

第 7 章 约束非线性规划

上一章介绍了无约束问题的最优化方法. 但实际问题中, 大多数都是有约束条件的问题. 一般说来, 求解带有约束条件的非线性规划问题比无约束问题和线性规划问题都要复杂得多, 非线性规划不像线性规划那样有统一的数学模型及单纯形法这一通用解法. 非线性规划目前还没有适合于各种问题的一般算法, 各种算法都有特定的适用范围, 带有一定的局限性, 它是人们需要更深入的领域. 本章将介绍约束非线性规划的基本理论和方法.

7.1 约束非线性规划问题的描述

下面通过实例来归纳出约束非线性规划问题的一般数学模型, 然后介绍与该类问题相关的一些基本概念.

例 7-1 (选址问题) 设有 n 个市场, 第 j 个市场的位置为 (a_j, b_j) , 对某种货物的需要量为 $q_j (j=1, 2, \dots, n)$, 先计划建立 m 个货栈, 第 i 个货栈的容量为 $c_i (i=1, 2, \dots, m)$, 试确定货栈的位置, 使各货栈到各市场的运输量与路程的乘积之和最小.

解 设货栈的位置为 (x_i, y_i) , 第 i 个货栈供给第 j 个市场的货物量为 z_{ij} , 第 i 个货栈到第 j 个市场的距离为 d_{ij} , 则 $d_{ij} = \sqrt{(x_i - a_j)^2 + (y_i - b_j)^2}$

因为每个货栈的供货量不能超过它的容量, 且每个市场得到的货物量应等于它的需求量, 所以需要满足下列的限制条件:

$$\sum_{j=1}^n z_{ij} \leq c_i (i=1, 2, \dots, m);$$

$$\sum_{i=1}^m z_{ij} \leq q_j (j=1, 2, \dots, n);$$

$$x_i \geq 0, y_i \geq 0, z_{ij} \geq 0.$$

故该问题可归结在上述限制条件下, 求 $x_i, y_i (i=1, 2, \dots, m)$, 使得

$$\sum_{i=1}^m \sum_{j=1}^n z_{ij} \sqrt{(x_i - a_j)^2 + (y_i - b_j)^2}$$

最小.

上述例题是在一组等式或不等式的约束下, 求一个函数的最大值 (或最小值) 问题, 其中目标函数或约束条件中至少有一个非线性函数, 这类问题称之为**约束最优化问题**或**约束非线性规划**.

其一般形式可概括为:

$$\begin{aligned} \min_{x \in R^n} z &= f(x) \\ \text{s.t.} \quad &\begin{cases} g_i(x) \leq 0, & i \in E = \{1, 2, \dots, m\} \\ h_k(x) = 0, & k \in I = \{1, 2, \dots, l\} \end{cases} \end{aligned} \quad (7.1)$$

其中, $f(x)$ 称为**目标函数**, n 维向量 $x = (x_1, x_2, \dots, x_n)^T$ 称为**决策变量**.

建立优化问题的数学模型, 首先要确定问题的**决策变量**, 然后构造模型的目标函数 $f(x)$ 和允许取值的范围 $x \in \Omega$, Ω 称**可行域**, 由约束条件界定.

对于一般约束最优化问题 (7.1) 式, 记其可行域为

$$\Omega = \{x \mid c_i(x) = 0, i \in E, c_j(x) \leq 0, j \in I\} \quad (7.2)$$

若对 $x^* \in \Omega$, 存在 $\varepsilon > 0$, 使当 $x \in \Omega$ 且 $\|x - x^*\| \leq \varepsilon$ 时, 总有

$$f(x) \geq f(x^*)$$

则称 x^* 为约束问题 (7.1) 式的**局部解**, 或简称 x^* 为**最优解**. 若当 $x \in \Omega$ 且 $0 < \|x - x^*\| \leq \varepsilon$, 总有

$$f(x) > f(x^*)$$

则称 x^* 为约束问题的**严格局部解**.

1. 等式约束的最优性条件

对于一般的等式约束问题,

$$\begin{aligned} \min_{x \in R^n} z &= f(x) \\ \text{s.t.} \quad &h_k(x) = 0, \quad k \in I = \{1, 2, \dots, l\}. \end{aligned} \quad (7.3)$$

