng trọng số là bé nhất.

Đưa Bài toán 2 về TSP như sau: Xây dựng đồ thị có hướng G^* từ đồ thị G bằng cách thay hai đỉnh s và t bởi một đỉnh u và định nghĩa trọng số của các cung liên quan với u của G^* như sau:

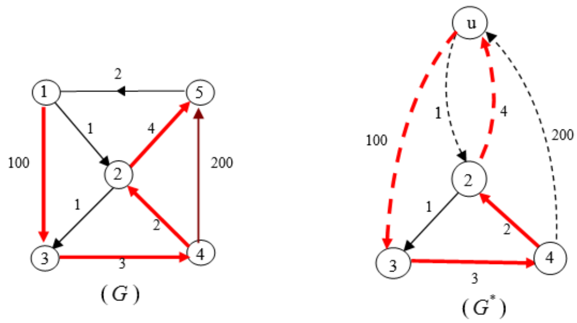
$$c_{ij}^* = \begin{cases} c_{sj} & khi\ \ i = u \\ c_{it} & khi\ \ j = u \end{cases} \tag{1}$$

Dễ thấy, việc xây dựng G^* từ đồ thị G như trên đảm bảo rằng, nếu G^* có chu trình Hamilton thì G cũng có đường Hamilton; Ngược lại, G^* không có chu trình Hamilton thì G cũng không có đường Hamilton một cách tương ứng. Đặc biệt, nếu trong G^* có chu trình Hamilton tối ưu thì đây cũng chính là đường Hamilton tối ưu trên G cần tìm.

Thật vậy, giả sử $C_{\min} = (u, v_1, v_2, ..., v_{n-2}, u)$ là một chu trình Hamilton tối ưu của G^*. Khi đó, theo cách xây dựng G^*, $P_{\min} = (s, v_1, v_2, ..., v_{n-2}, t)$ là đường Hamilton tối ưu của G thu được bằng cách thay đỉnh u bởi hai đỉnh s và t của chu trình $C_{\min}$ trên G^*. Dễ dàng nhận thấy rằng, việc xây dựng G^* từ G có thời gian tính toán là $O(n)$.

Nhận xét, Bài toán 2 là trường hợp riêng của Bài toán 1. Bài toán 2 có tập phương án (tập hợp của tất cả các đường Hamilton từ đỉnh s đến đỉnh t) là tập con của tập phương án của Bài toán 1 (tập hợp của tất cả các đường Hamilton giữa các cặp đỉnh). Do đó, có thể vận dụng thuật toán nhánh-cận một cách phù hợp để tìm đường Hamilton tối ưu từ đỉnh s đến đỉnh t cho trước (cần lưu ý khi áp dụng thuật toán, với một đường Hamilton tối ưu tìm được, thì chu trình Hamilton tương ứng sau khi bổ sung cung (t, s) chưa hẳn là tối ưu). Bài toán 2 cũng thuộc lớp NP-khó.

Ví dụ 2: Hình 3 là đồ thị G của Bài toán 2, trong đó s là đỉnh 1 và t là đỉnh 5 và đồ thị G^* tương ứng của TSP.



Hình 3. Đồ thị G của Bài toán 2 và G^* của TSP tương ứng

Giải TSP trên G^* bằng thuật toán nhánh-cận, chu trình Hamilton tối ưu thu được là $C_{\min} = (u,3,4,2,u)$ (các cung được tô đậm trên G^*) có tổng trọng số là $w(C_{\min}) =$ 100+3+2+4 =109. Khi đó, đường Hamilton tối ưu từ đỉnh 1 đến đỉnh 5 trên G cần tìm là $P_{\min} = (1,3,4,2,5)$ (các cung được tô đậm trên G) với tổng trọng số $w(P_{\min}) = w(C_{\min})$ = 109.

Bài toán 3: Giả sử G là đồ thị có hướng đầy đủ có trọng số của các cung. Tìm một hành trình đóng có tổng trọng số nhỏ nhất trong G sao cho mỗi đỉnh được thăm ít nhất một lần.

