

Mục lục

1 Chuẩn bị	1
1.1 Kiến thức giải tích	1
1.2 Sai số làm tròn và số học máy tính	3
1.3 Thuật toán và sự hội tụ	3
1.4 MATLAB: ngôn ngữ tính toán và lập trình	3
1.5 MATLAB: giải tích và đại số	5
2 Giải gần đúng phương trình một biến	19
2.1 Phương pháp chia đôi	19
2.2 Phương pháp Newton và mở rộng	22
2.3 Lập điểm bất động	27
2.4 Phân tích sai số của các phương pháp lặp	31
2.5 Tăng tốc độ hội tụ	31
2.6 Nghiệm của đa thức và phương pháp Müller	32
3 Nội suy và xấp xỉ bằng đa thức	33
3.1 Nội suy tổng quát	33
3.2 Đa thức nội suy	34
3.3 Xấp xỉ số liệu và phương pháp Neville	38
3.4 Sai phân chia	39
3.5 Nội suy Hermite	42
3.6 Nội suy spline bậc ba	42
3.7 Đường cong tham số	42
4 Tính gần đúng đạo hàm và tích phân	43
4.1 Đạo hàm bằng số	44
4.2 Ngoại suy Richardson	48
4.3 Tích phân bằng số	48
4.4 Tích phân Romberg	54
4.5 Phương pháp cầu phương thích ứng	54

4.6	Cầu phương Gauss	54
4.7	Tích phân bội	54
4.8	Tích phân suy rộng	54
5	Bài toán giá trị ban đầu của phương trình vi phân thường	55
5.1	Lý thuyết cơ bản về bài toán giá trị ban đầu	56
5.2	Phương pháp Picard	57
5.3	Phương pháp chuỗi Taylor	60
5.4	Phương pháp Euler	63
5.5	Phương pháp Taylor bậc cao	66
5.6	Phương pháp Runge–Kutta	66
5.7	Điều khiển sai số và phương pháp Runge–Kutta–Fehlberg	70
5.8	Phương pháp đa bước	70
5.9	Phương pháp đa bước với bước nhảy biến thiên	70
5.10	Phương pháp ngoại suy	70
5.11	Phương trình cấp cao và hệ phương trình vi phân	70
5.12	Sự ổn định	70
5.13	Phương trình vi phân cứng	70
6	Phương pháp trực tiếp giải hệ phương trình tuyến tính	71
6.1	Hệ phương trình tuyến tính	71
6.2	Chiến thuật chốt	72
6.3	Đại số tuyến tính và ma trận nghịch đảo	72
6.4	Định thức của ma trận	72
6.5	Phân tích ma trận	72
6.6	Các dạng ma trận đặc biệt	72
7	Kỹ thuật lặp trong đại số tuyến tính	73
7.1	Chuẩn của vectơ và ma trận	73
7.2	Giá trị riêng và vectơ riêng	75
7.3	Lặp điểm bất động	75
7.4	Kỹ thuật lặp Jacobi và Gauss–Seidel	79
7.5	Ma trận nghịch đảo	81
7.6	Kỹ thuật giảm dư giải hệ tuyến tính	82
7.7	Giới hạn sai số và tinh chỉnh phép lặp	82
7.8	Phương pháp gradient liên hợp	82
8	Lý thuyết xấp xỉ	83
8.1	Xấp xỉ bình phương nhỏ nhất	83
8.2	Đa thức trực giao và xấp xỉ bình phương nhỏ nhất	88

8.3 Đa thức Chebyshev và [Economization] chuỗi lũy thừa	88
8.4 Xấp xỉ hàm hữu tỷ	88
8.5 Xấp xỉ đa thức lượng giác	88
8.6 Biến đổi Fourier nhanh	88
9 Xấp xỉ giá trị riêng	89
9.1 Đại số tuyến tính và giá trị riêng	89
9.2 Ma trận trực giao và biến đổi đồng dạng	89
9.3 Phương pháp lũy thừa	89
9.4 Phương pháp Householder	89
9.5 Thuật toán QR	89
9.6 Phân tích giá trị kỳ dị	89
10 Giải gần đúng hệ phương trình phi tuyến	90
10.1 Phương pháp lặp điểm bất động	92
10.2 Phương pháp Newton	96
10.3 Phương pháp độ dốc nhất	98
10.4 Đồng luân và các phương pháp mở rộng	98
11 Bài toán giá trị biên của phương trình vi phân thường	99
11.1 Phương pháp bắn tuyến tính	99
11.2 Phương pháp bắn cho bài toán phi tuyến	99
11.3 Phương pháp sai phân hữu hạn cho bài toán tuyến tính	99
11.4 Phương pháp sai phân hữu hạn cho bài toán phi tuyến	100
11.5 Phương pháp Rayleigh–Ritz	100
12 Giải gần đúng phương trình đạo hàm riêng	101
12.1 Phương trình đạo hàm riêng Elliptic	101
12.2 Phương trình đạo hàm riêng Parabolic	107
12.3 Phương trình đạo hàm riêng Hyperbolic	107
12.4 Giới thiệu về phương pháp phần tử hữu hạn	107
13 Các phương pháp tối ưu	108
13.1 Tối ưu không ràng buộc một biến	108
13.2 Tối ưu không ràng buộc nhiều biến	110
13.3 Quy hoạch tuyến tính	111
13.4 Tối ưu có ràng buộc	132

Chương 13

Các phương pháp tối ưu

13.1 Tối ưu không ràng buộc một biến

13.1.1 Phương pháp Newton

Ta đã biết

$$\min_{x \in [a, b]} f(x) = \min_{f'(x^*)=0} \{f(a), f(b), f(x^*)\} \quad (13.1)$$

Do đó bài toán $\min_{x \in [a, b]} f(x)$ đưa về bài toán tìm điểm dừng của f

$$f'(x) = 0 \quad (13.2)$$

Công thức lặp Newton (2.1) cho phương trình (13.2):

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} \quad (13.3)$$

Ví dụ 13.1. Xét hàm số $f(x) = \frac{x^2}{10} - 2 \sin x$.

