

《运筹学讲义》

杨周旺

中国科学技术大学
数学科学学院

2021年9月

- A. Ravindran, D. Phillips, J. Solberg, "Operations Research: Principles and Practice". 2nd Ed. John Wiley, 1987.
- Hamdy A. Taha, "Operations Research: An Introduction". 8th Ed. Pearson Prentice Hall, 2007.
- Jorge Nocedal, Stephen J. Wright, "Numerical Optimization". Second Edition, Springer, 2006.
- Frederick S. Hillier, "Introduction to Operations Research". McGraw-Hill Education, 2012.
- Wikipedia (the free encyclopedia), "Operations research".
https://en.wikipedia.org/wiki/Operations_research
- ...

课程考核

- (1) 课堂测试 (10%)
- (2) 课后习题 (20%)
- (3) 课程作业 (20%)
- (4) 期末考试 (50%)

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

Outline

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

运筹学 (Operations Research, OR)是从二十世纪三四十年代（即二战期间）发展起来的一门应用性很强的学科。它的研究对象是人类对各种资源的运用及筹划活动，如战争、后勤、生产规划、经营管理、资本运作、工程设计等。研究内容就是资源筹划活动中各种问题的模型化及其数学方法。

Mathematical programming

Network optimization

Decision analysis

Queuing theory

Game theory

Industrial engineering

Logistics

Supply chain management

Financial engineering

...

解决实际问题的第一步往往是定性分析，具体的说

- 要有一个初步概念，即主要的决策是什么？
- 在选取决策方案时，有效性的度量是什么？
- 在对各方案进行比较时，这些度量之间可能要做哪些权衡？

.....

定量分析对管理决策过程有着重要意义...

定量分析方法的标准步骤

- 表达问题 (Definition of the problem)
- 建立模型 (Construction of the model)
- 分析求解 (Solution of the model)
- 检验和改进 (Validation of the model)
- 解的实施 (Implementation of the solution)

表达问题：列出问题的各种要素，它们包括一些可控的变量（或称决策变量），不可控变量（参数），各变量的约束条件以及为确定最佳解决方案拟采用的目标度量。

建立模型

建立模型：即把问题中的可控变量，参数，约束及目标的关系用一定的数学模型刻画出来。

分析求解：分析模型并用各种数学方法和手段来求解模型，进而确定解对模型的技术条件的灵敏度。

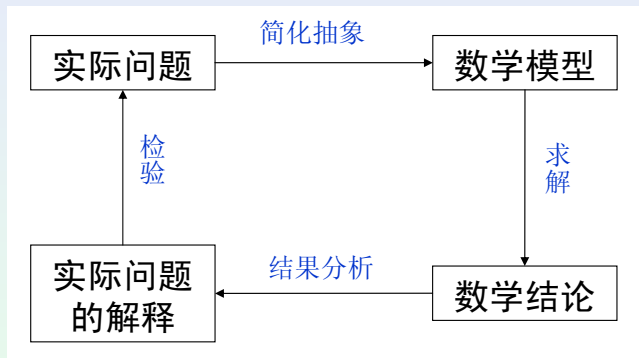
检验和改进

检验和改进：将模型的解应用到实际问题中，检验解的合理性和有效性，可能产生的问题和修改模型。

解的实施

解的实施：模型的解应用于实际问题，并最终解决实际问题。

建模过程



建模的重要性

建立模型：实际问题 \longrightarrow 反映实际问题的数学模型（如：微分方程模型，统计模型，最优化模型等）。

各种优化模型及其求解的数学方法构成了运筹学的大部分内容，这些代表性模型可供选择但并不能完全适用于所有实际问题。

一般的运筹学参考书不会着重叙述建立模型的过程，这并不是说建模不重要；相反，建模在任何时候任何场合都是极其重要的。

在很多实际应用问题中，从数学上看都是非适定的(ill-posed)，即解不唯一。对于这样的实际问题，人们往往通过制定相应的目标准则，然后从众多的解中选出在一定条件下最好的解。

这些正是最优化理论与方法所研究的内容，本节对最优化的基本概念作一些简要的介绍，并给出最优化建模方法的相关知识。

最优化讨论的是为找出实际问题的最优解决策略而采取的模型化及方法，其过程是：

- 先把待解决的问题用最优化形式描述为在给定的约束条件下找出使某个目标函数达到最大（小）的解；
- 然后再采用数学上严密的算法来求解。

在这里我们强调最优化方法作为数学工具在现实中被广泛应用的事实，大多数有代表性的最优化算法都已有程序库和软件包。但是，有效利用这些成果是以有待解决的问题已被模型化成最优化问题的形式为前提的，而这一过程并非简单。

这个模型化的过程要求：在收集客观数据的同时要有敏锐的洞察力和综合能力；兼蓄有针对性的领域知识和技能。

最优化方法在生产规划、经济管理、工业工程、交通运输、国防等重要领域中有广泛的应用，并已受到政府部门、科研机构 and 产业界的高度重视。

例如：选址问题，是运筹学和计算几何的一个交叉分支。

选址问题关注的是设施的最佳位置以减少诸多社会成本（如交通、时间等），同时要考虑类似“避免扰民或居民区附近放置危险物品”等因素。

很多机器学习任务本质上是（可转化为）最优化模型：

- Supervised learning (Regression, Classification)
- Unsupervised learning (Clustering, K-means, PCA)
- Deep learning
- Reinforcement learning
- Collaborative filtering
- Nonparametric Bayesian inference
- Social activity modeling for networks
-

“Optimization lies at the heart of machine learning.”

很多机器学习任务本质上是（可转化为）最优化模型：

- Supervised learning (Regression, Classification)
- Unsupervised learning (Clustering, K-means, PCA)
- Deep learning
- Reinforcement learning
- Collaborative filtering
- Nonparametric Bayesian inference
- Social activity modeling for networks
-

“Optimization lies at the heart of machine learning.”

最优化(optimization)一般是指在某种条件下作出最好的决策，或者是从众多的方案中选出最好的。这种问题经常用下面的数学模型描述：

在给定的约束条件(*constraint*)下，找出决策变量(*decision variable*)的一个值，使得被称为目标函数(*objective function*)的表达愿望尺度的函数值达到最小或最大。

最优化模型

一般说来决策变量有多个，因此用 n 维向量 $x = (x_1, \dots, x_n)^T$ 来表示，于是把问题写成如下形式

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in S \subset \mathbb{R}^n. \end{array} \quad (1)$$

在此，目标函数 f 是定义在包含 S 的适当集合上的实值函数。进一步， S 是该问题变量 x 的可取值之集合，称为问题的可行域(feasible region).

最优化问题的分类

- 如果可行域 $S = \mathbb{R}^n$, 则称为无约束最优化问题:

$$\min_{x \in \mathbb{R}^n} f(x) \quad (2)$$

- 带约束最优化问题通常可写为:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0, \quad i \in \mathcal{E}, \\ & c_i(x) \geq 0, \quad i \in \mathcal{I}. \end{aligned} \quad (3)$$

这里 $c_i(x)$ 是约束函数, \mathcal{E} 和 \mathcal{I} 分别是等式约束的指标集和不等式约束的指标集。

最优化问题的分类

- 当目标函数和约束函数均为线性函数时，问题称为线性规划(Linear Programming);
- 当目标函数与约束函数中至少有一个是变量 x 的非线性函数时，问题称为非线性规划(Nonlinear Programming)。
- 此外，根据决策变量、目标函数和要求的不同，最优化还分为整数规划、动态规划、网络规划、非光滑规划、随机规划、多目标规划等。

最优化问题的解

满足约束条件 $x \in S$ 的 x 称为问题的可行解(feasible solution), 如果可行解 $x^* \in S$ 进一步满足

$$f(x^*) \leq f(x), \forall x \in S. \quad (4)$$

则称 x^* 为问题(1)的全局最优解(global optimal solution). 另外, 在包含可行解 $x^* \in S$ 的适当邻域 $U(x^*)$ 里, 成立

$$f(x^*) \leq f(x), \forall x \in S \cap U(x^*). \quad (5)$$

此时称 x^* 为问题(1)的局部最优解(local optimal solution).

不少问题的目标函数或约束条件可能很复杂, 要找出全局最优解非常困难, 这时我们的目标就会是求出局部最优解。

最优性条件

最优性条件：问题的最优解所满足的必要或者充分条件。它为最优化问题求解算法的设计、分析提供必不可少的理论基础。

无约束问题的极值条件

一阶必要条件：设目标函数 $f(x)$ 在点 \bar{x} 处可微，若 \bar{x} 是局部极小点，则 $\nabla f(\bar{x}) = 0$ 。

二阶必要条件：设目标函数 $f(x)$ 在点 \bar{x} 处二次可微，若 \bar{x} 是局部极小点，则 $\nabla f(\bar{x}) = 0$ ，并且Hesse矩阵 $\nabla^2 f(\bar{x}) \geq 0$ 。

二阶充分条件：设目标函数 $f(x)$ 在点 \bar{x} 处二次可微，若 $\nabla f(\bar{x}) = 0$ 且 $\nabla^2 f(\bar{x}) > 0$ ，则 \bar{x} 是局部极小点。

充要条件：设 $f(x)$ 是定义在 \mathbb{R}^n 上的可微凸函数，则 \bar{x} 是整体极小点（全局最优解）的充要条件是 $\nabla f(\bar{x}) = 0$ 。

最优性条件

约束问题的最优性条件

Kuhn-Tucker必要条件：设 \bar{x} 为约束问题(3)的可行点，记 $\mathcal{I}_e = \{i \in \mathcal{I} \mid c_i(\bar{x}) = 0\}$ ， f 和 $c_i(i \in \mathcal{I}_e)$ 在点 \bar{x} 可微， $c_i(i \in \mathcal{I} \setminus \mathcal{I}_e)$ 在点 \bar{x} 连续， $c_i(i \in \mathcal{E})$ 在点 \bar{x} 连续可微，向量集 $\{\nabla c_i(\bar{x}) \mid i \in \mathcal{E} \cup \mathcal{I}_e\}$ 线性无关。如果 \bar{x} 是局部最优解，则存在数 $\lambda_i \geq 0(i \in \mathcal{I}_e)$ 和 $\mu_i(i \in \mathcal{E})$ 使得

$$\nabla f(\bar{x}) - \sum_{i \in \mathcal{I}_e} \lambda_i \nabla c_i(\bar{x}) - \sum_{i \in \mathcal{E}} \mu_i \nabla c_i(\bar{x}) = 0. \quad (6)$$

最优性条件

定义Lagrange函数 $L(x, \lambda, \mu) = f(x) - \sum_{i \in \mathcal{I}} \lambda_i c_i(x) - \sum_{i \in \mathcal{E}} \mu_i c_i(x)$. 若 \bar{x} 为问题局部最优解, 则存在乘子向量 $\bar{\lambda} \geq 0, \bar{\mu}$ 使得 $\nabla_x L(\bar{x}, \bar{\lambda}, \bar{\mu}) = 0$.

此时, 一阶必要条件可以表达为

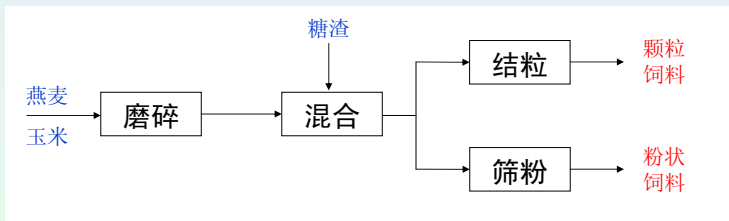
$$(K - T) \begin{cases} \nabla_x L(x, \lambda, \mu) = 0, \\ c_i(x) = 0, i \in \mathcal{E}, \\ c_i(x) \geq 0, i \in \mathcal{I}, \\ \lambda_i c_i(x) = 0, i \in \mathcal{I}, \\ \lambda_i \geq 0, i \in \mathcal{I}. \end{cases} \quad (7)$$

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

$$\begin{array}{ll}\min & c^T x \\ \text{s.t.} & A_e x = b_e \\ & Ax \leq b.\end{array} \quad (8)$$

例子：饲料配制

某饲料公司要生产两种类型的动物饲料：粉状饲料和颗粒饲料。生产这些饲料所需的原料由燕麦、玉米和糖渣组成。首先需要将这些原料（除糖渣外）磨碎，然后将所有原料混合形成饲料半成品。在最后一个生产工序中，将半成品制成颗粒状或粉末状，从而得到最终的饲料产品。



例子：饲料配制

饲料产品都要满足一些营养成分需求。表1列出了各种原料的蛋白质，脂肪和纤维含量百分比，以及最终产品的相应含量要求。

Table: 营养成分含量百分比

原料	蛋白质	脂肪	纤维素
燕麦	13.6	7.1	7.0
玉米	4.1	2.4	3.7
糖渣	5.0	0.3	25.0
含量要求	≥ 9.5	≥ 2.0	≤ 6.0

例子：饲料配制

各种原料的可用量也有限制。表2给出了各种原料的可用量以及对应的价格。

Table: 原料可用量与价格

原料	可用量（千克）	价格（元/千克）
燕麦	11900	1.3
玉米	23500	1.7
糖渣	750	1.2

例子：饲料配制

表3列出了各道工序的成本。

Table: 加工成本

磨碎	混合	结粒	筛粉
2.5	0.5	4.2	1.7

如果每天需求量为9吨颗粒饲料，12吨粉状饲料，则各种原料应分别使用多少进行混合才能使得总成本最低？

模型表达

在这里，燕麦、玉米、糖渣的使用量为决策变量，并分别设为 x_1 千克， x_2 千克， x_3 千克。

饲料配制最优化模型：

$$\begin{aligned} \min \quad & (1.3x_1 + 1.7x_2 + 1.2x_3) + 2.5(x_1 + x_2) + 0.5(x_1 + x_2 + x_3) \\ \text{s.t.} \quad & 13.6x_1 + 4.1x_2 + 5.0x_3 \geq 9.5(x_1 + x_2 + x_3), \\ & 7.1x_1 + 2.4x_2 + 0.3x_3 \geq 2.0(x_1 + x_2 + x_3), \\ & 7.0x_1 + 3.7x_2 + 25.0x_3 \leq 6.0(x_1 + x_2 + x_3), \\ & x_1 + x_2 + x_3 \geq 9000 + 12000, \\ & 0 \leq x_1 \leq 11900, \\ & 0 \leq x_2 \leq 23500, \\ & 0 \leq x_3 \leq 750. \end{aligned} \tag{9}$$

Solve a linear programming problem

$$\begin{array}{ll}\min_{x} & f^T x \\ \text{s.t.} & A x \leq b \\ & A_{eq} x = b_{eq} \\ & lb \leq x \leq ub\end{array}\quad (10)$$

where f , x , b , b_{eq} , lb , and ub are vectors and A and A_{eq} are matrices.

$[x, fval, exitflag, output, lambda] = \text{linprog}(f, A, b, A_{eq}, b_{eq}, lb, ub, x_0, options)$

```
f = [4.3; 4.7; 1.7];  
A = [-4.1, 5.4, 4.5; -5.1, -0.4, 1.7; 1, -2.3, 19; -1, -1, -1];  
b = [0; 0; 0; -21000];  
lb = zeros(3, 1);  
ub = [11900; 23500; 750];
```

```
[x,fval,exitflag,output,lambda] = linprog(f,A,b,[],[],lb,ub);
```

生产出需求的9吨颗粒饲料和12吨粉状饲料，最低成本为150868元。详细的结果如下所示：

Table: 饲料配制的最佳方案

原料	燕麦（千克）	玉米（千克）	糖渣（千克）
使用量	11897	8678.9	424.47

例子：生产计划

某农场在现有土地上种植不同农作物的生产计划安排

定性分析：合理分配土地资源，在约束条件及一定假设下获得好收成。

Table: 农作物生产相关数据

农作物	土地	用工	粪肥	化肥	利润
A	1	450	35	350	1500
B	1	600	25	400	1200
C	1	900	30	300	1300
总资源	100	63000	3300	33000	

例子：生产计划

分析并建立模型

设农作物A, B, C的种植面积分别为 x_1, x_2, x_3 公顷，则总利润是

$$1500x_1 + 1200x_2 + 1800x_3$$

资源约束：

土地限制 $x_1 + x_2 + x_3 \leq 100$,

劳力限制 $450x_1 + 600x_2 + 900x_3 \leq 63000$,

粪肥限制 $35x_1 + 25x_2 + 30x_3 \leq 3300$,

化肥限制 $350x_1 + 400x_2 + 300x_3 \leq 33000$,

此外，实际种植面积的非负性限制，即 $x_1, x_2, x_3 \geq 0$.

例子：生产计划

综上所述，得到农业生产计划安排问题的数学模型（线性规划）为：

$$\begin{aligned} \max \quad & 1500x_1 + 1200x_2 + 1800x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 \leq 100, \\ & 450x_1 + 600x_2 + 900x_3 \leq 63000, \\ & 35x_1 + 25x_2 + 30x_3 \leq 3300, \\ & 350x_1 + 400x_2 + 300x_3 \leq 33000, \\ & x_1, x_2, x_3 \geq 0. \end{aligned} \tag{11}$$

线性规划模型及其标准化

线性规划的一般形式：

$$\begin{array}{ll} \min(\max) & z = c_1x_1 + \cdots + c_nx_n \\ \text{s.t.} & a_{11}x_1 + \cdots + a_{1n}x_n \leq (=, \geq) b_1 \\ & \vdots \\ & a_{m1}x_1 + \cdots + a_{mn}x_n \leq (=, \geq) b_m \end{array} \quad (12)$$

线性规划问题总可以写成如下标准形式：

$$\begin{array}{ll} \min & \sum_{j=1}^n c_j x_j \\ \text{(LP)} \quad \text{s.t.} & \sum_{j=1}^n a_{ij} x_j = b_i, i = 1, \dots, m \\ & x_j \geq 0, j = 1, \dots, n. \end{array} \quad (13)$$

线性规划标准型

或者用矩阵表示为：

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0. \end{array} \quad (14)$$

其中矩阵 $A \in \mathbb{R}^{m \times n}$, c 是 n 维列向量, b 是 m 维列向量。为了后面单纯形方法的计算需要, 我们还可以假设 $b \geq 0$.

标准化步骤

可能的标准化步骤有：

- 目标函数 $\max f(x) \longrightarrow \min -f(x)$
- 不等式约束的等式化（引入松弛变量或者剩余变量）
- 自由变量的非负化 $x_j = x'_j - x''_j, x'_j, x''_j \geq 0$
- 变量上下界的处理

线性规划的基本理论

结论1: 在线性规划中, 约束条件均为线性等式及线性不等式, 所以可行域 S 是凸集。

线性规划的基本理论

极点： 设 S 是非空凸集，若 $x = \lambda x^{(1)} + (1 - \lambda)x^{(2)}$, $\lambda \in (0, 1)$, $x^{(1)}, x^{(2)} \in S$, 必有 $x = x^{(1)} = x^{(2)}$, 则称 x 是凸集 S 的极点。

结论2： 对于有界闭凸集，任何一点都能表示成极点的凸组合。

线性规划的基本理论

结论2对无界集并不成立，为此需引入极方向的概念。

设 $S \subset \mathbb{R}^n$ 是闭凸集（可能无界）， d 为非零向量。如果对 S 中的每个 x 都有射线 $\{x + \lambda d \mid \lambda \geq 0\} \subset S$ ，则称向量 d 为 S 的方向。

又设 $d^{(1)}$ 和 $d^{(2)}$ 是 S 的两个方向，若对任何正数 λ 有 $d^{(1)} \neq \lambda d^{(2)}$ ，则称 $d^{(1)}$ 和 $d^{(2)}$ 是两个不同的方向。

极方向： 若 S 的方向 d 不能表示成该集合的两个不同方向的正的线性组合，则称 d 为 S 的极方向

线性规划的基本理论

表示定理： 设 $S = \{x \mid Ax = b, x \geq 0\}$ 为非空多面集，则有

- (1) 极点集非空，且存在有限个极点 $x^{(1)}, \dots, x^{(k)}$;
- (2) 极方向集合为空集的充要条件是 S 有界；
- (3) 若 S 无界，则存在有限个极方向 $d^{(1)}, \dots, d^{(\ell)}$;
- (4) $x \in S$ 的充要条件是

$$x = \sum_{i=1}^k \lambda_i x^{(i)} + \sum_{j=1}^{\ell} \mu_j d^{(j)}$$

其中 $\lambda_i \geq 0, i = 1, \dots, k, \sum_{i=1}^k \lambda_i = 1, \mu_j \geq 0, j = 1, \dots, \ell$.

[习题1.1：证明上述凸集表示定理...]

线性规划的基本理论

考虑标准形式

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0. \end{array} \quad (\text{LP}) \quad (15)$$

设可行域 $\mathcal{S} = \{x \mid Ax = b, x \geq 0\}$ 的极点为 $x^{(1)}, \dots, x^{(k)}$, 极方向为 $d^{(1)}, \dots, d^{(\ell)}$. 根据表示定理, 任何可行点 x 可表示为

$$x = \sum_{i=1}^k \lambda_i x^{(i)} + \sum_{j=1}^{\ell} \mu_j d^{(j)}.$$

线性规划的基本理论

将其代入标准型中得到:

$$\begin{aligned} \min \quad & c^T x = \sum_{i=1}^k \lambda_i (c^T x^{(i)}) + \sum_{j=1}^{\ell} \mu_j (c^T d^{(j)}) \\ \text{s.t.} \quad & \lambda_i \geq 0, i = 1, \dots, k, \sum_{i=1}^k \lambda_i = 1, \mu_j \geq 0, j = 1, \dots, \ell. \end{aligned} \tag{16}$$

线性规划的基本理论

最优解的约定

可行域为空 \implies 无解

可行域有界 \implies 唯一解 或者 无穷多解

可行域无界 \implies 唯一解 或者 无穷多解 或者 $-\infty$

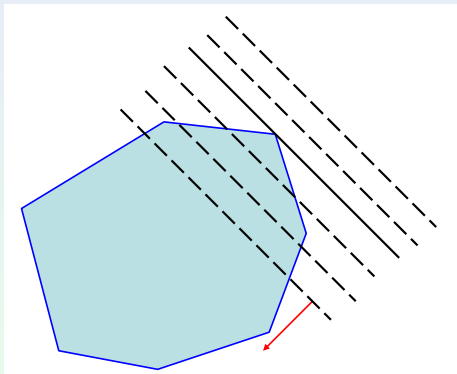
我们把“唯一解”和“无穷多解”称为模型存在最优解，而把“无界解 $-\infty$ ”归入不存在最优解的情形。

线性规划的基本理论

结论3: 标准形式线性规划模型的可行域非空, 存在有界最优解当且仅当所有 $c^T d^{(j)}$ 为非负数 (其中 $d^{(j)}$ 为可行域的极方向)。进一步有, 若存在有界最优解, 则目标函数的最优值一定能在某个极点 (最优极点) 处达到。

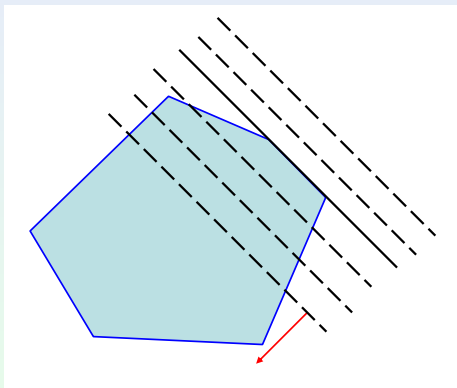
线性规划的基本理论

线性规划的直观图解



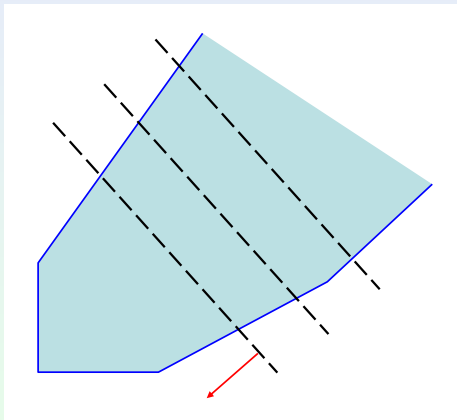
线性规划的基本理论

线性规划的直观图解



线性规划的基本理论

线性规划的直观图解



线性规划的基本理论

设线性规划标准形式

$$\begin{array}{ll} \text{(LP)} & \min \quad c^T x \\ & \text{s.t.} \quad Ax = b \\ & \quad x \geq 0. \end{array} \quad (17)$$

假设 $A = (B, N)$, 其中 B 是 m 阶可逆矩阵 (不失一般性)。同时记 $x = (x_B^T, x_N^T)^T$, 其中 x_B 的分量与 B 中的列对应, x_N 的分量与 N 中的列对应。

线性规划的基本理论

这样 $Ax = b$ 即可写成

$$(B, N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b,$$

即 $Bx_B + Nx_N = b \implies x_B = B^{-1}b - B^{-1}Nx_N$.

线性规划的基本理论

基解/基矩阵:

在上式中, x_N 的分量就是线性方程组 $Ax = b$ 的自由变量。特别地令 $x_N = 0$, 则得到解

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$$

称为方程组的一个基解, 对应的 B 称为基矩阵。

x_B 的各分量称为基变量, x_N 的各分量称为非基变量。

线性规划的基本理论

可行基解/可行基矩阵:

又若 $B^{-1}b \geq 0$, 则称 $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ 为(LP)的可行基解, 相应的称 B 为可行基矩阵, $x_B = \begin{pmatrix} x_{B_1} \\ \vdots \\ x_{B_m} \end{pmatrix}$ 为一组可行基变量。

线性规划的基本理论

例子:

$$\begin{array}{ll}\min & -x_1 + 3x_2 \\ \text{s.t.} & x_1 + 2x_2 \leq 8 \\ & x_2 \leq 2 \\ & x_1, x_2 \geq 0.\end{array}\tag{18}$$

[习题1.2: 试给出上述例子的所有极点和可行基解...]

线性规划的基本理论

极点集 -vs- 可行基解集

线性规划模型中，可行域的极点与可行基解之间是等价对应的。

定理： 设线性规划标准型(LP)的可行域 $S = \{x \mid Ax = b, x \geq 0\}$, 且 $\text{rank}(A_{m \times n}) = m$, 则可行域 S 的极点集与(LP)的可行基解集是等价的。

[习题1.3: 证明极点集-可行基解集等价定理...]

线性规划的基本理论

由**凸集表示定理**和**极点-可行基解等价定理**，我们易得：

当线性规划问题存在最优解时，目标函数的最优值一定能在可行域的某个极点处达到，即(LP)存在最优解时，则一定存在一个可行基解是最优解。

这样，线性规划模型的求解（最优解）归结为求最优可行基解。这一思想正是单纯形方法的基本出发点。但可行基解的个数往往很多，不宜一一枚举。该采取何种策略？而这正是单纯形算法的实质...

