非线性规划

Liam

2023 年 7 月 23 日

1 非线性规划模型

如果目标函数或者约束条件包含非线性函数，则称为非线性规划问题。

1.1 无约束极值问题

1.1.1 无约束极值问题的数值解

使用命令：

`[x,fval]=fminunc(fun,x0,options)`

`[x,fval]=fminsearch(fun,x0,options)`

1.2 约束极值问题

带约束条件的极值问题成为约束极值问题。一般都是将非线性化简为**线性问题**，约束问题化简为**无约束问题**。

1.2.1 二次规划

如果某个非线性规划的目标函数为子标量的二次函数，约束为线性。

$$\min \quad \frac{1}{2}x^T H x + f^T x,$$
$$\text{s.t.} \quad \begin{cases} \boldsymbol{A}\boldsymbol{x} \le b, \\ \boldsymbol{A}eq * \boldsymbol{x} = beq, \\ lb \le \boldsymbol{x} \le ub. \end{cases}$$

求解二次规划的命令为

`[x,fval] = quadprog(H,f,A,b,Aeq,beq,lb,ub,x0,options)`

1.2.2 罚函数法

罚函数法又称乘子法，是指将有约束最优化问题转化为求解无约束最优化问题：其中 σ 为足够大的正数，起" 惩罚" 作用，称之为罚因子，$F(x,\sigma)$ 称为罚函数。内部罚函数法也称为障碍罚函数法。

外罚函数法：适用于含等式或者不等式约束的情形。思想是，在目标函数上加上惩罚项，该惩罚项由约束函数构造，在可行域中**不予惩罚**，但是在不可行域中**施加惩罚**，从而使目标函数在不可行域中的值变得很大，此时可以去掉约束项，看成一个**无约束优化问题**。如果最优解出现在可行域的边界上，随着惩罚力度的加大，无约束问题的最优解会在可行域外部逼近原问题最优解。外罚函数法的主要缺点：

1. **近似的最优解往往不是可行解**。从上面的例子很容易看出，最优解如果在边界上，那么近似的最优解是从可行域外部逼近的，这在一些实际应用场合是不能接受的（函数边界外面没有定义）.

2. **惩罚力度往往需要趋于无穷才能让近似的最优解逼近真正的最优解，但随着的增大，海森阵容易趋于病态**。尤其在用数值迭代的过程中造成不稳定的现象（所谓的梯度悬崖），甚至是无法求解。

内罚函数法：仅适用于含有不等式的情形，也在目标函数上加上惩罚项（障碍项），障碍项作用在可行域内部，特别是在靠近边界时候目标函数 $\to \infty$，这相当于在可行域边界上筑起一道高墙，让迭代点跳不出去，此时可以去掉约束项，看成一个无约束优化问题。若最优值出现在可行域的边界上，由于障碍项会影响可行域内部的值，所以要减少障碍力度，让其尽量不影响原目标函数在可行域内部形成的值，但同时还要在边界上保持高墙，当障碍力度充分小的时候，无约束最优化问题会在可行域内部逼近原问题最优解。

内罚函数的优点：

1. 每个近似最优解都是可行解（因为迭代点始终处于可行域内部）

缺点:

1. 仅适用于不等式约束

2. 寻找初始可行点可能不容易

3. 障碍因子 r 不断减小也会导致海森阵趋于病态（梯度悬崖）
4. 在数值求解过程中造成很大麻烦

如果非线性规划问题要求实时算法，则可以使用罚函数方法，但计算精度低。

1.2.3 Matlab 求解约束极值问题

$$\begin{cases} \text{fminbnd(fun,x1,x2,options) 单变量非线性区间}[x_1, x_2] \\ \text{fmincon} \\ \text{quadprog} \\ \text{fseminf} \\ \text{fminimax} \end{cases}$$

```
[x,fval]=fseminf(fun,x0,ntheta,seminfcon,A,b,Aeq,beq,lb,ub)
```

1.2.4 飞行管理问题