6.1 最优化理论

最优化理论是关于系统的最优设计、最优控制、最优管理问题的理论与方法。最优化，就是在一定的约束条件下，使系统具有所期待的最优功能的组织过程，是从众多可能的选择中作出最优选择，使系统的目标函数在约束条件下达到最大或最小。最优化是系统方法的基本目的。

优化方法有几个基本因素：系统目标;实现目标的可能方案;实行各方案的支付代价;建立系统模型;制定系统评价标准等。

6.2 最优化问题的数学描述

最优化的基本数学模型如下：

$$
\begin{aligned}
\min \quad & f(\mathbf{x}) \\
\text{s.t.} \quad & h_i(\mathbf{x}) = 0 \\
& g_j(\mathbf{x}) \leqslant 0
\end{aligned}
$$

它有三个基本要素，即：

```
* 设置变量：x是一个实数域范围内的n维向量，被称为决策变量或问题的解
* 目标函数：f(x)为目标函数；
* 约束条件：hi(x)=0称为等式约束，gi(x)≤0为不等式约束，i=0,1,2,......
```

最优化理论的python实现

在 x,y,z>0，xyz=1 x,y,z>0，xyz=1x,y,z>0，xyz=1 的条件下，求解 (x−2/3)/(x+y+z−2) (x-2/3)/(x+y+z-2) (x−2/3)/(x+y+z−2) 的最小值。

from scipy.optimize import minimize import numpy as np e = 1e-10 # 非常接近0的值 fun = lambda x : (x[0] - 0.667) / (x[0] + x[1] + x[2] - 2) # 约束函数 cons = ({'type': 'eq', 'fun': lambda x: x[0] * x[1] * x[2] - 1}, # xyz=1 {'type': 'ineq', 'fun': lambda x: x[0] - e}, # x>=e，即 x > 0 {'type': 'ineq', 'fun': lambda x: x[1] - e}, {'type': 'ineq', 'fun': lambda x: x[2] - e}) x0 = np.array((1.0, 1.0, 1.0)) # 设置初始值 res = minimize(fun, x0, method='SLSQP', constraints=cons) print('最小值：',res.fun) print('最优解：',res.x) print('迭代终止是否成功：', res.success) print('迭代终止原因：', res.message) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 输出：

最小值： -0.18814357989751096 最优解： [0.29250894 1.84897232 1.84897233] 迭代终止是否成功： True 迭代终止原因： Optimization terminated successfully.