利用多元微分学中求解条件极值问题的 Lagrange 乘子法思想, 引进下面的辅助函数

$$L(x, \mu) = f(x) + \sum_{j=1}^l \mu_j h_j(x) \quad (7.4)$$

定理 1 (一阶必要条件) 对于问题 (7.3) 式, $f, h_j (j=1, 2, \dots, l)$ 在 x^* 处可微, 向量组 $\nabla h_1(x^*), \dots, \nabla h_l(x^*)$ 线性无关, 若 x^* 是最优解, 则必存在相应的 $\mu^* = (\mu_1^*, \mu_2^*, \dots, \mu_l^*)^T$, 使得

$$\nabla_x L(x^*, \mu^*) = \nabla f(x^*) - \sum_{j=1}^l \mu_j^* \nabla h_j(x^*) = 0. \quad (7.5)$$

2. 不等式约束问题的最优性条件

考虑问题

$$\begin{aligned} \min_{x \in R^n} z &= f(x) \\ \text{s.t.} \quad &g_i(x) \leq 0, \quad i \in E = \{1, 2, \dots, m\}, \end{aligned} \quad (7.6)$$

定理 2 (Kuhn-Tucker 必要条件) 对于问题 (7.6) 式, 设 $x^* \in \Omega$, $I(x^*) = \{i \mid g_i(x^*) = 0, 1 \leq i \leq m\}$, $f, g_i (i \in I(x^*))$ 在 x^* 处可微, $g_i (i \notin I(x^*))$ 在 x^* 连续, 向量组 $\nabla g_i(x^*) (i \in I(x^*))$ 线性无关, 则 x^* 是局部最优解, 则存在相应的 $\mu_i (i \in I(x^*))$, 使得

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^m \mu_i \nabla g_i(x^*) = 0 \\ \mu_i \nabla g_i(x^*) = 0, \quad i = 1, 2, \dots, m \\ \mu_i \geq 0, \quad i = 1, 2, \dots, m \end{cases} \quad (7.7)$$

3. 等式和不等式约束问题的最优性条件

对于一般的约束优化问题 (7.1) 式, 构造拉格朗日函数

$$L(x, \lambda, \mu) = f(x) - \sum_{i=1}^m \lambda_i g_i(x) - \sum_{j=1}^l \mu_j h_j(x) \quad (7.8)$$

利用 (7.4) 式和 (7.5) 式, 可得到同时含有等式与不等式约束条件的 Kuhn-Tucher 条件.

定理 3 (Kuhn-Tucher 必要条件, 简称 K-T 条件) 对于非线性规划问题 (7.1) 式, 设 $x^* \in \Omega$, $I(x^*) = \{i | g_i(x^*) = 0, 1 \leq i \leq m\}$, f, g_i, h_j 均可微且 $\nabla g_i(x^*)$, $i \in I(x^*)$, $\nabla h_j(x^*)$, $j = 1, \dots, l$ 线性无关, 则 x^* 为 (7.1) 式最优解的必要条件为: 存在相应的 $\mu_i (i \in I)$ 和 $v_j (j = 1, \dots, l)$, 使得

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^m \mu_i \nabla g_i(x^*) + \sum_{j=1}^l v_j \nabla h_j(x^*) = 0 \\ \mu_i \nabla g_i(x^*) = 0, i = 1, \dots, m \\ \mu_i \geq 0, i = 1, \dots, m \end{cases} \quad (7.9)$$

7.2 约束非线性规划问题的求解

求解约束问题方法的思路大致可分为三大类: 第一类是将有约束问题转化为无约束问题之后再求解, 例如罚函数法; 第二类是构造合适的迭代格式求解, 在迭代过程中, 不仅要使目标函数值有所下降, 而且要使迭代点都落在可行域内, 像可行方向法、投影梯度法、复合形法等就属于这类方法; 第三类是利用一系列简单函数的解点去近似原约束问题的最优解.

本节我们主要介绍第一类方法——罚函数法, 具体包括**外点罚函数法 (外点法)**和**内点罚函数法 (内点法)**. 其基本思想是通过构造惩罚函数把约束问题转化为一系列无约束最优化问题. 进而用无约束最优化方法求解, 对于无约束问题求解过程中企图违反约束的那些迭代点给以很大的目标数值, 迫使这一系列无约束问题的极小点或者不断地向可行域靠近, 或者一直保持在可行域内移动, 直到收敛于原约束问题的极小点.