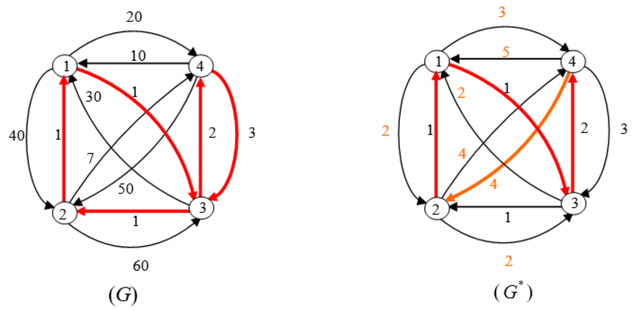
Hành trình nêu ở trên được hiểu đơn giản như sau: Xuất phát từ một đỉnh nào đó của đồ thị G, hành trình lần lượt đi qua tất cả các đỉnh còn lại, mỗi đỉnh đi qua ít nhất một lần và trở về đỉnh ban đầu. Hiển nhiên, hành trình này có thể phải đi qua vài cung nào đó của đồ thị nhiều hơn một lần.

Đưa Bài toán 3 về TSP như sau: Xây dựng đồ thị đầy đủ có hướng G^* từ đồ thị G bằng cách thay trọng số của cung (i, j) bằng độ dài của đường đi ngắn nhất từ i đến j trong đồ thị ban đầu G ($i, j = \overline{1,n}$, $i \neq j$). Lưu ý, từ G với ma trận kề $M = (c_{ij})$ tương ứng, thuật toán Floyd cung cấp kết quả là ma trận $D = (d_{ij})$ cho biết độ dài đường đi ngắn nhất d_{ij} giữa các cặp đỉnh (i, j) và ma trận $T = (t_{ij})$ xác định vết của đường đi ngắn nhất giữa các cặp đỉnh (i, j) trên G. Độ phức tạp thời gian tính toán của thuật toán Floyd là $O(n^3)$ (Korter & Vygen, 2002). Vì vậy, xây dựng G^* từ G có thời gian tính toán đa thức.

Việc giả thiết G là đồ thị có hướng đầy đủ là điều kiện cần và đủ cho G^* luôn tồn tại chu trình Hamilton (một hành trình du lịch). Giả sử $C_{\min}$ là chu trình Hamilton tối ưu của G^*. Ta xác định hành trình $H_{\min}$ tối ưu trên G từ chu trình $C_{\min}$ như sau: Tất cả những cung (i, j) nằm trên $C_{\min}$ mà có trọng số được gán bởi giá trị mới (bằng độ dài đường đi ngắn nhất từ i đến j trên G), tức là những cung (i, j) mà $c^*_{ij} = d_{ij} <$ c_{ij}, sẽ được thay thế bởi vết của đường đi nhắn nhất từ i đến j trên G một cách tương ứng.

Rõ ràng, $H_{\min}$ là đóng bởi $C_{\min}$ là một chu trình. Hơn nữa, $H_{\min}$ có tổng trọng số bé nhất $w(H_{\min}) = w(C_{\min})$ và hành trình này có thể phải đi qua vài cung nào đó của G nhiều hơn một lần. Ngược lại, giả sử H là một hành trình đóng tối ưu trên G. Khi đó, H có thể được biểu diễn bởi hữu hạn các đường đi phân biệt nối tiếp nhau trên G. Do H là tối ưu, các đường đi này là đường đi ngắn nhất tương ứng trên G. Lưu ý, với một đồ thị G cho trước, có thể có nhiều hành trình đóng tối ưu khác nhau có tổng trọng số là nhỏ nhất và bằng nhau. Tiếp theo, trên H, lần lượt thay mỗi đường đi ngắn nhất từ i đến j trên G có độ dài $d_{ij} = c^*_{ij}$ bởi cung (i, j) tương ứng trên G^*. Hành trình thu được không còn đỉnh nào được "thăm" quá một lần, đây chính là một chu trình Hamilton trên G^* và có tổng trọng số là nhỏ nhất.

