

ỨNG DỤNG MÔ HÌNH QUY HOẠCH NGUYÊN TUYẾN TÍNH TRONG THIẾT KẾ PHẦN MỀM CẮT THÉP THANH

Nguyễn Đình Định¹, Trịnh Thị Phú¹, Lê Đình Nghiệp¹

TÓM TẮT

Bài báo này trình bày một ứng dụng của mô hình quy hoạch nguyên tuyến tính cho bài toán cắt thép thanh trong công trình xây dựng với mục đích giảm hao phí thép trong các công trình xây dựng. Trong quá trình nghiên cứu đã chỉ ra được hai bài toán quy hoạch nguyên tuyến tính ứng với các yêu cầu cắt thép không nối và có nối.

Từ khoá: Quy hoạch nguyên tuyến tính, cắt vật liệu, cắt thép.

1. ĐẶT VẤN ĐỀ

Trong các công trình xây dựng, những vật liệu như: thép, ống nước, khung nhôm, dây điện... chiếm một phần không nhỏ trong tổng chi phí của công trình. Vấn đề về giảm hao phí do quá trình cắt vật liệu trong công trình xây dựng luôn được các nhà thi công quan tâm hàng đầu, bởi phần hao phí này ảnh hưởng không nhỏ đến lợi nhuận của họ đối với mỗi công trình.

Với những công trình nhỏ có thể thực hiện một cách thủ công để tìm ra phương án cắt tối ưu nhất cho mỗi loại vật liệu. Tuy nhiên với những công trình lớn thì giải quyết vấn đề này là không đơn giản. Bài toán có thể được mô hình hoá về dạng mô hình bài toán Quy hoạch nguyên tuyến tính, nhưng độ phức tạp tính toán của bài toán này là rất lớn, nó thuộc lớp bài toán NP - Hard nên với các thuật toán đã có hiện nay vẫn chưa thể thực hiện được trên máy tính. Trong bài báo này sẽ tập trung trình bày những kết quả nghiên cứu nhằm giải quyết một phần khó khăn trên.

2. BÀI TOÁN CẮT THÉP THANH KHÔNG NỐI

Trước hết ta giải quyết bài toán cắt thép thanh với giả thiết không nối, bài toán cắt có nối sẽ được trình bày trong mục 3.

2.1. Bài toán cắt vật liệu dạng thanh

Trong thực tế, ban đầu các vật liệu được sản xuất và lưu hành trên thị trường có một kích thước xác định theo tiêu chuẩn. Chẳng hạn, các loại thép thanh có chiều dài 11.7m; các thanh khung nhôm, ống nước có chiều dài 6m.

¹ Giảng viên khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Hồng Đức

Bài toán: Giả sử mỗi thanh vật liệu có chiều dài là L , trong công trình cần sử dụng các đoạn vật liệu loại i có chiều dài d_i với số lượng k_i , $i = \overline{1, m}$ như sau:

Bảng 1. Vật liệu cần sử dụng

Chiều dài	d_1	d_2	...	d_m
Số lượng	k_1	k_2	...	k_m

Hãy tìm phương án cắt sao cho tổng số thanh vật liệu sử dụng là ít nhất?

Mỗi loại bài toán cắt vật liệu lại có những yêu cầu riêng, chẳng hạn: Bài toán cắt khung nhôm thường không cho phép nối; các bài toán cắt thép, cắt ống nước thì cho phép có mối nối, nhưng theo yêu cầu kỹ thuật nên có một số đoạn không được phép nối; bài toán cắt dây lại tùy thuộc từng công trình mà có cho phép nối hay không.

Hiện nay trong các công trình xây dựng người ta thường cắt theo kiểu lần lượt: cắt hết loại này thì cắt tiếp loại khác cho đến khi đủ, nếu đoạn thừa còn dài thì cắt để lấy các đoạn ngắn hơn. Các đoạn thừa ngắn có thể nối với nhau để được các đoạn dài thích hợp.

Với những bài toán có quy mô nhỏ thì trước khi cắt tìm cách liệt kê các phương án rồi chọn phương án tốt nhất. Tuy nhiên, với những bài toán có quy mô lớn thì không đủ điều kiện để liệt kê thủ công tất cả các phương án nên phương án được thực hiện thường không tối ưu dẫn đến lượng vật liệu hao phí lớn hơn so với phương án tối ưu.