1. 外点法

对于等式约束问题

$$\begin{aligned} \min_{x \in R^n} f(x) \\ \text{s.t. } h_j(x) = 0, j = 1, \dots, l \end{aligned} \quad (7.10)$$

定义辅助函数

$$F_1(x, \sigma) = f(x) + \sigma \sum_{j=1}^l h_j^2(x) \quad (7.11)$$

参数 σ 是很大的正数. 当我们求解无约束问题

$$\min_{x \in R^n} F_1(x, \sigma) \quad (7.12)$$

得到的最优解 x^* , 满足 $h_j(x^*) = 0, j = 1, \dots, l$ 时, 显然 x^* 即为原约束问题 (7.10) 式的最优解. 而在求解 (7.12) 式的过程中, 若 x^k 不满足 $h_j(x^k) = 0$, 则 (7.11) 式中第二项将是很大正数,

限制它成为极小点，以迫使无约束问题 (7.11) 式的最优解满足 $h_j(x)=0(j=1,\dots,l)$ 。由此可见，求解问题 (7.12) 式能够得到问题 (7.10) 式的近似解。

对于不等式约束问题

$$\begin{aligned} \min_{x \in R^n} f(x) \\ \text{s.t. } g_i(x) \leq 0, i=1, \dots, m \end{aligned} \quad (7.13)$$

可用同样的思想方法构造辅助函数，即在可行点上，辅助函数值和 f 值相等；在不可行点上，辅助函数等于 f 的值再加上一个很大的正数。例如可定义

$$F_2(x, \sigma) = f(x) + \sigma \sum_{i=1}^m [\max\{0, g_i(x)\}]^2 \quad (7.14)$$

这样可将问题 (7.13) 式转化为无约束问题

$$\min F_2(x, \sigma) \quad (7.15)$$

对于一般情形

$$\begin{cases} \min f(x) \\ \text{s.t. } \begin{cases} g_i(x) \leq 0, i=1, \dots, p \\ h_j(x) = 0, j=1, \dots, q \end{cases} \end{cases} \quad (7.16)$$

可定义辅助函数

$$F(x, \sigma) = f(x) + \sigma P(x) \quad (7.17)$$

其中 $P(x)$ 可取如下形式

$$P(x) = \sum_{i=1}^m [\max\{0, g_i(x)\}]^2 + \sum_{j=1}^l |h_j(x)|^2 \quad (7.18)$$

这样就把约束问题 (7.16) 式转化为无约束问题

$$\min F(x, \sigma) \quad (7.19)$$

其中 σ 是很大的数。

由构造知，当 x 为 (7.16) 式的可行点时， $P(x)=0$ ，从而有 $F(x, \sigma)=f(x)$ ；当 x 不是 (7.16) 式的可行点时， $\sigma P(x)$ 是很大的正数，可视为 x 脱离可行域的一种惩罚，其作用是在极小化 (7.19) 式的过程中迫使迭代点靠近 (7.16) 式的近似解。而且 σ 越大，近似程度越好。通常将 $\sigma P(x)$ 称为惩罚项， σ 称为惩罚因子， $F(x, \sigma)$ 称为惩罚函数。

外点法具体计算步骤为：

(1) 给定初始点 x^0 ，允许误差 $\varepsilon > 0$ ，初始参数 $\sigma_1 > 0$ ， $k=1$ ；

(2) 以 x^{k-1} 为初始点，求解无约束问题

$$\min f(x) + \sigma_k P(x)$$

设求得极小点为 x^k ；

(3) 若 $\sigma_k P(x^k) < \varepsilon$ ，则停止迭代，得近似解 x^k ；否则令 $\sigma_{k+1} = c\sigma_k$ ， $c > 1$ ， $k = k+1$ ，返回步骤 (2)。

例 7-2 求解非线性规划

$$\begin{aligned} \min f(X) &= (x_1 - 1)^2 + x_2^2 \\ \text{s.t. } g(X) &= x_2 \geq 1 \end{aligned}$$

解 定义惩罚函数

$$F(X, \sigma) = (x_1 - 1)^2 + x_2^2 + \sigma [\max\{0, -(x_2 - 1)\}]^2$$

$$= \begin{cases} (x_1 - 1)^2 + x_2^2, & x_2 \geq 1 \\ (x_1 - 1)^2 + x_2^2 + \sigma(x_2 - 1)^2, & x_2 < 1 \end{cases}$$

用解析法求解 $\min F(X, \sigma)$, 有

$$\frac{\partial F}{\partial x_1} = 2(x_1 - 1),$$

$$\frac{\partial F}{\partial x_2} = \begin{cases} 2x_2, & x_2 \geq 1 \\ 2x_2 + 2\sigma(x_2 - 1), & x_2 < 1 \end{cases}$$

$$\text{令 } \frac{\partial F}{\partial x_1} = 0 \text{ 且 } \frac{\partial F}{\partial x_2} = 0, \text{ 得到 } X^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{\sigma}{1 + \sigma} \end{bmatrix}$$

显然, 当 $\sigma \rightarrow +\infty$ 时

$$X^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

X^* 确为最优解.

用此法求得的无约束问题的解开始往往是不满足约束条件的, 从可行域的外部随着 σ 的增大而趋于 x^* 的, 故称此法为**外点法**.