线性规划的基本理论

另外，我们在前面给出了**凸集表示定理**，其结论之一是“极点的存在有限性”，在这里将以另一种方式提出。

可行基解存在性定理： (LP)有可行解，则一定存在可行基解。

基本思想： 从一个可行基解出发，求下一个使目标函数值有所改善的可行基解；通过不断迭代改进可行基解力图达到最优可行基解。

主要步骤：

- 最优判定 (optimality)
- 转轴运算 (pivoting)

首先假定对线性规划标准形问题

$$\{\min c^T x \mid Ax = b, x \geq 0\}$$

已得到一个可行基的划分 $A = (B, N)$, $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$, $c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}$.

那么由 $x_B = B^{-1}b - B^{-1}Nx_N$, 目标函数值可表示为

$$\begin{aligned} z &= c_B^T x_B + c_N^T x_N \\ &= c_B^T B^{-1}b + (c_N^T - c_B^T B^{-1}N)x_N \\ &= z_0 + \sum_{j \in N} (c_j - z_j)x_j, \end{aligned} \tag{19}$$

这里,

$$z_0 = c_B^T B^{-1}b, \tag{20}$$

$$z_j = c_B^T B^{-1}a_j, \tag{21}$$

a_j 表示矩阵 A 的第 j 列向量。

对应于某个划分 $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$, 若由式(21)所决定的 z_j 满足 (最优判定条件)

$$c_j - z_j \geq 0, \forall j \in N. \quad (22)$$

记任意可行解 x (满足 $x_j \geq 0, j \in N$) 的目标函数值为 z , 则由(19)得

$$z = z_0 + \sum_{j \in N} (c_j - z_j) x_j \geq z_0 = c_B^T B^{-1} b.$$

即此时的可行基解 $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ 所对应目标函数值 z_0 不比任何一个可行解的目标函数值 z 大, 所以 $x = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}$ 即为问题的最优解。

反之，若式(22)不成立时，即存在某个 $p \in N$ 使得

$$c_p - z_p < 0. \quad (23)$$

因此把 x_p 以外的非基变量固定为现在的0值，将 x_p 值由0增加到 $\Delta (\geq 0)$ ，目标函数值 z 相应地可由 z_0 减至 $z_0 + (c_p - z_p)\Delta (\leq z_0)$ 。

单纯形法

为了保证解的可行性，有必要进一步分析 Δ 的取值及基变量 x_B 的变化情况。

若取 $x_p = \Delta$, $x_j = 0 (j \in N, j \neq p)$, 则由线性方程组 $Ax = b$ 得 x_B 的值为

$$x_B = B^{-1}b - B^{-1}a_p\Delta.$$

记 $\bar{b} = B^{-1}b$, $y_p = B^{-1}a_p$, 上式可写成

$$\begin{pmatrix} x_{B_1} \\ x_{B_2} \\ \vdots \\ x_{B_m} \end{pmatrix} = \begin{pmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \vdots \\ \bar{b}_m \end{pmatrix} - \begin{pmatrix} y_{1p} \\ y_{2p} \\ \vdots \\ y_{mp} \end{pmatrix} \Delta. \quad (24)$$

受制于非负约束，在式(24)右边不为负的范围里取最大的 Δ ，即

$$\Delta = \frac{\bar{b}_r}{y_{rp}} = \min\left\{\frac{\bar{b}_i}{y_{ip}} \mid y_{ip} > 0 (i = 1, \dots, m)\right\} \quad (25)$$

上式的 Δ 作为 x_p 的新值，同时意味在 $y_{ip} > 0$ 前提下取最小的 $\frac{\bar{b}_i}{y_{ip}}$ ，并将达到该最小值的下标 i 记为 r 。这样定下的

$$\begin{aligned} x_{B_i} &= \bar{b}_i - y_{ip}\Delta (i = 1, \dots, m) \\ x_p &= \Delta \\ x_j &= 0 (j \in N, j \neq p) \end{aligned} \quad (26)$$

构成问题的可行解，特别地成立 $x_{B_r} = 0$ 。

至此, x_{B_r} 和 $x_j (j \in N, j \neq p)$ 合起来有 $n - m$ 个变量取值为0; 取值为正的只能是除此之外的 $\{x_{B_1}, \dots, x_{B_{r-1}}, x_p, x_{B_{r+1}}, \dots, x_{B_m}\}$. 把这 m 个变量看成新的基变量, 另外 $n - m$ 个看成非基变量, 就得到了新的“划分”。

于是, 我们获得了新的可行基解

$$x = (x_{B_1}, \dots, x_{B_{r-1}}, 0, x_{B_{r+1}}, \dots, x_{B_m}, 0, \dots, x_p, \dots, 0)^T.$$

[习题1.4: 证明上述新得到的 x 是可行基解...]

以上操作是通过替换基变量 x_{B_r} 与非基变量 x_p 来确定新的可行基解，我们称之为**转轴运算**(pivoting).

另外，称这时的 x_{B_r} 为出基变量， x_p 为入基变量。这样得到的新可行基解，利用式(22)进行最优判定，并重复上述操作。

单纯形算法(SIMPLEX)

第一步（初始化）：

确定一个可行基划分 $A = (B, N)$, 并计算 $x_B = \bar{b} := B^{-1}b$.

单纯形算法(SIMPLEX)

第二步（最优判定）：

计算向量 $w = (B^T)^{-1}c_B$, 对所有的非基分量 $j \in N$ 求出 $z_j = w^T a_j$.

如果 $c_j - z_j \geq 0 (j \in N)$, 则当前的可行基解 $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} \bar{b} \\ 0 \end{pmatrix}$

已是最优解[stop!].

否则, 选取一个满足 $c_p - z_p < 0$ 的 $p \in N$ 进入下一步。

单纯形算法(SIMPLEX)

第三步（转轴运算）：

计算向量 $y_p = B^{-1}a_p$. 若 $y_p \leq 0$, 则问题无有界解[stop!!]。

否则, 找出 r 使得 $\Delta = \frac{\bar{b}_r}{y_{rp}} = \min\{\frac{\bar{b}_i}{y_{ip}} \mid y_{ip} > 0 (i = 1, \dots, m)\}$, 将基矩阵 B 的列向量 a_{B_r} 用 a_p 替换, 得到新的基矩阵

$$B = (a_{B_1}, \dots, a_{B_{r-1}}, a_p, a_{B_{r+1}}, \dots, a_{B_m}).$$

进而, 记新基变量的值为

$$x_{B_i} = \bar{b}_i - y_{ip}\Delta (i = 1, \dots, m, i \neq r), x_p = \Delta.$$

令基变量的下标集合与非基变量的下标集合分别为

$$B := (B \cup \{p\}) - \{B_r\}, \quad N := (N \cup \{B_r\}) - \{p\}.$$

回到第二步。

单纯形法的理论收敛性

如前所述, 经过转轴运算, 对应于 $c_p - z_p < 0$ 的非基变量 x_p 入基得到新的可行基解时, 目标函数值将减少 $|c_p - z_p|\Delta$.

在这里, $\Delta := \frac{\bar{b}_r}{y_{rp}} = \min\{\frac{\bar{b}_i}{y_{ip}} \mid y_{ip} > 0 (i = 1, \dots, m)\}$.

因为 $\bar{\mathbf{b}} = B^{-1}\mathbf{b} \geq 0$, 所以 $\Delta \geq 0$.

特别当 $\bar{\mathbf{b}} > 0$ 时, 有 $\Delta > 0$, 此时目标函数值严格减少。

另外, 可行基解的个数是有限的, 迭代不会无限重复下去, 必在有限次迭代后结束计算。

单纯形法的有限收敛性定理：（非退化情形）若所给的线性规划问题可以求出初始可行基解，而且转轴运算过程中的所有可行基解都是非退化的，则利用单纯形法在有限次迭代后，要么找到最优解，要么识别出问题无有界解，从而结束计算。

注：可行基解 $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} \bar{b} \\ 0 \end{pmatrix}$ 满足 $\bar{b} > 0$ 时，称它为非退化的。
反之，存在 $\bar{b}_i = 0$ ，称该可行基解为退化的。

初始可行基解： 前述单纯形法需要一个可行基解作为初始解。

一般问题中为给出初始可行基解，我们在这里介绍常用的：

- 两阶段法
- 大M法

所谓**两阶段法**，是第一阶段求一个辅助问题的最优解，第二阶段再以它
为原问题的初始可行基解进行计算。

在**两阶段法**的第一阶段里，针对线性规划问题

$$\{\min c^T x \mid Ax = b, x \geq 0\},$$

考虑如下的辅助问题

$$\begin{aligned} \min \quad & 1^T y \\ \text{s.t.} \quad & Ax + y = b \\ & x \geq 0, y \geq 0. \end{aligned} \tag{27}$$

这里， y 为新引进的 m 维人工变量向量， 1 是所有分量均为1的常数向量。

在辅助问题(27)中, $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}$ 是可行基解。把该可行基解当作初始解, 就可直接用上单纯形法进行迭代。

显然, 辅助问题(27)的目标函数 $1^T y$ 在可行域里取值是非负的。若其最优解是 $\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}$, 则成立 $1^T \bar{y} \geq 0$

如果原问题存在可行解，则任意可行解 x 和 $y = 0$ 的组合 $\begin{pmatrix} x \\ 0 \end{pmatrix}$ 都满足辅助问题(27)的约束条件，且目标函数值为 $1^T y = 0$.

这个事实意味着，当辅助问题的最优解 $\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}$ 成立 $1^T \bar{y} > 0$ 时，原问题没有可行解。

相反，如果辅助问题的最优解 $\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}$ 成立 $1^T \bar{y} = 0$ ，即 $\bar{y} = 0$ 。此时， \bar{x} 显然是原问题的一个可行解（也是可行基解）。

先求出初始可行基解后再进行第二阶段最优解的计算合并成一步作业的方法，称为**大M法**。

大M法考虑如下线性规划问题

$$\begin{array}{ll}\min & c^T x + M \cdot 1^T y \\ \text{s.t.} & Ax + y = b \\ & x \geq 0, y \geq 0.\end{array} \quad (28)$$

这里 M 为充分大的常数，人工变量 y 和常数向量 1 与两阶段法中的一样。

在**大M法**的问题(28)中，目标函数项 $M \cdot 1^T y$ 占支配地位，所以单纯形法迭代将首先让人工变量 y_i 的值减小。其结果是，若所有人工变量都成为0，则目标函数 $c^T x + M \cdot 1^T y$ 和原问题的目标函数 $c^T x$ 本质上一致，可认为在此之后单纯形法的迭代所进行的就是求原问题最优解的计算。

事实上，如果原问题有最优解，在 M 充分大时，得到**大M法**问题(28)的最优解 $\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}$ 中 $\bar{y} = 0$ ，且 \bar{x} 成为原问题的最优解。

用单纯形法求解如下问题：

$$\begin{array}{ll}\min & -4x_1 - x_2 \\ \text{s.t.} & -x_1 + 2x_2 \leq 4 \\ & 2x_1 + 3x_2 \leq 12 \\ & x_1 - x_2 \leq 3 \\ & x_1, x_2 \geq 0.\end{array}\tag{29}$$

[习题1.5：先列出单纯形法的计算步骤，然后求解上述线性规划问题...]

单纯形表（单纯形方法的计算结构）构建

$$\begin{array}{ll}\min & f \\ \text{s.t.} & f - c^T x = 0 \\ & Ax = b \\ & x \geq 0.\end{array}$$

单纯形表（单纯形方法的计算结构）构建

$$\begin{array}{ll} \min & f \\ \text{s.t.} & \begin{array}{rcl} & x_B & + B^{-1} N x_N = B^{-1} b \\ f & + 0 x_B & + (c_B^T B^{-1} N - c_N^T) x_N = c_B^T B^{-1} b \\ & x_B \geq 0, & x_N \geq 0. \end{array} \end{array}$$

单纯形法

Table: 单纯形表

f	x_B	x_N	RHS
0	I_m	$B^{-1}N$	$B^{-1}b$
1	0	$c_B^T B^{-1}N - c_N^T$	$c_B^T B^{-1}b$

单纯形法

Table: 单纯形表

	$x_{B_1} \cdots x_{B_r} \cdots x_{B_m}$	$\cdots x_j \cdots x_p \cdots$	
x_{B_1}	1 \cdots 0 \cdots 0	$\cdots y_{1j} \cdots y_{1p} \cdots$	\bar{b}_1
\vdots	$\vdots \quad \quad \vdots \quad \quad \vdots$	$\vdots \quad \quad \vdots$	\vdots
x_{B_r}	0 \cdots 1 \cdots 0	$\cdots y_{rj} \cdots y_{rp} \cdots$	\bar{b}_r
\vdots	$\vdots \quad \quad \vdots \quad \quad \vdots$	$\vdots \quad \quad \vdots$	\vdots
x_{B_m}	0 \cdots 0 \cdots 1	$\cdots y_{mj} \cdots y_{mp} \cdots$	\bar{b}_m
	0 \cdots 0 \cdots 0	$\cdots z_j - c_j \cdots z_p - c_p \cdots$	$c_B^T \bar{b}$

用大M法求解如下问题：

$$\begin{array}{ll}\min & x_1 + x_2 - 3x_3 \\ \text{s.t.} & x_1 - 2x_2 + x_3 \leq 11 \\ & 2x_1 + x_2 - 4x_3 \geq 3 \\ & x_1 - 2x_3 = 1 \\ & x_1, x_2, x_3 \geq 0.\end{array}\quad (30)$$

[以单纯形法的表格形式进行计算...]

标准化模型（大M法）：

$$\begin{array}{ll}\min & x_1 + x_2 - 3x_3 + M(x_6 + x_7) \\ \text{s.t.} & x_1 - 2x_2 + x_3 + x_4 = 11 \\ & 2x_1 + x_2 - 4x_3 - x_5 + x_6 = 3 \\ & x_1 - 2x_3 + x_7 = 1 \\ & x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0.\end{array}\quad (31)$$

Table: 迭代单纯形表-1

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_4	1	-2	1	1	0	0	0	11
x_6	2	1	-4	0	-1	1	0	3
x_7	1	0	-2	0	0	0	1	1
	$3M - 1$	$M - 1$	$-6M + 3$	0	$-M$	0	0	$4M$

单纯形法

Table: 迭代单纯形表-2

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_4	0	-2	3	1	0	0	-1	10
x_6	0	1	0	0	-1	1	-2	1
x_1	1	0	-2	0	0	0	1	1
	0	$M - 1$	1	0	$-M$	0	$1 - 3M$	$M + 1$

Table: 迭代单纯形表-3

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_4	0	0	3	1	-2	2	-5	12
x_2	0	1	0	0	-1	1	-2	1
x_1	1	0	-2	0	0	0	1	1
	0	0	1	0	-1	$1 - M$	$-1 - M$	2

单纯形法

Table: 迭代单纯形表-4 (opt)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
x_3	0	0	1	$1/3$	$-2/3$	$2/3$	$-5/3$	4
x_2	0	1	0	0	-1	1	-2	1
x_1	1	0	0	$2/3$	$-4/3$	$4/3$	$-7/3$	9
	0	0	0	$-1/3$	$-1/3$	$1/3 - M$	$2/3 - M$	-2

注：单纯形方法的所有计算都可对应于单纯形表（矩阵）的消元运算（即行初等变换）。

[习题1.6：证明在执行主元消去法前后两个不同可行基下的判别系数和目标函数值有如下关系]

$$\begin{cases} (z_j - c_j)^{new} = (z_j - c_j) - \frac{y_{rj}}{y_{rp}}(z_p - c_p); \\ (c_B^T B^{-1}b)^{new} = c_B^T B^{-1}b - \frac{\bar{b}_r}{y_{rp}}(z_p - c_p). \end{cases} \quad (32)$$

针对任意一个线性规划模型，可以定义一个称之为对偶问题 (Dual Problem) 的模型。

对偶理论将揭示原问题与对偶问题之间的内在联系，为进一步深入研究线性规划的求解算法提供理论依据。

原问题:

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0. \end{array} \quad (33)$$

对偶问题:

$$\begin{array}{ll} \max & b^T w \\ \text{s.t.} & A^T w \leq c \\ & w \geq 0. \end{array} \quad (34)$$

对偶关系描述:

- (1) 目标函数极小(min) \longleftrightarrow 极大(max)
- (2) 一方问题的目标函数系数向量 \longleftrightarrow 另一方问题的约束条件右边常数向量
- (3) 约束条件矩阵 $A \longleftrightarrow A^T$
- (4) 原问题的 \geq 约束条件 \longleftrightarrow 对偶问题的变量非负约束
- (5) 原问题的变量非负约束 \longleftrightarrow 对偶问题的 \leq 约束条件

对偶问题（等价形式）：

$$\begin{array}{ll} \text{(DP)} & \min \quad (-b)^T w \\ & \text{s.t.} \quad (-A)^T w \geq -c \\ & \quad \quad w \geq 0. \end{array} \quad (35)$$

对偶问题的对偶：

$$\begin{array}{ll} \text{(DDP)} & \max \quad (-c)^T x \\ & \text{s.t.} \quad (-A)x \leq -b \\ & \quad \quad x \geq 0. \end{array} \quad (36)$$

标准型对偶:

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0. \end{array} \quad \Rightarrow \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \geq b \\ & -Ax \geq -b \\ & x \geq 0. \end{array} \quad (37)$$

$$\begin{array}{ll} \max & b^T w \\ \text{s.t.} & A^T w \leq c. \end{array} \quad \Leftarrow \quad \begin{array}{ll} \max & b^T w' - b^T w'' \\ \text{s.t.} & A^T w' - A^T w'' \leq c \\ & w' \geq 0, w'' \geq 0. \end{array} \quad (38)$$

一般式对偶:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & A_G x \geq b_G \\ & A_E x = b_E \\ & A_L x \leq b_L \\ & x \geq 0. \end{aligned} \tag{39}$$

$$\begin{aligned} \max \quad & b_G^T w_G + b_E^T w_E + b_L^T w_L \\ \text{s.t.} \quad & A_G^T w_G + A_E^T w_E + A_L^T w_L \leq c \\ & w_G \geq 0, w_E \text{ free}, w_L \leq 0. \end{aligned} \tag{40}$$

原问题与对偶问题之间有着内密切的内在关系，记 x 和 w 分别为(LP)和(DP)的任意可行解。

弱对偶定理： 任意线性规划问题(LP)及其对偶问题(DP)之间成立以下关系

- (a) x 是(LP)问题的可行解, w 是(DP)问题的可行解, $\Rightarrow c^T x \geq b^T w$.
- (b) x 是(LP)问题的可行解 $\Rightarrow c^T x \geq$ 问题(DP)的最大值。
 w 是(DP)问题的可行解 $\Rightarrow b^T w \leq$ 问题(LP)的最小值。
- (c) x 是(LP)问题的可行解, w 是(DP)问题的可行解, 且 $c^T x = b^T w \Rightarrow$
 x 是(LP)问题的最优解, w 是(DP)问题的最优解。
- (d) 问题(LP)无界 \Rightarrow 问题(DP)无可行解。
问题(DP)无界 \Rightarrow 问题(LP)无可行解。

最优解存在性定理： 若(LP)与(DP)都有可行解，则它们均存在最优解。

对偶定理： 原问题(LP)和对偶问题(DP)只要一方有最优解，则另一方也有最优解，且此时两方的最优值一致。

[习题1.7：证明上述对偶定理...]

互补松弛定理： 设 \bar{x} 和 \bar{w} 分别是(LP)和(DP)的可行解，那么 \bar{x} 和 \bar{w} 是对应问题最优解的充要条件是：

$$\begin{cases} \bar{x}^T (A^T \bar{w} - c) = 0 \\ \bar{w}^T (A \bar{x} - b) = 0 \end{cases} \quad (41)$$

线性规划的Kuhn-Tucker条件为：

$$(K - T) \quad \begin{cases} Ax - b \geq 0 \\ x \geq 0 \\ A^T w - c \leq 0 \\ w \geq 0 \\ x^T (A^T w - c) = 0 \\ w^T (Ax - b) = 0 \end{cases} \quad (42)$$

对偶理论

利用互补松弛定理，当知道一个问题的最优解时，可求出其对偶问题的最优解。

$$\begin{array}{ll}\max & w_1 + 2w_2 \\ \text{s.t.} & 3w_1 + w_2 \leq 2 \\ & -w_1 + 2w_2 \leq 3 \\ & w_1 - 3w_2 \leq 1 \\ & w_1 \geq 0, w_2 \geq 0.\end{array} \quad \text{的最优解} \bar{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1/7 \\ 11/7 \end{pmatrix}.$$

$$\begin{array}{ll}\min & 2x_1 + 3x_2 + x_3 \\ \text{s.t.} & 3x_1 - x_2 + x_3 \geq 1 \\ & x_1 + 2x_2 - 3x_3 \geq 2 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.\end{array} \quad \text{的最优解} \bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4/7 \\ 5/7 \\ 0 \end{pmatrix}.$$

对偶理论

利用互补松弛定理，当知道一个问题的最优解时，可求出其对偶问题的最优解。

$$\begin{array}{ll}\max & w_1 + 2w_2 \\ \text{s.t.} & 3w_1 + w_2 \leq 2 \\ & -w_1 + 2w_2 \leq 3 \\ & w_1 - 3w_2 \leq 1 \\ & w_1 \geq 0, w_2 \geq 0.\end{array} \quad \text{的最优解 } \bar{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1/7 \\ 11/7 \end{pmatrix}.$$

$$\begin{array}{ll}\min & 2x_1 + 3x_2 + x_3 \\ \text{s.t.} & 3x_1 - x_2 + x_3 \geq 1 \\ & x_1 + 2x_2 - 3x_3 \geq 2 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.\end{array} \quad \text{的最优解 } \bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4/7 \\ 5/7 \\ 0 \end{pmatrix}.$$

对偶理论

利用互补松弛定理，当知道一个问题的最优解时，可求出其对偶问题的最优解。

$$\begin{array}{ll}\max & w_1 + 2w_2 \\ \text{s.t.} & 3w_1 + w_2 \leq 2 \\ & -w_1 + 2w_2 \leq 3 \\ & w_1 - 3w_2 \leq 1 \\ & w_1 \geq 0, w_2 \geq 0.\end{array} \quad \text{的最优解 } \bar{w} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 1/7 \\ 11/7 \end{pmatrix}.$$

$$\begin{array}{ll}\min & 2x_1 + 3x_2 + x_3 \\ \text{s.t.} & 3x_1 - x_2 + x_3 \geq 1 \\ & x_1 + 2x_2 - 3x_3 \geq 2 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.\end{array} \quad \text{的最优解 } \bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4/7 \\ 5/7 \\ 0 \end{pmatrix}.$$

实际问题中，很多时候是基于某些采集数据来决定模型的系数。在这种情况下，势必会出现系数的扰动及引起的变化。

所谓灵敏度分析就是利用解一个问题得到的结果，研究当系数有微小变化时最优解的反应。

灵敏度分析

为简单起见，考虑问题

$$\begin{array}{ll}\min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0\end{array}\quad (43)$$

约束条件右边常数向量作微小变化

$$b + \Delta b = (b_1 + \Delta b_1, \dots, b_m + \Delta b_m)^T,$$

得到新的线性规划问题

$$\begin{array}{ll}\min & c^T x \\ \text{s.t.} & Ax = b + \Delta b \\ & x \geq 0.\end{array}\quad (44)$$

现假设对问题(43)利用单纯形法已得到其最优解

$$x^* = \begin{pmatrix} x_B^* \\ x_N^* \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix}.$$

这时对于最优基 B , 根据可行性条件有

$$B^{-1}b \geq 0, \quad (45)$$

根据最优性条件有

$$c_N^T - c_B^T B^{-1}N \geq 0. \quad (46)$$

当 b 变为 $b + \Delta b$ 后, 如果还成立

$$B^{-1}(b + \Delta b) \geq 0, \quad (47)$$

则问题(43)的最优基 B 也是问题(44)的最优基。

此时问题(44)的最优解成为 $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} B^{-1}(b + \Delta b) \\ 0 \end{pmatrix}$.

进一步，记问题(43)的目标函数最优值为 $z(b)$ ，问题(44)的目标函数最优值为 $z(b + \Delta b)$ ，则有

$$\begin{aligned} z(b) &= c_B^T B^{-1} b \\ z(b + \Delta b) &= c_B^T B^{-1} (b + \Delta b) \end{aligned} \quad (48)$$

故成立

$$z(b + \Delta b) - z(b) = c_B^T B^{-1} \Delta b. \quad (49)$$

事实上, $(B^T)^{-1}c_B$ 是对应于基矩阵 B 的单纯形乘子向量 w . 根据对偶理论, 它即为问题(43)的对偶问题的最优解。

因此, 记 $w^* = (B^T)^{-1}c_B$, 则上式可写成

$$z(b + \Delta b) - z(b) = (w^*)^T \Delta b = \sum_{i=1}^m w_i^* \Delta b_i. \quad (50)$$

考虑 $\Delta b_i \rightarrow 0$ 时的极限，就可以得到

$$\frac{\partial z(b)}{\partial b_i} = w_i^*, (i = 1, \dots, m). \quad (51)$$

这里，实际上给出了对偶问题最优解的经济意义（影子价格）！

线性规划的多项式时间算法

从最坏计算量角度来看，单纯形方法不是多项式时间算法。关于单纯形方法的理论及实际上的计算效率可见综述文献：

R. Shamir. The efficiency of the simplex method: A survey. Management Science, Vol. 33, 1987: 301-334.

线性规划的多项式时间算法

针对线性规划问题的多项式时间算法：

- 椭球算法，最先在理论上明确线性规划问题恒在多项式时间内可解，因而它是意义深远的。

L.G. Khachiyan. A polynomial algorithm in linear programming. Soviet Mathematics Doklady, Vol. 20, 1979: 191-194.

- 内点算法，作为可替代单纯形法的有效方法而引入注目，同时它可被推广并应用到不限于线性规划的更一般问题。

N. Karmarkar. A new polynomial time algorithm for linear programming. Combinatorica, Vol. 4, 1984: 373-395.

本章作业（线性规划）

- **Exercise 1.1:** 证明线性规划可行域 $S = \{x \mid Ax = b, x \geq 0\}$ 的凸集表示定理。
- **Exercise 1.2:** 给出例子(18)的所有极点和可行基解。
- **Exercise 1.3:** 证明线性规划模型的基本可行解集与其可行域的极点集是等价的。
- **Exercise 1.4:** 证明单纯形法转轴运算得到的

$$x = (x_{B_1}, \dots, x_{B_{r-1}}, 0, x_{B_{r+1}}, \dots, x_{B_m}, 0, \dots, x_p, \dots, 0)^T$$

是一个新的可行基解。

本章作业（线性规划）

- **Exercise 1.5:** 先列出单纯形法的计算步骤，然后求解线性规划例子(29)。
- **Exercise 1.6:** 证明在执行主元消去法前后两个不同可行基下的判别系数和目标函数值满足(32)。
- **Exercise 1.7:** 证明线性规划问题的对偶定理，即原问题(LP)和对偶问题(DP)只要一方有最优解，则另一方也有最优解且此时两方的最优值一致。

- 1 绪论
- 2 线性规划
- 3 网络最优化**
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

网络最优化简介

运输问题，最短路问题，最大流问题，最小成本流问题等有关网络最优化通常应用于解决实际问题，并具有重要的经济意义。

就数学模型而言，它们是线性规划的几个重要特例。针对线性规划模型已有多项式时间算法，因为网络模型的特殊数学结构，利用其结构特性还可以设计出效率更高的求解算法。

运输模型是一类特殊的线性规划，即从货源地（如生产厂家）装运货物到目的地（如经销商仓库），其目标是确定运输表使得总运输成本最小并满足供应和需求的限制条件。

运输模型可扩展应用于其他领域，包括投资控制，工作调度，人员指派等。

经典的运输问题：

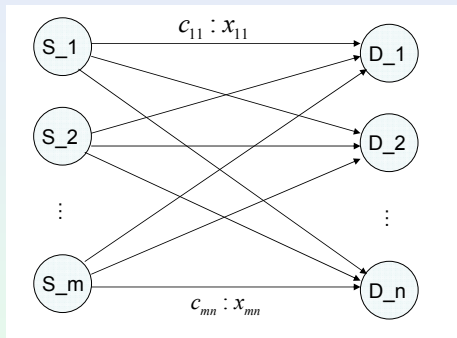
工厂 i 的货物量 s_i , $i = 1, 2, \dots, m$.