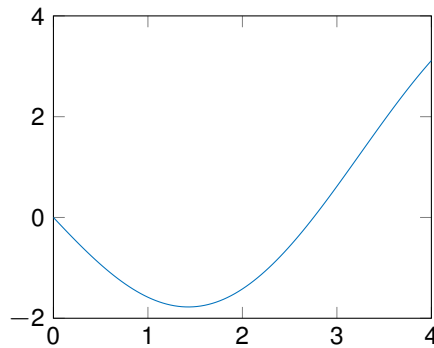
a) Vẽ đồ thị của $f(x)$ trên đoạn $[0, 4]$ (để chỉ ra hàm số đạt giá trị nhỏ nhất tại điểm dừng trên đoạn này)

b) Tìm $\min_{0 \leq x \leq 4} f(x)$.

Giải.

```
1 f = @(x) x.^2/10 - 2*sin(x)
```

a)



```

2 X = linspace(0, 4, 101)
3 Y = f(X)
4 plot(X, Y)

```

b) **Cách 1:** $\min_{0 \leq x \leq 4} f = -1.7757$ tại $x = 1.4275$

```

2 [x, fval] = fminbnd(f, 0, 4)

```

Cách 2: áp dụng phương pháp Newton $x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$ với $x_0 = 2.5$ và 3 lần lặp.

n	x_n	$f(x_n)$
0	2.5	-0.5719
1	0.9951	-1.5786
2	1.4690	-1.7738
3	1.4276	-1.7757

$\min_{0 \leq x \leq 4} f = -1.7757$ tại $x = 1.4276$

```

2 syms t
3 df = @(x) subs(diff(f(t)), x)
4 d2f = @(x) subs(diff(f(t), 2), x)

5 X = zeros(4, 1);
6 X(1) = 2.5;
7 f(X(1))
8 for n = 1:3
9     X(n+1) = X(n) - df(X(n)) / d2f(X(n));
10    f(X(n+1))
11 end
12 X

```

□

13.2 Tối ưu không ràng buộc nhiều biến

13.2.1 Phương pháp Newton

Xét $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $f(x) = f(x_1, x_2, \dots, x_n)$. Ký hiệu gradient và ma trận Hesse của f tại x

$$\begin{aligned}\nabla f(x) &= \left(\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)^T \\ \nabla^2 f(x) &= \left(\frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right)_{i,j=1,\dots,n}\end{aligned}\quad (13.4)$$

Bài toán tìm

$$\min f(x) \quad (13.5)$$

được đưa về bài toán tìm cực điểm dừng của f

$$\nabla f(x) = 0 \quad (13.6)$$

Phương pháp Newton

$$x^{(k+1)} = x^{(k)} - [\nabla^2 f(x^{(k)})]^{-1} \nabla f(x^{(k)}) \quad (13.7)$$

Ví dụ 13.2. Tìm $\min f(x) = f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, với xấp xỉ ban đầu $x^{(0)} = (-1.2, 1)^T$.

Giải.

```
1 f = @(x) 100*(x(2) - x(1)^2)^2 + (1 - x(1))^2
```

Cách 1: $\min f(x) = 8.1777 \cdot 10^{-10}$ tại $x_1 = 1, x_2 = 1$

```
2 x0 = [-1.2; 1]
3 [x, fval] = fminsearch(f, x0)
```

Cách 2: Áp dụng phương pháp Newton $x^{(k+1)} = x^{(k)} - [\nabla^2 f(x^{(k)})]^{-1} \nabla f(x^{(k)})$, với $x^{(0)} = (-1.2, 1)^T$ và 5 bước lặp. Ta có

$$\begin{aligned}\nabla f(x) &= \begin{bmatrix} 2x_1 - 400x_1(x_2 - x_1^2) - 2 \\ 200x_2 - 200x_1^2 \end{bmatrix} \\ \nabla^2 f(x) &= \begin{bmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{bmatrix}\end{aligned}$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$f(x^{(k)})$
0	-1.2	1	24.2000
1	-1.17528	1.38067	4.73188
2	0.763115	-3.17503	1411.85
3	0.76343	0.582825	0.0559655
4	0.999995	0.944027	0.313189
5	0.999996	0.999991	$1.85274 \cdot 10^{-11}$

```

2 x = sym('x', [2, 1])

3 g = sym('g', [2, 1])
4 for i = 1:2
5     g(i) = diff(f(x), x(i));
6 end
7 g

8 H = sym('H', [2, 2])
9 for i = 1:2
10    H(:, i) = diff(g, x(i));
11 end
12 H

13 X = [-1.2; 1]
14 f(X)
15 for k = 1:5
16    X = vpa(X - (subs(H, x, X))^-1 * subs(g, x, X), 6)
17    vpa(f(X), 6)
18 end

```

□

13.3 Quy hoạch tuyến tính

Bài toán quy hoạch tuyến tính dạng chuẩn: tìm

$$\min f = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \quad (13.8)$$

- Nếu có ràng buộc sau mà không có ràng buộc trước (trống), các tham số tương ứng của ràng buộc trống được đặt là [], cụ thể

```
[x, fval] = linprog(c, [], [], Ae, be, 1, u)
[x, fval] = linprog(c, [], [], Ae, be, 1, u)
[x, fval] = linprog(c, A, b, [], [], 1, u)
[x, fval] = linprog(c, [], [], [], [], 1, u)
[x, fval] = linprog(c, A, b, Ae, be, [], u)
```

Cách 2: nhập trực tiếp các thông tin (13.8) và (13.9) của bài toán. Cách này sẽ được mô tả cụ thể trong ví dụ ở dưới.