实际计算中, 惩罚因子 σ 的选择十分重要. 一般的方法是取一个趋向无穷大的严格递增正数列 $\{\sigma_k\}$, 逐个求解

$$\min f(x) + \sigma_k P(x)$$

而得一个极小点的序列 $\{x_k^*\}$, 如此通过一系列无约束问题来获得约束问题最优解的方法称为**序列无约束极小化方法**, 简称为 **SUMT 方法**.

2. 内点法

内点法从可行域的内部出发, 并保持在可行域内部搜索, 因此这种方法只适用于下列只有不等式约束的问题 (7.13) 式.

定义障碍函数

$$G(x, r) = f(x) + rH(x) \quad (7.20)$$

其中 $H(x)$ 是连续函数, 当点 x 趋向可行域的边界时, $H(x) \rightarrow +\infty$. 所以构造函数

$$H(x) = \sum_{i=1}^m \frac{1}{g_i(x)}$$

或

$$H(x) = -\sum_{i=1}^m \log g_i(x)$$

r 是很小的正数. 这样, 当点 x 趋向边界时, $G(x, r) \rightarrow +\infty$; 否则 $G(x, r) \approx f(x)$. 因此, 我们可以通过求解问题

$$\min G(x, r)$$

得到所求问题的解.

根据 $G(x, r)$ 的定义, 显然 r 越小, (7.20) 式越接近 (7.13) 式的解. 因此在实际计算中, 也是采取 SUMT 方法, 取一个严格单调减少且趋于零的惩罚因子数列 $\{r_k\}$, 对每个 r_k , 求解 $\min G(x, r_k)$.

内点法具体计算步骤为:

- (1) 给定初始点 x^0 , 允许误差 $\varepsilon > 0$, 初始参数 $r_1 > 0$, $k=1$;
- (2) 以 x^{k-1} 为初始点, 求解问题 $\min f(x) + r_k H(x)$ 设求得极小点为 x^k ;
- (3) 若 $r_k H(x^k) < \varepsilon$, 则停止迭代, 得近似解 x^k ; 否则令 $r_{k+1} = \beta r_k$, $\beta \in (0, 1)$, $k = k+1$, 返回步骤 (2).

例 7-3 用内点法求解

$$\begin{aligned} \min & \frac{1}{12}(x_1 + 1)^3 + x_2 \\ \text{s.t.} & \begin{cases} x_1 - 1 \geq 0 \\ x_2 \geq 0 \end{cases} \end{aligned}$$

解 定义障碍函数

$$G(X, r_k) = \frac{1}{12}(x_1 + 1)^3 + x_2 + r_k \left(\frac{1}{x_1 - 1} + \frac{1}{x_2} \right)$$

用解析法求解 $G(X, r_k)$, 令

$$\begin{aligned} \frac{\partial G}{\partial x_1} &= \frac{1}{4}(x_1 + 1)^2 - \frac{r_k}{(x_1 - 1)^2} = 0 \\ \frac{\partial G}{\partial x_2} &= 1 - \frac{r_k}{x_2^2} = 0 \end{aligned}$$

$$\text{解得 } X_{r_k}^* = (x_1, x_2) = \left(\sqrt{1 + 2\sqrt{r_k}}, \sqrt{r_k} \right)$$

当 $r_k \rightarrow 0$ 时, $X_{r_k}^* \rightarrow X^* = (1, 0)$, X^* 确为最优解.

7.3 用 MATLAB 求解非线性规划

1. 二次规划的求解

若某非线性规划的目标函数为自变量 x 的二次函数, 约束条件又全是线性的, 就称这种规划为二次规划.

MATLAB 中二次规划的数学模型可表述如下:

$$\begin{aligned} \min Z &= \frac{1}{2} X^T H X + C^T X \\ \text{s.t.} & \begin{cases} A X \leq b \\ A_{eq} \cdot X = b_{eq} \\ VLB \leq X \leq VUB \end{cases} \end{aligned}$$

这里 H 是实对称矩阵, C, b 是列向量, A 是相应维数的矩阵.