需求点 j 的需求量 d_j , $j = 1, 2, \dots, n$.

从工厂 i 到需求点 j 的单位货运费用 c_{ij} 及其发货量 x_{ij} .

选取一个能使运输总费用达到最小的路径规划。

运输模型



运输问题的数学描述

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq s_i, i = 1, \dots, m \\ & \sum_{i=1}^m x_{ij} \geq d_j, j = 1, \dots, n \\ & x_{ij} \geq 0, \quad i = 1, \dots, m \quad j = 1, \dots, n. \end{aligned} \tag{52}$$

Table: Mileage Chart

	Denver	Miami
Los Angeles	1000	2690
Detroit	1250	1350
New Orleans	1275	850

Table: Transportation Cost per Car

	Denver($j = 1$)	Miami($j = 2$)
Los Angeles($i = 1$)	\$80	\$215
Detroit($i = 2$)	\$100	\$108
New Orleans($i = 3$)	\$102	\$68

Table: MG Auto Transportation Model

	Denver	Miami	Supply
Los Angeles	x_{11} 80	x_{12} 215	1000
Detroit	x_{21} 100	x_{22} 108	1500
New Orleans	x_{31} 102	x_{32} 68	1200
Demand	2300	1400	

$$\begin{array}{ll}\min & z = 80x_{11} + 215x_{12} + 100x_{21} + 108x_{22} + 102x_{31} + 68x_{32} \\ \text{s.t.} & x_{11} + x_{12} \leq 1000 \quad (\text{LosAngeles}) \\ & x_{21} + x_{22} \leq 1500 \quad (\text{Detroit}) \\ & x_{31} + x_{32} \leq 1200 \quad (\text{NewOreleans}) \\ & x_{11} + x_{21} + x_{31} \geq 2300 \quad (\text{Denver}) \\ & x_{12} + x_{22} + x_{32} \geq 1400 \quad (\text{Miami}) \\ & x_{ij} \geq 0, \quad i = 1, 2, 3, \quad j = 1, 2\end{array} \quad (53)$$

考虑产销平衡的情形

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} = s_i, i = 1, \dots, m \\ & \sum_{i=1}^m x_{ij} = d_j, j = 1, \dots, n \\ & x_{ij} \geq 0, \quad i = 1, \dots, m \quad j = 1, \dots, n. \end{aligned} \tag{54}$$

运输问题的对偶

$$\begin{aligned} \max \quad & \sum_{i=1}^m s_i u_i + \sum_{j=1}^n d_j v_j \\ \text{s.t.} \quad & u_i + v_j \leq c_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n \\ & u_i, v_j \quad \text{无限制} \end{aligned} \tag{55}$$

利用对偶关系：

对基变量 x_{ij} 而言， $u_i + v_j = c_{ij}$ ，即 $\sigma_{ij} = u_i + v_j - c_{ij} = 0$ 。

对非基变量 x_{ij} 而言，若 $\sigma_{ij} = u_i + v_j - c_{ij} \leq 0$ ，已对偶可行；
若 $\sigma_{ij} = u_i + v_j - c_{ij} > 0$ ，非对偶可行，则引进基。

运输模型的单纯形法求解步骤:

1. 选取一组 $m + n - 1$ 个路径, 作为初始可行基解。
2. 检验当前解是否可改进, 如果可改进, 则引进一个非基变量进行步3, 否则停止。
3. 当把步2中挑选的变量引进时, 确定哪个路径应当由基解中退出。
4. 调整其他基本路径的流量 (满足可行性), 返回到步2.

我们将以例子来说明运输模型的单纯形算法。

Table: 某公司的运输表

	D(1)	D(2)	D(3)	D(4)	Supply
S(1)	<div>10</div> <div>x_{11}</div>	<div>2</div> <div>x_{12}</div>	<div>20</div> <div>x_{13}</div>	<div>11</div> <div>x_{14}</div>	15
S(2)	<div>12</div> <div>x_{21}</div>	<div>7</div> <div>x_{22}</div>	<div>9</div> <div>x_{23}</div>	<div>20</div> <div>x_{24}</div>	25
S(3)	<div>4</div> <div>x_{31}</div>	<div>14</div> <div>x_{32}</div>	<div>16</div> <div>x_{33}</div>	<div>18</div> <div>x_{34}</div>	10
Demand	5	15	15	15	

Table: 初始可行基解 (Northwest-Corner Starting Solution)

	D(1)	D(2)	D(3)	D(4)	Supply
S(1)	10 5	2 10	20	11	15
S(2)	12	7 5	9 15	20 5	25
S(3)	4	14	16	18 10	10
Demand	5	15	15	15	

Table: 迭代-1

	$v_1 = 10$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Supply
$u_1 \equiv 0$	10 5	2 10	20 [-16]	11 [4]	15
$u_2 = 5$	12 [3]	7 5	9 15	20 5	25
$u_3 = 3$	4 [9]	14 [-9]	16 [-9]	18 10	10
Demand	5	15	15	15	

Table: 迭代-1 (回路)

	$v_1 = 10$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Supply
$u_1 \equiv 0$	10 $5 - \theta$	2 $10 + \theta$	20 [-16]	11 [4]	15
$u_2 = 5$	12 [3]	7 $5 - \theta$	9 15	20 $5 + \theta$	25
$u_3 = 3$	4 θ [9]	14 [-9]	16 [-9]	18 $10 - \theta$	10
Demand	5	15	15	15	

Table: 迭代-2

	$v_1 = 1$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Supply
$u_1 \equiv 0$	10 [-9]	2 15	20 [-16]	11 [4]	15
$u_2 = 5$	12 [-6]	7 0	9 15	20 10	25
$u_3 = 3$	4 5	14 [-9]	16 [-9]	18 5	10
Demand	5	15	15	15	

Table: 迭代-2 (回路)

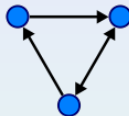
	$v_1 = 1$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Supply
$u_1 \equiv 0$	10 [-9]	2 $15 - \theta$	20 [-16]	11 θ [4]	15
$u_2 = 5$	12 [-6]	7 $0 + \theta$	9 15	20 $10 - \theta$	25
$u_3 = 3$	4 5	14 [-9]	16 [-9]	18 5	10
Demand	5	15	15	15	

Table: 迭代-3 (最优解)

	$v_1 = -3$	$v_2 = 2$	$v_3 = 4$	$v_4 = 11$	Supply
$u_1 \equiv 0$	10 [-13]	2 5	20 [-16]	11 10	15
$u_2 = 5$	12 [-10]	7 10	9 15	20 [-4]	25
$u_3 = 7$	4 5	14 [-5]	16 [-5]	18 5	10
Demand	5	15	15	15	

网络模型

设 $G = (V, E)$ 为有向图, 其中 V 是节点的集合, E 是边的集合。



有时把某个节点作为初始点 s , 另一个作为终点 t 而特殊对待。

各边 $e \in E$ 上赋有成本 $c(e)$ 以及容量 $u(e)$, 且都取实数值。

由它们组成的 $|E|$ 维相应向量分别记为

$$c = (c(e) | e \in E), \quad u = (u(e) | e \in E).$$

统括这些元素的 $\mathcal{N} = (G, s, t, c, u)$ 称为网络。

最短路径问题

最短路径问题中，成本 $c(e)$ 解释为边 e 的长度。最短路径问题的典型形式是：

在网络 $\mathcal{N} = (G, s, c)$ 中求出始点 s 到其它各点 $v \in V$ 的最短路径及其长度。

另外，定义 $w \in V$ 到 $v \in V$ 的路径(path)

$$\pi = v_{i_0}(= w), v_{i_1}, \dots, v_{i_k}(= v)$$

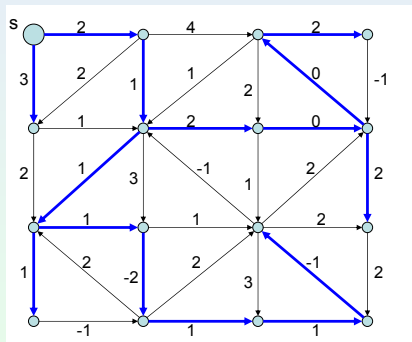
长度为

$$c(\pi) = \sum_{j=0}^{k-1} c(v_{i_j}, v_{i_{j+1}}).$$

最短路径问题

当 \mathcal{N} 中没有长度为负的回路时，存在有从 s 到所有点 v 的最短路径，它们可用以 s 为根的**最短路树**来表示。

求 s 到 v 的最短路径，只要沿着 s 到 v 的最短路树的边走下去就行。



如何求得该最短路树？最短路问题的模型算法？

最短路径问题

最短路径问题的数学描述：

$$\begin{aligned} \min \quad & \sum_{(i,j) \in E} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{(s,j) \in E} x_{sj} = 1 \\ & \sum_{(k,j) \in E} x_{kj} - \sum_{(i,k) \in E} x_{ik} = 0, \forall k \in V - \{s, t\} \\ & - \sum_{(i,t) \in E} x_{it} = -1 \\ & x_{ij} \geq 0, \forall (i,j) \in E \end{aligned} \tag{56}$$

最短路径问题

最短路径问题的对偶为:

$$\begin{array}{ll} \max & (y_s - y_t) \\ \text{s.t.} & y_i - y_j \leq c_{ij}, \quad \forall (i, j) \in E \end{array} \quad (57)$$

最短路径问题的算法

一般网络的最短路径问题可以看成是一个线性规划模型（事实上是一个更特殊的运输模型），可依据对偶性构造其求解算法。

如果基于动态规划的思想，可给出最短路径问题的强多项式时间算法。以下仅说明具有代表性的算法之一：

Dijkstra's algorithm.

最短路径问题的算法

算法 DIJKSTRA

输入 有向图 $G = (V, E)$, 各边长度 $c: E \rightarrow \mathbb{R}_+$, 始点 $s \in V$.

输出 从始点到所有节点 $v \in V$ 的最短路径及其长度 $c^*(v)$.

步一 初始化: 令 $d(s) := 0, d(v) := \infty (v \in V - \{s\})$, 以及 $P := \emptyset$.

步二 迭代: 选取一个满足 $d(v^*) = \min\{d(v) \mid v \in V - P\}$ 的节点 $v^* \in V - P$.

步三 更新: $c^*(v^*) := d(v^*), P := P \cup \{v^*\}$.
进一步对 $w \in V - P$ 的各边 $e = (v^*, w) \in E$ 作如下更新:

$$d(w) := \min\{d(w), d(v^*) + c(e)\}.$$

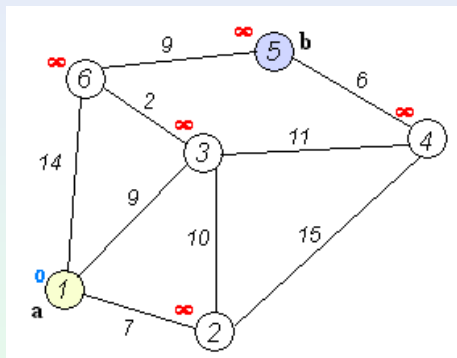
步四 结束判定: 如果 $P = V$ 则结束计算; 否则回到**步二**。

最短路径问题的算法

计算过程中求出的 $d(v)$ 恒为从 s 到 v 的最短路径长度 $c^*(v)$ 的上界，即 $d(v) \geq c^*(v)$ 。

但在已执行节点集合 $P \subset V$ 里，具有性质 $d(v) = c^*(v), v \in P$ 。于是，当 $P = V$ 时即求出了 s 到所有节点的最短路径及其长度，结束计算。

最短路径问题算例



[dist, path, pred] = graphshortestpath(G,S)

This procedure determines the single-source shortest paths from node *S* to all other nodes in the graph represented by matrix *G*.

Input *G* is an *N*-by-*N* sparse matrix that represents a graph. Nonzero entries in matrix *G* represent the weights of the edges.

Output *dist* is the vector of *N* distances from the source to every node (using *Inf*s for nonreachable nodes and 0 for the source node).

Output *path* contains the winning paths to every node.

Output *pred* contains the predecessor nodes of the winning paths.

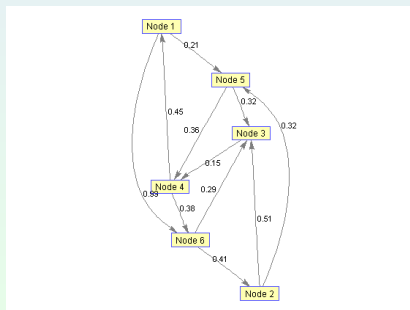
graphshortestpath

Create and view a directed graph with 6 nodes and 11 edges.

```
W = [.41 .99 .51 .32 .15 .45 .38 .32 .36 .29 .21];
```

```
DG = sparse([6 1 2 2 3 4 4 5 5 6 1],[2 6 3 5 4 1 6 3 4 3 5],W);
```

```
h = view(biograph(DG,[],'ShowWeights','on'))
```



graphshortestpath

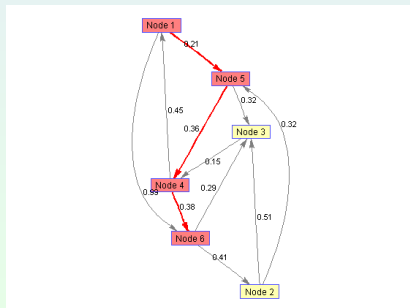
Find the shortest path in the graph from node 1 to node 6.

```
[dist,path,pred] = graphshortestpath(DG,1,6)
```

```
set(h.Nodes(path),'Color',[1 0.5 0.5])
```

```
edges = getedgesbynodeid(h,get(h.Nodes(path),'ID'));
```

```
set(edges,'LineColor',[1 0 0]); set(edges,'LineWidth',2)
```



最短路径问题实例

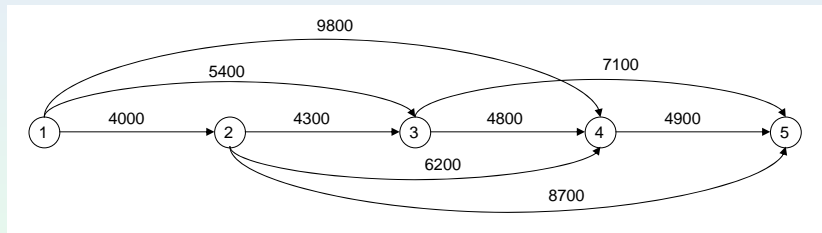
最短路径问题的应用实例：设备更新

Table: Equipment replacement

Equipment acquired at start of year	Replacement cost for given years in operation		
	+1	+2	+3
1	4000	5400	9800
2	4300	6200	8700
3	4800	7100	—
4	4900	—	—

最短路径问题实例

最短路径问题的应用实例：设备更新



最大流问题

网络 $\mathcal{N} = (G, s, t, u)$ 里, 所谓从源点 s 到汇点 t 的流(flow) 是指实数向量 $x = (x(e) \mid e \in E)$, 满足如下**流量守恒**条件:

$$\sum_{e \in \text{Out}(v)} x(e) - \sum_{e \in \text{In}(v)} x(e) = 0, v \in V - \{s, t\} \quad (58)$$

这里 $\text{Out}(v)$ 和 $\text{In}(v)$ 分别表示 G 中流出和流入 v 节点的边集合。

最大流问题

从 s 的纯流出量

$$x(s) \triangleq \sum_{e \in \text{Out}(s)} x(e) - \sum_{e \in \text{In}(s)} x(e)$$

与到 t 的纯流入量

$$x(t) \triangleq \sum_{e \in \text{In}(t)} x(e) - \sum_{e \in \text{Out}(t)} x(e)$$

相等，即成立

$$x(s) = x(t). \quad (59)$$

这个 $x(s)$ 称为 x 的流值。

最大流问题

当流 x 进一步满足**容量约束** (capacity constraint) 条件

$$0 \leq x(e) \leq u(e), e \in E \quad (60)$$

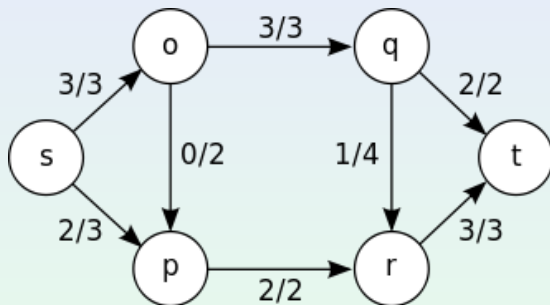
则称它为**可行流** (feasible flow)。

最大流问题

所谓最大流问题，就是求出使流值达最大的可行流问题，数学上可描述如下：

$$\begin{aligned} \max \quad & x(s) \\ \text{s.t.} \quad & \sum_{e \in \text{Out}(v)} x(e) - \sum_{e \in \text{In}(v)} x(e) = 0, v \in V - \{s, t\} \\ & x(s) = \sum_{e \in \text{Out}(s)} x(e) - \sum_{e \in \text{In}(s)} x(e) \\ & 0 \leq x(e) \leq u(e), e \in E. \end{aligned} \tag{61}$$

最大流问题



最小割问题

把一个流网络的顶点集划分成两个集合 S 和 T , 使得源点 $s \in S$ 且汇点 $t \in T$. 分割 (S, T) 的容量为

$$U(S, T) = \sum_{\substack{(i,j) \in E \\ i \in S, j \in T}} u(i,j).$$

所谓最小割, 就是指所有分割中容量最小的一个割。

最大流最小割定理

从直观上看，割集 (S, T) 是从源点 s 到汇点 t 的必经之路，如果该路堵塞则流从 s 无法到达 t 。于是我们可以得到下面的定理。

最大流最小割定理： 任意一个流网络的最大流量等于该网络的最小割的容量。

注：定理的证明可以参考图论方面的文献资料。

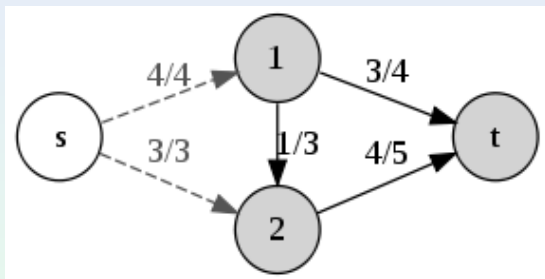
最大流最小割定理

从直观上看，割集 (S, T) 是从源点 s 到汇点 t 的必经之路，如果该路堵塞则流从 s 无法到达 t 。于是我们可以得到下面的定理。

最大流最小割定理： 任意一个流网络的最大流量等于该网络的最小割的容量。

注：定理的证明可以参考图论方面的文献资料。

最大流最小割定理



最大流问题的算法

网络 $\mathcal{N} = (G, s, t, u)$ 中给定一个可行流 x ，通过构造余网络和流扩充路来迭代更新达到最大流。

从 \mathcal{N} 的各边 $e = (v, w) \in E$ 出发，按照以下规则生成边 (v, w) 或 (w, v) ，得到的有向边集合记为 E_x ，同时确定其容量 \bar{u} 。

R-1 如果 $u(e) - x(e) > 0$ ，则生成 $(v, w) \in E_x$ ，并令其容量 $\bar{u}(v, w) = u(e) - x(e)$ 。

R-2 如果 $x(e) > 0$ ，则生成 $(w, v) \in E_x$ ，并令其容量 $\bar{u}(w, v) = x(e)$ 。

所得到的有向网中所有容量 $\bar{u} > 0$ 的边构成余网络 $\mathcal{N}_x = (G_x, s, t, \bar{u})$ ，其中有向图 $G_x = (V, E_x)$ ，容量 $\bar{u} = \{\bar{u}(e) > 0 \mid \forall e \in E_x\}$ 。那么， \mathcal{N}_x 中从 s 到 t 的路称为流扩充路。

最大流问题的算法

引理（流扩充路）给定网络 $\mathcal{N} = (G, s, t, u)$, 其可行流 x 为最大流的充要条件是, 余网络 \mathcal{N}_x 中不存在有流扩充路。

如果存在有流扩充路, 则 x 可修正为流值更大的流 x' 。

最大流问题的算法

假定在 \mathcal{N}_x 中存在从 s 到 t 的路 π , 则利用

$$\Delta := \min\{\bar{u}(e) \mid e \in \pi\}, \quad (62)$$

对各边 $e = (v, w) \in E$ 作如下修正

$$x'(e) = \begin{cases} x(e) + \Delta, & (v, w) \in \pi \text{ (由Rule 1 生成的边)} \\ x(e) - \Delta, & (w, v) \in \pi \text{ (由Rule 2 生成的边)} \\ x(e), & \text{其它情形} \end{cases} \quad (63)$$

易证 x' 也是 \mathcal{N} 的可行流, 其流值是

$$x'(s) = x(s) + \Delta.$$

[思考练习题]

验证上述 x' 是 \mathcal{N} 的可行流, 并且其流值为 $x'(s) = x(s) + \Delta$.

最大流问题的算法

算法 MAX-FLOW

输入 网络 $\mathcal{N} = (G, s, t, u)$, 其中有向图 $G = (V, E)$.

输出 从 s 到 t 的最大流值 f_{\max} .

步一 初始化: 令 $x(e) := 0 (e \in E)$ 以及 $f_{\max} := 0$.

步二 余网络: 构造当前可行流 x 的余网络 $\mathcal{N}_x = (G_x, s, t, \bar{u})$.

步三 流的扩充: 如果 \mathcal{N}_x 中没有流的扩充路则结束计算。相反, 如果存在有流的扩充路, 则选取其一 π , 根据(62)式和(63)式修正 x 得到流 x' 。
令 $f_{\max} := f_{\max} + \Delta$, $x := x'$ 回到**步二**。

[MaxFlow, FlowMatrix, Cut] =

graphmaxflow(G, SNode, TNode)

This procedure calculates the maximum flow of directed graph G from node SNode to node TNode.

Input G is an N-by-N sparse matrix that represents a directed graph. Nonzero entries in matrix G represent the capacities of the edges.

Output MaxFlow is the maximum flow, and FlowMatrix is a sparse matrix with all the flow values for every edge. FlowMatrix(X,Y) is the flow from node X to node Y.

Output Cut is a logical row vector indicating the nodes connected to SNode after calculating the minimum cut between SNode and TNode. If several solutions to the minimum cut problem exist, then Cut is a matrix.

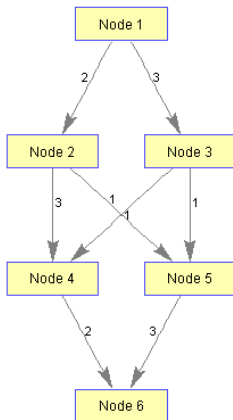
Create a directed graph with six nodes and eight edges.

```
cm = sparse([1 1 2 2 3 3 4 5],[2 3 4 5 4 5 6 6],...  
[2 3 3 1 1 1 2 3],6,6);
```

graphmaxflow

View the graph with the original capacities.

```
h = view(biograph(cm,[],'ShowWeights','on'))
```



Calculate the maximum flow in the graph from node 1 to node 6.

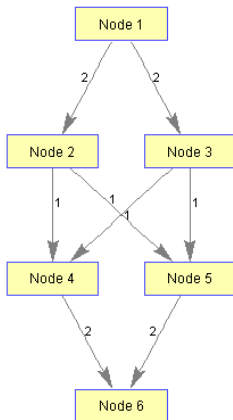
```
[M,F,K] = graphmaxflow(cm,1,6);
```

Notice that K is a two-row matrix because there are two possible solutions to the minimum cut problem.

graphmaxflow

View the graph with the calculated maximum flows.