Ví dụ 13.3. Tìm min $\left(-x_1 - \frac{x_2}{3}\right)$, biết

- a) $x_1 + x_2 \leq 2$ $x_1 - x_2 \leq 2$ $-x_1 - x_2 \leq -1$
 $x_1 + \frac{x_2}{4} \leq 1$ $-\frac{x_1}{4} - x_2 \leq 1$ $-x_1 + x_2 \leq 2$
- b) Điều kiện (a) và $x_1 + \frac{x_2}{4} = \frac{1}{2}$
- c) Điều kiện (a) và $-1 \leq x_1 \leq 1.5, -0.5 \leq x_2 \leq 1.25$.

Giải. a) Lệnh

```
1 c = [-1; -1/3]
2 A = [1, 1; 1, 1/4; 1, -1; -1/4, -1; -1, -1; -1, 1]
3 b = [2; 1; 2; 1; -1; 2]
4 [x, f] = linprog(c, A, b)
```

cho kết quả

```
Optimal solution found.
x = 2x1
    0.6667
    1.3333
f = -1.1111
```

Ta có min $\left(-x_1 - \frac{x_2}{3}\right) = -1.1111$ tại $x_1 = 0.6667, x_2 = 1.3333$.

b) min $f = -0.6667$ tại $x_1 = 0, x_2 = 2$.

```

5 Ae = [1, 1/4]
6 be = [1/2]

7 [x, f] = linprog(c, A, b, Ae, be)

```

c) $\min f = -1.1042$ tại $x_1 = 0.6875, x_2 = 1.25$.

```

8 l = [-1; -0.5]
9 u = [1.5; 1.25]

10 [x, f] = linprog(c, A, b, [], [], l, u)

```

□

Ví dụ 13.4. Tìm $\max f = x_1 + 3x_2 + 2x_3$, biết

$$\begin{aligned}
 7x_1 + 2x_2 + 3x_3 &\leq 20 \\
 x_1 + 5x_2 + 4x_3 &\leq 30 \\
 x_1 + x_2 + x_3 &= 1 \\
 x_1, x_2, x_3 &\geq 0
 \end{aligned}$$

Giải. Đổi hàm mục tiêu để đưa về dạng $\min (-f) = -x_1 - 3x_2 - 2x_3$

Cách 1:

```

1 c = [-1; -3; -2]
2 A = [7, 2, 3; 1, 5, 4]
3 b = [20; 30]
4 Ae = [1, 1, 1]
5 be = [1]
6 l = [0; 0; 0]

7 [x, f] = linprog(c, A, b, Ae, be, l)

```

Cách 2:

```

1 x = optimvar('x', 3, 'LowerBound', 0)

2 prob = optimproblem('Objective', x(1) + 3*x(2) + 2*x(3), 'ObjectiveSense', 'max')

3 prob.Constraints.c1 = 7*x(1) + 2*x(2) + 3*x(3) <= 20
4 prob.Constraints.c2 = x(1) + 5*x(2) + 4*x(3) <= 30
5 prob.Constraints.c3 = x(1) + x(2) + x(3) == 1

```

```

6 problem = prob2struct(prob)
7 [x, f] = linprog(problem)

```

Hai cách đều cho kết quả

```

Optimal solution found.
x = 3x1
      0
     1.0000
      0
f = -3.0000

```

Ta được $\min(-f) = -3 \Rightarrow \max f = 3$ tại $x_1 = 0, x_2 = 1, x_3 = 0$. □

Ví dụ 13.5 (Bài toán kế hoạch sản xuất). Công ty Reddy Mikks sản xuất sơn nội thất và sơn ngoài trời. Nguyên liệu gồm 2 loại A và B với trữ lượng là 6 tấn và 8 tấn tương ứng. Để sản xuất một tấn sơn nội thất cần 2 tấn nguyên liệu A và 1 tấn nguyên liệu B. Hai số tương ứng của sơn ngoài trời là 1 tấn và 2 tấn. Qua tiếp thị được biết nhu cầu thị trường như sau (cho một ngày):

- Nhu cầu sơn nội thất không hơn nhu cầu sơn ngoài trời quá 1 tấn;
- Nhu cầu cực đại của sơn nội thất là 2 tấn.

Giá bán sỉ là 2000 USD một tấn sơn nội thất và 3000 USD một tấn sơn ngoài trời.

Cần sản xuất mỗi ngày như thế nào để doanh thu là lớn nhất?

Giải. Đặt x_1, x_2 lần lượt là số tấn sơn nội thất và ngoài trời cần sản xuất trong ngày. Doanh thu trong ngày là

$$f = 2x_1 + 3x_2$$

Nguyên liệu sử dụng không được quá trữ lượng

$$2x_1 + x_2 \leq 6 \quad (\text{nguyên liệu A})$$

$$x_1 + 2x_2 \leq 8 \quad (\text{nguyên liệu B})$$

Sản xuất không nhiều hơn nhu cầu thị trường

$$x_1 - x_2 \leq 1$$

$$x_1 \leq 2$$

Sản lượng phải không âm

$$x_1 \geq 0, x_2 \geq 0$$

Ta có bài toán

$$\max f = 2x_1 + 3x_2$$

hay

$$\min (-f) = -2x_1 - 3x_2$$

với

$$2x_1 + x_2 \leq 6$$

$$x_1 + 2x_2 \leq 8$$

$$x_1 - x_2 \leq 1$$

$$x_1 \leq 2$$

$$x_1 \geq 0, x_2 \geq 0$$

Lệnh

```
1 c = [-2; -3]
2 A = [2, 1; 1, 2; 1, -1; 1, 0]
3 b = [6; 8; 1; 2]
4 l = [0; 0]
5 [x, f] = linprog(c, A, b, [], [], l)
```

cho kết quả

```
Optimal solution found.
x = 2x1
    1.3333
    3.3333

f = -12.6667
```

Vậy $\min(-f) = -12.6667$ hay $\max f = 12.6667$ tại $x_1 = 1.3333, x_2 = 3.3333$. □

Ví dụ 13.6 (Bài toán khẩu phần ăn). Giả sử món ăn có n thành phần thực phẩm, với giá một đơn vị (khối lượng) thành phần j là $c_j, j = 1, \dots, n$. Đồng thời có m chất (như chất đạm, chất béo, đường, ...). Biết rằng một đơn vị thành phần j chứa a_{ij} đơn vị chất $i, i = 1, \dots, m$, và mức chấp nhận được số đơn vị chất i trong một đơn vị khối lượng của món ăn là nằm giữa $l_i \geq 0$ và $u_i \geq 0, i = 1, \dots, m$.