MATLAB 中求解二次规划的命令是:

- (1) $x = \text{quadprog}(H, C, A, b);$
- (2) $x = \text{quadprog}(H, C, A, b, Aeq, beq);$
- (3) $x = \text{quadprog}(H, C, A, b, Aeq, beq, VLB, VUB);$
- (4) $x = \text{quadprog}(H, C, A, b, Aeq, beq, VLB, VUB, X0);$
- (5) $x = \text{quadprog}(H, C, A, b, Aeq, beq, VLB, VUB, X0, options);$
- (6) $[x, fval] = \text{quadprog}(\dots);$
- (7) $[x, fval, exitflag] = \text{quadprog}(\dots);$
- (8) $[x, fval, exitflag, output] = \text{quadprog}(\dots);$

其中命令(6)~(8)的等式右边可选用命令(1)~(5)的等式右边。

例 7-4 求解二次规划

$$\begin{aligned} \min f(x_1, x_2) &= 2x_1^2 - 4x_1x_2 + 4x_2^2 - 6x_1 - 3x_2 \\ s.t. \quad &\begin{cases} x_1 + x_2 \leq 3 \\ 4x_1 + x_2 \leq 9 \\ x_1, x_2 \geq 0 \end{cases} \end{aligned}$$

解 写成标准形式

$$\begin{aligned} \min z &= \frac{1}{2}(x_1, x_2) \begin{bmatrix} 4 & -4 \\ -4 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -2 \\ -6 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ s.t. \quad &\begin{cases} \begin{bmatrix} 1 & 1 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 3 \\ 9 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{cases} \end{aligned}$$

在 command. window 编写命令如下:

```
H=[4 -4;-4 8];
C=[-6;-3];
A=[1,1;4,1];
b=[3;9];
Aeq=[];Beq=[];
VLB=[0;0]; VUB=[];
[x,z]=quadprog(H,C,A,b, Aeq, Beq,VLB,VUB,zeros(2,1))
```

运算结果为

```
x =
    1.9500
    1.0500
z =
   -11.0250.
```

2. 一般非线性规划的解法

MATLAB 中非线性规划的数学模型写成以下形式

$$\min f(x)$$

$$\begin{cases} Ax \leq b \\ Aeq \cdot X = Beq \\ C(x) \leq 0 \\ Ceq(x) = 0 \\ VLB \leq X \leq VUB \end{cases}$$

其中 $f(x)$ 是标量函数, A, b, Aeq, Beq 是相应维数的矩阵和向量, $C(x), Ceq(x)$ 是非线性向量函数.

用 MATLAB 求解上述问题, 分 3 个基本步骤:

(1) 首先建立 M 文件 fun1.m 定义目标函数 F(x):

```
function f=fun1(x);
f=F(X);
```

(2) 若约束条件中有非线性约束: $C(x) \leq 0$ 或 $Ceq(x) = 0$, 则建立 M 文件 nonlcon.m 定义函数 C(x) 与 Ceq(x):

```
Function [C,Ceq]=fun2(x);
C=...
Ceq=...
```

(3) 在 MATLAB 命令窗口输入的基本格式如下:

- 1) $x = \text{fmincon}('fun1', X0, A, b)$
- 2) $x = \text{fmincon}('fun1', X0, A, b, Aeq, Beq)$
- 3) $x = \text{fmincon}('fun1', X0, A, b, Aeq, Beq, VLB, VUB)$
- 4) $x = \text{fmincon}('fun1', X0, A, b, Aeq, Beq, VLB, VUB, 'fun2')$
- 5) $x = \text{fmincon}('fun1', X0, A, b, Aeq, Beq, VLB, VUB, 'fun2', options)$
- 6) $[x, fval] = \text{fmincon}(...)$

例 7-5 $\min f = -x_1 - 2x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$

$$s.t. \begin{cases} 2x_1 + 3x_2 \leq 6 \\ x_1 + 4x_2 \leq 5 \\ x_1, x_2 \geq 0 \end{cases}$$

解 先建立 M 文件 fun2.m:

```
function f=fun2(x);
f=-x(1)-2*x(2)+(1/2)*x(1)^2+(1/2)*x(2)^2
```

输入命令:

```
x0=[1;1];
A=[2 3 ;1 4]; b=[6;5];
Aeq=[];beq=[];
VLB=[0;0]; VUB=[];
[x,fval]=fmincon('fun2',x0,A,b,Aeq,beq,VLB,VUB)
```

运算结果为:

```
x = 0.7647      1.0588
fval = -2.0294
```

例 7-6 $\min f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$

$$s.t. \begin{cases} x_1 + x_2 = 0 \\ 1.5 + x_1 x_2 - x_1 - x_2 \leq 0 \\ -x_1 x_2 - 10 \leq 0 \end{cases}$$

解 先建立 M 文件 fun3.m, 定义目标函数:

```
function f=fun3(x);  
f=exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
```

再建立 M 文件 mycon.m 定义非线性约束:

```
function [g,ceq]=mycon(x);  
g=[1.5+x(1)*x(2)-x(1)-x(2);-x(1)*x(2)-10];  
ceq=[];
```

