Numerical Optimization

Lecture 1: Introduction

王浩

信息科学与技术学院

Email: wanghao1@shanghaitech.edu.cn

本节内容：

- 课程简介
- 实例：从传统到前沿
- 优化问题分类、优化求解器简介

一、课程简介

王浩

信息学院2号楼401

wanghao1@shanghaitech.edu.cn

请登录学校教学互助平台查询最新的课程资料

作业：

- 可能每周课后均有一点点作业
- 作业尽量用latex完成，如有程序附源代码
- 布置后一周交

助教答疑：左世纪、韩子睿

- 时间：待定
- 地点：待定

教师答疑：不固定时间

作业50% + 考试50%

简单的最优化问题（人教版小学数学四年级上册）

简单的最优化问题（苏教版高中数学）

3.3 二元一次不等式组与简单的线性规划问题

我们先考察生产中遇到的一个问题：

某工厂生产甲、乙两种产品，生产 1 t 甲种产品需要 A 种原料4 t、B 种原料 12 t，产生的利润为 2 万元；生产 1 t 乙种产品需要 A 种原料 1 t、B 种原料 9 t，产生的利润为 1 万元. 现有库存 A 种原料 10 t、B 种原料 60 t，如何安排生产才能使利润最大？

为理解题意，可将已知数据整理成下表：

	A 种原料(t)	B 种原料(t)	利　润(万元)
甲种产品(1 t)	4	12	2
乙种产品(1 t)	1	9	1
现有库存(t)	10	60	

设计划生产甲、乙两种产品的吨数分别为 x，y，利润为 P(万元). 根据题意，A，B 两种原料分别不得超过 10 t 和 60 t，又产量不可能是负数，于是可得二元一次不等式组

$$\begin{cases}4x+y\leqslant 10,\\ 12x+9y\leqslant 60,\\ x\geqslant 0,\\ y\geqslant 0,\end{cases}$$

即

$$\begin{cases}4x+y\leqslant 10,\\ 4x+3y\leqslant 20,\\ x\geqslant 0,\\ y\geqslant 0.\end{cases}$$

因此，上述问题转化为如下的一个数学问题：在约束条件

$$\begin{cases}4x+y\leqslant 10,\\ 4x+3y\leqslant 20,\\ x\geqslant 0,\\ y\geqslant 0\end{cases}$$

下，求出 x，y，使利润

$$P=2x+y$$

达到最大.

这是一个含有两个变量 x 和 y 的函数，称为目标函数.

● 如何解决这个问题？

3.3.1 二元一次不等式表示的平面区域

我们分两步求解上面的问题：

第一步　研究问题中的约束条件，确定数对 (x, y) 的范围；

第二步　在第一步得到的数对 (x, y) 的范围中，找出使 P 达到最大的数对 (x, y).

先讨论第一步.

如图 3-3-1(1)，直线 l：$4x+y=10$ 将平面分成上、下两个半平面区域，直线 l 上的点的坐标满足方程 $4x+y=10$，即 $y=10-4x$，直线 l 上方的平面区域中的点的坐标满足不等式 $y>10-4x$，直线 l 下方的平面区域中的点的坐标满足不等式 $y<10-4x$.

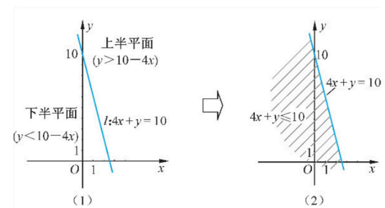


图 3-3-1

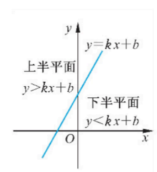


图 3-3-2

因此，$4x+y\leqslant 10$ 在平面上表示的是直线 l 及直线 l 下方的平面区域，即图 3-3-1(2) 中的阴影部分(包括边界直线 l).

一般地，直线 $y=kx+b$ 把平面分成两个区域(图 3-3-2)：

$y>kx+b$ 表示直线上方的平面区域；

$y<kx+b$ 表示直线下方的平面区域.

思　考

对于二元一次不等式 $Ax+By+C>0$ $(A^2+B^2\neq 0)$，如何确定它所表示的平面区域？

例 1　画出下列不等式所表示的平面区域：

(1) $y>-2x+1$；　　(2) $x-y+2>0$.

解　(1)，(2) 两个不等式所表示的平面区域如图 3-3-3(1)，(2) 所示.