Lập phương án chế biến món ăn từ các thành phần thực phẩm sao cho đủ các chất, mà giá thành rẻ nhất.

Giải. Gọi x_j là số đơn vị khối lượng của thành phần j trong một đơn vị khối lượng của món ăn. Ta có bài toán

$$\min z = \sum_{j=1}^n c_j x_j \quad (1)$$

với

$$\begin{aligned} l_i &\leq \sum_{j=1}^n a_{ij} x_j \leq u_i, \quad i = 1, \dots, m \\ \sum_{j=1}^n x_j &= 1 \\ x_j &\geq 0, \quad j = 1, \dots, n \end{aligned} \quad (2)$$

□

Ví dụ 13.7 (Bài toán vận tải Hitchcock). Hàng hóa được vận chuyển từ m kho đến n cửa hiệu bán lẻ. Lượng hàng ở kho i là $s_i \geq 0$ (tấn), $i = 1, \dots, m$, và cửa hiệu j có nhu cầu $d_j \geq 0$ (tấn), $j = 1, \dots, n$. Cước vận chuyển một tấn hàng từ kho i đến cửa hiệu j là c_{ij} đồng. Giả sử tổng hàng ở các kho và tổng nhu cầu bằng nhau:

$$\sum_{i=1}^m s_i = \sum_{j=1}^n d_j \quad (1)$$

- a) Lập kế hoạch vận chuyển để tiền cước là nhỏ nhất, với điều kiện là mỗi cửa hàng đều nhận đủ và mỗi kho đều trao hết hàng.

Giải bài toán với

$$\text{b) } s = \begin{bmatrix} 150 \\ 250 \\ 50 \end{bmatrix}, \quad d = \begin{bmatrix} 50 \\ 150 \\ 150 \\ 100 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 5 & 6 & 2 \\ 3 & 1 & 3 & 4 \\ 4 & 5 & 7 & 2 \end{bmatrix}$$

c) s, d, C cho bởi bảng sau

$s_i \backslash d_j$	10	23	15
7	5	6	*
11	8	4	3
18	*	9	*
12	*	3	6

trong đó dấu * ở hàng i cột j ngụ ý

không chuyển hàng từ kho i đến cửa hàng j .

Giải. a) Gọi lượng hàng vận chuyển từ kho i đến cửa hàng j là x_{ij} , thì kế hoạch vận

chuyển là ma trận $(x_{ij})_{m \times n}$. Bài toán

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2)$$

với

$$\sum_{j=1}^n x_{ij} = s_i, \quad i = 1, \dots, m \quad (3)$$

$$\sum_{i=1}^m x_{ij} = d_j, \quad j = 1, \dots, n$$

$$x_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

b)

```

1 s = [150; 250; 50]
2 d = [50; 150; 150; 100]
3 C = [3, 5, 6, 2; 3, 1, 3, 4; 4, 5, 7, 2]

4 m = length(s)
5 n = length(d)

6 x = optimvar('x', m, n, 'LowerBound', 0)

7 prob = optimproblem('Objective', sum(C .* x, 'all'), '
    ObjectiveSense', 'min')

8 x_r = sum(x, 2)
9 x_c = sum(x)

10 for i = 1:m
11     field = sprintf('s%d', i)
12     prob.Constraints.(field) = x_r(i) == s(i)
13 end

14 for j = 1:n
15     field = sprintf('d%d', j)
16     prob.Constraints.(field) = x_c(j) == d(j)
17 end

18 problem = prob2struct(prob)
19 [x, f] = linprog(problem)

20 x = reshape(x, m, n)

```

```
Optimal solution found.
```

```
x = 12x1
```

```
50
```

```
0
```

```
0
```

```
0
```

```
150
```

```
0
```

```
50
```

```
100
```

```
0
```

```
50
```

```
⋮
```

```
f = 1100
```

```
x = 3x4
```

```
50      0      50      50
```

```
0      150     100      0
```

```
0      0      0      50
```

$$\min f = 1100 \text{ tại } x = \begin{bmatrix} 50 & 0 & 50 & 50 \\ 0 & 150 & 100 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix}$$

c)

```
1 s = [7; 11; 18; 12]
2 d = [10; 23; 15]
3 C = [5, 6, inf; 8, 4, 3; inf, 9, inf; inf, 3, 6]

4 m = length(s)
5 n = length(d)

6 x = optimvar('x', m, n, 'LowerBound', 0)

7 prob = optimproblem('Objective', sum(C(C < inf) .* x(C
    < inf)), 'ObjectiveSense', 'min')

8 for i = 1:m
9     row = x(i, :)
10    x_r(i) = sum(row(C(i, :) < inf))
11 end
```

```

12 for j = 1:n
13     col = x(:, j)
14     x_c(j) = sum(col(C(:, j) < inf))
15 end

16 for i = 1:m
17     field = sprintf('s%d', i)
18     prob.Constraints.(field) = x_r(i) == s(i)
19 end

20 for j = 1:n
21     field = sprintf('d%d', j)
22     prob.Constraints.(field) = x_c(j) == d(j)
23 end

24 problem = prob2struct(prob)
25 [x, f] = linprog(problem)

26 x = reshape(x, m, n)

```

```

f = 302
x = 4x3
    7    0    0
    3    0    8
    0   18    0
    0    5    7

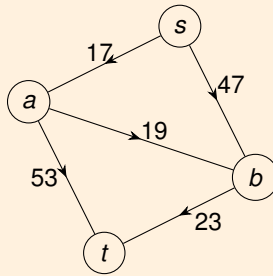
```

$$\min f = 302 \text{ tại } x = \begin{bmatrix} 7 & 0 & 0 \\ 3 & 0 & 8 \\ 0 & 18 & 0 \\ 0 & 5 & 7 \end{bmatrix}$$

□

Mô hình trong ví dụ trên gọi là *mô hình vận tải đóng*. Nếu không có giả thiết (1) và ràng buộc (3) đổi lại là $\sum_{j=1}^n x_{ij} \leq s_i$, tức là các kho có thể không trao hết, thì mô hình được gọi là *mô hình vận tải mở*.