2.2. Mô hình hoá bài toán dưới dạng quy hoạch nguyên tuyến tính

Bước 1: Liệt kê tất cả cách cắt một thanh vật liệu ban đầu thành các đoạn $d_1, d_2, \dots$

Các cách cắt này được cho bởi một ma trận $A = [a_{ij}]$ gồm n hàng $m+1$ cột. Trong đó: n là số cách cắt một thanh vật liệu; m là số loại đoạn vật liệu yêu cầu; a_{ij} là số đoạn loại d_j trong cách cắt thứ i ; a_{ij} ở cột thứ $m+1$ chứa phần dư trong cách cắt thứ i .

Ma trận A gọi là *ma trận cắt*.

Bước 2: Mô hình hoá bài toán

Gọi x_i là số thanh được dùng để cắt theo cách cắt thứ i ($i=1, 2, \dots, n$).

Ta có:

- Tổng số thanh vật liệu cần dùng là: $f = \sum_{i=1}^n x_i$.

- Số đoạn vật liệu loại d_j cắt được là: $a_{1j}x_1 + a_{2j}x_2 + \dots + a_{nj}x_n$. Số lượng này cần phải lớn hơn hoặc bằng số lượng yêu cầu k_j ($j=1, 2, \dots, m$).

Từ đó ta có bài toán dạng Quy hoạch tuyến tính:

$$\begin{cases} f = \sum_{i=1}^n x_i \rightarrow \min \\ a_{11} x_1 + a_{21} x_2 + \dots + a_{n1} x_n \geq k_1 \\ a_{12} x_1 + a_{22} x_2 + \dots + a_{n2} x_n \geq k_2 \\ \dots\dots\dots \dots\dots\dots \dots\dots\dots \dots\dots\dots \dots\dots\dots \dots\dots\dots \\ a_{1m} x_1 + a_{2m} x_2 + \dots + a_{nm} x_n \geq k_m \\ x_i \in \{0,1,2,\dots\}, \quad i = 1, 2, \dots, n. \end{cases} \quad (1)$$

2.3. Thuật toán sinh ma trận cắt

Trước tiên cần sắp xếp dữ liệu vào theo chiều giảm dần chiều dài các đoạn d_1 : $d_1 > d_2 > \dots > d_m$, tiếp theo thực hiện sinh ma trận cắt bởi thủ tục đệ quy "k cat" sau:

```

procedure k_cat(Lk :real; k: integer);
var i,j : integer;
begin
    i:=phần nguyên((Lk+delta)/(d[k]+delta));
    if k=m then
        begin
            t[k]:=t[k]+i; (* Cắt i đoạn dk *)
            n:=n+1; (* Thêm 1 cách cắt mới *)
            for j:=1 to m do A[n][j]:=t[j];
            A[n][m+1] := Lk - i*(d[k]+delta);
            t[k]:=t[k]-i;
        end
    else
        for j:=0 to i do
            begin
                t[k]:=t[k]+j; (* Cắt j đoạn dk *)
                k_cat(Lk-j*(d[k]+delta), k+1); (* Tiếp tục cắt đoạn dk+1 *)
                t[k]:=t[k]-j;
            end;
        end;
end;

procedure sinhA; (* Thủ tục sinh ma trận cắt *)
var i: integer; (* L, m, n là các biến toàn cục *)
begin
    n:=0;
    for i:=1 to m do t[i]:=0;

```

```

    k_cat(L,1);
end;

```

Tham số delta là chiều dài hao phí tại mỗi lát cắt, mỗi lần cắt được đoạn d_k thì độ dài của thanh vật liệu bị giảm đi một lượng không phải là d_k mà là $d_k + \text{delta}$, còn đối với bài toán có thể bỏ qua hao phí tại lát cắt thì delta được gán giá trị bằng 0.