主程序 youh3.m 为:

```
x0=[-1;1];  
A=[];b=[];  
Aeq=[1 1];beq=[0];  
VLB=[]; VUB=[];  
[x,fval]=fmincon('fun3',x0,A,b,Aeq,beq, VLB, VUB,'mycon')
```

运算结果为:

```
x = -1.2247  
      1.2247  
fval =1.8951
```

7.4 案例分析

7.4.1 飞行管理问题

1. 问题提出

在约 10,000m 高空的某边长 160km 的正方形区域内, 经常有若干架飞机作水平飞行. 区域内每架飞机的位置和速度向量均由计算机记录其数据, 以便进行飞行管理. 当一架欲进入该区域的飞机到达区域边缘时, 记录其数据后, 要立即计算并判断是否会与区域内的飞机发生碰撞. 如果会碰撞, 则应计算如何调整各架(包括新进入的)飞机飞行的方向角, 以避免碰撞.

2. 基本假设

- (1) 不碰撞的标准为任意两架飞机的距离大于 8km;
- (2) 飞机飞行方向角调整的幅度不应超过 30 度;
- (3) 所有飞机飞行速度均为每小时 800km;
- (4) 进入该区域的飞机在到达区域边缘时, 与区域内飞机的距离应在 60km 以上;
- (5) 最多需考虑 6 架飞机;
- (6) 不必考虑飞机离开此区域后的状况.

请你对这个避免碰撞的飞行管理问题建立数学模型, 列出计算步骤, 对以下数据进行计算(方向角误差不超过 0.01 度), 要求飞机飞行方向角调整的幅度尽量小.

设该区域4个顶点的坐标为(0,0), (160,0), (160,160), (0,160). 记录数据为:

飞机编号	横坐标 x	纵坐标 y	方向角 (度)
1	150	140	243
2	85	85	236
3	150	155	220.5
4	145	50	159
5	130	150	230
新进入	0	0	52

注: 方向角指飞行方向与 x 轴正向的夹角.

试根据实际应用背景对你的模型进行评价与推广.

3. 模型求解

以各飞机调整的飞行角度平方和作为目标函数, 而以每两架飞机之间的最小距离不超过 8km, 各飞机飞行角度调整的值不超过 30° 为约束条件. 如此得出的是一个非线性模型. 以 t 表示时间; x_i 与 y_i 分别表示第 i 架飞机的横纵坐标 (问题中已经给出); θ_i 表示第 i 架飞机的飞行方向角 (问题中已经给出); $d_{ij}(t)$ 表示 t 时刻第 i 架飞机与第 j 架飞机间的距离; v 表示飞机的飞行速度 ($v = 800\text{km/h}$).

则目标函数为: $f = \sum_{i=1}^6 \Delta\theta_i^2$.

$$d_{ij}^2 = (x_i - x_j + vt(\cos(\theta_i + \Delta\theta_i) - \cos(\theta_j + \Delta\theta_j)))^2 \\ + (y_i - y_j + vt(\sin(\theta_i + \Delta\theta_i) - \sin(\theta_j + \Delta\theta_j)))^2$$

则约束条件为: $D_{ij} \triangleq \min_{t \geq 0} d_{ij}^2(t) > 64, i, j = 1, \dots, 6, i \neq j$.

$$\frac{dd_{ij}^2}{dt} = 0 \Rightarrow t = -\frac{a}{b}, \text{ 其中} \\ a = (x_i - x_j)(\cos(\theta_i + \Delta\theta_i) - \cos(\theta_j + \Delta\theta_j)) \\ + (y_i - y_j)(\sin(\theta_i + \Delta\theta_i) - \sin(\theta_j + \Delta\theta_j)) \\ b = v[(\cos(\theta_i + \Delta\theta_i) - \cos(\theta_j + \Delta\theta_j))^2 \\ + (\sin(\theta_i + \Delta\theta_i) - \sin(\theta_j + \Delta\theta_j))^2]$$

将 t 代入即可求出 D_{ij} . 于是本问题的一个数学模型为:

$$\min f = \sum_{i=1}^6 \Delta\theta_i^2, \\ s.t. \begin{cases} D_{ij} > 64 \\ |\Delta\theta_i| \leq \frac{\pi}{6} \\ i, j = 1, \dots, 6, i \neq j \end{cases}$$

引入记号:

$\Delta\theta = (\Delta\theta_1, \dots, \Delta\theta_6)$, $g = (g_1, \dots, g_{15})$ (g 是由 $64 - D_{ij}$ 按 $i = 1, \dots, 6, j = 1, \dots, 6, i \neq j$ 构成的

向量, 在下面的程序中计算), 则模型为:

$$\begin{aligned} \min f &= \Delta \theta^T \Delta \theta \\ \text{s.t.} \quad &\begin{cases} g < 0 \\ VLB \leq \Delta \theta \leq VUB \end{cases} \end{aligned}$$

其中 $VLB = -\frac{\pi}{6}(1,1,1,1,1,1)^T$, $VUB = \frac{\pi}{6}(1,1,1,1,1,1)^T$.