Ví dụ 13.8 (Bài toán đường đi ngắn nhất). Cho đồ thị $G = (V, E)$, mỗi cạnh $(u, v) \in E$ có độ dài c_{uv} . Tìm đường đi ngắn nhất từ đỉnh s tới đỉnh t . Áp dụng với đồ thị sau



Giải. Dùng biến $x_{uv} \in \{1, 0\}$ để đánh dấu đường đi có qua / không qua cạnh $(u, v) \in E$.

Bài toán

$$\min I = \sum_{(u,v) \in E} c_{uv} x_{uv} \quad (1)$$

trong đó

- 1) với mỗi đỉnh $v \in V - \{s, t\}$, số cạnh đi vào v bằng số cạnh ra khỏi v .

$$\sum_{(u,v) \in E} x_{uv} = \sum_{(v,u) \in E} x_{vu} \quad (2)$$

- 2) có đúng một cạnh ra khỏi s

$$\sum_{(s,v) \in E} x_{sv} = 1 \quad (3)$$

- 3) có đúng một cạnh đi vào t

$$\sum_{(v,t) \in E} x_{vt} = 1 \quad (4)$$

```

1 V = ['s', 'a', 'b', 't']
2 C = [inf, 17, 47, inf;
3      inf, inf, 19, 53;
4      inf, inf, inf, 23;
5      inf, inf, inf, inf]
6 n = length(V)
7 s = find(V == 's')
8 t = find(V == 't')
9 x = optimvar('x', n, n, 'Type', 'integer', 'LowerBound',
10             0, 'UpperBound', 1)
11 prob = optimproblem('Objective', sum(C(C < inf) .* x(C <
12             inf), 'all'), 'ObjectiveSense', 'min')
```

```

11 for i = 1:n
12     row = x(i, :)
13     x_r(i) = sum(row(C(i, :) < inf))
14 end
15
16 for j = 1:n
17     col = x(:, j)
18     x_c(j) = sum(col(C(:, j) < inf))
19 end
20
21 for v = 1:4
22     if v ~= s & v ~= t
23         field = sprintf('c%d', v)
24         prob.Constraints.(field) = x_r(v) == x_c(v)
25     end
26 end
27
28 prob.Constraints.s = x_r(s) == 1
29 prob.Constraints.t = x_c(t) == 1
30
31 problem = prob2struct(prob)
32 [x, l] = intlinprog(problem)
33
34 x = reshape(x, n, n)

```

```

l = 59
x = 4x4
    0     1     0     0
    0     0     1     0
    0     0     0     1
    0     0     0     0

```

$$\min / = 59 \text{ tại } x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

□

Để dò đường đi từ ma trận x

```

1 v = s;
2 V(s)
3 while v ~= t

```

```

4     row = x(v, :);
5     v = find(row == 1);
6     V(v)
7 end

```

```

ans = 's'
ans = 'a'
ans = 'b'
ans = 't'

```

ta được đường đi ngắn nhất từ s tới t là $sabt$.

Ví dụ 13.9 (Bài toán Sudoku). Một bảng cỡ 9×9 ô chia thành 3×3 khối, mỗi khối cỡ 3×3 ô. Điền số từ 1 đến 9 vào các ô sao cho

- 1) Mỗi ô chỉ điền một số
- 2, 3) Trên mỗi hàng / cột, mỗi số chỉ xuất hiện đúng một lần
- 4) Trên mỗi khối, mỗi số xuất hiện đúng một lần.

			5				6	
8		9					1	
1	6			8	7			
3				2	6			
		7		1		6		
			8	5				3
			4	7			2	1
	4					9		8
	8				3			

4	7	2	5	3	1	8	6	9
8	5	9	6	4	2	3	1	7
1	6	3	9	8	7	2	5	4
3	1	8	7	2	6	4	9	5
5	9	7	3	1	4	6	8	2
6	2	4	8	5	9	1	7	3
9	3	6	4	7	8	5	2	1
7	4	1	2	6	5	9	3	8
2	8	5	1	9	3	7	4	6

Giải. Đặt

$$x_{ijk} = \begin{cases} 1, & \text{nếu } k \text{ điền vào ô } (i, j) \\ 0, & \text{ngược lại} \end{cases} \quad (1)$$

Tập các manh mối $C = \{(i, j, k) \mid k \text{ đã điền vào ô } (i, j)\}$, thì

$$x_{ijk} = 1 \quad \forall (i, j, k) \in C \quad (2)$$

Mỗi ô (i, j) chỉ điền được một số k

$$\sum_{k=1}^9 x_{ijk} = 1 \quad \forall i, j = 1, \dots, 9 \quad (3)$$

Trên hàng i , mỗi số k xuất hiện đúng một lần

$$\sum_{j=1}^9 x_{ijk} = 1 \quad \forall i, k = 1, \dots, 9 \quad (3)$$

Trên cột j , mỗi số k xuất hiện đúng một lần

$$\sum_{i=1}^9 x_{ijk} = 1 \quad \forall j, k = 1, \dots, 9 \quad (4)$$

B là tọa độ ô trên bên trái của các khối.