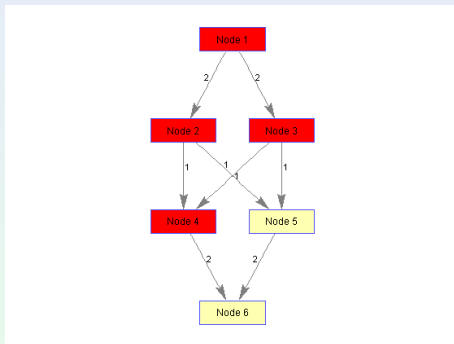
```
g = biograph(F,[],'ShowWeights','on'); view(g)
```



graphmaxflow

Show one solution to the minimum cut problem in the graph.

```
set(g.Nodes(K(1,:)),'Color',[1 0 0]); view(g)
```

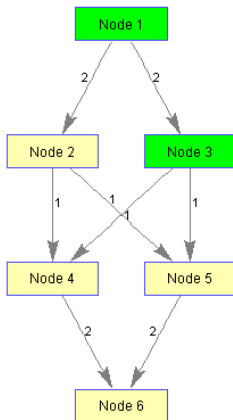


Notice that in the three edges that connect the source nodes (red) to the destination nodes (yellow), the original capacities and the calculated maximum flows are the same.

graphmaxflow

Show another solution to the minimum cut problem in the graph.

```
set(g.Nodes(K(2,:)), 'Color', [0 1 0]); view(g)
```



最小成本流问题

考虑网络 $\mathcal{N} = (G, s, t, c, u)$ 上的流 x , 其中流值固定为

$$x(s) = f^*.$$

在流值 $x(s) = f^*$ 不超过网络的最大流的条件下, 求出使成本达最小的流就是所谓最小成本流问题。

最小成本流问题

最小成本流问题的数学模型：

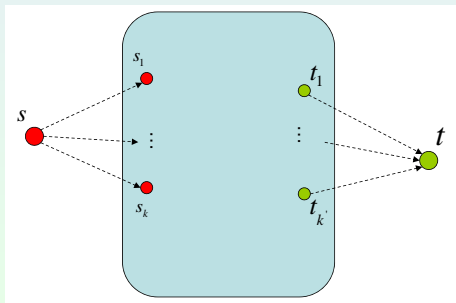
$$\begin{aligned} \min \quad & \sum_{e \in E} c(e)x(e) \\ \text{s.t.} \quad & \sum_{e \in \text{Out}(v)} x(e) - \sum_{e \in \text{In}(v)} x(e) = 0, v \in V - \{s, t\} \\ & \sum_{e \in \text{Out}(s)} x(e) - \sum_{e \in \text{In}(s)} x(e) = f^* \\ & 0 \leq x(e) \leq u(e), e \in E. \end{aligned} \tag{64}$$

最小成本流问题

在最小成本流问题中，可以有多个源点和汇点。如果最小成本流问题具有 k 个源点 $s_i, i = 1, \dots, k$ 以及各自的流出量 $x(s_i) = f^*(s_i)$ ，还有 k' 个汇点 $t_j, j = 1, \dots, k'$ 以及各自的流出量 $x(t_j) = f^*(t_j)$ ，则可以引入哑元源点 s 和哑元汇点 t ，并添加 $(k + k')$ 条边

$$(s, s_i) : c(s, s_i) = 0, u(s, s_i) = f^*(s_i), i = 1, \dots, k;$$

$$(t_j, t) : c(t_j, t) = 0, u(t_j, t) = f^*(t_j), j = 1, \dots, k'.$$



最小成本流问题

最小成本流问题具有很好的模型化能力：

通过引入哑元和构造相应的边（及其量化赋值），可将最短路径问题和最大流问题转化为求解特殊的最小成本流问题。

最小成本流问题

最小成本流问题具有很好的模型化能力:

(一) 通过引入哑元终点 t' , 加入从 $v \in V$ 出发的边 (v, t') 且满足

$$c(v, t') = 0, \quad u(v, t') = 1.$$

那么, 最短路径问题成为求解从 s 到 t' 的具有流值 $f^* = n$ 的最小成本流问题。这里, $n = |V|$, 并假定原网络各边的容量全为 ∞ 。

最小成本流问题

最小成本流问题具有很好的模型化能力：

（二）通过引入哑元始点 s' ，并构造如下两条边：

$$(s', s); \quad c(s', s) = 0, \quad u(s', s) = \infty$$

$$(s', t); \quad c(s', t) = 1, \quad u(s', t) = \infty$$

那么，最大流问题成为求解从 s' 到 t 的最小成本流问题。这里，设定原网络里 $c(e) = 0 (e \in E)$ ，并取最大流值的适当上界（比如 $\sum_{e \in E} u(e)$ ）为从 s' 出发的流值 f^* 。

最小成本循环流问题

如不考虑特定的节点为源点或者汇点，并在各边 $e \in E$ 上引入流的下界值 $l = (l(e) \mid e \in E)$ ，从而得到的网络 $\mathcal{N} = (G, c, l, u)$ 。

求在这个网络上使成本达到最小的流，即所谓的最小成本循环流问题：

$$\begin{aligned} \min \quad & \sum_{e \in E} c(e)x(e) \\ \text{s.t.} \quad & \sum_{e \in \text{Out}(v)} x(e) - \sum_{e \in \text{In}(v)} x(e) = 0, v \in V \\ & l(e) \leq x(e) \leq u(e), e \in E. \end{aligned} \tag{65}$$

[思考练习题]

最小成本循环流问题的模型化能力与最小成本流问题等价。

最小成本循环流问题

最小成本循环流问题 (65) 相比于最小成本流问题，其约束条件更加简洁，因而具有更清晰的求解算法描述等优点。

[编程练习]

调研最小成本循环流问题的算法研究进展，并选择实现其一求解算法。

本章作业（网络最优化）

- **Exercise 2.1:** 依据对偶理论，给出最短路径问题的一种求解方法。
- **Exercise 2.2:** 证明最大流最小割定理：任意一个流网络的最大流量等于该网络的最小割的容量。
- **Exercise 2.3:** 验证(63)式给出的 x' 是原网络 N 的可行流，并且其流值为 $x'(s) = x(s) + \Delta$.
- **Exercise 2.4:** 通过构造说明最小成本流问题作为其特殊情况包含：最短路问题和最大流问题。
- **Exercise 2.5:** 试证明最小成本循环流问题与最小成本流问题具有等价的模型化能力。

本章作业（网络最优化）

Project 2.6: 调研并实现最小成本循环流问题的强多项式时间求解算法。要求提交程序源代码和程序报告。

程序报告的提纲（包括但不限于）：

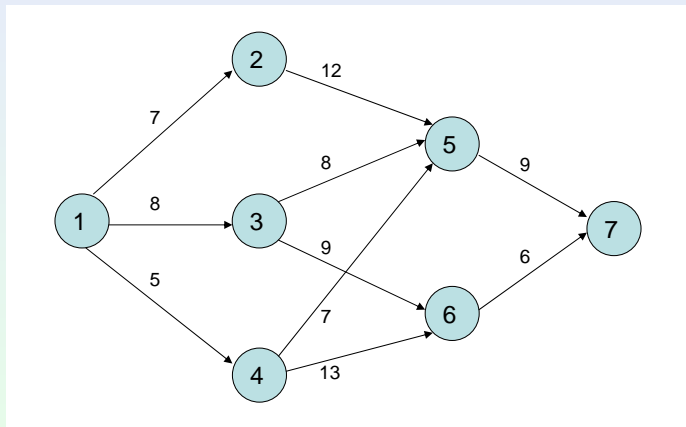
- 问题描述
- 算法原理
- 程序测试
- 调用指南
- 结论

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划**
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

动态规划研究的是“决策过程的最优化”，具有广泛应用背景，并已建立了严密的理论基础。

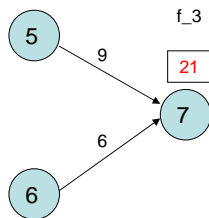
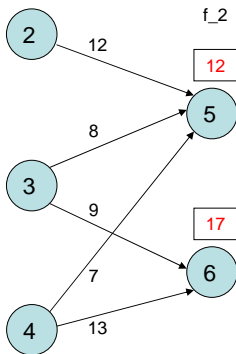
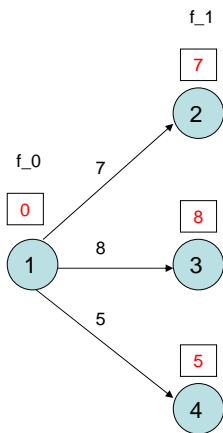
动态规划

最短路径问题的一个例子



动态规划

分阶段考虑



递归方程(Recursive Equation)

$$\begin{cases} f_i(x_i) = \min_{(x_{i-1}, x_i) \in E} \{d(x_{i-1}, x_i) + f_{i-1}(x_{i-1})\}, & i = 1, 2, 3, \\ f_0(x_0) = 0. \end{cases}$$

动态规划的几个关键要素：

- Stages
- Alternatives
- States
- Recursive Equations

背包/货物装载(Knapsack/Cargo-Loading)

A general (n -Item, W -LB) knapsack problem can be represented by the following Integer Linear Programming:

$$\begin{aligned} \max \quad & z = \sum_{i=1}^n r_i m_i \\ \text{s.t.} \quad & \sum_{i=1}^n w_i m_i \leq W \\ & m_1, \dots, m_n \in \mathbb{Z}_+ \cup \{0\} \end{aligned}$$

1. Stage i is represented by item i , $i = 1, 2, \dots, n$.
2. The alternatives at stage i are represented by m_i , the number of units of item i included in the knapsack. It follows that $m_i = 0, 1, \dots, \lfloor \frac{W}{w_i} \rfloor$.
3. The state at stage i is represented by x_i , the total weight assigned to stages (items) $i, i + 1, \dots$, and n . This definition reflects the fact that the weight constraint is the only restriction that links all n stages together.

向后递归(backward recursion)

$$\begin{cases} f_i(x_i) = \max_{\substack{m_i = 0, 1, \dots, \lfloor \frac{W}{w_i} \rfloor \\ x_i \leq W}} \{r_i m_i + f_{i+1}(x_{i+1})\}, & i = n, \dots, 1 \\ f_{n+1}(x_{n+1}) = 0. \end{cases}$$

向后递归(backward recursion)

$$\begin{cases} f_i(x_i) = \max_{\substack{m_i = 0, 1, \dots, \lfloor \frac{W}{w_i} \rfloor \\ x_i \leq W}} \{r_i m_i + f_{i+1}(x_i - w_i m_i)\}, & i = n, \dots, 1 \\ f_{n+1}(x_{n+1}) = 0. \end{cases}$$

例子: 7-ton Vessel

Item i	1	2	3
w_i	2	3	1
r_i	31	47	15

设备更新模型(Equipment Replacement Model)

假设我们考虑的是一个跨度为 n 年的设备更新问题。每年初我们需要决定是否保留当前设备再使用一年或者更换一个新的设备。令 $r(t)$, $c(t)$, $s(t)$ 分别表示一台 t -年龄设备的年营业收入, 年运营成本及其残值。另外, 在规划期内的任何一年购置一台新设备的成本是 l 。

设备更新模型

- Stage i is represented by year i where $i = 1, \dots, n$.
- The alternatives at stage i are either *keeping* or *replacing* the machine at the start of year i .
- The state at stage i is the age of the machine at the start of year i .
- Recursive equation:

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), & \text{if Keeping} \\ r(0) - c(0) + s(t) - I + f_{i+1}(1), & \text{if Replacing} \end{cases}$$
$$f_{n+1}(t) = s(t)$$

where $f_i(t)$ is defined as the maximum net income for years $i, i+1, \dots, n$ by given a t -year-old machine at the start of year i .

动态规划

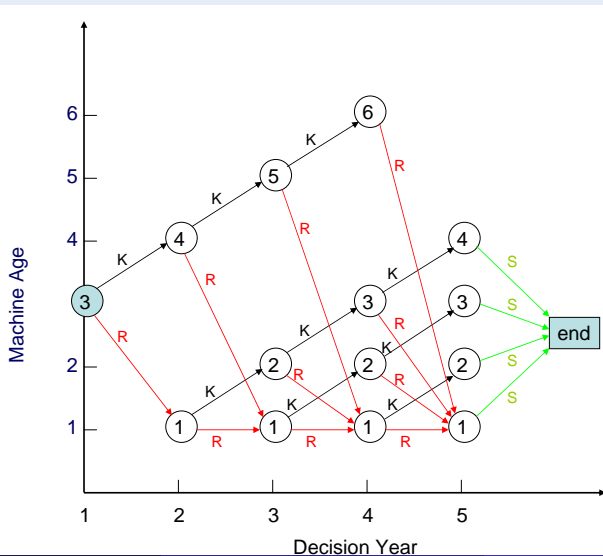
设备更新模型：设某公司现有一台3年龄的设备，需制定一个未来4年($n = 5$)的设备更新最优策略。该公司还规定6年龄的设备必须得以更换。一台新设备的成本是\$100,000. 下表给出的是设备更新问题的相关数据，其中 t 是机器年龄， $r(t)$, $c(t)$, $s(t)$ 分别表示 t 年龄机器的年营业收入，年运营成本及残值。

Table: 设备更新问题数据表

t	$r(t)$	$c(t)$	$s(t)$
0	20,000	200	—
1	19,000	600	80,000
2	18,500	1,200	60,000
3	17,200	1,500	50,000
4	15,500	1,700	30,000
5	14,000	1,800	10,000
6	12,200	2,200	5,000

动态规划

设备更新模型



本章作业（动态规划）

- **Exercise 3.1:** 设现有一台2年龄的设备，另规定5年龄的设备必须更换。在规划期购置新设备的成本分别是

$$(p_1, p_2, p_3, p_4, p_5) = (100, 105, 110, 115, 120).$$

试构建如下设备更新的动态规划模型并求其最优更新策略。

Table: 五年期设备更新

设备年龄 t	残值 v_t	运行费用 c_t
0	-	30
1	50	40
2	25	50
3	10	75
4	5	90
5	2	-

本章作业（动态规划）

- **Exercise 3.2:** 调研分析动态规划问题的求解算法及其时间复杂性。
- **Project 3.3[可选]:** 实现动态规划的一种快速算法并提交程序源代码和程序报告。

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论**
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化
- 9 总结

非线性规划

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & x \in S \subset \mathbb{R}^n.\end{array}\tag{66}$$

在此，目标函数 f 是定义在 \mathbb{R}^n 上的实值函数， S 是决策变量 x 的可取值之集合，称为问题的可行域(feasible region).

非线性规划

最优化问题从属性上可以分为两大类：一类是具有连续变量的问题，另一类是离散变量的问题（即组合优化问题）。

非线性规划属于连续型最优化问题的范畴，通常可行域 S 可由一组方程来描述，即

$$S = \{x \in \mathbb{R}^n \mid g_i(x) \geq 0, i = 1, \dots, m; h_j(x) = 0, j = 1, \dots, \ell\}.$$

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & g_i(x) \geq 0, i = 1, \dots, m \\ & h_j(x) = 0, j = 1, \dots, \ell\end{array} \quad (67)$$

这里, $f(x)$, $g_i(x)$, $h_j(x)$ 都是 n 变量、实值、确定的函数, 且至少有一个是非线性的。

为求解一个非线性规划问题（即找出其最优解），与此相关的研究分两个方面：一是研究最优解的性质，二是设计有效算法来获得问题的解。

回顾最优解的定义

满足约束条件 $x \in S$ 的 x 称为问题的可行解(feasible solution), 如果可行解 $x^* \in S$ 进一步满足

$$f(x^*) \leq f(x), \forall x \in S. \quad (68)$$

则称 x^* 为问题(66)的全局最优解(global optimal solution). 另外, 在包含可行解 $x^* \in S$ 的适当邻域 $U(x^*)$ 里, 成立

$$f(x^*) \leq f(x), \forall x \in S \cap U(x^*). \quad (69)$$

此时称 x^* 为问题(66)的局部最优解(local optimal solution).

最优性条件

最优性条件：问题的最优解所满足的必要或者充分条件。

最优性条件将为各种求解算法的设计、分析提供必不可少的理论基础。

无约束问题的极值条件

一阶必要条件：设目标函数 $f(x)$ 在点 \bar{x} 处可微，若 \bar{x} 是局部极小点，则 $\nabla f(\bar{x}) = 0$.

二阶必要条件：设目标函数 $f(x)$ 在点 \bar{x} 处二次可微，若 \bar{x} 是局部极小点，则 $\nabla f(\bar{x}) = 0$ ，并且Hesse矩阵 $\nabla^2 f(\bar{x}) \geq 0$.

二阶充分条件：设目标函数 $f(x)$ 在点 \bar{x} 处二次可微，若 $\nabla f(\bar{x}) = 0$ 且 $\nabla^2 f(\bar{x}) > 0$ ，则 \bar{x} 是局部极小点。

充要条件：设 $f(x)$ 是定义在 \mathbb{R}^n 上的可微凸函数，则 \bar{x} 是整体极小点（全局最优解）的充要条件是 $\nabla f(\bar{x}) = 0$.

约束问题的最优性条件

可行方向

设 $\bar{x} \in S$, $d \in \mathbb{R}^n$ 是非零向量。若存在 $\delta > 0$ 使得:

$$\bar{x} + \lambda d \in S, \forall \lambda \in (0, \delta) \quad (70)$$

则称 d 是 S 在 \bar{x} 处的可行方向。

记 S 在 \bar{x} 处的所有可行方向的集合为 $F(\bar{x}, S)$.

约束问题的最优性条件

下降方向

设 $f(x)$ 是 \mathbb{R}^n 上的实函数, $\bar{x} \in \mathbb{R}^n$, d 是非零向量。若存在 $\delta > 0$ 使得:

$$f(\bar{x} + \lambda d) < f(\bar{x}), \forall \lambda \in (0, \delta) \quad (71)$$

则称 d 为函数 $f(x)$ 在 \bar{x} 处的下降方向。

下降方向集的子集

如果 $f(x)$ 是可微函数, 且 $\nabla f(\bar{x})^T d < 0$. 显然, 此处的 d 为 $f(x)$ 在 \bar{x} 处的下降方向。记这样的方向集合为

$$D(\bar{x}, f) = \{d \mid \nabla f(\bar{x})^T d < 0\}.$$

结论

对于问题 $\min\{f(x) \mid x \in S\}$, 设 $\bar{x} \in S$, $f(x)$ 在 \bar{x} 处可微。如果 \bar{x} 是问题的局部最优解, 则

$$F(\bar{x}, S) \cap D(\bar{x}, f) = \emptyset. \quad (72)$$

约束问题的最优性条件

$$\mathcal{I}(\bar{x}) = \{i \in \{1, \dots, m\} \mid g_i(\bar{x}) = 0\},$$

$$D_f = D(\bar{x}, f) = \{d \mid \nabla f(\bar{x})^T d < 0\}, \quad (73)$$

$$F_g = F(\bar{x}, g) = \{d \mid \nabla g_i(\bar{x})^T d > 0, i \in \mathcal{I}(\bar{x})\}, \quad (74)$$

$$F_h = F(\bar{x}, h) = \{d \mid \nabla h_j(\bar{x})^T d = 0, j = 1, \dots, \ell\}. \quad (75)$$

设 \bar{x} 为问题(67)的局部最优解, f 和 $g_i, i \in \mathcal{I}(\bar{x})$ 在点 \bar{x} 可微, $g_i, i \notin \mathcal{I}(\bar{x})$ 在点 \bar{x} 连续, h_j 在点 \bar{x} 连续可微, 且 $\{\nabla h_j(\bar{x})\}_{j=1}^\ell$ 线性无关, 则

$$D_f \cap F_g \cap F_h = \emptyset. \quad (76)$$

约束问题的最优性条件

Fritz-John条件：在问题(67)中，设 \bar{x} 为可行点， f 和 $g_i, i \in \mathcal{I}(\bar{x})$ 在点 \bar{x} 可微， $g_i, i \notin \mathcal{I}(\bar{x})$ 在点 \bar{x} 连续， h_j 在点 \bar{x} 连续可微。如果 \bar{x} 是局部最优解，则存在不全为零的数 $\lambda_0, \lambda_i, i \in \mathcal{I}(\bar{x})$ 和 $\mu_j, j = 1, \dots, \ell$ 使得

$$\lambda_0 \nabla f(\bar{x}) - \sum_{i \in \mathcal{I}(\bar{x})} \lambda_i \nabla g_i(\bar{x}) - \sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{x}) = 0. \quad (77)$$

其中 $\lambda_0 \geq 0, \lambda_i \geq 0, i \in \mathcal{I}(\bar{x})$.

约束问题的最优性条件

证明(1): 如果 $\{\nabla h_j(\bar{x})\}_{j=1}^{\ell}$ 线性相关, 则存在不全为零的数 $\mu_j, j = 1, \dots, \ell$ 使得

$$\sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{x}) = 0.$$

这时可令 $\lambda_0 = 0, \lambda_i = 0, i \in \mathcal{I}(\bar{x})$, 结论成立。

约束问题的最优性条件

证明(2): 如果 $\{\nabla h_j(\bar{x})\}_{j=1}^{\ell}$ 线性无关, 则必有 $D_f \cap F_g \cap F_h = \emptyset$.
即不等式组

$$\begin{cases} \nabla f(\bar{x})^T d < 0 \\ \nabla g_i(\bar{x})^T d > 0, i \in \mathcal{I}(\bar{x}) \\ \nabla h_j(\bar{x})^T d = 0, j = 1, \dots, \ell \end{cases} \quad (78)$$

无解。

约束问题的最优性条件

证明(2)续: 令 A 是以 $\{\nabla f(\bar{x}), -\nabla g_i(\bar{x}), i \in \mathcal{I}(\bar{x})\}$ 为列组成的矩阵, B 是以 $\{-\nabla h_j(\bar{x}), j = 1, \dots, \ell\}$ 为列组成的矩阵。
于是得

$$\begin{cases} A^T d < 0 \\ B^T d = 0 \end{cases} \quad (79)$$

无解。

下证

$$\begin{cases} Ap_1 + Bp_2 = 0 \\ p_1 \geq 0 \end{cases} \quad (80)$$

有解。

约束问题的最优性条件

证明(2)续：现定义

$$S_1 = \left\{ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \mid y_1 = A^T d, y_2 = B^T d, d \in \mathbb{R}^n \right\},$$

$$S_2 = \left\{ \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \mid y_1 < 0, y_2 = 0 \right\}.$$

显然 S_1 和 S_2 为非空凸集，且 $S_1 \cap S_2 = \emptyset$.

由凸集分离定理知，对 $\forall d \in \mathbb{R}^n, \forall \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \in CL(S_2)$ ，存在非零向量 $\begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$ 使得 $p_1^T A^T d + p_2^T B^T d \geq p_1^T y_1 + p_2^T y_2$.

约束问题的最优性条件

证明(2)续:

首先令 $y_2 = 0$, 由 d 的任意性 (取 $d = 0$) 及 $y_1 < 0$, $\implies p_1 \geq 0$.

再令 $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \text{CL}(S_2)$, $\implies p_1^T A^T d + p_2^T B^T d \geq 0$.

最后取 $d = -(Ap_1 + Bp_2)$, $\implies Ap_1 + Bp_2 = 0$.

综上所述, 即得(80)有解。

约束问题的最优性条件

证明(2)续:

把 p_1 的分量记作 λ_0 和 $\lambda_i, i \in \mathcal{I}(\bar{x})$, p_2 的分量记作 $\mu_j, j = 1, \dots, \ell$. 立即得到

$$\lambda_0 \nabla f(\bar{x}) - \sum_{i \in I(\bar{x})} \lambda_i \nabla g_i(\bar{x}) - \sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{x}) = 0. \quad (81)$$

约束问题的最优性条件

Kuhn-Tucker条件：设 \bar{x} 为约束问题(67)的可行点， f 和 $g_i, i \in \mathcal{I}(\bar{x})$ 在点 \bar{x} 可微， $g_i, i \notin \mathcal{I}(\bar{x})$ 在点 \bar{x} 连续， h_j 在点 \bar{x} 连续可微，向量集 $\{\nabla g_i(\bar{x}), i \in \mathcal{I}(\bar{x}); \nabla h_j(\bar{x}), j = 1, \dots, \ell\}$ 线性无关。如果 \bar{x} 是局部最优解，则存在数 $\lambda_i \geq 0$ 和 μ_j 使得

$$\nabla f(\bar{x}) - \sum_{i \in \mathcal{I}(\bar{x})} \lambda_i \nabla g_i(\bar{x}) - \sum_{j=1}^{\ell} \mu_j \nabla h_j(\bar{x}) = 0. \quad (82)$$

约束问题的最优性条件

定义Lagrange函数 $L(x, \lambda, \mu) = f(x) - \sum_{i=1}^m \lambda_i g_i(x) - \sum_{j=1}^{\ell} \mu_j h_j(x)$.

若 \bar{x} 为问题局部最优解, 则存在乘子向量 $\bar{\lambda} \geq 0, \bar{\mu}$ 使得

$$\nabla_x L(\bar{x}, \bar{\lambda}, \bar{\mu}) = 0.$$

此时, 一阶必要条件可表达为

$$(K - T) \begin{cases} \nabla_x L(x, \lambda, \mu) = 0 \\ g_i(x) \geq 0, i = 1, \dots, m \\ \lambda_i g_i(x) = 0, i = 1, \dots, m \\ \lambda_i \geq 0, i = 1, \dots, m \\ h_j(x) = 0, j = 1, \dots, \ell \end{cases} \quad (83)$$

理论上可以用最优性条件求“非线性规划问题”的最优解，但在实践中并不切实可行。

在求解最优化问题时最常用的计算方法是“迭代下降算法”。

算法映射：算法 \mathcal{A} 是定义在空间 X 上的**点到集**的映射，即对每一点 $x^{(k)} \in X$ ，经算法 \mathcal{A} 作用后产生一个点集 $\mathcal{A}(x^{(k)}) \subset X$ ，任意选择一个点 $x^{(k+1)} \in \mathcal{A}(x^{(k)})$ 作为 $x^{(k)}$ 的后续点。

引入所谓算法的闭性，其实质是点到点映射的连续性的推广。

设 X 和 Y 分别是空间 \mathbb{E}^p 和 \mathbb{E}^q 中的非空闭集， $\mathcal{A} : X \rightarrow Y$ 为点到集的映射。如果 $x^{(k)} \in X, x^{(k)} \rightarrow x, y^{(k)} \in \mathcal{A}(x^{(k)}), y^{(k)} \rightarrow y$ 蕴涵着 $y \in \mathcal{A}(x)$ ，则称映射 \mathcal{A} 在 $x \in X$ 处是闭的。

为研究算法的收敛性，首先要明确解集合的概念。

在许多情况下，要使算法产生的点列收敛于全局最优解是极为困难的。因此，一般把满足某些条件的点集定义为解集合。当迭代点属于这个集合时，就停止迭代。

例如，在无约束最优化问题中，可以定义解集合为

$$\Omega = \{\bar{x} \mid \|\nabla f(\bar{x})\| = 0\},$$

在约束最优化问题中，解集合取为

$$\Omega = \{\bar{x} \mid \bar{x} \text{ 是 K-T 点}\}.$$

下降函数

一般地，下降算法总是与某个函数在迭代过程中函数值的减小联系在一起，因此需要给出下降函数的概念。

设 $\Omega \subset X$ 为解集合， \mathcal{A} 为 X 上的一个算法映射， $\psi(x)$ 是定义在 X 上的连续实函数，若满足

$$\text{当 } x \notin \Omega \text{ 且 } y \in \mathcal{A}(x) \text{ 时, } \psi(y) < \psi(x)$$

$$\text{当 } x \in \Omega \text{ 且 } y \in \mathcal{A}(x) \text{ 时, } \psi(y) \leq \psi(x)$$

则称 ψ 是关于解集合 Ω 和算法 \mathcal{A} 的下降函数。

算法收敛性

设 Ω 为解集合， \mathcal{A} 为 X 上的算法映射。若以任意初始点 $x^{(0)} \in Y \subset X$ 出发，算法产生的序列的任一收敛子列的极限属于解集合，则称算法映射 \mathcal{A} 在 Y 上收敛于解集合 Ω .

算法收敛性

定理： 设 \mathcal{A} 为 X 上的一个算法， Ω 为解集合，给定初始点 $x^{(0)} \in X$ ，进行如下迭代：如果 $x^{(k)} \in \Omega$ ，则停止迭代；否则取 $x^{(k+1)} \in \mathcal{A}(x^{(k)})$ ， $k := k + 1$ ，重复以上过程。这样产生迭代序列 $\{x^{(k)}\}$ 。又设：

- ① 序列 $\{x^{(k)}\}$ 含于 X 的紧子集中；
- ② 存在一个连续函数 ψ ，它是关于 Ω 和 \mathcal{A} 的下降函数；
- ③ 映射 \mathcal{A} 在 Ω 的补集上是闭的。

则序列 $\{x^{(k)}\}$ 的任一收敛子列的极限属于 Ω 。

算法收敛性定理证明

证明： 先证序列 $\{x^{(k)}\}$ 对应的下降函数值数列 $\{\psi(x^{(k)})\}$ 有极限。

$\{x^{(k)}\}$ 含于 X 的紧子集，因此有收敛子列 $\{x^{(k)}\}_K$ ，设其极限为 $x \in X$ 。
由 ψ 的连续性得， $\psi(x^{(k)}) \rightarrow \psi(x), k \in K$ 。即对 $\forall \epsilon > 0, \exists N$ 使得
当 $k \geq N$ 时，有 $0 < \psi(x^{(k)}) - \psi(x) < \epsilon, k \in K$ 。特别地，
 $0 < \psi(x^{(N)}) - \psi(x) < \epsilon$ 。

又由 ψ 的下降性知， $\psi(x^{(k)}) - \psi(x^{(N)}) < 0, \forall k > N$ 。于是有

$$0 < \psi(x^{(k)}) - \psi(x) = \psi(x^{(k)}) - \psi(x^{(N)}) + \psi(x^{(N)}) - \psi(x) < \epsilon, \forall k > N.$$

即得 $\lim_{k \rightarrow \infty} \psi(x^{(k)}) = \psi(x)$ 。

算法收敛性定理证明

证明（继）： 下证 $x \in \Omega$ （反证法）。

假设 $x \notin \Omega$, 考虑序列 $\{x^{(k+1)}\}_K$. 由于它包含于紧集, 所以也存在收敛子列 $\{x^{(k+1)}\}_{\bar{K}}, \bar{K} \subset K$, 且设其极限为 $\bar{x} \in X$. 显然（同理前述证明）
 $\lim_{k \rightarrow \infty} \psi(x^{(k+1)}) = \psi(\bar{x})$. 由 $\psi(x^{(k)})$ 极限的唯一性知,

$$\psi(x) = \psi(\bar{x}).$$

另外, 对 $k \in \bar{K} \subset K$ 有

$$x^{(k)} \longrightarrow x, x^{(k+1)} \in \mathcal{A}(x^{(k)}), x^{(k+1)} \longrightarrow \bar{x}.$$

算法 \mathcal{A} 在 Ω 的补集上是闭的, $x \notin \Omega$, 因此 \mathcal{A} 在 x 处是闭的, 即有 $\bar{x} \in \mathcal{A}(x)$.