课程简介

数学规划(Mathematical Programming)，也称数学最优化(Mathematical Optimization)，是应用数学、计算数学与运筹学的交叉领域。

学科归属

- 国内：数学系（少数在控制、金融、管理系，上科大在CS）

数学 →

理论数学	方程、泛函、线性/非线性泛函、集合论、微分几何、黎曼几何、代数、群论、拓扑、概率论、数论……
计算数学	数值代数、数值分析、偏微分方程数值解、有限元、并行计算、数值最优化、数理逻辑……
应用数学	概率统计、数学建模、数值方法、运筹学、数学实验、计算机基础、金融数学………………
概率统计	

学科归属

- 国外：工业工程系（绝大多数，比如CMU就有例外）
- 何为工业工程 Industrial Engineering（IE）？

世界第一万金油专业：IE&OR (Industrial Engineering & Operations Research)

生物、化工、管理
data（数据）
Control（控制）
Supply Chain Management（供货链）
Inventory（库存）
Scheduling（调度）
Transportation（运输）
Financial Optimization（金融）
Simulation（仿真）
Communication（通信）
....

数学建模

建模能力

最优化问题
Optimization Problem

计算、分析、动手能力

开发最优化算法

本门课目的

IEOR的历史

- 1908年，宾州州立大学第一次开设了工业工程课程。1909年成系。1933年Cornell第一次授予工业工程博士学位。
- 20世纪二战期间：战争中的战术效率研究和试验。诞生了（运筹学）Operations Research行业。
- 1949年：兰德公司(RAND)，军用转民用。
- 1950–1960：第一个工业界的成功案例：shell、Exxon、BP石油公司的downstream石油冶炼中的pooling问题。
- 欧美大学的运筹系、管理科学系、工业工程系、系统科学系纷纷涌现。
- 现代计算机的诞生和发展，促使运筹学能够研究求解更加复杂的最优化问题。

欧美IEOR强校

- 美国：Georgia Tech、Northwestern、Berkeley、UIUC、Cornell、Stanford、PennState、VirginiaTech、Purdue、Lehigh、Columbia、TAMU、Rice、Florida、CMU……

欧美IEOR强校


UC Berkeley | Berkeley Engineering | Alumni | Industry | Give
IEOR - Designing a More Efficient World
Bookmarks
chrome://bookmarks
Berkeley
Industrial Engineering & Operations Research
COLLEGE OF ENGINEERING
About
Research
Academics
News
People
Welcome to UC Berkeley's Industrial Engineering and Operations Research Department. At Berkeley IEOR, we expand the frontiers of optimization, stochastics and data science enabling transformative decision analytics and technologies to solve grand challenges in transportation, supply chains, healthcare, energy, robotics, finance and risk management.
Optimization and Algorithms
Machine Learning and Data Science
Stochastic Modeling and Simulation
Robotics and Automation
Supply Chain Systems
Financial Systems
Energy Systems
Healthcare Systems

欧美IEOR强校

- 英国：Oxford、Manchester、Bristol、Edinburgh……
- 德国：Technische Universität Berlin、Technische Universität München
- 加拿大：Toronto，McGill
- ………
-

Manufacturing Technology Laboratory: Automation and Robotics Laboratory (Lehigh)

Industrial and Systems Engineering (ISE) Computer Lab (Lehigh)

COR@L

COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

WELCOME! PEOPLE PUBLICATIONS DATA SETS LINKS

Data Sets

MIP Instances
A Collection of Mixed Integer Programming Instances Collected by Jeff Linderoth from a number of sources, including instances submit to the NEOS Server. Files are in bzipped MPS format.

SPP Instances
A Collection of Set Partioning Problem Instances, from the paper A Parallel, Linear Programming Based Heuristic for Large Scale Set Partitioning Problems, by J. T. Linderoth, E. K. Lee, and M. W. P. Savelsbergh, that appeared in INFORMS Journal on Computing, 13 (2001), pp. 191-209.

SP Instances
Stochastic linear programming instances created or collected by Jeff Linderoth. Files are in (bzipped) SMPS format.