Dễ dàng nhận thấy thuật toán sinh ma trận cắt có độ phức tạp tính toán là $O(m*n)$, với m là số loại đoạn vật liệu theo yêu cầu bài toán, n là số tất cả cách cắt một thanh vật liệu thành những đoạn vật liệu d_k ($k=1, 2, \dots, m$). Với những bài toán lớn trong thực tế thì m cỡ <100 và n cỡ $<10^6$ nên thời gian chạy thủ tục sinh ma trận cắt trên máy tính cá nhân thực hiện rất tốt.

3. BÀI TOÁN CẮT THÉP THANH CÓ NỐI

3.1. Đặt vấn đề

Trong số các vật liệu dạng thanh sử dụng trong công trình xây dựng thì thép là một trong số vật liệu thường có yêu cầu nối vì thực tế việc nối thép sẽ giảm được đáng kể chi phí cho loại vật liệu này. Mặt khác việc nối thép không thể thiếu trong các công trình xây dựng vì có những kết cấu có độ dài lớn hơn độ dài của một thanh thép ($d_j > L$), khi đó bắt buộc phải thực hiện nối một hoặc nhiều thanh thép cùng với những đoạn thép ngắn khác mới có được một đoạn thép d_j .

Quy định trong xây dựng, việc nối thép có 2 loại: nối hàn với chiều dài mỗi nối $5d$ (d là đường kính thanh thép); nối buộc có chiều dài mỗi nối là $30d$. Với nối hàn thì lượng thép sử dụng sẽ ít hơn so với nối buộc, nhưng chi phí cho các mối nối hàn lại cao hơn bởi que hàn, công hàn, kiểm định mỗi nối...

Trong xây dựng không quy định rõ về số mối nối trong một loại kết cấu, nghĩa là ta có thể thực hiện nhiều đoạn ngắn nối lại để có được một đoạn thép d_j ($j = 1, 2, \dots, m$). Khi đó, trên mỗi đoạn thép d_j ($j = 1, 2, \dots, m$) nhận được có thể sẽ có nhiều mối nối. Tuy nhiên, đối với mỗi công trình cụ thể thì phải căn cứ vào bản thiết kế để đưa ra những yêu cầu riêng về việc nối cho từng kết cấu thép như: có những kết cấu không được nối; có những loại kết cấu chỉ cho phép có tối đa một mối nối; có những kết cấu chỉ cho nối ở 2 đầu; có những kết cấu chỉ cho nối ở giữa; có những kết cấu chỉ cho phép nối không quá một tỷ lệ nào đó tại mỗi lát cắt.

Như đã đề cập ở trên, việc nối thép sẽ giảm được hao phí trong công trình xây dựng nhưng điều đó cũng gây ra nhiều khó khăn và phức tạp cho việc giải quyết bài toán cắt thép. Tính phức tạp ở đây không chỉ là tăng kích thước bài toán mà còn khó khăn trong việc đáp ứng các yêu cầu về nối thép.

Trong quá trình nghiên cứu cài đặt thuật toán, chúng tôi nhận thấy trong số các thuật toán có thể tiếp cận bài toán quy hoạch nguyên tuyến tính thì *thuật toán lát cắt Gomory* (xem [1], tr 146 - 152) tỏ ra hiệu quả nhất, có thể giải tốt những bài toán cắt vật liệu với số ràng buộc cỡ vài chục và số ẩn cỡ hàng nghìn. Tuy nhiên, các bài toán cỡ lớn thực tế thường dẫn về bài toán quy hoạch nguyên tuyến tính có số ẩn cỡ hàng triệu, do vậy muốn sử dụng thuật toán lát cắt cho bài toán này thì phải tìm cách cải tiến thuật toán.

Trong bài viết này chúng tôi đề xuất giải pháp cải tiến thuật toán như sau: trước mỗi lần thực hiện lặp thuật toán đơn hình trong thuật toán lát cắt ta tìm cách loại bỏ bớt các ẩn bằng cách chỉ giữ lại những ẩn ứng với những cách cắt có phần dư nhỏ, nhưng phải thoả mãn có đủ số cách cắt để tạo được tất cả các đoạn d_k ($k=1, \dots, m$).