由于 ψ 是解集合 Ω 和算法 \mathcal{A} 的下降函数, $x \notin \Omega$, 则有 $\psi(\bar{x}) < \psi(x)$. 这显然矛盾, 所以 $x \in \Omega$.

实用收敛准则

在迭代下降算法里，当 $x^{(k)} \in \Omega$ 时才终止迭代。在实践中许多情况下，这是一个取极限的过程，需要无限次迭代。因此为了解决实际问题，需要规定一些实用的终止迭代过程的准则，一般称为收敛准则或停机准则。

常用的收敛准则有以下几种：

- ① $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$ or $\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} < \varepsilon$
- ② $f(x^{(k)}) - f(x^{(k+1)}) < \varepsilon$ or $\frac{f(x^{(k)}) - f(x^{(k+1)})}{|f(x^{(k)})|} < \varepsilon$
- ③ $\|\nabla f(x^{(k)})\| < \varepsilon$

在这里， ε 为事先给定的充分小的正数。除此之外，还可以根据收敛定理，制定出其它的收敛准则。

收敛速率

评价算法优劣的标准之一是收敛的快慢，通常称为收敛速率。

一般定义如下：设序列 $\{x^{(k)}\}$ 收敛与 x^* ，满足

$$0 \leq \lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|^p} = \beta < \infty \quad (84)$$

的非负数 p 的上确界称为序列 $\{x^{(k)}\}$ 的收敛阶。

若在定义式(84)中， $p = 1$ 且 $\beta < 1$ ，则称序列是（收敛比 β ）线性收敛的。

若在定义式(84)中， $p > 1$ ，或者 $\{p = 1, \beta = 0\}$ ，则称序列是超线性收敛的。

最优化可以追溯到十分古老的极值问题，然而它成为一门独立的学科是在二十世纪四十年代末，当时Dantzig提出了求解一般线性规划问题的单纯形法。此后各种最优化问题的理论及应用研究得到迅速发展，特别是线性规划由于其模型的普遍性和实用性，相关算法的进展引起广泛的重视。

随着实际问题的规模越来越大以及在计算机技术的推动下，人们开始从复杂性角度研究线性规划和非线性规划的算法。

迭代方法

最优化方法通常采用迭代方法求问题的最优解，其基本思想是：

给定一个初始点 $x^{(0)} \in \mathbb{R}^n$ ，按照某一迭代规则产生一个点列 $\{x^{(k)}\}$ ，使得当 $\{x^{(k)}\}$ 是有穷点列时，其最后一个点是最优化模型问题的最优解，当 $\{x^{(k)}\}$ 是无穷点列时，它有极限点且其极限点是最优化模型问题的最优解。

一个好的迭代算法应具备的典型特征是：

迭代点 $x^{(k)}$ 能稳定地接近局部极小点 x^* 的小邻域，然后迅速收敛于 x^* 。一般地，对于某种算法我们需要证明其迭代点列 $x^{(k)}$ 的聚点（即子列的极限点）为一局部极小点。在实际计算中，当指定的收敛准则满足时，迭代即终止。

搜索方向与步长因子

设 $x^{(k)}$ 为第 k 次迭代点， $d^{(k)}$ 为第 k 次搜索方向， α_k 为第 k 次步长因子，则第 $k+1$ 次迭代为：

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}. \quad (85)$$

从上述迭代格式可以看出，不同的搜索方向和不同的步长策略构成不同的方法。

搜索方向与步长因子

在最优化方法中，搜索方向 $d^{(k)}$ 一般选取的是某价值函数 (merit function) ψ 在 $x^{(k)}$ 处的下降方向，即 $d^{(k)}$ 满足

$$\nabla\psi(x^{(k)})^T d^{(k)} < 0. \quad (86)$$

步长因子的确定一般归结为解一维最优化问题

$$\min_{\alpha \geq 0} \psi(x^{(k)} + \alpha d^{(k)}) \quad (87)$$

最优化迭代算法的基本结构之一

- (a) 给定初始点 $x^{(0)}$
- (b) 计算搜索方向 $d^{(k)}$, 即构造某价值函数 ψ 在 $x^{(k)}$ 点处的下降方向作为搜索方向;
- (c) 确定步长因子 α_k , 使该价值函数值有某种程度的下降;
- (d) 迭代更新, 令 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$.
- (e) 判断停机准则, 若 $x^{(k+1)}$ 满足某种终止条件, 则停止迭代, 得到近似最优解 $\bar{x} = x^{(k+1)}$. 否则, 返回(b)重复以上步骤。

最优化迭代算法的基本结构之二

- (a) 给定初始点 $x^{(0)}$
- (b) 构造某价值函数 ψ 在 $x^{(k)}$ 附近（如一定半径内）的二次近似模型；
- (c) 求解该近似模型得到 $s^{(k)}$ 作为更新位移向量；
- (d) 迭代更新，令 $x^{(k+1)} = x^{(k)} + s^{(k)}$.
- (e) 判断停机准则，若 $x^{(k+1)}$ 满足某种终止条件，则停止迭代，得到近似最优解 $\bar{x} = x^{(k+1)}$. 否则，返回(b)重复以上步骤。

- ① 绪论
- ② 线性规划
- ③ 网络最优化
- ④ 动态规划
- ⑤ 非线性规划基础理论
- ⑥ 无约束最优化**
- ⑦ 二次规划
- ⑧ 非线性约束最优化
- ⑨ 总结

无约束最优化问题

$$\min_{x \in \mathbb{R}^n} f(x) \quad (88)$$

其目标函数 f 是定义在 \mathbb{R}^n 上的实值函数，决策变量 x 的可取值之集合是空间 \mathbb{R}^n .

梯度类求解算法

梯度向量 $\nabla f(x)$ 是函数 f 在点 x 处增加最快的方向，故它成为最优化时的重要工具。实际上针对无约束最优化问题，大家所知的求解算法中大多属于下面的梯度方法类。

GRADIENT（梯度法类）

- (0) 初始化：选取适当的初始点 $x^{(0)} \in \mathbb{R}^n$ ，令 $k := 0$ 。
- (1) 计算搜索方向：利用适当的正定对称阵 H_k 计算搜索方向向量 $d^{(k)} := -H_k \nabla f(x^{(k)})$ 。（如果 $\nabla f(x^{(k)}) = 0$ ，则结束计算）
- (2) 确定步长因子：解一维最优化问题 $\min_{\alpha \geq 0} f(x^{(k)} + \alpha d^{(k)})$ ，求出步长 $\alpha = \alpha_k$ ，令 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$ ， $k := k + 1$ ，回到第(1)步。

构造搜索方向：负梯度方向

无约束最优化问题： $\min_{x \in \mathbb{R}^n} f(x)$

$$f(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + O(\|x - x^{(k)}\|^2) \quad (89)$$

取负梯度方向

$$d^{(k)} = -\nabla f(x^{(k)}),$$

则当 α_k 足够小时，总能使

$$f(x^{(k)} + \alpha_k d^{(k)}) < f(x^{(k)}).$$

构造搜索方向：牛顿方向

$$\begin{aligned} f(x) = & f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) \\ & + \frac{1}{2} (x - x^{(k)})^T \nabla^2 f(x^{(k)}) (x - x^{(k)}) + O(\|x - x^{(k)}\|^3) \end{aligned} \quad (90)$$

取搜索方向

$$d^{(k)} = -G_k^{-1} \nabla f(x^{(k)}),$$

其中 $G_k = \nabla^2 f(x^{(k)})$ 为函数 f 在 $x^{(k)}$ 点处的Hesse矩阵。

确定步长因子：一维搜索

在迭代格式中，通过解一维最优化问题

$$\min_{\alpha \geq 0} \varphi(\alpha) = f(x^{(k)} + \alpha d^{(k)}) \quad (91)$$

确定步长因子的方法称为**一维搜索**(Line Search).

确定步长因子：一维搜索

若以问题(91)的最优解为步长，此时称为**精确一维搜索**(Exact Line Search).

经常用到的精确一维搜索有黄金分割法和插值迭代法。即使说是精确一维搜索，通过有限次计算求出问题(91)的严密解一般也是不可能的，实际上在得到有足够精度的近似解时，就采用它作为步长。

确定步长因子：一维搜索

在实际计算中，往往不是求解一维最优化问题(91)，而是找出满足某些适当条件的粗略近似解作为步长，此时称为**非精确一维搜索**(Inexact Line Search)。

与精确一维搜索相比，在很多情况下采用非精确一维搜索可以提高整体计算效率。

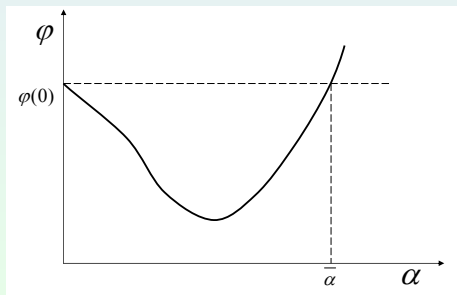
确定步长因子：一维搜索

设 $\bar{\alpha}_k$ 是使得

$$f(x^{(k)} + \alpha d^{(k)}) = f(x^{(k)})$$

的最小正数 α .

于是，我们将在区间 $[0, \bar{\alpha}_k]$ 内求得满足适当条件的可接受的步长因子，即 $\alpha \in [0, \bar{\alpha}_k]$.



确定步长因子：一维搜索

Goldstein(1965) conditions:

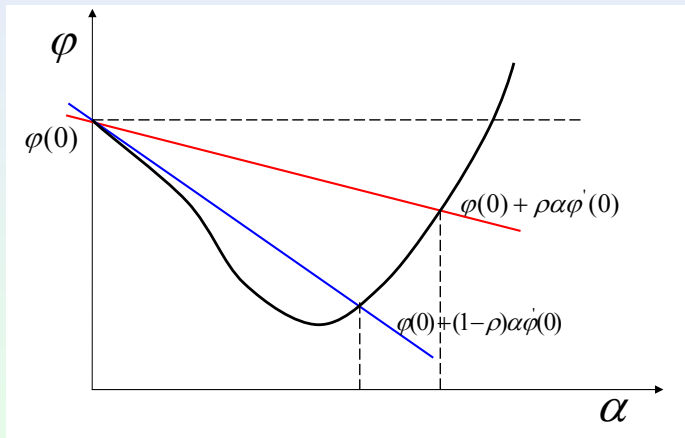
$$\varphi(\alpha) \leq \varphi(0) + \rho\alpha\varphi'(0) \quad (92)$$

$$\varphi(\alpha) \geq \varphi(0) + (1 - \rho)\alpha\varphi'(0) \quad (93)$$

其中 $\rho \in (0, 1/2)$ 是一个固定参数。

确定步长因子：一维搜索

Goldstein(1965) conditions:



确定步长因子：一维搜索

Wolfe(1968)-Powell(1976) conditions:

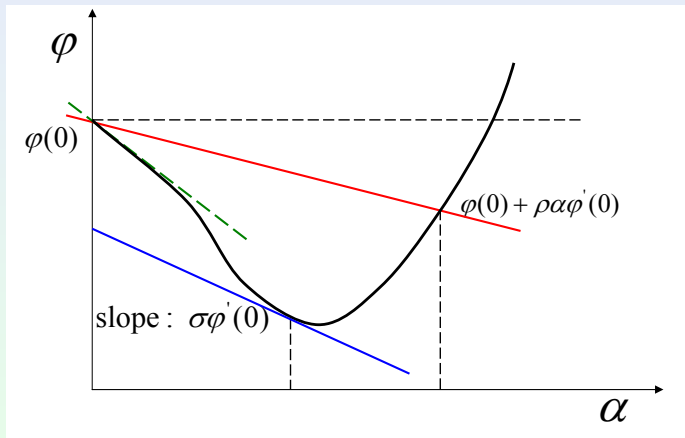
$$\varphi(\alpha) \leq \varphi(0) + \rho\alpha\varphi'(0) \quad (94)$$

$$\varphi'(\alpha) \geq \sigma\varphi'(0) \quad (95)$$

其中 $\sigma \in (\rho, 1)$ 是另一个固定参数。

确定步长因子：一维搜索

Wolfe(1968)-Powell(1976) conditions:



确定步长因子：一维搜索

在很多实际算法中，式(95)常被强化的双边条件所取代

$$|\varphi'(\alpha)| \leq -\sigma\varphi'(0) \quad (96)$$

确定步长因子：一维搜索

基于Wolfe-Powell准则的非精确一维搜索算法：

- (0) 给定初始一维搜索区间 $[0, \bar{\alpha}]$, 以及 $\rho \in (0, 1/2)$, $\sigma \in (\rho, 1)$.
计算 $\varphi_0 = \varphi(0) = f(\mathbf{x}^{(k)})$, $\varphi'_0 = \varphi'(0) = \nabla f(\mathbf{x}^{(k)})^T \mathbf{d}^{(k)}$.
并令 $a_1 = 0, a_2 = \bar{\alpha}, \varphi_1 = \varphi_0, \varphi'_1 = \varphi'_0$.
选取适当的 $\alpha \in (a_1, a_2)$.

确定步长因子：一维搜索

基于Wolfe-Powell准则的非精确一维搜索算法：

- (1) 计算 $\varphi = \varphi(\alpha) = f(x^{(k)} + \alpha d^{(k)})$. 若 $\varphi(\alpha) \leq \varphi(0) + \rho \alpha \varphi'(0)$, 则转到第(2)步。否则, 由 $\varphi_1, \varphi'_1, \varphi$ 构造两点二次插值多项式 $p^{(1)}(t)$, 并得其极小点

$$\hat{\alpha} = a_1 + \frac{1}{2} \frac{(a_1 - \alpha)^2 \varphi'_1}{(\varphi_1 - \varphi) - (a_1 - \alpha) \varphi'_1}.$$

于是置 $a_2 = \alpha, \alpha = \hat{\alpha}$, 重复第(1)步。

确定步长因子：一维搜索

基于Wolfe-Powell准则的非精确一维搜索算法：

- (2) 计算 $\varphi' = \varphi'(\alpha) = \nabla f(x^{(k)} + \alpha d^{(k)})^T d^{(k)}$. 若 $\varphi'(\alpha) \geq \sigma \varphi'(0)$, 则输出 $\alpha_k = \alpha$, 并停止搜索。否则, 由 $\varphi, \varphi', \varphi'_1$ 构造两点二次插值多项式 $p^{(2)}(t)$, 并得其极小点

$$\hat{\alpha} = \alpha - \frac{(a_1 - \alpha)\varphi'}{\varphi'_1 - \varphi'}.$$

于是置 $a_1 = \alpha, \alpha = \hat{\alpha}, \varphi_1 = \varphi, \varphi'_1 = \varphi'$, 返回第(1)步。

确定步长因子：一维搜索

[思考题：请写出上述基于Wolfe-Powell准则的非精确一维搜索算法中插值多项式 $p^{(1)}(t)$, $p^{(2)}(t)$ 的具体表达式。]

从任意初始点出发，如果某迭代算法产生的点列的极限（聚点），在适当假定下可保证恒为问题的最优解（或者稳定点），则称该迭代法具有全局收敛性(Global Convergence).

与此相对，如果仅在解的附近选取初始点时，才可以保证所生成的点列收敛于该解，则称这样的迭代法有局部收敛性(Local Convergence).

为了证明迭代法的下降性，我们应尽量避免搜索方向与负梯度方向几乎正交的情形，即要求 $d^{(k)}$ 偏离 $g^{(k)} = \nabla f(x^{(k)})$ 的正交方向远一些。否则， $g^{(k)T} d^{(k)}$ 接近于零， $d^{(k)}$ 几乎不是下降方向。

为此，我们假设 $d^{(k)}$ 与 $-g^{(k)}$ 的夹角 θ_k 满足

$$\theta_k \leq \frac{\pi}{2} - \mu, \quad \forall k \quad (97)$$

其中 $\mu > 0$ （与 k 无关）。

显然 $\theta_k \in [0, \pi/2)$, 其定义为

$$\cos \theta_k = \frac{-\mathbf{g}^{(k)T} \mathbf{d}^{(k)}}{\|\mathbf{g}^{(k)}\| \|\mathbf{d}^{(k)}\|} = \frac{-\mathbf{g}^{(k)T} \mathbf{s}^{(k)}}{\|\mathbf{g}^{(k)}\| \|\mathbf{s}^{(k)}\|} \quad (98)$$

这里 $\mathbf{s}^{(k)} = \alpha_k \mathbf{d}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$.

下面给出各种步长准则下的下降算法的全局收敛性结论。

全局收敛性定理：

设 $\nabla f(x)$ 在水平集 $L(x^{(0)}) = \{x \mid f(x) \leq f(x^{(0)})\}$ 上存在且一致连续。下降算法的搜索方向 $d^{(k)}$ 与 $-\nabla f(x^{(k)})$ 之间的夹角 θ_k 满足式(97), 其中步长 α_k 由三种方法之一确定：

- (1) 精确一维搜索
- (2) Goldstein准则 (92),(93)
- (3) Wolfe-Powell准则 (94),(95)

那么, 或者对某个 k 有 $\nabla f(x^{(k)}) = 0$, 或者 $f(x^{(k)}) \rightarrow -\infty$, 或者 $\nabla f(x^{(k)}) \rightarrow 0$.

全局收敛性证明：（只证明Wolfe-Powell准则的情形）

假设对所有的 k , $g^{(k)} = \nabla f(x^{(k)}) \neq 0$ 和 $f(x^{(k)})$ 有下界,
故 $f(x^{(k)}) - f(x^{(k+1)}) \rightarrow 0$. 由式(94)得, $-g^{(k)T}s^{(k)} \rightarrow 0$.

（反证）若 $g^{(k)} \rightarrow 0$ 不成立, 那么存在 $\varepsilon > 0$ 和子列 $\{x^{(k)}\}_{k \in K}$ 使得 $\|g^{(k)}\| \geq \varepsilon$. 从而由

$$-g^{(k)T}s^{(k)} = \|g^{(k)}\| \|s^{(k)}\| \cos \theta_k \geq \varepsilon \|s^{(k)}\| \sin \mu$$

以及式(97)有 $\|s^{(k)}\| \rightarrow 0$.

全局收敛性证明（续）：

又因为 $g(x) = \nabla f(x)$ 在 $L(x^{(0)})$ 上一致连续，所以

$$\begin{aligned} g^{(k+1)T} s^{(k)} &= g^{(k)T} s^{(k)} + o(\|s^{(k)}\|) \\ &\Downarrow \\ \frac{g^{(k+1)T} s^{(k)}}{g^{(k)T} s^{(k)}} &\rightarrow 1. \end{aligned} \tag{99}$$

全局收敛性证明（续）：

而这与Wolfe-Powell准则的式(95)

$$\frac{g^{(k+1)T} s^{(k)}}{g^{(k)T} s^{(k)}} \leq \sigma < 1 \quad (100)$$

相矛盾。因此有 $g^{(k)} \rightarrow 0$

[思考题：请补充证明基于Goldstein准则的非精确一维搜索算法的全局收敛性。]

最速下降法

最速下降法取负梯度作为迭代算法的搜索方向，其迭代格式为

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)}).$$

最速下降法

算法:

- (0) 选取初始点 $x^{(0)}$, 设置终止误差 $\varepsilon > 0$, 令 $k := 0$.
- (1) 计算 $g^{(k)} = \nabla f(x^{(k)})$. 若 $\|g^{(k)}\| < \varepsilon$, 则停止迭代并输出 $x^{(k)}$.
否则进行第(2)步。
- (2) 令 $d^{(k)} = -g^{(k)}$, 并由一维搜索确定步长因子 α_k 使得

$$f(x^{(k)} + \alpha_k d^{(k)}) = \min_{\alpha > 0} f(x^{(k)} + \alpha d^{(k)}).$$

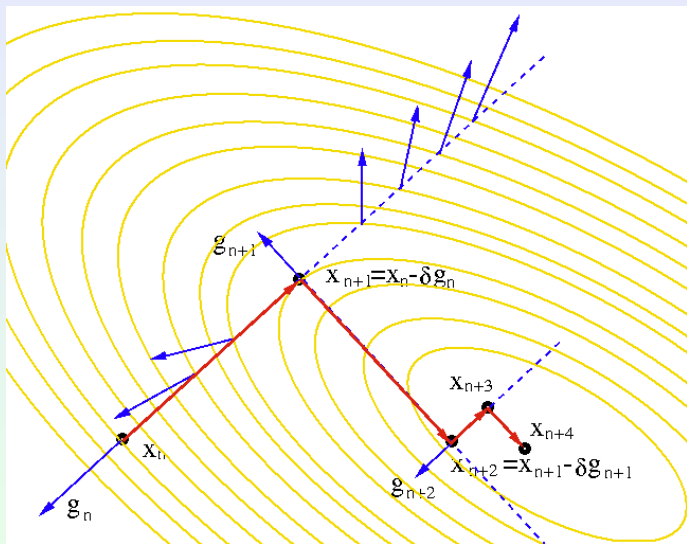
- (3) 迭代更新 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$, 置 $k := k + 1$, 回到第(1)步。

最速下降法全局收敛性定理:

设 $f(x) \in C^1$, 在最速下降法中采用（精确或非精确）一维搜索, 则产生的迭代点列 $\{x^{(k)}\}$ 的每一个聚点都是驻点。

Project: 编程实现基于Wolfe-Powell准则的非精确一维搜索算法。要求提交程序源代码和程序报告。

最速下降法



一般地，最速下降法只有线性收敛速度。

如下例子是一个非常著名的测试函数 (Rosenbrock function)

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

设 $f(x)$ 是二次可微实函数，在 $x^{(k)}$ 附近作二阶Taylor展开近似

$$f(x^{(k)} + s) \approx q^{(k)}(s) = f(x^{(k)}) + g^{(k)T}s + \frac{1}{2}s^T G_k s \quad (101)$$

其中 $g^{(k)} = \nabla f(x^{(k)})$, $G_k = \nabla^2 f(x^{(k)})$.

将 $q^{(k)}(s)$ 极小化便得

$$s = -G_k^{-1}g^{(k)}. \quad (102)$$

上式给出的搜索方向 $-G_k^{-1}g^{(k)}$ 称为牛顿方向(Newton Direction).

在目标函数是正定二次函数

$$f(x) = \frac{1}{2}x^T Gx - c^T x$$

的情况下(G 为正定阵), 对任意的 x 有 $\nabla^2 f(x) = G$.

在第一次迭代里令 $H_0 = G^{-1}$, 则有

$$d^{(0)} = -H_0 \nabla f(x^{(0)}) = -G^{-1}(Gx^{(0)} - c) = -(x^{(0)} - x^*).$$

这里, $x^* = G^{-1}c$ 是问题的最优解。若 $x^{(0)} \neq x^*$, 取步长 $\alpha_0 = 1$, 于是得 $x^{(1)} = x^{(0)} + \alpha_0 d^{(0)} = x^*$. 由此知道, 不管初始点 $x^{(0)}$ 如何取, 在一次迭代后即可到达最优解 x^* .

根据以上事实，可以认为即使对于一般的非线性函数 $f(x)$ ，在迭代中令搜索方向

$$d^{(k)} = -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$$

也是较合适的。

特别地，步长 $\alpha_k \equiv 1$ 的迭代公式为

$$x^{(k+1)} = x^{(k)} + d^{(k)} = x^{(k)} - G_k^{-1} g^{(k)}. \quad (103)$$

这就是经典的牛顿迭代法

对于正定二次函数而言，牛顿法一步即可达到最优解。对于非二次函数，牛顿法并不能保证经有限次迭代求得最优解。但由于目标函数在极小点附近可用二次函数较好地近似，故当初始点靠近极小点时，牛顿法的收敛速度一般会很快。

可以证明牛顿法的局部收敛性和二阶收敛速率。

牛顿法收敛定理:

设 $f \in C^2$, $x^{(k)}$ 充分靠近 x^* , 其中 $\nabla f(x^*) = 0$. 如果 $\nabla^2 f(x^*)$ 正定, 目标函数的Hesse矩阵 $G(x)$ 满足Lipschitz条件, 即存在 $\beta > 0$ 使得对所有 (i, j) 有

$$|G_{ij}(x) - G_{ij}(y)| \leq \beta \|x - y\|. \quad (104)$$

则对一切的 k , 牛顿迭代(103)有定义, 所得序列 $\{x^{(k)}\}$ 收敛到 x^* , 且具有二阶收敛速率。

证明一：

记 $g(x) = \nabla f(x)$, 因为 $f \in C^2$, 我们有

$$g(x - h) = g(x) - G(x)h + O(\|h\|^2).$$

令 $x = x^{(k)}$, $h = h^{(k)} = x^{(k)} - x^*$ 代入上式得

$$0 = g(x^*) = g(x^{(k)} - h^{(k)}) = g(x^{(k)}) - G(x^{(k)})h^{(k)} + O(\|h^{(k)}\|^2). \quad (105)$$

证明一（续）：

由于 $G(x)$ 满足Lipschitz条件，易证 $[G(x^{(k)})]^{-1}$ 有界。方程(105)两边同时乘以 $[G(x^{(k)})]^{-1}$ 得

$$\begin{aligned} 0 &= [G(x^{(k)})]^{-1} g(x^{(k)}) - h^{(k)} + O(\|h^{(k)}\|^2) \\ &= x^* - (x^{(k)} - [G(x^{(k)})]^{-1} g(x^{(k)})) + O(\|h^{(k)}\|^2) \\ &= x^* - x^{(k+1)} + O(\|h^{(k)}\|^2) \\ &= -h^{(k+1)} + O(\|h^{(k)}\|^2) \end{aligned}$$

所以 $\|h^{(k+1)}\| = O(\|h^{(k)}\|^2)$ ，即牛顿迭代法具有二阶收敛速率。

证明二:

对于牛顿迭代法, 我们记

$$x^{(k+1)} = x^{(k)} - G_k^{-1}g^{(k)} \triangleq \mathcal{A}(x^{(k)}). \quad (106)$$

注意到 $g(x^*) = 0$, $G(x^*)$ 正定 (非奇异), 有 $\mathcal{A}(x^*) = x^*$.