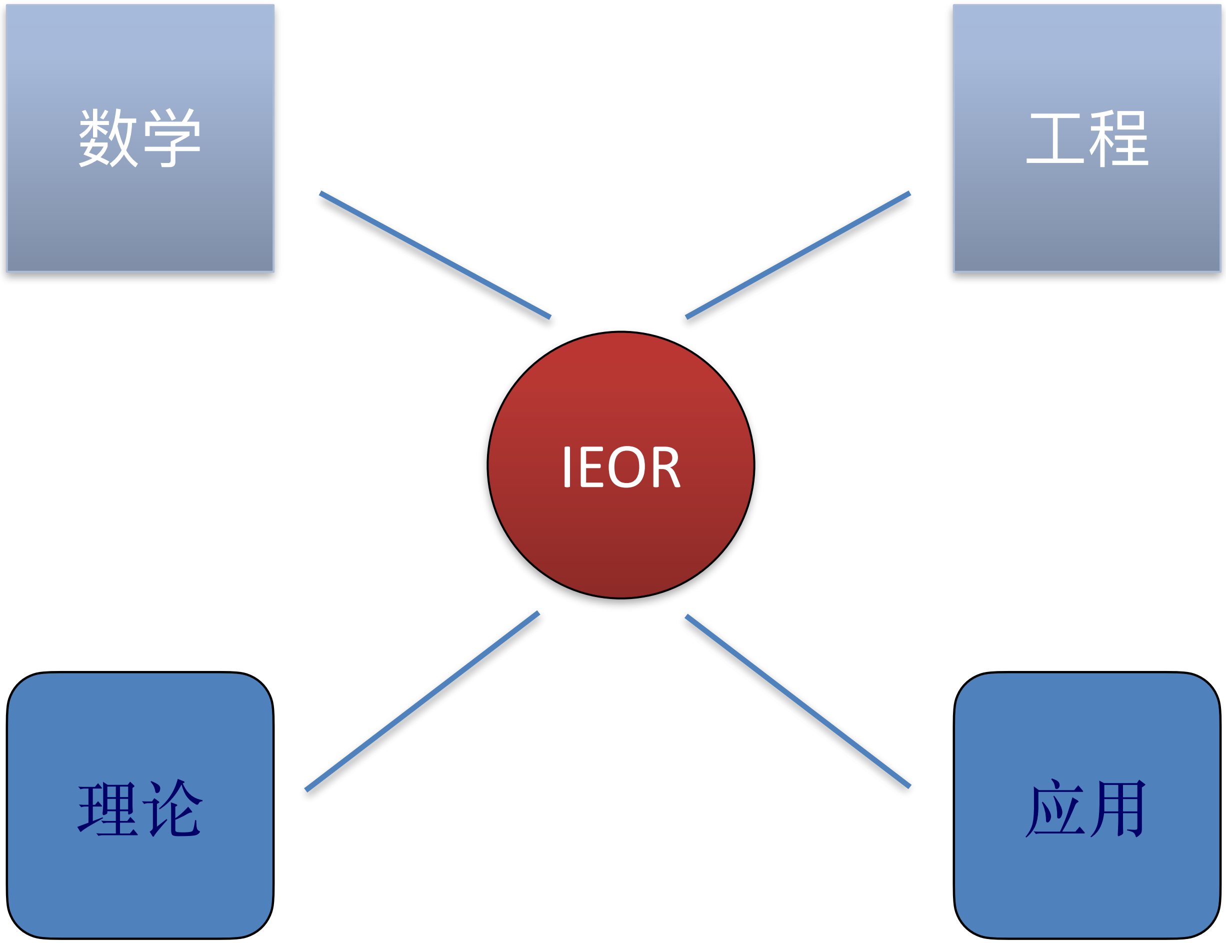
QAPLIB Instances
A collection of Quadratic Assignment Problem instances collected by R.E. BURKARD, E. ÇELA, S.E. KARISCH and F. RENDL.

Bilevel Instances
A collection of Bilevel Optimization Instances collected by Ted Ralphs. Files are in MPS format with a bilevel auxiliary file.

中国运筹学发展历史

- 1956年第一个运筹学小组：中科学力学所，钱学森、许国志先生1955年回国成立。
- 1959年第二个运筹学小组：力学所小组+数学所小组，主要研究排队论、非线性规划和图论，运输理论、动态规划和经济分析。
- 1963年数学所的运筹学研究室为中国科技大学应用数学系的第一届学生（58 届）开设了较为系统的运筹学专业课
- 1950年末，主要研究运输问题，“打麦场的选址问题”
- 1963年，华罗庚先生带队到农村、工厂讲解基本的优化技术和统筹方法，使之用于日常的生产和生活中
- 改开后，组合优化、生产系统优化、图论和非线性规划领域长足发展