调用 MATLAB 命令 `fmincon` 求解, 先写两个 M 函数 `airfun.m` 和 `airfunco.m` 如下:

M 函数 airfun.m

```
function f=airfun(delta)           %目标函数 airfun.m
f=delta*delta';
```

M 函数 airfunco.m

```
function [c,ceq]=airfunco(delta)    %约束条件函数 airfunco.m
x0=[150 85 150 145 130 0];y0=[140,85,155,50,150,0];
alpha0=[243 236 220.5 159 230 52]*pi/180; v=800;
co=cos(alpha0+delta);si=sin(alpha0+delta);
for i=2:6
    for j=1:i-1
        t(i,j)=(x0(i)-x0(j))*(co(i)-co(j));
        t(i,j)=t(i,j)+(y0(i)-y0(j))*(si(i)-si(j));
        t(i,j)=-t(i,j)/v;
        t(i,j)=t(i,j)/((co(i)-co(j))^2+(si(i)-si(j))^2);
        if t(i,j)<0, d(i,j)=1000;
        else,
            d(i,j)=(x0(i)-x0(j)+v*t(i,j)*(co(i)-co(j)))^2;
            d(i,j)=(y0(i)-y0(j)+v*t(i,j)*(si(i)-si(j)))^2+d(i,j);
    end; end; end;
c=64-[d(2,1),d(3,1:2),d(4,1:3),d(5,1:4),d(6,1:5)]; ceq=[];
```

由于非线性规划求解对初值依赖性较大, 我们可在零点随机生成若干个初值计算来获取可能的最优解. 为此可编写一个 M 文件 `egl.m` 如下:

```
clear;
deltaini=[0,0,0,0,0,0];
vlb=-10*ones(1,6);vub=10*ones(1,6);
options = optimset('LargeScale','off');
n=10
for i=1:n-1
    deltaini=(rand(1,6)-0.5)*20;

    [dt,feval]=fmincon(@airfun,deltaini,[],[],[],[],vlb,vub,@airfunco,options);
    if feval<fval, fval=feval;d1=dt;
end;end;
d1=d1*180/pi,fval
```

在 MATLAB 命令窗口计算结果如下:

```
>>egl
n =
```

```

10          %十个初值
dl =          %最优解
-0.0000    0.0000    2.0624   -0.4954    0.0000    1.5670
fval =          %最优解函数值
6.9547

```

7.4.2 节约洗衣机用水问题

我国淡水资源有限，节约用水人人有责。洗衣机在家庭用水中占有很大份额，目前洗衣机已非常普及。节约洗衣机用水十分必要，假设再放入衣物和洗涤液后洗衣机的运行过程为：加水—洗涤—脱水—加水—洗涤—脱水……加水—洗涤—脱水（称“加水—洗涤—脱水”为运行一轮）。请为洗衣机设计一种程序（包括运行多少轮，每轮加水量等），使得在满足一定洗涤效果的前提下，总用水量最少。选用合理的数据进行计算。对照目前常用的洗衣机的运行情况，对你的模型和结果做出评价。

1. 洗衣机的基本原理和过程

洗衣机的基本原理就是将吸附在衣物上的污物溶于水，通过脱去污水而带走污物。“溶污物—脱污水”是由两个基本要素完成的一个“元动作”，无论是如何设计的洗衣方式和程序都是以此为基础的。洗衣的过程就是通过加水来实现上述动作的，使得残留在衣物上的污物越来越少，直到满意的程度。

通常洗衣要加入洗涤剂，它能帮助溶解污物。但是洗涤剂本身也是不能留在衣物上的东西。因此“污物”应是衣物上原有污物和洗涤剂的总和。有了这种认识以后，我们就可以把“洗涤”（即通过加洗涤剂的首轮洗衣）和“漂洗”（即以后各轮洗衣，不再加洗涤剂，但水中还有剩余洗涤剂）看作“溶污物”环节。

“脱污水”在洗衣中通常称为“脱水”，常由排水和甩干两个步骤组成。

2. “节水洗衣机”要点分析

- (1) 污物的溶解情况如何？我们将用溶解特性来刻画。
- (2) 每轮脱去污水后污物减少情况如何？这将有系统的动态方程表示。
- (3) 如何设计由一系列“溶污物—脱污水”的节水洗衣程序？这将通过用水程序反映。