于是由 $x^{(k+1)} - x^* = \mathcal{A}(x^{(k)}) - \mathcal{A}(x^*)$ 得

$$\begin{aligned} \|x^{(k+1)} - x^*\| &= \|\mathcal{A}(x^{(k)}) - \mathcal{A}(x^*)\| \\ &\leq \|\mathcal{A}'(x^*)(x^{(k)} - x^*)\| + \frac{1}{2}\|\mathcal{A}''(\bar{x})\|\|x^{(k)} - x^*\|^2, \end{aligned}$$

其中 \bar{x} 位于 $x^{(k)}$ 和 x^* 之间的线段上。

证明二（续）：

显然

$$\mathcal{A}'(x) = [x - G(x)^{-1}g(x)]' = -[G(x)^{-1}]'g(x)$$

所以 $\mathcal{A}'(x^*) = 0$. 从而有

$$\|h^{(k+1)}\| = \|x^{(k+1)} - x^*\| \leq \gamma \|x^{(k)} - x^*\|^2 = \gamma \|h^{(k)}\|^2$$

其中常数 γ 仅依赖于 $f(x)$ 在 x^* 附近的三阶导数。

在式(103)的牛顿迭代法里，如果选取的初始点 $x^{(0)}$ 不在解 x^* 的附近，那么生成的点列 $\{x^{(k)}\}$ 未必收敛于最优解。

为保证算法的全局收敛性，有必要对牛顿法作某些改进。

比如，在牛顿法中也可采用一维搜索来确定步长。

阻尼牛顿法：

- (0) 选取初始点 $x^{(0)}$ ，设置终止误差 $\varepsilon > 0$ ，令 $k := 0$ 。
- (1) 计算 $g^{(k)} = \nabla f(x^{(k)})$ 。若 $\|g^{(k)}\| < \varepsilon$ ，停止迭代并输出 $x^{(k)}$ 。
否则进行第(2)步。
- (2) 解线性方程组 $G_k d = -g^{(k)}$ ，求出牛顿方向 $d^{(k)}$ 。
- (3) 采用一维搜索确定步长因子 α_k ，令 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$ ，
置 $k := k + 1$ ，回到第(1)步。

牛顿法面临的主要困难是Hesse矩阵 $G_k = \nabla^2 f(x^{(k)})$ 不正定。这时二阶近似模型不一定有极小点，即二次函数 $q^{(k)}(s)$ 是无界的。

为了克服这些困难，人们提出了很多修正措施。

Goldstein & Price (1967)

$$d^{(k)} = \begin{cases} -G_k^{-1}g^{(k)}, & \text{if } \cos \theta_k > \eta \\ -g^{(k)}, & \text{otherwise} \end{cases} \quad (107)$$

Levenberg(1944), Marquardt(1963), Goldfeld et. al(1966)

$$(G_k + \mu_k I)d^{(k)} = -g^{(k)} \quad (108)$$

设 x 是函数 f 的一个不定点，若方向 d 满足

$$d^T \nabla^2 f(x) d < 0,$$

则称 d 为 f 在 x 处的负曲率方向。

当Hesse矩阵 $\nabla^2 f(x^{(k)})$ 不正定时，负曲率方向法是修正牛顿法的另一种途径。

牛顿法的突出优点是局部收敛很快（具有二阶收敛速率），但运用牛顿法需要计算二阶导，而且目标函数的Hesse矩阵 $\nabla^2 f(x^{(k)})$ 可能非正定，甚至奇异。为了克服这些缺点，人们提出了拟牛顿法。其基本思想是：用不含二阶导数的矩阵 H_k 近似牛顿法中的Hesse矩阵的逆 $G(x^{(k)})^{-1}$ 。

由构造近似矩阵的方法不同，将出现不同的拟牛顿法。

回顾牛顿法的迭代

$$\begin{cases} G_k \mathbf{d} = -\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)} \end{cases}$$

为了构造Hesse矩阵逆 G_k^{-1} 的近似 H_k , 我们先分析二阶导 $\nabla^2 f(\mathbf{x}^{(k)})$ 与一阶导 $\nabla f(\mathbf{x}^{(k)})$ 的关系。

设第 k 次迭代后得到 $x^{(k+1)}$, 将目标函数 $f(x)$ 在 $x^{(k+1)}$ 处二阶Taylor展开:

$$f(x) \approx f(x^{(k+1)}) + \nabla f(x^{(k+1)})^T (x - x^{(k+1)}) + \frac{1}{2} (x - x^{(k+1)})^T \nabla^2 f(x^{(k+1)}) (x - x^{(k+1)}),$$

进一步有

$$\nabla f(x) \approx \nabla f(x^{(k+1)}) + \nabla^2 f(x^{(k+1)}) (x - x^{(k+1)}),$$

于是令 $x = x^{(k)}$ 得

$$\nabla f(x^{(k)}) \approx \nabla f(x^{(k+1)}) + \nabla^2 f(x^{(k+1)}) (x^{(k)} - x^{(k+1)}).$$

记 $s^{(k)} = x^{(k+1)} - x^{(k)}$, $y^{(k)} = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$, 则有

$$\nabla^2 f(x^{(k+1)})s^{(k)} \approx y^{(k)} \quad \text{or} \quad \nabla^2 f(x^{(k+1)})^{-1}y^{(k)} \approx s^{(k)}.$$

这样, 计算出 $s^{(k)}$ 和 $y^{(k)}$ 后, 可依上式估计在 $x^{(k+1)}$ 处的 Hesse 矩阵的逆。我们有理由要求在迭代中构造出 Hesse 矩阵逆的近似 H_{k+1} , 使其满足

$$H_{k+1}y^{(k)} = s^{(k)}. \quad (109)$$

通常把式(109)称作正割条件, 也称为拟牛顿条件。

拟牛顿迭代算法的一般格式:

- (0) 选取初始点 $x^{(0)}$, 令 $H_0 = I$, $k := 0$.
- (1) 计算搜索方向 $d^{(k)} = -H_k \nabla f(x^{(k)})$.
- (2) 采用一维搜索确定步长因子 α_k , 令 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$.
- (3) 基于 $x^{(k)}$ 到 $x^{(k+1)}$ 的梯度变化, 更新Hesse矩阵逆的近似, 即确定满足正割条件的 H_{k+1} . 置 $k := k + 1$, 返回第(1)步。

下面我们就来讨论怎样构造及确定满足拟牛顿条件的Hesse矩阵逆的近似 H_{k+1} .

设 H_k 是第 k 次迭代的Hesse矩阵逆的近似, 我们希望以 H_k 来产生 H_{k+1} , 即

$$H_{k+1} = H_k + E_k,$$

其中 E_k 是一个低秩的矩阵。

为此, 可采用对称秩一(SR1)校正

$$H_{k+1} = H_k + a u u^T, \quad (a \in \mathbb{R}, u \in \mathbb{R}^n).$$

由拟牛顿条件(109)知

$$H_{k+1}y^{(k)} = H_k y^{(k)} + (a u^T y^{(k)})u = s^{(k)}$$

故 u 必与方向 $s^{(k)} - H_k y^{(k)}$ 一致, 且假定 $s^{(k)} - H_k y^{(k)} \neq 0$.

不妨取 $u = s^{(k)} - H_k y^{(k)}$, 此时 $a = \frac{1}{u^T y^{(k)}}$, 从而得到

$$H_{k+1} = H_k + \frac{(s^{(k)} - H_k y^{(k)})(s^{(k)} - H_k y^{(k)})^T}{(s^{(k)} - H_k y^{(k)})^T y^{(k)}}. \quad (110)$$

上式称为对称秩一校正。

二次终止性

定义： 如果一种迭代法能在确知的有限步内找到二次函数的极小点，则称这种方法具有二次终止性。

对称秩一校正的突出性质:

- ① 针对二次函数具有遗传性, 即 $H_k y^{(\ell)} = s^{(\ell)}, \ell = 0, 1, \dots, k-1$.
- ② 具有二次终止性, 即对于二次函数不需要进行一维搜索而具有 n 步终止性质, 且 $H_n = [\nabla^2 f(x^*)]^{-1}$.

[思考题: 请证明对称秩一校正拟牛顿法的上述性质。]

对称秩一校正的缺点是，不能保持迭代矩阵 H_{k+1} 的正定性。

仅当 $(s^{(k)} - H_k y^{(k)})^T y^{(k)} > 0$ 时，对称秩一校正才能保持正定性。而这个条件往往很难保证，即使 $(s^{(k)} - H_k y^{(k)})^T y^{(k)} > 0$ 满足，它也可能很小从而导致数值上的困难。

这些都使得对称秩一校正的拟牛顿法应用有较大局限性。

采用对称秩二(SR2)校正

$$H_{k+1} = H_k + auu^T + bv v^T,$$

并使得拟牛顿条件(109)成立, 则有

$$H_{k+1}y^{(k)} = H_k y^{(k)} + (au^T y^{(k)})u + (bv^T y^{(k)})v = s^{(k)}.$$

这里 u, v 显然不是唯一确定的, 但有一种明显的选择是:

$$\begin{cases} u = s^{(k)}, & au^T y^{(k)} = 1; \\ v = H_k y^{(k)}, & bv^T y^{(k)} = -1. \end{cases}$$

因此有

$$H_{k+1} = H_k + \frac{s^{(k)}s^{(k)T}}{s^{(k)T}y^{(k)}} - \frac{H_k y^{(k)}y^{(k)T} H_k}{y^{(k)T} H_k y^{(k)}}. \quad (111)$$

上式称为 DFP(Davidon-Fletcher-Powell)校正公式, 由Davidon(1959)提出, 后经Fletcher & Powell(1963)修改而来。

DFP校正(111)是典型的拟牛顿校正公式，它有很多重要性质。

(一) 对于二次函数（采用精确一维搜索）

- ① 遗传性，即 $H_k y^{(\ell)} = s^{(\ell)}, \ell = 0, 1, \dots, k-1$.
- ② 二次终止性，即 $H_n = [\nabla^2 f(x^*)]^{-1}$.
- ③ 共轭性，即当取 $H_0 = I$ 时，迭代产生共轭方向。

(二) 对于一般非线性函数

- ① 校正保持正定性，因而 $d^{(k)}$ 总是下降方向。
- ② 每次迭代需要 $3n^2 + O(n)$ 次乘法运算。
- ③ 方法具有超线性收敛速度。

拟牛顿（正割）条件

$$H_{k+1}y^{(k)} = s^{(k)}$$

其中 H_{k+1} 是 Hesse 矩阵逆的近似；

$$B_{k+1}s^{(k)} = y^{(k)}$$

其中 B_{k+1} 是 Hesse 矩阵的近似。

由对称秩二校正和拟牛顿条件 $H_{k+1}y^{(k)} = s^{(k)}$ 可得到 H_k 的DFP校正公式

$$H_{k+1}^{(DFP)} = H_k + \frac{s^{(k)}s^{(k)T}}{s^{(k)T}y^{(k)}} - \frac{H_k y^{(k)}y^{(k)T} H_k}{y^{(k)T} H_k y^{(k)}}.$$

BFGS (Broyden-Fletcher-Goldfarb-Shanno) 校正

类似地，我们可从拟牛顿条件 $B_{k+1}s^{(k)} = y^{(k)}$ 得到关于 B_k 的对称秩二校正公式

$$B_{k+1}^{(BFGS)} = B_k + \frac{y^{(k)}y^{(k)T}}{y^{(k)T}s^{(k)}} - \frac{B_k s^{(k)}s^{(k)T} B_k}{s^{(k)T} B_k s^{(k)}}. \quad (112)$$

把(112)式称为关于 B_k 的BFGS校正。

如果我们对 B_k 的BFGS校正“求逆”，就可以得到关于 H_k 的BFGS校正公式

$$H_{k+1}^{(BFGS)} = H_k + \left(1 + \frac{y^{(k)T} H_k y^{(k)}}{s^{(k)T} y^{(k)}}\right) \frac{s^{(k)} s^{(k)T}}{s^{(k)T} y^{(k)}} - \frac{H_k y^{(k)} s^{(k)T} + s^{(k)} y^{(k)T} H_k}{s^{(k)T} y^{(k)}}. \quad (113)$$

[思考题：请给出 $H_{k+1}^{(BFGS)}$ 的对称秩二校正的特解，即 a, u, b, v .]

进一步, 若将(113)式中 $\{H \leftrightarrow B, s \leftrightarrow y\}$ 互换, 便得到关于 B_k 的DFP校正公式

$$B_{k+1}^{(DFP)} = B_k + \left(1 + \frac{s^{(k)T} B_k s^{(k)}}{y^{(k)T} s^{(k)}}\right) \frac{y^{(k)} y^{(k)T}}{y^{(k)T} s^{(k)}} - \frac{B_k s^{(k)} y^{(k)T} + y^{(k)} s^{(k)T} B_k}{y^{(k)T} s^{(k)}}. \quad (114)$$

秩一校正的求逆公式

Sherman-Morrison定理: 设 $A \in \mathbb{R}^{n \times n}$ 是非奇异阵, $u, v \in \mathbb{R}^n$ 是任意向量。若 $1 + v^T A^{-1} u \neq 0$, 则 A 的秩一校正 $A + uv^T$ 非奇异, 且其逆可以表示为

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u}. \quad (115)$$

[思考题: 利用秩一校正的求逆公式, 由 $H_{k+1}^{(DFP)}$ 推导 $B_{k+1}^{(DFP)}$]

进一步的参考资料

- R. Fletcher, Practical Methods of Optimization (2nd Edition). John Wiley & Sons, 1987.
- D. C. Liu and J. Nocedal, On the Limited Memory Method for Large Scale Optimization. Mathematical Programming B, 45(3), pp. 503-528, 1999.
- ...

共轭方向

定义： 设 G 是 $n \times n$ 正定阵， \mathbb{R}^n 中的任一组非零向量 $\{d^{(0)}, d^{(1)}, \dots, d^{(k)}\}$ ，如果 $d^{(i)T} G d^{(j)} = 0 (i \neq j)$ ，则称 $d^{(0)}, d^{(1)}, \dots, d^{(k)}$ 是 G -共轭的。

显然共轭是正交概念的推广，当取 $G = I$ 时，共轭即为正交。

共轭方向法（类）：

- (0) 给定正定阵 G , 选取初始点 $x^{(0)}$, 计算 $g^{(0)} = \nabla f(x^{(0)})$ 并构造 $d^{(0)}$ 使得 $g^{(0)T} d^{(0)} < 0$. 令 $k := 0$.
- (1) 求精确的一维搜索步长 α_k , 即 $\alpha_k = \arg \min_{\alpha > 0} f(x^{(k)} + \alpha d^{(k)})$.
- (2) 更新迭代点 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$, 并构造 $d^{(k+1)}$ 使得 $d^{(k+1)T} G d^{(j)} = 0, j = 0, 1, \dots, k$.
- (3) 置 $k := k + 1$, 返回第(1)步。

共轭方向法是从研究二次函数的极小化问题中产生的，但它可以推广到处理非二次函数的极小化问题。

共轭方向法的一个重要性质是，只要执行精确一维搜索，迭代算法就具有二次终止性。

共轭方向法基本定理： 严格凸二次函数 $f(x) = \frac{1}{2}x^T Gx + c^T x$,
共轭方向法执行精确一维搜索，则每步迭代点 $x^{(k+1)}$ 是 $f(x)$ 在线性流形

$$\mathcal{V} = \{x \mid x = x^{(0)} + \sum_{j=0}^k \beta_j d^{(j)}, \forall \beta_j \in \mathbb{R}\}$$

中的唯一极小点。

证明： 设共轭方向法产生的 G -共轭方向为 $d^{(0)}, d^{(1)}, \dots, d^{(k)}$.
由共轭方向的定义知, $\{d^{(0)}, d^{(1)}, \dots, d^{(k)}\}$ 线性无关。

下面只要证: 对所有 $k < n$ 成立

$$g^{(k+1)T} d^{(j)} = 0, \quad j = 0, 1, \dots, k.$$

即在点 $x^{(k+1)}$ 处的函数梯度 $g^{(k+1)} = \nabla f(x^{(k+1)})$ 与子空间 $\text{span}\{d^{(0)}, d^{(1)}, \dots, d^{(k)}\}$ 正交。

由此易得出定理的结论。

证明（续）： 直接由精确一维搜索知，对 $\forall j$ 成立

$$\mathbf{g}^{(j+1)T} \mathbf{d}^{(j)} = 0.$$

特别地，当 $j = k$ 时， $\mathbf{g}^{(k+1)T} \mathbf{d}^{(k)} = 0$.

证明（续）：事实上，由于

$$y^{(k)} = g^{(k+1)} - g^{(k)} = G(x^{(k+1)} - x^{(k)}) = Gs^{(k)} = \alpha_k Gd^{(k)}.$$

故当 $j < k$ 时有

$$\begin{aligned} g^{(k+1)T} d^{(j)} &= g^{(j+1)T} d^{(j)} + \sum_{i=j+1}^k y^{(i)T} d^{(j)} \\ &= g^{(j+1)T} d^{(j)} + \sum_{i=j+1}^k \alpha_i d^{(i)T} Gd^{(j)} \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

证明（续）： 综合上述，从而证明了

$$\mathbf{g}^{(k+1)T} \mathbf{d}^{(j)} = 0, \quad j = 0, 1, \dots, k.$$

推论： 对于严格凸的二次函数，若沿着一组共轭方向搜索，经有限步迭代必达到极小点。

共轭梯度法

由于共轭方向法具有二次终止性，人们希望能给出一个具体的算法（属于共轭方向法类）。通过修改最速下降法，使其搜索方向具有共轭性质，这便是共轭梯度法。

下面我们先针对二次函数，给出共轭梯度法的具体描述。

共轭梯度法

设二次函数 $f(x) = \frac{1}{2}x^T Gx + c^T x$, 其中 G 是 $n \times n$ 正定阵, c 是 n 维向量。
函数 f 的梯度向量为

$$g(x) = \nabla f(x) = Gx + c.$$

取 $d^{(0)} = -g^{(0)}$, 因为 $x^{(1)} = x^{(0)} + \alpha_0 d^{(0)}$ 中步长 α_0 由精确一维搜索决定,
所以 $g^{(1)T} d^{(0)} = 0$.

现设 $d^{(1)} = -g^{(1)} + \beta_0^{(1)} d^{(0)}$, 选择 $\beta_0^{(1)}$ 使 $d^{(1)T} G d^{(0)} = 0$, 即得

$$\beta_0^{(1)} = \frac{g^{(1)T} g^{(1)}}{g^{(0)T} g^{(0)}}.$$

同理, 令 $d^{(2)} = -g^{(2)} + \beta_0^{(2)} d^{(0)} + \beta_1^{(2)} d^{(1)}$, 选择 $\beta_0^{(2)}, \beta_1^{(2)}$ 使得 $d^{(2)T} G d^{(j)} = 0, j = 0, 1$. 从而有

$$\begin{aligned}\beta_0^{(2)} &= 0, \\ \beta_1^{(2)} &= \frac{g^{(2)T} g^{(2)}}{g^{(1)T} g^{(1)}}.\end{aligned}$$

共轭梯度法

一般地, 在第 k 次迭代中, 令

$$\mathbf{d}^{(k)} = -\mathbf{g}^{(k)} + \sum_{j=0}^{k-1} \beta_j^{(k)} \mathbf{d}^{(j)},$$

选择 $\beta_j^{(k)}$ 使得 $\mathbf{d}^{(k)T} \mathbf{G} \mathbf{d}^{(j)} = 0, j = 0, 1, \dots, k-1$, 则有

$$\beta_j^{(k)} = \frac{\mathbf{g}^{(k)T} \mathbf{G} \mathbf{d}^{(j)}}{\mathbf{d}^{(j)T} \mathbf{G} \mathbf{d}^{(j)}} = \frac{\mathbf{g}^{(k)T} (\mathbf{g}^{(j+1)} - \mathbf{g}^{(j)})}{\mathbf{d}^{(j)T} (\mathbf{g}^{(j+1)} - \mathbf{g}^{(j)})}.$$

又由于 $\mathbf{g}^{(k)T} \mathbf{g}^{(j)} = 0, j = 0, 1, \dots, k-1$, 故得

$$\begin{aligned} \beta_j^{(k)} &= 0, j = 0, 1, \dots, k-2 \\ \beta_{k-1}^{(k)} &= \frac{\mathbf{g}^{(k)T} \mathbf{g}^{(k)}}{\mathbf{g}^{(k-1)T} \mathbf{g}^{(k-1)}}. \end{aligned}$$

共轭梯度法

针对二次函数的共轭梯度算法 (Fletcher & Reeves, 1964)

(0) 给定初始点 $x^{(0)}$, 计算 $g^{(0)} = g(x^{(0)})$, 令 $d^{(0)} = -g^{(0)}$, $k := 0$.

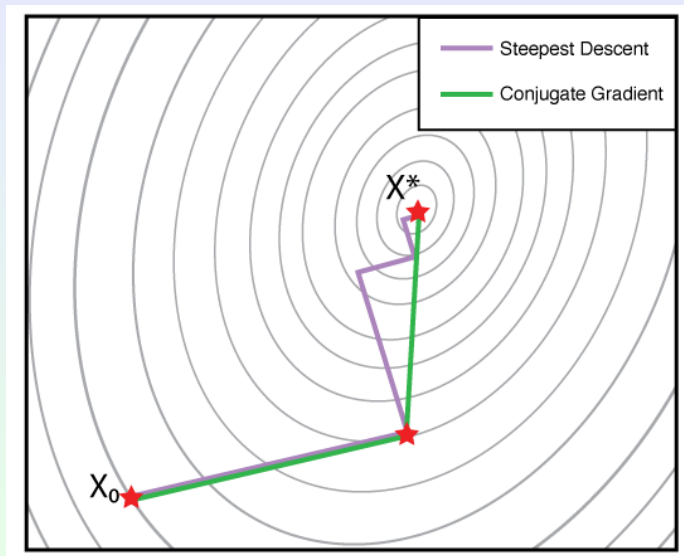
(1) 迭代更新 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$, 其中 $\alpha_k = \frac{g^{(k)T} g^{(k)}}{d^{(k)T} G d^{(k)}}$.

(2) 计算 $g^{(k+1)} = g(x^{(k+1)})$, 构造共轭梯度方

向 $d^{(k+1)} = -g^{(k+1)} + \beta_k d^{(k)}$, 其中 $\beta_k = \frac{g^{(k+1)T} g^{(k+1)}}{g^{(k)T} g^{(k)}}$.

(3) 置 $k := k + 1$, 返回第(1)步。

共轭梯度法



共轭梯度法

共轭梯度法性质定理： 设目标函数 $f(x) = \frac{1}{2}x^T Gx + c^T x$, 则采用精确一维搜索的共轭梯度法经 $m \leq n$ 步迭代后终止, 且对所有的 $1 \leq k \leq m$ 成立下列关系式:

$$d^{(k)T} G d^{(j)} = 0, \quad g^{(k)T} g^{(j)} = 0, \quad j = 0, 1, \dots, k-1$$

$$d^{(k)T} g^{(k)} = -g^{(k)T} g^{(k)}$$

$$\text{span}\{g^{(0)}, g^{(1)}, \dots, g^{(k)}\} = \text{span}\{g^{(0)}, Gg^{(0)}, \dots, G^k g^{(0)}\}$$

$$\text{span}\{d^{(0)}, d^{(1)}, \dots, d^{(k)}\} = \text{span}\{d^{(0)}, Gg^{(0)}, \dots, G^k g^{(0)}\}$$

[思考题：证明上述定理...]

将共轭梯度法推广到非二次函数的极小化问题，其迭代为

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}.$$

步长 α_k 由精确或者非精确一维搜索决定，而 $d^{(k+1)}$ 的构造如下：

$$d^{(k+1)} = -g^{(k+1)} + \beta_k d^{(k)}.$$

共轭梯度法

其中

$$\beta_k := \frac{\mathbf{g}^{(k+1)T} \mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)T} \mathbf{g}^{(k)}} \quad (\text{Fletcher} - \text{Reeves})$$

$$\beta_k := \frac{\mathbf{g}^{(k+1)T} (\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})}{\mathbf{d}^{(k)T} (\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})} \quad (\text{Hestenes} - \text{Stiefel})$$

$$\beta_k := \frac{\mathbf{g}^{(k+1)T} (\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})}{\mathbf{g}^{(k)T} \mathbf{g}^{(k)}} \quad (\text{Polak} - \text{Ribiere} - \text{Polyak})$$

$$\beta_k := \frac{\mathbf{g}^{(k+1)T} \mathbf{g}^{(k+1)}}{-\mathbf{d}^{(k)T} \mathbf{g}^{(k)}} \quad (\text{Dixon})$$

$$\beta_k := \frac{\mathbf{g}^{(k+1)T} \mathbf{g}^{(k+1)}}{\mathbf{d}^{(k)T} (\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)})} \quad (\text{Dai} - \text{Yuan})$$

对于非二次函数，共轭梯度法迭代 n 步以后所产生的搜索方向 $\mathbf{d}^{(k+1)} = -\mathbf{g}^{(k+1)} + \beta_k \mathbf{d}^{(k)}$ 可能不再是下降方向（由非精确一维搜索造成的）。

因此， n 步以后我们应该周期性采用最速下降方向作为搜索方向，即令 $\mathbf{d}^{(\ell n)} = -\mathbf{g}^{(\ell n)}$, $\ell = 1, 2, \dots$

这种策略称为重启动策略，这样的共轭梯度法也称作重启动共轭梯度法。

共轭梯度法

如上所述的共轭梯度法迭代对于一般的非线性函数的最小化也是照样适用的，但迭代更新的步长因子无法显式表达，需要执行数值近似的非精确一维搜索。

由于每 n 步迭代执行重启动策略，若记重新启动时得到的点列为 $\{z^{(j)}\}$ ，则可证明这些相隔 n 次的迭代点列超线性收敛。受实际计算误差的影响，在很多情形下仅能取得类似线性的收敛速率。

共轭梯度法

从实际计算效率及稳定性来看，共轭梯度法未必比拟牛顿法好。但是，共轭梯度法中搜索方向的计算仅仅用到目标函数的梯度，而不必像拟牛顿法那样在每次迭代中更新Hesse矩阵（或其逆）的近似阵并记忆之。所以，当问题的规模大而且有稀疏结构时，共轭梯度法有高效执行计算的好处。

The preconditioned conjugate gradient method

在大多数情况下，为确保共轭梯度法的快速收敛，预条件处理是必要的。

进一步的参考资料

- M. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards, 49 (6), 1952.
- K. Atkinson, An Introduction to Numerical Analysis (2nd Edition). John Wiley & Sons, 1988.
- M. Avriel, Nonlinear Programming: Analysis and Methods. Dover Publishing, 2003.
- G. Golub and C. Van Loan, Matrix Computations (3rd Edition). Johns Hopkins University Press.
- ...