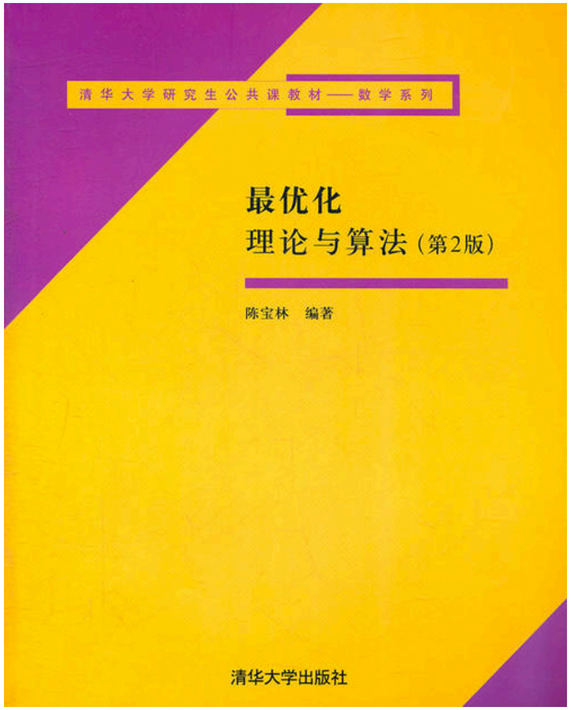
运筹学发展的回顾与展望，胡晓东，袁亚湘，章祥荪


数学
工程
IEOR
理论
应用

学完课程后，收获以下能力

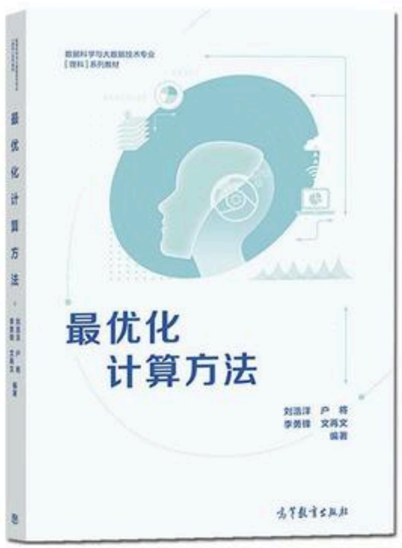
- 调用求解器：作为工程师理解并运用最新最有效的优化方法求解特定的应用
- 建模：为具有挑战性的科学问题、管理问题和工程问题研发更好的方法，提供更恰当的表述方式
- 算法：作为数学最优化的专家(应用数学和运筹学)发展更好的求解方法

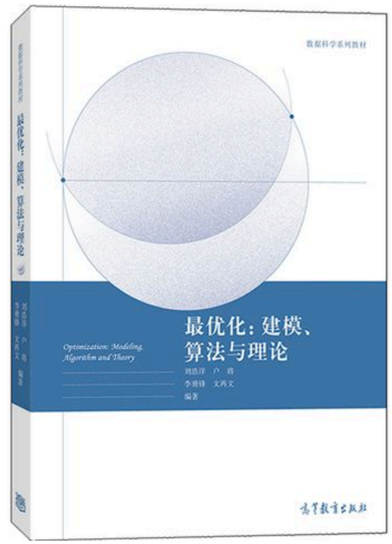
推荐参考书： 线性规划部分

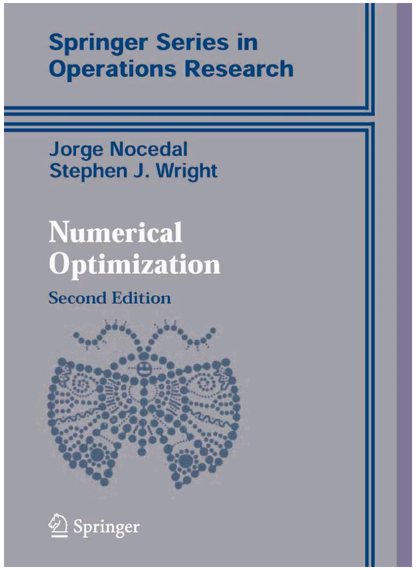


[1] 陈宝林，最优化理论与算法(第2版)，清华大学出版社, 2005.10.

推荐参考书： 非线性规划部分







[1] Numerical Optimization, Jorge Nocedal and Stephen Wright, 2nd Edition (2006)

[2] 最优化计算方法，高等教育出版社，文再文 等编著。

其他优秀教材：Course Notes for EE227C (Spring 2018): Convex Optimization and Approximation. https://ee227c.github.io/

二、实例：从传统到前沿

数学描述与例子

数学规划问题(基本形式)

$$
\begin{aligned}
\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\
\text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\
& c_i(x) \leq 0, \quad i \in \mathcal{I} \\
& x \in \Omega
\end{aligned}
$$

一般考虑极小化minimize
也有极大化问题maximize
分别简写为min、max

另外也有inf问题、sup问题

subject to简写为s.t.