Quá trình loại ẩn được thực hiện bởi thủ tục sau:

{*Dùng mảng *Danhdau* để ghi nhận những cách cắt được chọn, biến h là số đảm bảo đủ cách cắt để tạo được các đoạn d_k , tập các cách cắt trong ma trận cắt được sắp xếp theo thứ tự tăng dần phần dư *}.

procedure *loai_an*();

var i, j, k : integer;

begin

for $i:=1$ to n do *Danhdau*[i]:=0;

for $j:=1$ to m do

begin

$k:=0$; $i:=1$;

while ($k < h$) and ($i \leq n$) do

begin

if $A[i][j] > 0$ then

begin

$k:=k+1$;

Danhdau[i]:=1;

end;

$i:=i+1$;

end;

end;

end;

Dễ thấy độ phức tạp thuật toán của thủ tục *loai_an* là $O(m*n)$.

Nhận xét: Thực tế cho thấy với số h thích hợp, sau khi thực hiện thủ tục trên thì số ẩn của bài toán giảm đi đáng kể, có thể từ hàng trăm nghìn chỉ còn vài nghìn. Do vậy,

với giải pháp cải tiến này thì bài toán quy hoạch nguyên tuyến tính đã nêu là giải được trên máy tính.

4.2. Kết quả thử nghiệm

Qua thử nghiệm cài đặt chương trình, chúng tôi nhận thấy các bài toán cắt khung nhôm, cắt thép không nổi hoàn toàn giải quyết được trên các máy tính cá nhân có tốc độ trung bình hiện nay. Thời gian tính toán của chương trình chủ yếu là ở giai đoạn giải bài toán quy hoạch nguyên tuyến tính. Đối với bài toán cắt vật liệu có nổi thì thuật toán chỉ đáp ứng được các công trình nhỏ với số loại kết cấu cỡ $m < 20$, còn các công trình lớn thì thường chúng tôi phải dùng phương pháp chia số liệu thành 2 hoặc 3 phần rồi giải quyết từng phần. Sau đây là kết quả thử nghiệm cắt thép của một công trình thực tế quy mô trung bình, đối với loại thép có đường kính 22mm.

Bảng 2. Số liệu thép yêu cầu sử dụng (Đơn vị đo độ dài: mm)

Chiều dài	12480	4670	4450	4350	4152	3700	2900	2820	2400	2020	1980
Số lượng	12	16	16	16	16	3	174	3	140	25	150

Kết quả chạy chương trình với yêu cầu cắt thép có nổi buộc (độ dài mỗi nổi là 30d):

KET QUUA CAT THEP:

Thép can dung la: $142(\text{Thanh}) = 1661.4(\text{m}) = \mathbf{4.958 \text{ (Tan)}}$

16 thanh: $[1: 4.45] + [1: 4.35] + [1: 2.90]$

25 thanh: $[1: 2.90] + [2: 2.40] + [1: 2.02] + [1: 1.98]$

41 thanh: $[1: 2.90] + [2: 2.40] + [2: 1.98] + [\text{Du: .04}]$

3 thanh: $[1: 4.67] + [1: 4.15] + [1: 2.82] + [\text{Du: .06}]$

17 thanh: $[4: 2.90] + [\text{Du: .10}]$

13 thanh: $[1: 4.67] + [1: 2.90] + [2: 1.98] + [\text{Du: .17}]$

1(x2) thanh: $[5: 4.15] + [1: 1.98]$

3(x2) thanh: $[1: 12.48] + [1: 4.15] + [1: 3.70] + [1: 2.40] + [\text{Du: .01}]$

5(x2) thanh: $[1: 12.48] + [2: 2.90] + [1: 2.40] + [1: 1.98] + [\text{Du: .08}]$

3(x2) thanh: $[1: 12.48] + [1: 4.15] + [3: 1.98] + [\text{Du: .17}]$

1 thanh: $[2: 4.15] + [1: 2.90] + [\text{Du: .50}]$

1(x2) thanh: $[1: 12.48] + [2: 1.98] + [\text{Du: 6.30}]$

Thép vun la: $7.15(\text{m}) = \mathbf{.021 \text{ (Tan)}}$

Thép su dung con du: $[1, 6.30]$

Thép vun: $[1, .50] [13, .17] [3, .17] [17, .10] [5, .08] [3, .06] [41, .04] [3, .01]$