Ví dụ 3*:* Hình 4 là ví dụ minh họa cho Bài toán 3, một đồ thị G có hướng đầy đủ gồm 4 đỉnh cho trước và đồ thị G^* tương ứng thu được với các cung (i, j) đã được cực tiểu hóa trọng số bởi độ dài của đường đi ngắn nhất từ i đến j trên G.



Hình 4. Đồ thị G của Bài toán 3 và G^* của TSP tương ứng

Giải TSP trên G^* bằng thuật toán nhánh-cận, một chu trình Hamilton tối ưu tìm được là $C_{\min}$ = (1, 3, 4, 2, 1) gồm các cung được tô đậm trên G^* và $w(C_{\min}) = 8$. Hành trình đóng tối ưu tương ứng $H_{\min}$ cần tìm trên G được xác định từ $C_{\min}$ như sau: Do $c^*_{42} = 4 < 50 = c_{42}$, thay cung (4, 2) bởi vết của đường đi ngắn nhất từ đỉnh 4 đến đỉnh 2 là (4, 3, 2). Vì vậy, $H_{\min}$ = (1, 3, 4, 3, 2, 1), bao gồm các cung được tô đậm trên G. Tổng trọng số của hành trình tối ưu là $w(H_{\min}) = w(C_{\min}) = 8$. Để ý, trong hành trình này, đỉnh 3 được "thăm" hai lần.

Mô tả thuật toán giải Bài toán 3:

- *Input:* Ma trận kề của G: $M = (c_{ij})$, $i, j = \overline{1,n}$; n là số đỉnh của đồ thị, chính là kích thước của dữ liệu đầu vào của bài toán.
- *Output:* Hành trình đóng tối ưu $H_{\min}$ của *G*.
- *Phương pháp*:

Bước 1. Từ $M = (c_{ij})$ trên *G*, xác định $D = (d_{ij})$ và $T = (t_{ij})$ ($i, j = \overline{1,n}$) bởi thuật toán Floyd. Thời gian tính toán ở Bước 1 là $O(n^3)$.

Bước 2. Xây dựng đồ thị G^*, nghĩa là xác định ma trận $M^* = (c^*_{ij})$ của G^* từ $M = (c_{ij})$ và $D = (d_{ij})$ như sau: Với mỗi $i, j = \overline{1,n}$:

$$c_{ij}^* = \begin{cases} d_{ij} & khi\ d_{ij} < c_{ij} \\ c_{ij} & khi\ d_{ij} \geq c_{ij} \end{cases} \tag{2}$$

Thời gian tính toán ở Bước 2 là $O(n^2)$.

Bước 3. Giải TSP trên G^* bởi thuật toán nhánh-cận, thời gian tính toán là $O(n!)$. Gọi $C_{\min}$ là chu trình Hamilton tối ưu tìm được.

Bước 4. Xác định hành trình đóng tối ưu $H_{\min}$ trên G tương ứng từ $C_{\min}$ trên G^* như sau: Lần lượt xét từng cung trên $C_{\min}$, nếu một cung (i, j) mà $c_{ij}^* < c_{ij}$ thì thay cung (i, j) bởi vết của đường đi ngắn nhất từ i đến j trên G tương ứng. Thời gian tính toán của Bước 4 là $O(n^2)$.

3. KẾT LUẬN

Hầu hết các bài toán tối ưu liên quan đến đường Hamilton hay chu trình Hamilton trong đồ thị đều là những vấn đề phức tạp. Các bài toán đã nêu ở trên được xếp vào lớp các bài toán NP-khó. Phương pháp tiếp cận để giải quyết các bài toán tối ưu được trình bày như trên là công việc cần thiết, mặc dù không làm giảm độ khó của bài toán đặt ra, song từ đó giúp ta khẳng định được độ khó của vấn đề đặt ra và bổ sung thêm các giải pháp để giải quyết chúng.