三要素

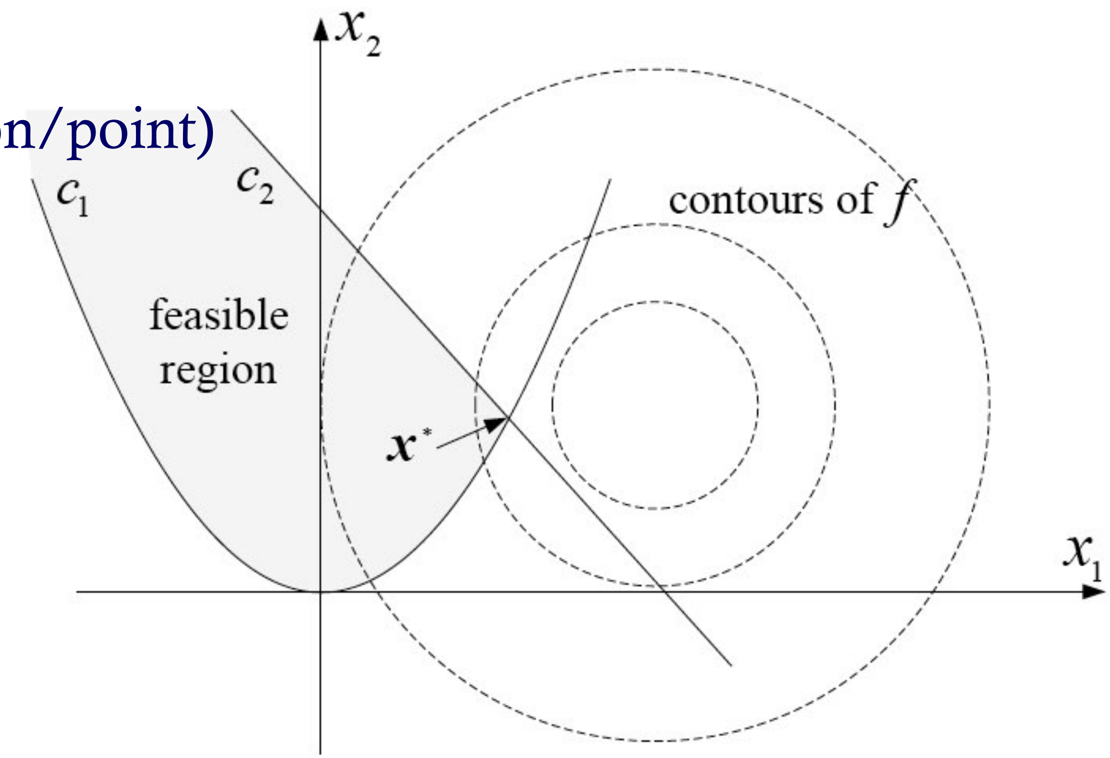
- 目标(objective)：系统性能的一种“量的度量”(利润、时间、势能)
- 变量(variables)：目标所依赖的系统的“某些可控的特征”
- 约束条件(constraints)：变量经常以某种方式受限制(如贷款利率的量不能是负的)

小例子 (toy example):

$$
\begin{aligned}
\text{minimize} \quad & (x_1-2)^2+(x_2-1)^2 \\
\text{subject to} \quad & x_1^2-x_2\le 0 \\
& x_1+x_2\le 2.
\end{aligned}
$$

图解法
- 画出可行域
- 做目函数的等高线
- 确定临界点

可行域(feasible region)
可行集(feasible set)
可行解/点(feasible solution/point)
等高线/等值线(contour)
最优解(optimal solution)
极小点(minimizer)
解(solution)
最小值(minima)