$$\begin{aligned} B &= \{(1, 1), (1, 4), (1, 7), (4, 1), (4, 4), (4, 7), (7, 1), (7, 4), (7, 7)\} \\ &= \{(p, q) \mid p, q = 1, 4, 7\} \end{aligned}$$

thì

$$\sum_{i=p}^{p+2} \sum_{j=q}^{q+2} x_{ijk} = 1 \quad \forall k = 1, \dots, 9, (p, q) \in B \quad (5)$$

```

1 C = [1 4 5; 1 8 6;
2       2 1 8; 2 3 9; 2 8 1;
3       3 1 1; 3 2 6; 3 5 8; 3 6 7;
4       4 1 3; 4 5 2; 4 6 6;
5       5 3 7; 5 5 1; 5 7 6;
6       6 4 8; 6 5 5; 6 9 3;
7       7 4 4; 7 5 7; 7 8 2; 7 9 1;
8       8 2 4; 8 7 9; 8 9 8;
9       9 2 8; 9 6 3]

10 x = optimvar('x', 9, 9, 9, 'Type', 'integer', 'LowerBound',
    , 0, 'UpperBound', 1)

11 prob = optimproblem('Objective', 0*x(1), 'ObjectiveSense',
    , 'min')

12 for r = C' % duyệt hàng của C
13     field = sprintf('clue%d%d%d', r(1), r(2), r(3));
14     prob.Constraints.(field) = x(r(1), r(2), r(3)) == 1;
15 end

16 for i = 1:9
17     for j = 1:9
18         s = 0;
19         for k = 1:9

```

```
20         s = s + x(i, j, k);
21     end
22     field = sprintf('cell%d%d', i, j);
23     prob.Constraints.(field) = s == 1;
24 end
25 end

26 for i = 1:9
27     for k = 1:9
28         s = 0;
29         for j = 1:9
30             s = s + x(i, j, k);
31         end
32         field = sprintf('row%d%d', i, k);
33         prob.Constraints.(field) = s == 1;
34     end
35 end

36 for j = 1:9
37     for k = 1:9
38         s = 0;
39         for i = 1:9
40             s = s + x(i, j, k);
41         end
42         field = sprintf('col%d%d', j, k);
43         prob.Constraints.(field) = s == 1;
44     end
45 end

46 for p = 1:3:7
47     for q = 1:3:7
48         for k = 1:9
49             s = 0;
50             for i = p:p+2
51                 for j = q:q+2
52                     s = s + x(i, j, k);
53                 end
54             end
55             field = sprintf('block%d%d%d', p, q, k);
56             prob.Constraints.(field) = s == 1;
57         end
58     end
59 end
```

```

60 problem = prob2struct(prob)
61 [x, f] = intlinprog(problem)

62 x = reshape(x, 9, 9, 9);

63 sol = zeros(9, 9);
64 for i = 1:9
65     for j = 1:9
66         for k = 1:9
67             if x(i, j, k) == 1
68                 sol(i, j) = k;
69             end
70         end
71     end
72 end
73 sol

```



Bài tập 13.3

13.1. Một máy cán thép có thể sản xuất hai sản phẩm thép tấm và thép cuộn với công suất cho mỗi loại là (nếu chỉ sản xuất một sản phẩm): 200 tấn/giờ, thép cuộn: 140 tấn/giờ. Lợi nhuận bán sản phẩm là: thép tấm 25 USD/tấn, thép cuộn: 30 USD/tấn. Theo tiếp thị, một tuần chỉ tiêu thụ được tối đa 6000 tấn thép tấm và 4000 tấn thép cuộn. Biết rằng máy làm việc 40 giờ một tuần.

Cần sản xuất mỗi loại bao nhiêu trong một tuần để có lợi nhuận cao nhất?

x_1, x_2 là số tấn thép tấm, thép cuộn sản xuất trong 1 tuần. Bài toán

$$\max f = 25x_1 + 30x_2 \Leftrightarrow \min (-f) = -25x_1 - 30x_2$$

với

$$\begin{aligned} \frac{x_1}{200} + \frac{x_2}{140} &\leq 40 \\ x_1 &\leq 6000 \\ x_2 &\leq 4000 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Lệnh

```

1 c = [-25; -30]
2 A = [1/200, 1/140]
3 b = [40]
4 l = [0; 0]
5 u = [6000; 4000]

6 [x, f] = linprog(c, A, b, [], [], l, u)

```

cho kết quả

Optimal solution found.

```

x = 2x1
103 ×
    6.0000
    1.4000

f = -192000

```

Vậy $\max f = -(-192000) = 192000$ tại $x_1 = 10^3 \times 6.0000 = 6000$, $x_2 = 10^3 \times 1.4000 = 1400$.

13.2 (Bài toán xe đạp). Có n người cùng phải đi quãng đường 10 dặm mà chỉ có một xe đạp một chỗ ngồi. Tốc độ đi bộ của người j là w_j và đi xe đạp là b_j , $j = 1, \dots, n$. Giả sử người đạp xe chậm nhất cũng nhanh hơn người đi bộ nhanh nhất.

Làm sao để thời gian người cuối cùng đến đích là ngắn nhất? Giải bài toán với $n = 3$, $w_1 = 4$, $w_2 = w_3 = 2$, $b_1 = 16$, $b_2 = b_3 = 12$.

Đặt x_j là quãng đường đạp xe của người j , $j = 1, \dots, n$ (sau đó để xe lại cho người kế tiếp đi, người cuối cùng đạp xe một mạch về đích). Khi đó $\sum_{j=1}^n x_j = 10$, và thời gian hoàn thành quãng đường của người j là

$$\frac{x_j}{b_j} + \frac{10 - x_j}{w_j}.$$

Thời gian để cả n người hoàn thành là $t = \max \left(\frac{x_j}{b_j} + \frac{10 - x_j}{w_j} \right)$. Bài toán tìm min t có dạng

$$\min f = t$$

trong đó

$$\begin{aligned} \sum_{j=1}^n x_j &= 10 \\ \frac{x_j}{b_j} + \frac{10 - x_j}{w_j} &\leq t, \quad j = 1, \dots, n \\ x_j &\geq 0, \quad j = 1, \dots, n \end{aligned}$$

Với $n = 3$, $w_1 = 4$, $w_2 = w_3 = 2$, $b_1 = 16$, $b_2 = b_3 = 12$:

Cách 1:

```

1 x = optimvar('x', 'LowerBound', 0)
2 y = optimvar('y', 'LowerBound', 0)
3 z = optimvar('z', 'LowerBound', 0)
4 t = optimvar('t')

5 prob = optimproblem('Objective', t, 'ObjectiveSense', 'min')

6 prob.Constraints.c0 = x + y + z == 10
7 prob.Constraints.c1 = x/16 + (10-x)/4 <= t
8 prob.Constraints.c2 = y/12 + (10-y)/2 <= t
9 prob.Constraints.c3 = z/12 + (10-z)/2 <= t

10 problem = prob2struct(prob)
11 [sol, f] = linprog(problem)

```

cho kết quả

```
Optimal solution found.
sol = 4x1      4 biến theo thứ tự chữ cái
      2.9167 → t
           0 → x
      5.0000 → y
      5.0000 → z

f = 2.9167
```

Cách 2: lập trình cho cách 1

```
1 w = [ 4,  2,  2]
2 b = [16, 12, 12]

3 x = optimvar('x', 3, 'LowerBound', 0)
4 t = optimvar('t')

5 prob = optimproblem('Objective', t, 'ObjectiveSense', 'min')

6 prob.Constraints.c0 = sum(x) == 10

7 for i = 1:3
8     field = sprintf('c%d', i)
9     prob.Constraints.(field) = x(i) / b(i) + (10 - x(i)) / w(i) <= t
10 end

11 problem = prob2struct(prob)

12 [sol, fval] = linprog(problem)
```

cho kết quả

```
Optimal solution found.
sol = 4x1      4 biến theo thứ tự chữ cái
      2.9167 → t
           0 → x1
      5.0000 → x2
      5.0000 → x3

f = 2.9167
```

Vậy quãng đường đạp xe của từng người là 0, 5, và 5 dặm.

13.3. Một nhà máy chế biến thịt sản xuất ba loại thịt bò, lợn và cừu, với tổng lượng mỗi ngày 480 tấn bò, 400 tấn lợn và 230 tấn cừu. Mỗi loại đều có thể bán được ở dạng tươi hoặc nấu chín. Tổng lượng các loại thịt có thể nấu chín (để bán) trong giờ làm việc là 420 tấn. Ngoài ra, có thể nấu thêm ngoài giờ 250 tấn (với giá cao hơn). Lợi nhuận thu được trên một tấn được cho bằng bảng bên sau, với đơn vị là triệu đồng.

	tươi	nấu chín	nấu chín ngoài giờ
bò	8	14	11
lợn	4	12	7
cừu	4	13	9

Ví dụ: phương án sản xuất sau cho lợi nhuận 9965 triệu đồng

	tươi	nấu chín	nấu chín ngoài giờ
bò	165 tấn	280 tấn	35 tấn
lợn	295 tấn	70 tấn	35 tấn
cừu	55 tấn	70 tấn	105 tấn

Tìm phương án sản xuất làm cực đại lợi nhuận.

Đặt $p = \begin{bmatrix} 8 & 14 & 11 \\ 4 & 12 & 7 \\ 3 & 13 & 9 \end{bmatrix}$, $m = (480, 400, 230)^T$, $x = (x_{ij})_{3 \times 3}$ với x_{ij} là số tấn loại thịt i chế biến theo kiểu j . Bài

toán max $\sum_{i,j=1}^3 p_{ij} x_{ij}$ biết

$$\sum_{j=1}^3 x_{ij} = m_i, \quad i = 1, 2, 3$$

$$\sum_{i=1}^3 x_{i2} = 420$$

$$\sum_{i=1}^3 x_{i3} \leq 250$$

```

1 p = [8, 14, 11; 4, 12, 7; 4, 13, 9]
2 m = [480; 400; 230]

3 x = optimvar('x', 3, 3, 'LowerBound', 0)
4 prob = optimproblem('Objective', sum(p .* x, 'all'), 'ObjectiveSense', 'max')

5 x_r = sum(x, 2) %lấy tổng các phần tử trên mỗi hàng của x
6 x_c = sum(x)    %.....cột.....

7 for i = 1:3
8     field = sprintf('c%d', i)
9     prob.Constraints.(field) = x_r(i) == m(i)
10 end

11 prob.Constraints.nau_chin = x_c(2) == 420
12 prob.Constraints.nau_ngoai_gio = x_c(3) <= 250

13 problem = prob2struct(prob)
14 [x, f] = linprog(problem)

```

cho kết quả

Optimal solution found.

```

x = 9x1
    440 → x11 cột 1
         0 → x21
         0 → x31
         0 → x12 cột 2
    400 → x22
     20 → x32

```

```

40 → x13 cột 3
0 → x23
210 → x33. Để đưa x về dạng ma trận 3 × 3, dùng lệnh reshape(x, 3, 3)

```

```
f = -10910
```

Ta được $\max f = -(-10910) = 10910$ (triệu) tại $x = \begin{bmatrix} 440 & 0 & 40 \\ 0 & 400 & 0 \\ 0 & 20 & 210 \end{bmatrix}$ (tần)

13.4. Một xưởng mộc làm bàn và ghế. Để một công nhân làm xong một bàn cần hai giờ còn một ghế thì cần 30 phút. Khách hàng thường mua nhiều lắm là bốn ghế kèm theo một bàn. Do đó, tỷ lệ sản xuất ghế và bàn nhiều nhất là 4:1. Giá bán một bàn là 135 USD và một ghế là 50 USD.

Hãy xác định xem số lượng sản xuất bàn và ghế hàng ngày là bao nhiêu để làm cực đại doanh thu. Biết rằng xưởng có bốn công nhân làm việc 8 giờ một ngày.