Việc giải một bài toán cho trước bằng cách biến đổi về bài toán tương đương để giải là việc làm thường gặp trong toán học. Trong lý thuyết độ phức tạp tính toán, đặc biệt trong lý thuyết NP-đầy đủ, cách làm như vậy được gọi là phép qui dẫn đa thức (*polynomial reductions*) trong lớp bài toán quyết định (*decision problem*). Bài toán quyết định yêu cầu trả lời đúng/sai với mỗi dữ kiện đầu vào (*instance*) cho trước. Một phép qui dẫn F từ bài toán quyết định A về bài toán quyết định B phải đảm bảo tính chất sau đây: Với mỗi dữ kiện đầu vào I_A cho A, xác định dữ kiện đầu vào $I_B = F(I_A)$ cho B tương ứng. Khi đó, câu trả lời chính xác đúng (sai) cho B cũng là câu trả lời đúng (sai) cho A một cách tương ứng. Hơn nữa, việc biến đổi từ A sang B phải được thực hiện chỉ trong thời gian đa thức mà thôi. Với một phép qui dẫn đa thức từ bài toán A về bài toán B được thiết lập, chúng ta có thể đánh giá độ khó (dễ) của A theo B một cách tương ứng (A khó nhiều nhất là bằng B), hoặc khẳng định được rằng A và B là hai vấn đề có cùng độ khó (dễ) sai khác nhau một khoảng thời gian tính toán đa thức.

Một bài toán tối ưu rời rạc có thể phát biểu thành bài toán quyết định bằng cách so sánh giá trị của hàm mục tiêu với một giá trị cụ thể nào đó (được gọi là cận trên hoăc cận dưới của hàm mục tiêu) của bài toán đặt ra.

Lớp NP-đầy đủ là tập con của lớp NP-khó và từ các nhận xét ở trên, vấn đề đặt ra là cần mở rộng khái niệm phép qui dẫn đa thức trong lớp NP-đầy đủ cho các bài toán thuộc NP-khó.

LỜI CẢM ƠN

Tác giả trân trọng và cám ơn PGS.TSKH. Vũ Đình Hòa (Khoa Công nghệ Thông tin, Trường Đại học Sư phạm Hà Nội) đã có nhiều ý kiến góp ý cho bài viết.

TÀI LIỆU THAM KHẢO

Dantzig, G., Fulkerson, D., & Johnson, S. (1954). Solution of large-scale traveling salesman problem. *Journal of the Operations Research Society of America*, *10*(3), 393-410.

Ding, Z. D. (2001). *Theory of computational complexity*. New Jersey, USA: Wiley-Interscience.

Korter, B., & Vygen, J. (2002). *Combinatorial optimization: Theory and algorithms.* Berlin, Gemany: Springer.

Lawler, E. L., & Lenstra, J. K. (1985). *The traveling salesman problem*. New Jersey, USA: John Wiley & Sons.

APPLICATIONS OF BRANCH-BOUND ALGORITHM TO SOLVE SOME OPTIMAL PROBLEMS RELATED TO THE HAMILTONIAN CYCLE BASED ON THE TSP

Do Nhu An[a*]

[a]*The Faculty of Information Technology, Nhatrang University, Khanhhoa, Vietnam*
[*]*Corresponding author: Email: andn@ntu.edu.vn*

Article history
Received: January 05th, 2017 | Received in revised form: April 08th, 2017
Accepted: May 10th, 2017

Abstract

The Traveling Salesman Problem (TSP) is the most prominent of the combinatorial optimization problems that belongs to NP-Hard. The best algorithm for solving TSP is the branch-bound algorithm with exponential-time complexity. This paper presents how to use the branch-bound algorithm to solve some of the combinatorial optimization problems related to the Hamiltonian cycle based on the TSP.

Keywords: Branch-bound algorithm; Combinatorial optimization problem; Hamiltonian cycle; NP-C (Non-deterministic Polynomial Complete; NP-Hard; Traveling Salesman Problem (TSP).