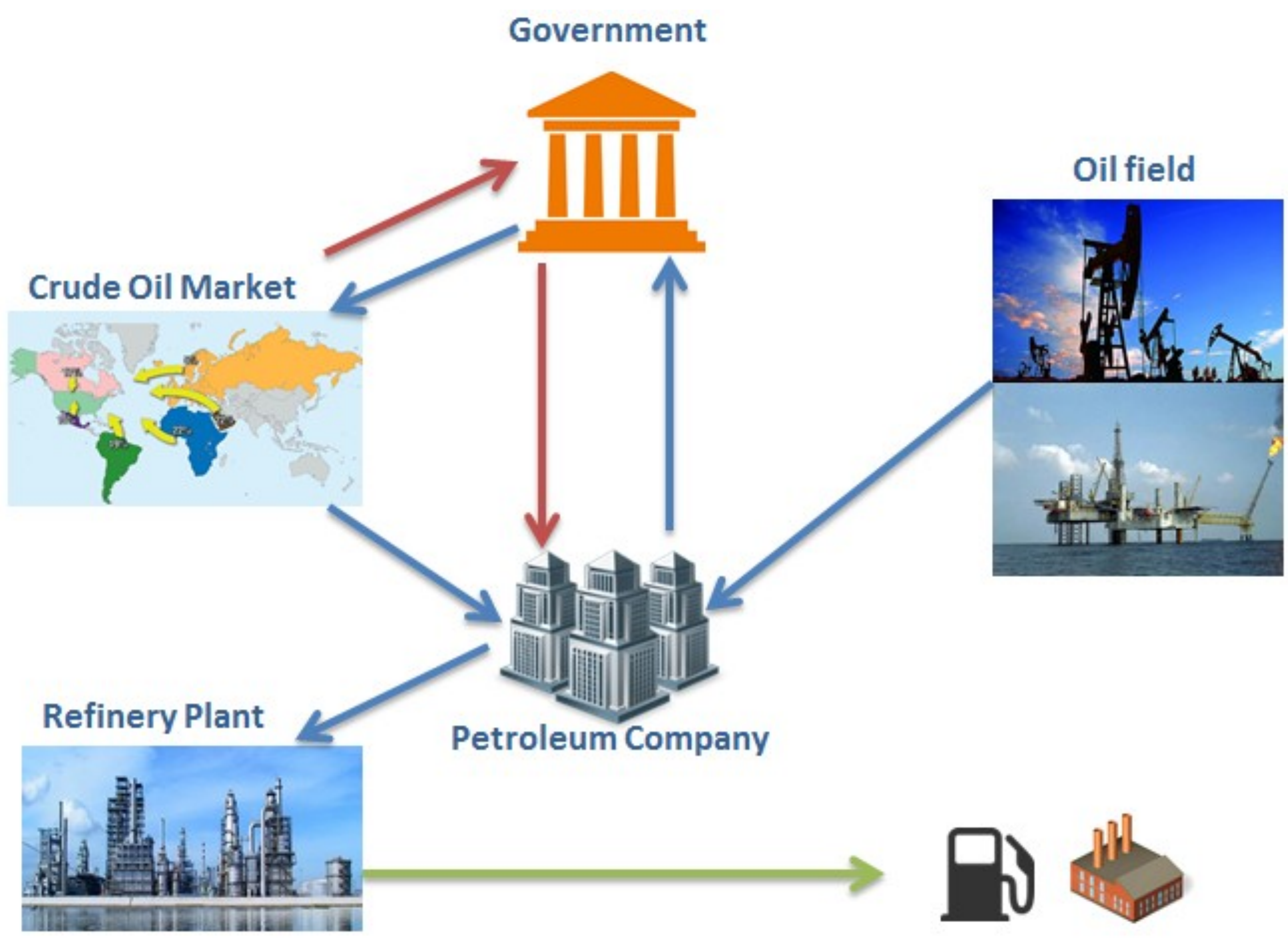


优化建模(modeling)：识别出给定问题的目标、变量和约束的过程

- 建立恰当模型：第一步、最重要的一步(太简单,不能为实际问题提供有用的信息；太复杂, 不易求解)

- 选择特定算法：很重要，决定求解速度及质量(无通用优化算法，有求解特定类型优化问题的算法)

实例1：第一个工业界的成功案例、石油公司的下线(downstream)原油冶炼中的池化(pooling)问题

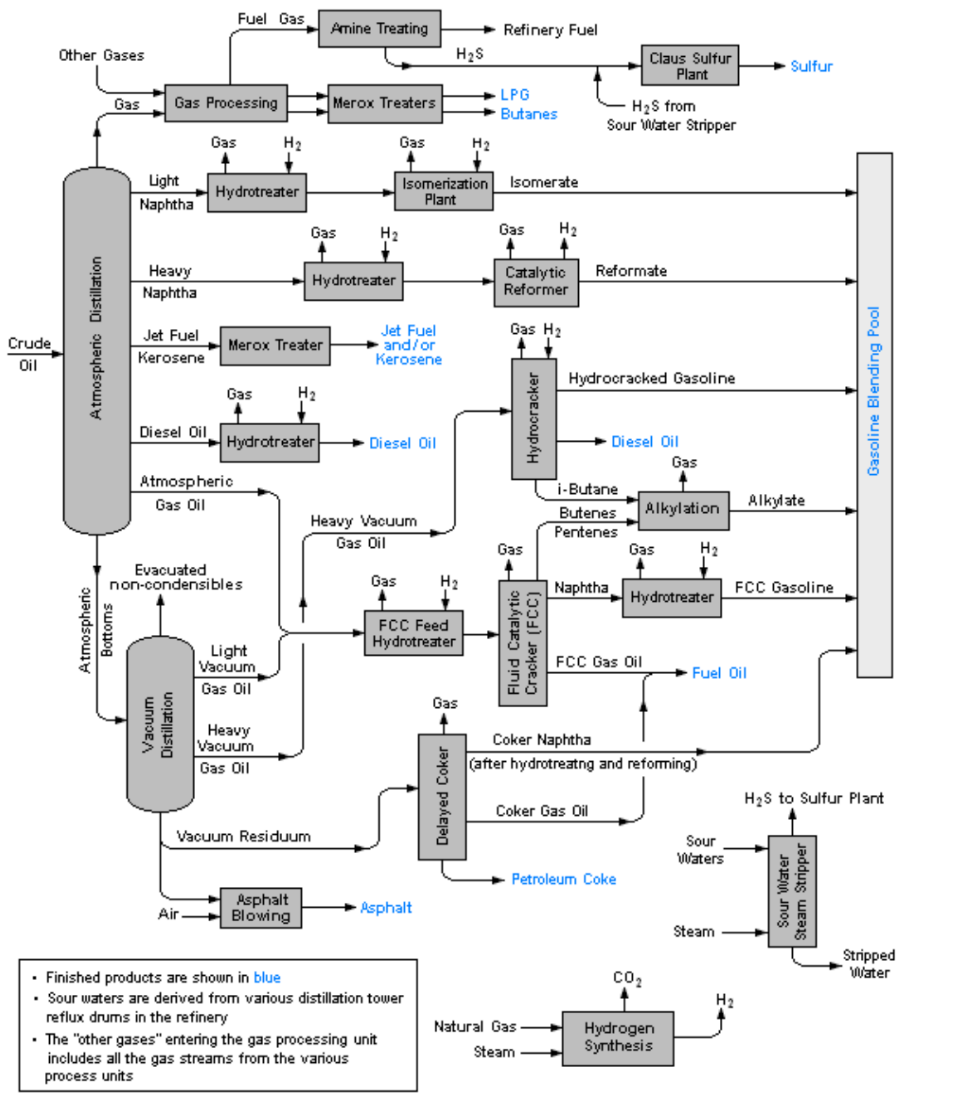


Crude Oil: Type 1
Crude Oil: Type 1
Crude Oil: Type 1
Crude Oil: Type 1
....