为了保证迭代法的全局收敛性，之前我们采用了一维搜索策略。

一维搜索策略先确定一个搜索方向 $d^{(k)}$ ，然后沿着这个方向选择适当的步长因子 α_k ，新的迭代点 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$ 。

现在，我们讨论另一种全局收敛策略
— 信赖域方法 (Trust-Region Method).

为了保证迭代法的全局收敛性，之前我们采用了一维搜索策略。

一维搜索策略先确定一个搜索方向 $d^{(k)}$ ，然后沿着这个方向选择适当的步长因子 α_k ，新的迭代点 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$ 。

现在，我们讨论另一种全局收敛策略
— 信赖域方法 (Trust-Region Method).

为了保证迭代法的全局收敛性，之前我们采用了一维搜索策略。

一维搜索策略先确定一个搜索方向 $d^{(k)}$ ，然后沿着这个方向选择适当的步长因子 α_k ，新的迭代点 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$ 。

现在，我们讨论另一种全局收敛策略
— 信赖域方法 (Trust-Region Method).

信赖域方法首先定义当前迭代点 $x^{(k)}$ 的邻域

$$\Omega_k = \{x \in \mathbb{R}^n \mid \|x - x^{(k)}\| \leq e_k\},$$

这里 Ω_k 称为信赖域， e_k 是信赖域半径。

假定在这个邻域里，二次模型 $q^{(k)}(s)$ 是目标函数 $f(x)$ 的一个合适的近似，则在信赖域中极小化二次模型，得到近似极小点 $s^{(k)}$ ，并取 $x^{(k+1)} = x^{(k)} + s^{(k)}$ 。

信赖域方法利用二次模型在信赖域内求得方向步 $s^{(k)}$ ，使得目标函数的下降比一维搜索更有效。

信赖域方法不仅具有全局收敛性，而且不要求目标函数的Hesse矩阵（或其近似）是正定的。

信赖域子问题

$$\begin{aligned} \min \quad & q^{(k)}(s) = f(x^{(k)}) + g^{(k)T}s + \frac{1}{2}s^TB_k s \\ \text{s.t.} \quad & \|s\| \leq e_k. \end{aligned} \tag{116}$$

其中 $s = x - x^{(k)}$, $g^{(k)} = \nabla f(x^{(k)})$, 对称阵 B_k 是 Hesse 矩阵 $\nabla^2 f(x^{(k)})$ 或其近似, $e_k > 0$ 为信赖域半径, $\|\cdot\|$ 为某一范数。

如何选择信赖域半径 e_k ?

我们将根据二次模型 $q^{(k)}(s)$ 对目标函数 $f(x)$ 的拟合程度自适应地调整信赖域半径。

设子问题(116)的解 $s^{(k)}$, 令目标函数的下降量

$$\text{Act}_k = f(x^{(k)}) - f(x^{(k)} + s^{(k)})$$

为实际下降量, 令二次模型函数的下降量

$$\text{Pre}_k = q^{(k)}(0) - q^{(k)}(s^{(k)})$$

为预测下降量。定义比值

$$r_k = \frac{\text{Act}_k}{\text{Pre}_k} = \frac{f(x^{(k)}) - f(x^{(k)} + s^{(k)})}{q^{(k)}(0) - q^{(k)}(s^{(k)})}.$$

它衡量了二次模型与目标函数之间的一致程度。

当 r_k 越接近1, 表明二次模型函数 $q^{(k)}(s)$ 与目标函数 f 的一致性程度越好, 此时可以增大半径 e_k 以扩大信赖域。

如果 $r_k > 0$ 但不接近1, 我们保持信赖域半径 e_k 不变。

如果 r_k 接近零或取负值, 表明 $q^{(k)}(s)$ 与目标函数 f 的一致性程度不理想, 就减小半径 e_k 以缩小信赖域。

信赖域算法

- (0) 给定初始点 $x^{(0)}$, 信赖域半径的上界 \bar{e} ,
 $\varepsilon > 0, 0 < \gamma_1 < \gamma_2 < 1, 0 < \eta_1 < 1 < \eta_2$. 取 $e_0 \in (0, \bar{e})$, 令 $k := 0$.
- (1) 如果 $\|g^{(k)}\| \leq \varepsilon$, 停止迭代。否则, 求解信赖域子问题(116)得到 $s^{(k)}$.
- (2) 计算比值 r_k , 更新迭代点

$$x^{(k+1)} = \begin{cases} x^{(k)} + s^{(k)} & \text{if } r_k > 0, \\ x^{(k)} & \text{otherwise.} \end{cases}$$

信赖域算法

(3) 调整信赖域半径，令

$$e_{k+1} = \begin{cases} \eta_1 e_k & \text{if } r_k < \gamma_1, \\ e_k & \text{if } \gamma_1 \leq r_k < \gamma_2, \\ \min(\eta_2 e_k, \bar{e}) & \text{if } r_k \geq \gamma_2. \end{cases}$$

(4) 置 $k := k + 1$, 返回第(1)步。

信赖域方法的全局收敛性定理:

设水平集 $L(x^{(0)}) = \{x \mid f(x) \leq f(x^{(0)})\}$ 有界, 且 $f(x)$ 在其上 C^2 连续, 则由信赖域算法产生的迭代序列存在聚点 x^∞ , 满足一阶和二阶必要条件, 即

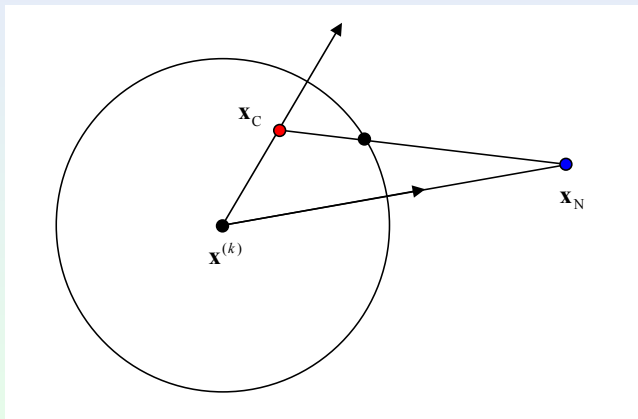
$$g^\infty = \nabla f(x^\infty) = 0, \quad G_\infty = \nabla^2 f(x^\infty) \geq 0.$$

在信赖域算法中关键的一步是，解信赖域子问题(116).

这里我们介绍一种求解信赖域子问题的方法，即由 Powell (1970) 提出的折线法。

所谓折线法，是连接 Cauchy 点（由最速下降法产生的极小点）和牛顿点（由牛顿法产生的极小点），其连线与信赖域边界的交点取为 $x^{(k+1)}$ 。

折线法图示



对于二次模型

$$q^{(k)}(-\alpha \mathbf{g}^{(k)}) = f(\mathbf{x}^{(k)}) - \alpha \|\mathbf{g}^{(k)}\|^2 + \frac{1}{2} \alpha^2 \mathbf{g}^{(k)T} B_k \mathbf{g}^{(k)},$$

精确一维搜索的步长因子可表达为

$$\alpha_k = \frac{\|\mathbf{g}^{(k)}\|^2}{\mathbf{g}^{(k)T} B_k \mathbf{g}^{(k)}}.$$

于是 Cauchy 步为

$$\mathbf{s}_C^{(k)} = -\alpha_k \mathbf{g}^{(k)} = -\frac{\|\mathbf{g}^{(k)}\|^2}{\mathbf{g}^{(k)T} B_k \mathbf{g}^{(k)}} \mathbf{g}^{(k)}.$$

如果 $\|s_C^{(k)}\| = \|\alpha_k g^{(k)}\| \geq e_k$, 取

$$s^{(k)} = -\frac{e_k}{\|g^{(k)}\|} g^{(k)},$$

便得

$$x^{(k+1)} = x^{(k)} - \frac{e_k}{\|g^{(k)}\|} g^{(k)}.$$

如果 $\|s_C^{(k)}\| = \|\alpha_k g^{(k)}\| < e_k$, 再计算牛顿步

$$s_N^{(k)} = -B_k^{-1} g^{(k)}.$$

如果 $\|s_N^{(k)}\| \leq e_k$, 取

$$s^{(k)} = s_N^{(k)} = -B_k^{-1}g^{(k)},$$

否则, 取

$$s^{(k)} = s_C^{(k)} + \lambda(s_N^{(k)} - s_C^{(k)}),$$

其中 λ 使得

$$\|s_C^{(k)} + \lambda(s_N^{(k)} - s_C^{(k)})\| = e_k.$$

综上所述，我们得到

$$x^{(k+1)} = \begin{cases} x^{(k)} - \frac{e_k}{\|g^{(k)}\|} g^{(k)} & \text{当 } \|s_C^{(k)}\| \geq e_k, \\ x^{(k)} + s_C^{(k)} + \lambda(s_N^{(k)} - s_C^{(k)}) & \text{当 } \|s_C^{(k)}\| < e_k \text{ 且 } \|s_N^{(k)}\| > e_k, \\ x^{(k)} - B_k^{-1} g^{(k)} & \text{当 } \|s_C^{(k)}\| < e_k \text{ 且 } \|s_N^{(k)}\| \leq e_k. \end{cases} \quad (117)$$

折线法满足下列性质：

- 1) 沿着 Cauchy 点 $x_C^{(k+1)}$ 和牛顿点 $x_N^{(k+1)}$ 的连线，到 $x^{(k)}$ 的距离单调增加；
- 2) 沿着 Cauchy 点 $x_C^{(k+1)}$ 和牛顿点 $x_N^{(k+1)}$ 的连线，子问题模型函数值单调减少。

[思考题：证明上述性质...]

本章作业（无约束最优化）

- **Exercise 5.1:** 请写出上述基于Wolfe-Powell准则的非精确一维搜索算法中插值多项式 $p^{(1)}(t)$, $p^{(2)}(t)$ 的具体表达式。
- **Exercise 5.2:** 请证明基于Goldstein准则的非精确一维搜索算法的全局收敛性。
- **Exercise 5.3:** 试将非线性方程组求根 $F(x) = 0$ 的牛顿迭代，用于求解无约束最优化问题 $\min_{x \in \mathbb{R}^n} f(x)$. 请给出相应的迭代格式，并说明理由。
- **Exercise 5.4:** 请证明对称秩一校正拟牛顿法具有二次终止性和遗传性。
- **Exercise 5.5:** 利用秩一校正的求逆公式（Sherman-Morrison定理），由 $H_{k+1}^{(DFP)}$ 推导 $B_{k+1}^{(DFP)}$ 。

本章作业（无约束最优化）

- **Exercise 5.6:** 请证明共轭梯度法的性质定理。
- **Exercise 5.7:** 在信赖域方法中，请给出一种与调整信赖域半径等效的自适应模式算法。
- **Exercise 5.8:** 请证明折线法（信赖域方法）子问题模型的函数单调性。
- **Project 5.9:** 编程实现基于非精确一维搜索(line search)的某一种（搜索方向）迭代下降算法，并用于求解无约束最优化问题。要求提交程序代码，用户指南及测试报告。

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划**
- 8 非线性约束最优化
- 9 总结

二次规划

二次规划(Quadratic Programming)是指, 在变量的线性等式和/或不等式限制下求二次函数的极小点问题

$$\begin{aligned} \min \quad & Q(x) = \frac{1}{2}x^T Gx + c^T x \\ \text{s.t.} \quad & a_i^T x = b_i, i \in \mathcal{E} = \{1, \dots, m_e\} \\ & a_i^T x \geq b_i, i \in \mathcal{I} = \{m_e + 1, \dots, m\} \end{aligned} \tag{118}$$

我们假定 G 为对称阵, $a_i(i \in \mathcal{E})$ 是线性无关的。

二次规划

二次规划的约束可能不相容，也可能没有有限的最小值，这时称二次规划问题无解。

如果矩阵 G 半正定，问题(118)是凸二次规划问题，它的任意局部解也是整体解。

如果矩阵 G 正定，问题(118)是正定二次规划问题，只要存在解即是唯一的。

如果矩阵 G 不定，问题(118)是一般的二次规划问题，有可能出现非整体解的局部解。

等式约束二次规划问题

$$\begin{array}{ll} \min & Q(x) = \frac{1}{2}x^T Gx + c^T x \\ \text{s.t.} & Ax = b \end{array} \quad (119)$$

这里 A 是 $m \times n$ 矩阵, 且不失一般性可设 $\text{rank}(A) = m$.

二次规划

设有一种基分解 $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$, 其中 $x_B \in \mathbb{R}^m$, $x_N \in \mathbb{R}^{n-m}$, 使得其约束矩阵的对应分块 $A = (A_B, A_N)$ 中 A_B 可逆。于是, 等式约束条件可写成

$$x_B = A_B^{-1}(b - A_N x_N),$$

并将上式代入目标函数中得到无约束问题

$$\min_{x_N \in \mathbb{R}^{n-m}} \frac{1}{2} x_N^T \hat{G}_N x_N + \hat{c}_N^T x_N. \quad (120)$$

在上式中

$$\hat{G}_N = G_{NN} - G_{NB}A_B^{-1}A_N - A_N^T A_B^{-T} G_{BN} + A_N^T A_B^{-T} G_{BB} A_B^{-1} A_N,$$

$$\hat{c}_N = c_N - A_N^T A_B^{-T} c_B + G_{NB} A_B^{-1} b - A_N^T A_B^{-T} G_{BB} A_B^{-1} b,$$

以及对应分块形式

$$G = \begin{pmatrix} G_{BB} & G_{BN} \\ G_{NB} & G_{NN} \end{pmatrix}, \quad c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}.$$

(1) 如果 \hat{G}_N 正定, 则无约束问题的解可唯一地给出

$$x_N^* = -\hat{G}_N^{-1} \hat{c}_N,$$

进一步得原问题(119)的解为

$$x^* = \begin{pmatrix} x_B^* \\ x_N^* \end{pmatrix} = \begin{pmatrix} A_B^{-1} b \\ 0 \end{pmatrix} + \begin{pmatrix} A_B^{-1} A_N \\ -I \end{pmatrix} \hat{G}_N^{-1} \hat{c}_N.$$

设 x^* 对应的Lagrange乘子向量为 λ^* , 则有

$$Gx^* + c = A^T \lambda^* \implies \lambda^* = A_B^{-T} (G_{BB} x_B^* + G_{BN} x_N^* + c_B).$$

- (2) 如果 \hat{G}_N 是半正定的, 则在 $(I - \hat{G}_N \hat{G}_N^+) \hat{c}_N = 0$ 时, 无约束问题有界, 且它的解可表示为

$$x_N^* = -\hat{G}_N^+ \hat{c}_N + (I - \hat{G}_N^+ \hat{G}_N) \tilde{y},$$

其中 $\tilde{y} \in \mathbb{R}^{n-m}$ 为任意向量, \hat{G}_N^+ 表示 \hat{G}_N 的广义逆矩阵。此时, 原问题的解 x^* 和相应最优乘子 λ^* 可类似确定。

当 $(I - \hat{G}_N \hat{G}_N^+) \hat{c}_N = 0$ 不成立时, 则可推出无约束问题无下界, 从而原问题也无下界。

- (3) 如果 \hat{G}_N 不定（即存在负的特征根），显然无约束问题无下界，故原问题不存在有限最优解。

上述消去法的不足之处是，当 A_B 接近奇异时，容易导致数值计算的不稳定。

广义消去法

设 $Z = \{z_{m+1}, \dots, z_n\}$ 解空间 $\text{Ker}(A)$ 的一组基,
 $Y = \{y_1, \dots, y_m\}$ 是商空间 $\mathbb{R}^n / \text{Ker}(A)$ 的一组基,
则 $\forall x \in \mathbb{R}^n$ 可作如下分解表达

$$x = Yx_Y + Zx_Z.$$

从而有

$$Ax = b \implies AYx_Y + AZx_Z = b \implies x_Y = (AY)^{-1}b,$$

所以得

$$x = Y(AY)^{-1}b + Zx_Z,$$

其中 $x_Z \in \mathbb{R}^{n-m}$ 是自由变量。

广义消去法

将上式代入目标函数中得无约束问题

$$\min_{\mathbf{x}_Z \in \mathbb{R}^{n-m}} \frac{1}{2} \mathbf{x}_Z^T (\mathbf{Z}^T \mathbf{G} \mathbf{Z}) \mathbf{x}_Z + [\mathbf{Z}^T \mathbf{G} \mathbf{Y} (\mathbf{A} \mathbf{Y})^{-1} \mathbf{b} + \mathbf{Z}^T \mathbf{c}]^T \mathbf{x}_Z. \quad (121)$$

假定 $\mathbf{Z}^T \mathbf{G} \mathbf{Z}$ 正定, 则有

$$\mathbf{x}_Z^* = -(\mathbf{Z}^T \mathbf{G} \mathbf{Z})^{-1} \mathbf{Z}^T [\mathbf{G} \mathbf{Y} (\mathbf{A} \mathbf{Y})^{-1} \mathbf{b} + \mathbf{c}].$$

广义消去法

从而得到原问题的最优解

$$x^* = Y(AY)^{-1} - Z(Z^T GZ)^{-1}Z^T [GY(AY)^{-1}b + c],$$

相应的Lagrange乘子为

$$\lambda^* = (AY)^{-T} Y^T (Gx^* + c).$$

二次规划

Lagrange方法是基于求解可行域内的(K-T)点, 即Lagrange函数的稳定点。

对于等式约束问题(119), 其Lagrange函数的稳定点就是如下线性方程组的解

$$\begin{cases} Gx + c = A^T \lambda, \\ Ax = b. \end{cases}$$

写成矩阵形式得

$$\begin{pmatrix} G & -A^T \\ -A & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = - \begin{pmatrix} c \\ b \end{pmatrix}.$$

二次规划

设矩阵 $\begin{pmatrix} G & -A^T \\ -A & 0 \end{pmatrix}$ 可逆, 则存在矩阵 $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{m \times m}$, $W \in \mathbb{R}^{m \times n}$ 使得

$$\begin{pmatrix} G & -A^T \\ -A & 0 \end{pmatrix}^{-1} = \begin{pmatrix} U & W^T \\ W & V \end{pmatrix}$$

从而可求得问题的唯一解

$$\begin{cases} x^* = -Uc - W^T b, \\ \lambda^* = -Wc - Vb. \end{cases}$$

上述Lagrange方法中的矩阵非奇异性并不一定要求 G^{-1} 存在，可用不同的方法给出分块矩阵 U, V, W 的表达形式，从而导致不同的计算公式。

二次规划

当 G 可逆, A 行满秩, 则 $(AG^{-1}A^T)^{-1}$ 存在, 不难验证

$$\begin{cases} U = G^{-1} - G^{-1}A^T(AG^{-1}A^T)^{-1}AG^{-1}, \\ V = -(AG^{-1}A^T)^{-1}, \\ W = -(AG^{-1}A^T)^{-1}AG^{-1}. \end{cases}$$

于是我们得到求解公式

$$\begin{cases} x^* = -G^{-1}c + G^{-1}A^T(AG^{-1}A^T)^{-1}(AG^{-1}c + b), \\ \lambda^* = (AG^{-1}A^T)^{-1}(AG^{-1}c + b). \end{cases}$$

二次规划

如果取 Y, Z 满足 $(Y, Z) = \begin{pmatrix} A \\ B \end{pmatrix}^{-1}$, 即 $AY = I_{m \times m}, AZ = 0$.

若另有 $Z^T GZ$ 可逆, 则知 $\begin{pmatrix} G & -A^T \\ -A & 0 \end{pmatrix}$ 可逆。此时

$$\begin{cases} U = Z(Z^T GZ)^{-1}Z^T, \\ V = -Y^T G P^T Y, \\ W = -Y^T P. \end{cases}$$

其中 $P = I - GZ(Z^T GZ)^{-1}Z^T$.

二次规划

基于 A^T 的QR分解, 可给出 (Y, Z) 的一种特殊取法:

设

$$A^T = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = (Q_1, Q_2) \begin{pmatrix} R \\ 0 \end{pmatrix},$$

即

$$A = (R^T, 0) \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix}.$$

其中 Q 为 $n \times n$ 正交阵, R 为 $m \times m$ 上三角阵。于是令 $Y = Q_1 R^{-T}$, $Z = Q_2$, 则有

$$AY = R^T Q_1^T Q_1 R^{-T} = I_{m \times m}, \quad AZ = R^T Q_1^T Q_2 = 0_{m \times (n-m)}.$$

一般的二次规划

$$\begin{aligned} \min \quad & Q(x) = \frac{1}{2}x^T Gx + c^T x \\ \text{s.t.} \quad & a_i^T x = b_i, i \in \mathcal{E} = \{1, \dots, m_e\} \\ & a_i^T x \geq b_i, i \in \mathcal{I} = \{m_e + 1, \dots, m\} \end{aligned} \tag{122}$$

直观上，不积极的不等式约束在解的附近不起作用，可去掉不予考虑；而积极的不等式约束，由于它在解处等号成立，故我们可以用等式约束来代替这些积极的不等式约束。

积极集基本定理： 设 x^* 是一般的二次规划问题(122)的局部极小点，则 x^* 也必是等式约束问题

$$(EQ) \begin{cases} \min & Q(x) = \frac{1}{2}x^T Gx + c^T x \\ \text{s.t.} & a_i^T x = b_i, i \in \mathcal{E} \cup \mathcal{I}(x^*) \end{cases}$$

的局部极小点。反之，如果 x^* 是一般问题(122)的可行点，同时是(EQ)的K-T点，且相应的Lagrange乘子 λ^* 满足 $\lambda_i^* \geq 0, i \in \mathcal{I}(x^*)$ ，则 x^* 必是原问题(122)的K-T点。

[习题6.1：证明上述定理...]

二次规划

设 $x^{(k)}$ 为当前迭代点, 且是问题(122)的可行点。

记 $\mathcal{E}_k = \mathcal{E} \cup \mathcal{I}(x^{(k)})$, 考虑等式约束问题

$$(\text{EQ1}) \begin{cases} \min & \frac{1}{2} s^T G s + (G x^{(k)} + c)^T s \\ \text{s.t.} & a_i^T s = 0, i \in \mathcal{E}_k \end{cases}$$

求得(EQ1)的解 $s^{(k)}$, 及其相应的Lagrange乘子 $\lambda_i^{(k)}, i \in \mathcal{E}_k$.

二次规划

(a) $s^{(k)} \neq 0$ 时, $x^{(k)}$ 不可能是原问题的K-T点。

(b) $s^{(k)} = 0$ 时, $x^{(k)}$ 是问题

$$(EQ2) \begin{cases} \min & \frac{1}{2}x^T Gx + c^T x \\ \text{s.t.} & a_i^T x = b_i, i \in \mathcal{E}_k \end{cases}$$

的K-T点; 如果 $\lambda_i^{(k)} \geq 0, i \in \mathcal{I}(x^{(k)})$, 则 $x^{(k)}$ 也是原问题的K-T点。

(c) 否则, 由 $\lambda_{i_q}^{(k)} = \min_{i \in \mathcal{I}(x^{(k)})} \lambda_i^{(k)} < 0$ 确定 i_q , 那么如下问题

$$(EQ3) \begin{cases} \min & \frac{1}{2}s^T Gs + (Gx^{(k)} + c)^T s \\ \text{s.t.} & a_i^T s = 0, i \in \hat{\mathcal{E}} = \mathcal{E}_k \setminus \{i_q\}. \end{cases}$$

的解 \hat{s} 是原问题在当前点 $x^{(k)}$ 处的可行方向, 即 $a_{i_q}^T \hat{s} \geq 0$.

[思考题: 证明上述(c)的结论...]

积极集方法(Active Set Method)

- (0) 给出可行点 $x^{(0)}$, 令 $\mathcal{E}_0 = \mathcal{E} \cup \mathcal{I}(x^{(0)})$, $k := 0$.
- (1) 求解等式约束问题(EQ1)得 $s^{(k)}$, 若 $s^{(k)} \neq 0$, 转第(3)步。
- (2) 如果 $\lambda_i^{(k)} \geq 0, i \in \mathcal{I}(x^{(k)})$, 则停止; 否则由 $\lambda_{i_q}^{(k)} = \min_{i \in \mathcal{I}(x^{(k)})} \lambda_i^{(k)} < 0$ 确定 i_q 并令 $\mathcal{E}_k := \mathcal{E}_k \setminus \{i_q\}$, $x^{(k+1)} = x^{(k)}$, 转第(4)步。
- (3) 由 $\alpha_k = \min\{1, \min_{i \notin \mathcal{E}_k, a_i^T s^{(k)} < 0} \frac{b_i - a_i^T x^{(k)}}{a_i^T s^{(k)}}\}$, 计算 $x^{(k+1)} = x^{(k)} + \alpha_k s^{(k)}$. 如果 $\alpha_k = 1$, 转第(4)步; 不然定可找到 $p \notin \mathcal{E}_k$ 使得 $a_p^T(x^{(k)} + \alpha_k s^{(k)}) = b_p$, 并令 $\mathcal{E}_k := \mathcal{E}_k \cup \{p\}$.
- (4) $\mathcal{E}_{k+1} := \mathcal{E}_k, k := k + 1$, 返回第(1)步。

本章作业（二次规划）

- **Exercise 6.1:** 请证明积极集基本定理。
- **Exercise 6.2:** 请证明前述(c)的结论。
- **Exercise 6.3（选做题）:** 试证明存在半正定的能量阵 $E_{T,m}$ 满足如下模型解

$$\frac{1}{n}y^TE_{T,m}y = \min_{\substack{\phi \in W_2^m(\Omega) \\ \phi(X_i) = y_i, i = 1, \dots, n}} |\phi|_{T,m}^2$$

其中 $y = (y_1, \dots, y_n)^T$, $T = \{X_i\}_{i=1}^n$,

$$|f|_{T,m}^2 = \frac{1}{n} \sum_{i=1}^n \sum_{|\alpha|=m} \frac{m!}{\alpha!} |D^\alpha f(X_i)|^2.$$

- 1 绪论
- 2 线性规划
- 3 网络最优化
- 4 动态规划
- 5 非线性规划基础理论
- 6 无约束最优化
- 7 二次规划
- 8 非线性约束最优化**
- 9 总结

非线性约束最优化

先考虑等式约束问题

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & c(x) = 0,\end{array}\tag{123}$$

其中 $c(x) = (c_1(x), \dots, c_m(x))^T$.

记 $A(x) = [\nabla c(x)]^T = (\nabla c_1(x), \dots, \nabla c_m(x))^T$.

由最优性条件知: x 是等式约束问题(123)的K-T点当且仅当存在乘子 $\lambda \in \mathbb{R}^m$ 使得

$$\nabla f(x) - A(x)^T \lambda = 0,$$

且 x 是一可行点, 即 $c(x) = 0$.