3. 基本假设

- (1) 仅考虑“加水—溶污物—脱污水”（以下称为“加水—洗涤—脱水”）这 3 个环节，并认为它们是独立的，这 3 个环节构成一个洗衣周期，称为“一轮”。
- (2) 每轮用水量不能低于 L ，否则洗衣机无法转动；用水量不能高于 H ，否则会溢出。
- (3) 每轮的洗涤时间是足够的，以便衣物上的污物充分溶于水。

注：除首轮外，每轮的用水量实际上都包括该轮加水量和衣物上脱水后残留的水量，即残留水被自然地利用了。

4. 变量定义

- (1) 设共进行 n 轮“加水—溶污物—脱污水”的过程，依次为第 0 轮，第 1 轮，……第 $n-1$ 轮。
- (2) 第 k 轮用水量为 u_k ($k=1,2,\dots,n-1$)。
- (3) 衣物上的初始污物量为 x_0 ，在第 1 轮脱水之后仍吸附在衣物上的污物量为 x_1 ，所以

在第 k 轮脱水后仍吸附在衣物上的污物量为 x_k ($k=0,1,2,\cdots,n-1$).

5. 溶解特性和动态方程

第 1 轮加水洗涤之后脱水之前, x_1 可以用两部分表示

$$x_1 = x_0 - p_0$$

其中 p_0 表示已溶入水中的污物, p_0 与第 1 轮的加水量有关, 总的规律是 u_1 越大 p_0 越大, 当 $u_1 \rightarrow H$ 时, $p_0 = Qx_0$ (Q 为溶解率), 此时 p_0 最大, 当 $u_1 \rightarrow L$ 时, 洗衣机处于转动的临界状态, 此时 $p_0 = 0$ 最小. 由此得 p_0 的关系式:

$$p_0 = Qx_0 \left(\frac{u_1 - L}{H - L} \right)$$

所以

$$x_1 = x_0 - Qx_0 \left(\frac{u_1 - L}{H - L} \right)$$

那么同理第 k 轮脱水后, 系统的动态方程为

$$x_k = x_{k-1} \left(1 - Q \left(\frac{u_k - L}{H - L} \right) \right)$$

6. 优化模型

由于 x_0 是衣物上的初始污物量, x_n 是洗衣过程结束后衣物上最终残留的污物量, 所以 $\frac{x_n}{x_0}$

反映了洗涤效果, 由动态方程得

$$\frac{x_n}{x_0} = \prod_{k=1}^n \left(1 - Q \left(\frac{u_k - L}{H - L} \right) \right)$$

又总用水量为

$$\sum_{k=1}^n u_k$$

于是可得优化模型如下

$$\begin{aligned} \min & \sum_{k=1}^n u_k \\ \text{s.t.} & \prod_{k=1}^n \left[1 - Q \left(\frac{u_k - L}{H - L} \right) \right] \leq \varepsilon, 0 < \varepsilon < 1 \end{aligned}$$

7.5 习题

1. 某工厂制造产品 I, 6 小时可加工 100 箱, 若制造 II, 5 小时可加工 100 箱, 设产品 I 和产品 II 占用生产场地分别是 10 和 20 个体积单位, 而生产场地允许 15000 个体积单位的存储量, 若加工产品的机器每周工时数不超过 60 小时, 产品 I 生产 x_1 百箱的收益为 $(60 - 5x_1)x_1$ 元, 产品 II 生产 x_2 百箱的收益为 $(80 - 4x_2)x_2$ 元, 又由于收购部门的限制, 产品 I 和 II 的生产量每周都不能超过 800 箱, 试制定最优的周生产计划, 使产品收益最大.

2. 设有 400 万元资金, 要求 4 年内使用完, 若在一年内使用资金 x 万元, 则可得效益 $\sqrt{x}$

万元（效益不能再使用），当年不用的资金可存入银行，年利率为 10%。试制定出资金的使用计划，以使 4 年效益之和为最大。

3. 对边长为 3 米的正方形铁板，在四个角剪去相等的正方形以制成方形无盖水槽，问如何剪法使水槽的容积最大？

4. 用外点法求下列非线性规划

$$\begin{aligned} \min f(x) &= x_1^2 + x_2^2 + 8 \\ s.t. \begin{cases} x_1^2 - x_2 \geq 0 \\ -x_1 - x_2^2 + 2 = 0 \\ x_1, x_2 \geq 0 \end{cases} \end{aligned}$$

5. 用内点法求下列非线性规划

$$\begin{aligned} \min f(x) &= x_1 + x_2 \\ s.t. \begin{cases} -x_1^2 + x_2 \geq 0 \\ x_1 \geq 0 \end{cases} \end{aligned}$$