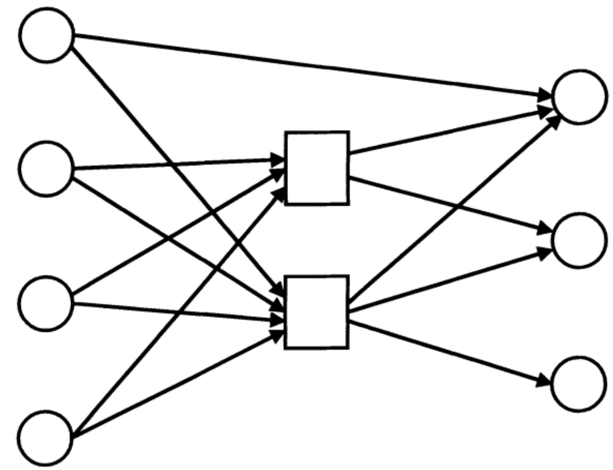
Products:
Motor fuel (87, 89, 91)
Diesel oil
Liquefied petroleum gases (LPG)
Jet aircraft fuel
Kerosene
Heating fuel oils
Lubricating oils
Asphalt
Petroleum coke
....

- Petroleum industry has the longest history of making profit by OR techniques
- The use of optimization techniques in refinery models and decisions is the first triumph of OR in industry
- To maximize the profit by deciding the amount of crude oil to purchase for each type

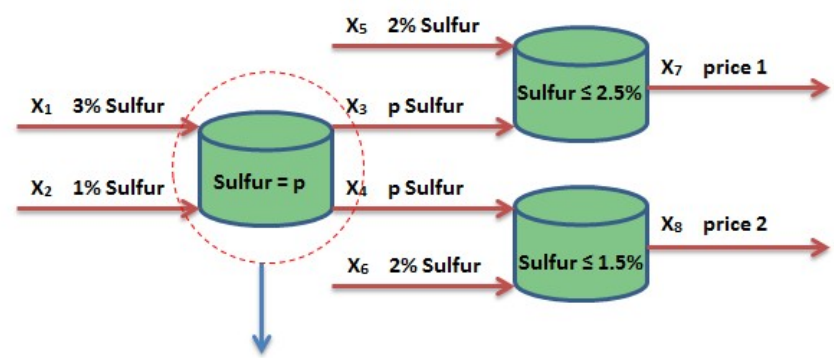
Refinery Flowchart



Pooling model

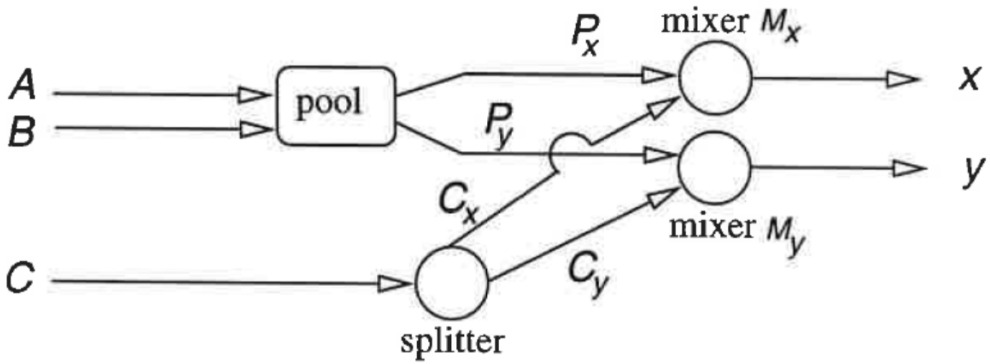


Raw Materials Pools Products



$p X_3 + p X_4 = 3\% X_1 + 1\% X_2$ **Bilinearity: nonlinear! nonconvex!**

Example: Haverly Pooling Problem



A, B, C : quantities of crude oil A, B, C, respectively
x, y : quantities of products x and y
p : sulfur content of outstream of the pool
P_x, P_y : outstreams of the pool
C_x, C_y : outstreams of the splitter

$$\max_{x,y,A,B,p,C_x,C_y,P_x,P_y} \quad 9x + 15y - 6A - 8B - 10(C_x + C_y)$$

s.t.