于是得到联立方程组

$$\begin{cases} \nabla f(x) - A(x)^T \lambda = 0, \\ -c(x) = 0. \end{cases}$$

我们可用Newton-Raphson迭代法求解上述联立方程组。

记 x 和 λ 的计算增量分别为 δ_x, δ_λ , Newton-Raphson迭代满足:

$$\begin{pmatrix} W(x, \lambda) & -A(x)^T \\ -A(x) & 0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - A(x)^T \lambda \\ -c(x) \end{pmatrix}, \quad (124)$$

其中 $W(x, \lambda) = \nabla^2 f(x) - \sum_{i=1}^m \lambda_i \nabla^2 c_i(x)$.

上述方法称为Lagrange-Newton法，最早由Wilson(1963)提出的。

其实质上是用Newton-Raphson迭代求问题(123)的Lagrange函数 $L(x, \lambda)$ 的稳定点。

在此，我们定义价值函数

$$\psi(\mathbf{x}, \lambda) = \|\nabla f(\mathbf{x}) - A(\mathbf{x})^T \lambda\|^2 + \|c(\mathbf{x})\|^2. \quad (125)$$

显然， $\psi(\mathbf{x}, \lambda)$ 是关于Lagrange-Newton法的下降函数，即满足

$$\nabla \psi(\mathbf{x}, \lambda)^T \begin{pmatrix} \delta_{\mathbf{x}} \\ \delta_{\lambda} \end{pmatrix} = -2\psi(\mathbf{x}, \lambda) \leq 0.$$

Lagrange-Newton法:

- (0) 给定 $x^{(0)}$, $\lambda \in \mathbb{R}^m$, $\beta \in (0, 1)$, $\varepsilon \geq 0$, 令 $k := 0$.
- (1) 计算价值函数 $\psi(x^{(k)}, \lambda^{(k)})$, 如果 $\psi(x^{(k)}, \lambda^{(k)}) \leq \varepsilon$, 则停止; 否则在 $(x^{(k)}, \lambda^{(k)})$ 处求解(124)得到 $(\delta_{x^{(k)}}, \delta_{\lambda^{(k)}})$, 并令 $\alpha_k = 1$.
- (2) 若 $\psi(x^{(k)} + \alpha_k \delta_{x^{(k)}}, \lambda^{(k)} + \alpha_k \delta_{\lambda^{(k)}}) \leq (1 - \beta \alpha_k) \psi(x^{(k)}, \lambda^{(k)})$, 转第(3)步; 否则令 $\alpha_k = \frac{1}{4} \alpha_k$, 返回第(2)步。
- (3) 置 $x^{(k+1)} = x^{(k)} + \alpha_k \delta_{x^{(k)}}$, $\lambda^{(k+1)} = \lambda^{(k)} + \alpha_k \delta_{\lambda^{(k)}}$, $k := k + 1$, 返回第(1)步。

非线性约束最优化

Lagrange-Newton法的收敛性结果

定理： 设Lagrange-Newton法产生的迭代点列 $\{(x^{(k)}, \lambda^{(k)})\}$ 有界，如果 $f(x)$ 和 $c_i(x)$ 都是二次连续可微，且逆矩阵

$$\begin{pmatrix} W(x, \lambda) & -A(x)^T \\ -A(x) & 0 \end{pmatrix}^{-1}$$

一致有界，则 $\{(x^{(k)}, \lambda^{(k)})\}$ 的任何聚点都是方程 $\psi(x, \lambda) = 0$ 的根，从而 $\{x^{(k)}\}$ 的聚点是问题(123)的K-T点。

注：在一定条件下，还可进一步证明Lagrange-Newton法具有二阶收敛速度。

非线性约束最优化

逐步二次规划法

Lagrange-Newton法的一大重要贡献是，在其基础上发展出了逐步二次规划方法(Sequential Quadratic Programming Methods)。而后者已成为求解一般非线性约束最优化问题的一类十分重要的方法。

非线性约束最优化

逐步二次规划法

我们可将式(124)写成如下形式：

$$\begin{cases} W(x, \lambda)\delta_x + \nabla f(x) = A(x)^T(\lambda + \delta_\lambda), \\ c(x) + A(x)\delta_x = 0. \end{cases}$$

由最优性条件知， $\delta_{x^{(k)}}$ 即为下列二次规划问题

$$\begin{aligned} \min \quad & \frac{1}{2}d^T W(x^{(k)}, \lambda^{(k)})d + \nabla f(x^{(k)})^T d \\ \text{s.t.} \quad & c(x^{(k)}) + A(x^{(k)})d = 0 \end{aligned} \tag{126}$$

的K-T点。

非线性约束最优化

逐步二次规划法

Lagrange-Newton法可以理解为逐步求解上述等式约束二次规划的方法。

设 $d^{(k)}$ 是二次规划问题(126)的最优解，那么可迭代更新

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)},$$

其中 α_k 为第 k 次迭代的步长。

设 $\bar{\lambda}^{(k)}$ 是(126)对应的Lagrange乘子向量，那么对 $k \geq 1$ 有

$$\lambda^{(k+1)} = \lambda^{(k)} + \alpha_k (\bar{\lambda}^{(k)} - \lambda^{(k)}).$$

非线性约束最优化

逐步二次规划法

现考虑一般的非线性约束最优化问题

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0, i \in \mathcal{E} = \{1, \dots, m_e\}, \\ & c_i(x) \geq 0, i \in \mathcal{I} = \{m_e + 1, \dots, m\}. \end{aligned} \tag{127}$$

类似地，在第 k 次迭代里求解子问题

$$\begin{aligned} \min \quad & \frac{1}{2} d^T W_k d + g^{(k)T} d \\ \text{s.t.} \quad & c_i(x^{(k)}) + a_i(x^{(k)})^T d = 0, i \in \mathcal{E}, \\ & c_i(x^{(k)}) + a_i(x^{(k)})^T d \geq 0, i \in \mathcal{I}. \end{aligned} \tag{128}$$

非线性约束最优化

逐步二次规划法

在这里, W_k 是原问题Lagrange函数的Hesse阵或其近似,
 $g^{(k)} = \nabla f(x^{(k)})$, $A(x^{(k)}) = (a_1(x^{(k)}), \dots, a_m(x^{(k)}))^T = [\nabla c(x^{(k)})]^T$.

记子问题(128)的解为 $d^{(k)}$, 相应Lagrange乘子向量为 $\bar{\lambda}^{(k)}$, 故有

$$\begin{cases} W_k d^{(k)} + g^{(k)} = A(x^{(k)})^T \bar{\lambda}^{(k)}, \\ \bar{\lambda}_i^{(k)} \geq 0, i \in \mathcal{I}, \\ c(x^{(k)}) + A(x^{(k)})d^{(k)} = 0. \end{cases}$$

非线性约束最优化

逐步二次规划法

逐步二次规划法的迭代以 $d^{(k)}$ 作为搜索方向。该搜索方向有很好的性质。它是许多罚函数的下降方向，例如 L_1 罚函数

$$P(x, \sigma) = f(x) + \sigma \left(\sum_{i=1}^{m_e} |c_i(x)| + \sum_{i=m_e+1}^m |c_i(x)_-| \right).$$

其中 $c_i(x)_-$ 定义如下：

$$\begin{cases} c_i(x)_- = c_i(x), & i \in \mathcal{E}, \\ c_i(x)_- = \min\{0, c_i(x)\}, & i \in \mathcal{I}. \end{cases}$$

非线性约束最优化

逐步二次规划法

下面的算法是Han(1977)提出的逐步二次规划方法:

- (0) 给定 $x^{(0)}$, $W_0 \in \mathbb{R}^{n \times n}$, $\sigma > 0$, $\rho \in (0, 1)$, $\varepsilon \geq 0$, 令 $k := 0$.
- (1) 求解子问题(128)给出 $d^{(k)}$, 如果 $\|d^{(k)}\| \leq \varepsilon$, 则停止; 否则求 $\alpha_k \in [0, \rho]$ 使得

$$P(x^{(k)} + \alpha_k d^{(k)}, \sigma) \leq \min_{0 \leq \alpha \leq \rho} P(x^{(k)} + \alpha d^{(k)}, \sigma) + \epsilon_k.$$

- (2) 置 $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$, 计算 W_{k+1} , 令 $k := k + 1$, 返回第(1)步。

非线性约束最优化

逐步二次规划法

可证明前述逐步二次规划法的收敛性结果如下:

定理: 假定 $f(x)$ 和 $c_i(x)$ 连续可微, 且存在常数 $M_1, M_2 > 0$ 使得

$$M_1 \|d\|^2 \leq d^T W_k d \leq M_2 \|d\|^2, \forall k \in \mathbb{N}, \forall d \in \mathbb{R}^n,$$

如果 $\|\lambda^{(k)}\|_\infty \leq \sigma$ 均成立, 则Han(1977)算法产生的点列 $\{x^{(k)}\}$ 的任何聚点都是问题(127)的K-T点。

非线性约束最优化 罚函数法

对于非线性约束最优化问题

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & c_i(x) = 0, i \in \mathcal{E} = \{1, \dots, m_e\} \\ & c_i(x) \geq 0, i \in \mathcal{I} = \{m_e + 1, \dots, m\}\end{array} \quad (129)$$

的罚函数，是指利用目标函数 $f(x)$ 和约束方程 $c(x)$ 所构造的具有“罚性质”的函数

$$P(x) = P(f(x), c(x)).$$

非线性约束最优化

罚函数法

所谓“罚性质”，即要求对问题的可行点 $x \in S$ 均有 $P(x) = f(x)$ ，而当约束条件破坏时有 $P(x) > f(x)$ 。

非线性约束最优化

罚函数法

为了描述约束条件被破坏的程度，我们定义 $c(x)_-$ 如下：

$$\begin{cases} c_i(x)_- = c_i(x), & i \in \mathcal{E}, \\ c_i(x)_- = \min\{0, c_i(x)\}, & i \in \mathcal{I}. \end{cases}$$

非线性约束最优化

罚函数法

罚函数一般可取为目标函数与“罚项”之和，即

$$P(x) = f(x) + \phi(c(x)_-).$$

罚项 $\phi(c(x)_-)$ 是定义在 \mathbb{R}^m 上的函数，它满足

$$\phi(0) = 0, \quad \lim_{\|c\| \rightarrow \infty} \phi(c) = +\infty.$$

非线性约束最优化

罚函数法

如Courant罚函数：

$$P_{\sigma}(x) = f(x) + \frac{\sigma}{2} \|c(x)\|_2^2,$$

其中 $\sigma > 0$ 是罚因子。

非线性约束最优化

罚函数法

考虑简单罚函数

$$P_{\sigma}(x) = f(x) + \frac{\sigma}{2} \|c(x)_{-}\|^2.$$

记 $x(\sigma)$ 是无约束问题 $\min_{x \in \mathbb{R}^n} P_{\sigma}(x)$ 的最优解，我们有如下引理。

引理： 若 $x(\sigma)$ 同时是非线性约束最优化问题(129)的可行点，则 $x(\sigma)$ 也是原问题的最优解。

非线性约束最优化

罚函数法

上述引理表明，只要选取充分大的罚因子 $\sigma > 0$ ，则通过求解无约束最优化问题应可找到相应约束最优化问题的最优解。

然而在实际计算中，确定大小合适的 σ 往往比较困难，故通常是选取一个单调增的罚因子序列 $\{\sigma_k\}$ 。

通过求解一系列无约束问题来获得约束最优化问题的解，这称为序贯无约束极小化技术(SUMT)。

非线性约束最优化

罚函数法

至此，我们可以给出罚函数法的迭代步骤：

(0) 任选初始点 $x^{(0)}$ ，给定初始罚因子 $\sigma_0 > 0$ 及 $\beta > 1, \varepsilon > 0$. 令 $k := 0$.

(1) 以 $x^{(k)}$ 作为初始迭代点求解无约束问题的极小点，即

$$x(\sigma_k) = \arg \min_{x \in \mathbb{R}^n} P_{\sigma_k}(x).$$

(2) 若 $\|c(x(\sigma_k))\| < \varepsilon$ ，则停止迭代并取 $x(\sigma_k)$ 为原约束问题的近似最优解；否则，置 $x^{(k+1)} = x(\sigma_k)$ ， $\sigma_{k+1} = \beta \sigma_k$ ，令 $k := k + 1$ 返回第(1)步。

非线性约束最优化

罚函数法

易得如下三个引理

引理1: 设 $\sigma_{k+1} > \sigma_k > 0$, 则有 $P_{\sigma_k}(x(\sigma_k)) \leq P_{\sigma_{k+1}}(x(\sigma_{k+1}))$,

$$\|c(x(\sigma_k))_-\| \geq \|c(x(\sigma_{k+1}))_-\|, \quad f(x(\sigma_k)) \leq f(x(\sigma_{k+1})).$$

引理2: 设令 \bar{x} 是原问题(129)的最优解, 则对任意的 $\sigma_k > 0$ 成立

$$f(\bar{x}) \geq P_{\sigma_k}(x(\sigma_k)) \geq f(x(\sigma_k)).$$

引理3: 令 $\delta = \|c(x(\sigma))_-\|$, 则 $x(\sigma)$ 也是约束问题

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & \|c(x)_-\| \leq \delta \end{array}$$

的最优解。

非线性约束最优化

罚函数法

[思考题：证明上述引理1...]

[思考题：证明上述引理3...]

非线性约束最优化

罚函数法

这里只给出引理2的证明:

由题设易得

$$P_{\sigma_k}(x(\sigma_k)) \geq f(x(\sigma_k)).$$

因为 \bar{x} 是原问题的最优解, 自然为可行点, 于是 $\sigma_k \|c(\bar{x})_-\|^2 = 0$.

又因为 $x(\sigma_k) = \arg \min_{x \in \mathbb{R}^n} P_{\sigma_k}(x)$, 则有

$$f(\bar{x}) = P_{\sigma_k}(\bar{x}) \geq P_{\sigma_k}(x(\sigma_k)).$$

非线性约束最优化

罚函数法

关于罚函数法的收敛性，我们有如下结果

定理1： 设罚函数法中的 ε 满足

$$\varepsilon > \min_{x \in \mathbb{R}^n} \|c(x) -\|,$$

则算法必有限终止。

[思考题：证明上述定理...]

非线性约束最优化

罚函数法

该定理表明，如果原约束问题存在可行点，则对任意给定的 $\varepsilon > 0$ ，算法都将有限终止于问题

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & \|c(x)_-\| \leq \delta\end{array}$$

的解，且 $\delta \leq \varepsilon$ 。

非线性约束最优化

罚函数法

定理2: 如果算法不有限终止, 则必有 $\min_{x \in \mathbb{R}^n} \|c(x)_-\| \geq \varepsilon$,

且 $\lim_{k \rightarrow \infty} \|c(x(\sigma_k))_-\| = \min_{x \in \mathbb{R}^n} \|c(x)_-\|$. 此时, $\{x(\sigma_k)\}$ 的任何聚点 x^* 都是问题

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & \|c(x)_-\| = \min_{y \in \mathbb{R}^n} \|c(y)_-\| \end{array}$$

的解。

非线性约束最优化

乘子罚函数

为了叙述简单，仅考虑等式约束问题

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & c(x) = 0\end{array}\quad (130)$$

其中 $c(x) = (c_1(x), \dots, c_{m_e}(x))^T$.

设 x^* 是上述问题的最优解且 λ^* 是相应的Lagrange乘子，由Kuhn-Tucher定理知， x^* 必是Lagrange函数

$$L(x, \lambda^*) = f(x) - (\lambda^*)^T c(x)$$

的稳定点。但一般而言， x^* 并不是Lagrange函数的极小点。

非线性约束最优化

乘子罚函数

我们考虑乘子罚函数（也称增广Lagrange函数）

$$P(x, \lambda, \sigma) = L(x, \lambda) + \frac{\sigma}{2} \|c(x)\|_2^2.$$

由于增广Lagrange函数的性态，只要取足够大的罚因子 σ 而不必趋向无穷大，就可通过极小化 $P(x, \lambda, \sigma)$ 求得原问题的最优解。

非线性约束最优化

乘子罚函数

我们事先并不知道最优乘子向量 λ^* ，因此用乘子 λ 代替，得到增广Lagrange罚函数：

$$P(x, \lambda, \sigma) = f(x) - \lambda^T c(x) + \frac{\sigma}{2} \|c(x)\|_2^2.$$

一般的策略是，先给定充分大的 σ 和乘子向量的初始估计 λ ，然后在迭代过程中修正乘子 λ 力图使之趋向最优乘子 λ^* 。

如何修正？

非线性约束最优化

乘子罚函数

基于增广Lagrange函数的迭代算法:

(0) 给定初始点 $x^{(0)}$ 和乘子向量初始估计 $\lambda^{(0)}$, 给定罚因子 $\sigma_0 > 0$, 常数 $\beta > 1$, $0 < \gamma < 1$ 及容许误差 $\varepsilon > 0$. 令 $k := 0$, $\sigma := \sigma_0$.

(1) 以 $x^{(k)}$ 为初点求解无约束问题的极小点, 即

$$x^{(k+1)} = \arg \min_{x \in \mathbb{R}^n} P(x, \lambda^{(k)}, \sigma).$$

(2) 若 $\|c(x^{(k+1)})\| < \varepsilon$, 则停止迭代并取 $x^{(k+1)}$ 作为原问题的近似最优解; 否则, 更新乘子向量

$$\lambda^{(k+1)} = \lambda^{(k)} - \sigma c(x^{(k+1)}).$$

(3) 如果 $\frac{\|c(x^{(k+1)})\|}{\|c(x^{(k)})\|} \geq \gamma$, 则置 $\sigma := \beta\sigma$. 令 $k := k + 1$ 返回第(1)步。

非线性约束最优化

乘子罚函数

记 $A(x) = [\nabla c(x)]^T$, 由于 $c(x^*) = 0$, 我们易得

$$\nabla_x P(x^*, \lambda^*, \sigma) = \nabla_x L(x^*, \lambda^*) = 0,$$

$$\nabla_{xx}^2 P(x^*, \lambda^*, \sigma) = \nabla_{xx}^2 L(x^*, \lambda^*) + \sigma A(x^*) A(x^*)^T.$$

设在 x^* 处满足二阶充分条件, 即对 $\forall d$ 使得 $A(x^*)^T d = 0$ 的非零向量, 均有

$$d^T \nabla_{xx}^2 L(x^*, \lambda^*) d > 0.$$

那么在二阶充分条件的假定下, 对于充分大的 σ , 可证

$$\nabla_{xx}^2 P(x^*, \lambda^*, \sigma)$$

是正定阵。

非线性约束最优化

乘子罚函数

定理： 设 x^* 和 λ^* 满足等式约束问题(130)局部最优解的二阶充分条件，则存在 σ_0 使得当 $\sigma > \sigma_0$ 时， x^* 是函数 $P(x, \lambda^*, \sigma)$ 的严格局部极小点。

[思考题：证明上述定理...]

非线性约束最优化

乘子罚函数

定理： 若 \bar{x} 是等式约束问题(130)的可行解，且对于某个 $\bar{\lambda}$ ， \bar{x} 满足 $P(x, \bar{\lambda}, \sigma)$ 的极小点二阶充分条件，则 \bar{x} 是问题(130)的严格局部最优解。

[思考题：证明上述定理...]

非线性约束最优化

障碍函数

基本思想 在迭代中总是从内点出发，并通过引入障碍函数使之保持在可行域内部进行搜索。因此，这种方法适用于不等式约束的非线性最优化问题。

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & g_i(x) \geq 0, i = 1, \dots, m.\end{array}\quad (131)$$

现将可行域内部记作 $\text{int}S$, 其中 $S = \{x \mid g_i(x) \geq 0, i = 1, \dots, m\}$. 保持迭代点含于可行域内部的方法是定义如下障碍函数:

$$B(x, \theta) = f(x) + \theta\psi(x)$$

其中障碍因子 θ 是很小的正数, $\psi(x)$ 是连续函数, 当 x 趋于可行域边界时, $\psi(x) \rightarrow +\infty$.

非线性约束最优化

障碍函数

两种重要的障碍形式是:

$$\psi(x) = \sum_{i=1}^m \frac{1}{g_i(x)} \quad \text{and} \quad \psi(x) = - \sum_{i=1}^m \log g_i(x)$$

这样, 当 x 趋向可行域边界时, 函数 $B(x, \theta) \rightarrow +\infty$. 否则, 由于 θ 很小, 函数 $B(x, \theta)$ 的取值近似于 $f(x)$.

非线性约束最优化

障碍函数

因此，我们可通过求解下列问题得到原问题(131)的近似解：

$$\begin{array}{ll}\min & B(x, \theta) \\ \text{s.t.} & x \in \text{int} S\end{array} \quad (132)$$

由于 $\psi(x)$ 的存在，在可行域边界形成一道“围墙”，因此上述障碍问题(132)的解 $\bar{x}(\theta)$ 必含于可行域的内部。

需要解释的是，障碍问题(132)表面上看起来仍是带约束的最优化问题，且它的约束条件比原来的约束还要复杂。但是，由于函数 $\psi(x)$ 的障碍阻挡作用是自动实现的，因此从计算观点看，求解(132)完全可当作无约束问题来处理。

非线性约束最优化

障碍函数

因此，我们可通过求解下列问题得到原问题(131)的近似解：

$$\begin{array}{ll}\min & B(x, \theta) \\ \text{s.t.} & x \in \text{int} S\end{array} \quad (132)$$

由于 $\psi(x)$ 的存在，在可行域边界形成一道“围墙”，因此上述障碍问题(132)的解 $\bar{x}(\theta)$ 必含于可行域的内部。

需要解释的是，障碍问题(132)表面上看起来仍是带约束的最优化问题，且它的约束条件比原来的约束还要复杂。但是，由于函数 $\psi(x)$ 的障碍阻挡作用是自动实现的，因此从计算观点看，求解(132)完全可当作无约束问题来处理。

非线性约束最优化

障碍函数

于是，我们可以给出障碍函数法的计算步骤如下：

(0) 给定初始点 $x^{(0)} \in \text{int}S$, 初始障碍因子 $\theta_0 > 0$, $\beta \in (0, 1)$, $\varepsilon > 0$.
令 $k := 0$.

(1) 以 $x^{(k)}$ 作为初始迭代点求解下列问题：

$$\begin{array}{ll} \min & f(x) + \theta_k \psi(x) \\ \text{s.t.} & x \in \text{int}S \end{array}$$

记求得的极小点为 $x(\theta_k)$.

(2) 若 $\theta_k \psi(x(\theta_k)) < \varepsilon$, 则停止计算并取 $x(\theta_k)$ 为原问题的近似最优解；否则，置 $x^{(k+1)} = x(\theta_k)$, $\theta_{k+1} = \beta \theta_k$, 令 $k := k + 1$ 返回第(1)步。

非线性约束最优化

障碍函数

定理： 设 $\theta_k > \theta_{k+1} > 0$, 记 $x(\theta) = \arg \min_x B(x, \theta)$, 则有

$$B(x(\theta_k), \theta_k) \geq B(x(\theta_{k+1}), \theta_{k+1}),$$

$$\psi(x(\theta_k)) \leq \psi(x(\theta_{k+1})),$$

$$f(x(\theta_k)) \geq f(x(\theta_{k+1})).$$

[思考题：证明上述定理...]

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & c_E(x) = 0 \\ & c_I(x) \geq 0\end{array}\quad (133)$$

$$\begin{array}{ll}\min & f(x) \\ \text{s.t.} & c_E(x) = 0 \\ & c_I(x) - s = 0 \\ & s \geq 0\end{array}\quad (134)$$

The Karush-Kuhn-Tucker (KKT) conditions for the nonlinear program (134) can be written as

$$\begin{aligned}\nabla f(x) - A_E(x)^T y - A_I(x)^T z &= 0 \\ Sz - \mu 1 &= 0 \\ c_E(x) &= 0 \\ c_I(x) - s &= 0\end{aligned}\tag{135}$$

with $\mu = 0$, together with $s \geq 0, z \geq 0$.

Here $A_E(x)$ and $A_I(x)$ are the Jacobian matrices of the functions $c_E(x)$ and $c_I(x)$, respectively, and y and z are their Lagrange multipliers. We define S and Z to be the diagonal matrices whose diagonal entries are given by the vectors s and z , respectively, and let $1 = (1, \dots, 1)^T$.

Applying Newton's method to the KKT system (135), in the variables x, s, y, z , we obtain

$$\begin{pmatrix} \nabla_{xx}^2 L & 0 & -A_E(x)^T & -A_I(x)^T \\ 0 & Z & 0 & S \\ -A_E(x) & 0 & 0 & 0 \\ -A_I(x) & I & 0 & 0 \end{pmatrix} \begin{pmatrix} p_x \\ p_s \\ p_y \\ p_z \end{pmatrix} = - \begin{pmatrix} \nabla f(x) - A_E(x)^T y - A_I(x)^T z \\ Sz - \mu 1 \\ -c_E(x) \\ -c_I(x) + s \end{pmatrix} \quad (136)$$

where $L(x, s, y, z)$ denotes the Lagrange function

$$L(x, s, y, z) = f(x) - y^T c_E(x) - z^T (c_I(x) - s).$$

The system (136) is called the primal-dual system. After the step $p = (p_x, p_s, p_y, p_z)$ has been determined, we compute the new iterate (x^+, s^+, y^+, z^+) as

$$\begin{aligned}x^+ &= x + \alpha_s^{\max} p_x, & s^+ &= s + \alpha_s^{\max} p_s, \\y^+ &= y + \alpha_z^{\max} p_y, & z^+ &= z + \alpha_z^{\max} p_z,\end{aligned}$$

where

$$\begin{aligned}\alpha_s^{\max} &= \max\{\alpha \in (0, 1] : s + \alpha p_s \geq (1 - \tau)s\}, \\ \alpha_z^{\max} &= \max\{\alpha \in (0, 1] : z + \alpha p_z \geq (1 - \tau)z\},\end{aligned}\tag{137}$$

with $\tau \in (0, 1)$ (A typical value of τ is 0.995). The condition (137), called the fraction to the boundary rule, prevents the variables s and z from approaching their lower bounds of 0 too quickly.

本章作业（非线性约束最优化）

- **Exercise 7.1:** 请写出非线性方程组求根 $F(x) = 0$ 的牛顿迭代算法，并用于求解无约束最优化问题

$$\min_{x \in \mathbb{R}^n} f(x),$$

请给出相应的迭代格式，并说明理由。

- **Exercise 7.2:** 证明(125)中定义的 $\psi(x, \lambda)$ 是关于Lagrange-Newton法的下降函数。
- **Exercise 7.3:** 证明罚函数法迭代求解的引理1。
- **Exercise 7.4:** 证明罚函数法迭代求解的引理3。
- **Exercise 7.6:** 证明罚函数法求解带误差界近似问题的算法有限终止性。
- **Exercise 7.7:** 给出约束最优化问题的二阶充分最优性条件，并用于说明增广Lagrange函数的极小点与原问题最优解的等价性。

- 无约束最优化问题

- 最速下降法
- 牛顿法
- 拟牛顿法
- 共轭梯度法
- 信赖域方法

- 带约束最优化问题

- Lagrange-Newton法
- 逐步二次规划法
- 罚函数法
- 乘子罚函数法
- 障碍函数法
- 内点法

- 模型对问题的近似刻画与可求解性之间的平衡
- 求解算法的针对性选择
- 基于问题背景的初始解构造

谢谢!