Gọi x_i, y_i là số bàn, ghế công nhân i được phân công làm trong ngày, $i = 1, \dots, 4$. Bài toán

$$\max f = 135 \sum_{i=1}^4 x_i + 50 \sum_{i=1}^4 y_i$$

sao cho

$$\sum_{i=1}^4 y_i - 4 \sum_{i=1}^4 x_i \leq 0$$

$$2x_i + 0.5y_i \leq 8, \quad i = 1, \dots, 4$$

$$x_i, y_i \geq 0, \quad i = 1, \dots, 4$$

```

1 x = optimvar('x', 4, 'LowerBound', 0)
2 y = optimvar('y', 4, 'LowerBound', 0)

3 prob = optimproblem('Objective', 135 * sum(x) + 50 * sum(y), 'ObjectiveSense', 'max')

4 prob.Constraints.ty_le = sum(y) <= 4 * sum(x)

5 for i = 1:4
6     field = sprintf('c%d', i)
7     prob.Constraints.(field) = 2*x(i) + 0.5*y(i) <= 8
8 end

9 problem = prob2struct(prob)
10 [xy, f] = linprog(problem)

```

Optimal solution found.

xy = 8x1 → biến sắp theo thứ tự

0 → x₁

0 → x₂

4 → x₃

4 → x₄

16 → y₁

$$\begin{aligned} 16 &\rightarrow y_2 \\ 0 &\rightarrow y_3 \\ 0 &\rightarrow y_4 \end{aligned}$$

$$f = -2680$$

Doanh thu cực đại $\max f = -(-2680) = 2680$ USD, khi số bàn, ghế mỗi công nhân làm mỗi ngày lần lượt là 0, 16; 0, 16; 4, 0; 0, 4, 0. Số bàn, ghế tối ưu xưởng sản xuất mỗi ngày là $\sum_{i=1}^4 x_i = 8$, $\sum_{i=1}^4 y_i = 32$.

13.5. Một nhà máy sản xuất hai kiểu mũ. Thời gian lao động để làm ra một mũ kiểu thứ nhất nhiều gấp hai lần để làm xong một mũ kiểu hai. Nếu sản xuất toàn mũ kiểu thứ hai thì nhà máy làm được 400 mũ một ngày. Thị trường tiêu thụ được trong mỗi ngày nhiều nhất là 150 mũ kiểu một và 200 mũ kiểu hai. Tiền lãi một mũ kiểu một là 8 USD và kiểu hai là 5 USD.

Cần sản xuất bao nhiêu mũ mỗi kiểu trong ngày để tổng tiền lãi lớn nhất?

Đặt x_1, x_2 là số mũ kiểu một, kiểu hai sản xuất trong ngày. Bài toán $\max f = 8x_1 + 5x_2$ hay $\min(-f) = -8x_1 - 5x_2$ biết

$$1) \ 0 \leq x_1 \leq 150, \ 0 \leq x_2 \leq 200$$

$$2) \ t \text{ là thời gian sản xuất một mũ kiểu hai} \Rightarrow 2tx_1 + tx_2 \leq 400t \Rightarrow 2x_1 + x_2 \leq 400$$

$$1 \quad c = [-8; -5]$$

$$2 \quad A = [2, \ 1]$$

$$3 \quad b = [400]$$

$$4 \quad l = [0; 0]$$

$$5 \quad u = [150; 200]$$

$$6 \quad [x, f] = \text{linprog}(c, A, b, [], [], l, u)$$

Optimal solution found.

$$x = 2 \times 1$$

$$100$$

$$200$$

$$f = -1800$$

$\max f = -(-1800) = 1800$ USD tại $x_1 = 100, x_2 = 200$.

13.6 (Bài toán gà và trứng của Dantzig). Trong hai tuần một con gà mái đẻ được 12 trứng để bán hoặc ấp được 4 trứng nở ra gà con. Sau 8 tuần thì bán tất cả gà và trứng với giá 0.6 USD một gà (lớn hoặc bé đều cùng giá) và 0.1 USD một trứng. Cần phải bố trí gà đẻ và ấp trứng như thế nào để doanh thu lớn nhất. Giải bài toán trong hai trường hợp

a) ban đầu có 100 gà mái và 100 trứng

b) ban đầu có 100 gà mái và không có trứng

13.7. Một công ty điện tử sản xuất hai kiểu radio trên hai dây chuyền độc lập. Công suất của dây chuyền một 60 radio/ngày và dây chuyền hai là 75 radio/ngày. Để sản xuất một chiếc radio kiểu một cần 10 linh kiện điện tử E, và một chiếc radio kiểu hai cần 8 linh kiện này. Số linh kiện này không quá 800. Tiền lãi khi bán một radio

kiểu một là 30 USD và kiểu hai là 20 USD.

Xác định phương án sản xuất cho lãi nhiều nhất trong ngày.

Đặt x_1, x_2 số radio kiểu một, kiểu hai sản xuất trong ngày. Bài toán $\max f = 30x_1 + 20x_2$ hay $\min(-f) = -30x_1 - 20x_2$ với

$$10x_1 + 8x_2 \leq 800$$

$$0 \leq x_1 \leq 60, 0 \leq x_2 \leq 75$$

```

1 c = [-30; -20]
2 A = [10, 8]
3 b = [800]
4 l = [0; 0]
5 u = [60; 75]
6 [x, f] = linprog(c, A, b, [], [], l, u)

```

Optimal solution found.

```

x = 2x1
    60
    25

f = -2300

```

$\max f = -(-2300) = 2300$ USD tại $x_1 = 60, x_2 = 25$.

13.4 Tối ưu có ràng buộc

Tài liệu tham khảo

- [1] Phạm Kỳ Anh. *Giải tích số*. Đại học Quốc gia Hà Nội, 2002. 284 trang.
- [2] Steven C. Chapra **and** Raymond P. Canale. *Numerical Methods for Engineers*. phiên bản 8. Cengage Learning, 2020. 1006 trang.
- [3] Phan Văn Hạp **and** Lê Đình Thịnh. *Phương pháp tính và các thuật toán*. Nhà xuất bản Giáo dục, 2000. 400 trang.
- [4] Doãn Tam Hòe. *Toán học tính toán*. Đại học Quốc gia Hà Nội, 2009. 240 trang.
- [5] Trần Huệ Nương Phan Quốc Khánh. *Quy hoạch tuyến tính*. phiên bản 2. Nhà xuất bản Giáo dục, 2003. 457 trang.