$$P_x + P_y - A - B = 0$$
$$x - P_x - C_x = 0$$
$$y - P_y - C_y = 0$$
$$pP_x + 2C_x - 2.5x \le 0$$
$$pP_y + 2C_y - 1.5y \le 0$$
$$pP_x + pP_y - 3A - B = 0$$
$$0 \le x \le 200, 0 \le y \le 200, 0 \le p \le 100$$
$$0 \le A, B, C_x, C_y, P_x, P_y \le 500$$

sulfur content of A=3
sulfur content of B=1
sulfur content of C=2
sulfur content of x ≤ 2.5
sulfur content of y ≤ 1.5
Demands of x and y are ≤ 200
Supplies of A, B, C_x, C_y, P_x, P_y ≤ 500

优化实例2：田忌赛马

齐使者如梁，孙膑以刑徒阴见，说齐使。齐使以为奇，窃载与之齐。齐将田忌善而客待之。忌数与齐诸公子驰逐重射。孙子见其马足不甚相远，马有上、中、下辈。于是孙子谓田忌曰："君弟重射，臣能令君胜。"田忌信然之，与王及诸公子逐射千金。及临质，孙子曰："今以君之下驷(sì)与彼上驷，取君上驷与彼中驷，取君中驷与彼下驷。"既驰三辈毕，而田忌一不胜而再胜，卒得王千金。于是忌进孙子于威王。威王问兵法，遂以为师。——《史记》卷六十五：《孙子吴起列传第五》

解答． 用1, 2, 3分别表示上、中、下等马.

令 $x_{ij} = \begin{cases} 1, & \text{田}i \to \text{齐王的}j\text{等马} \\ 0, & \text{否则} \end{cases}$

$$C = (c_{ij}) = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

收益矩阵(payoff matrix)

指派问题(assignment problem)

线性规划

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{3} \sum_{j=1}^{3} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{3} x_{ij} = 1, \ i = 1,2,3 \\ & \sum_{i=1}^{3} x_{ij} = 1, \ j = 1,2,3 \end{aligned}$$

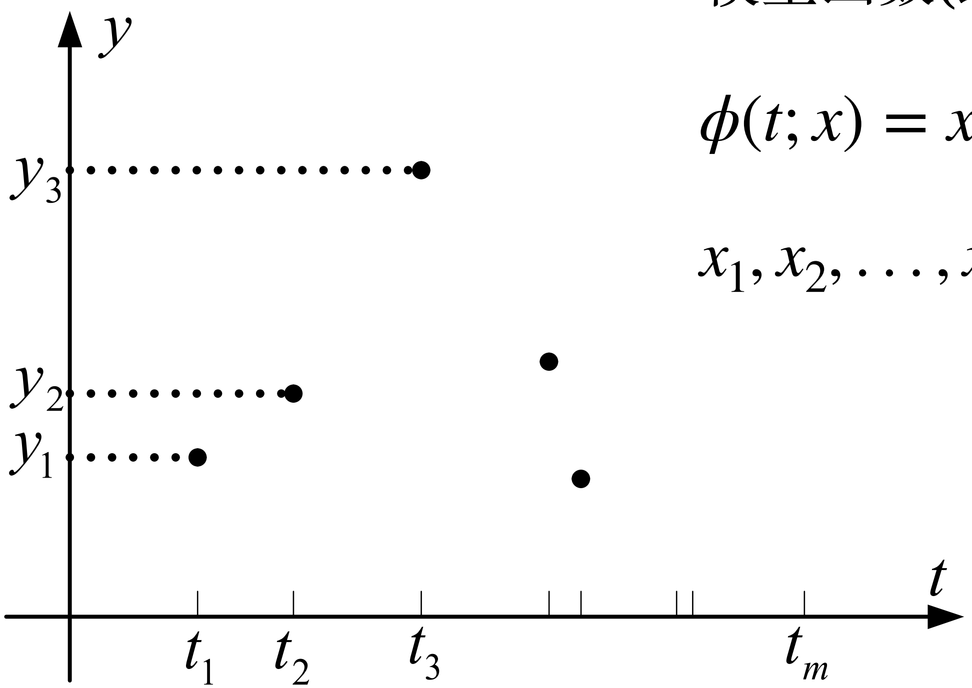
$$x_{ij} \in \{0,1\}, \ i,j = 1,2,3.$$

0-1整数线性规划

实例3：数据拟合(data fitting)、稀疏优化

已知：$t_1, t_2, \ldots, t_m$处的测量值$y_1, y_2, \ldots, y_m$

推测：信号具有指数衰减和振荡行为！



模型函数(经验函数)：

$$\phi(t;x) = x_1 + x_2 e^{-(x_3-t)^2/x_4} + x_5 \cos(x_6 t)$$

$x_1, x_2, \ldots, x_6$: 待定参数

- 变量： $x = (x_1, x_2, \ldots, x_6)^T$
- 余量/残差(residual)：

$$\underset{x\in\mathbb{R}^6}{\text{minimize}}\, f(x) = r_1^2(x) + \cdots + r_m^2(x)$$

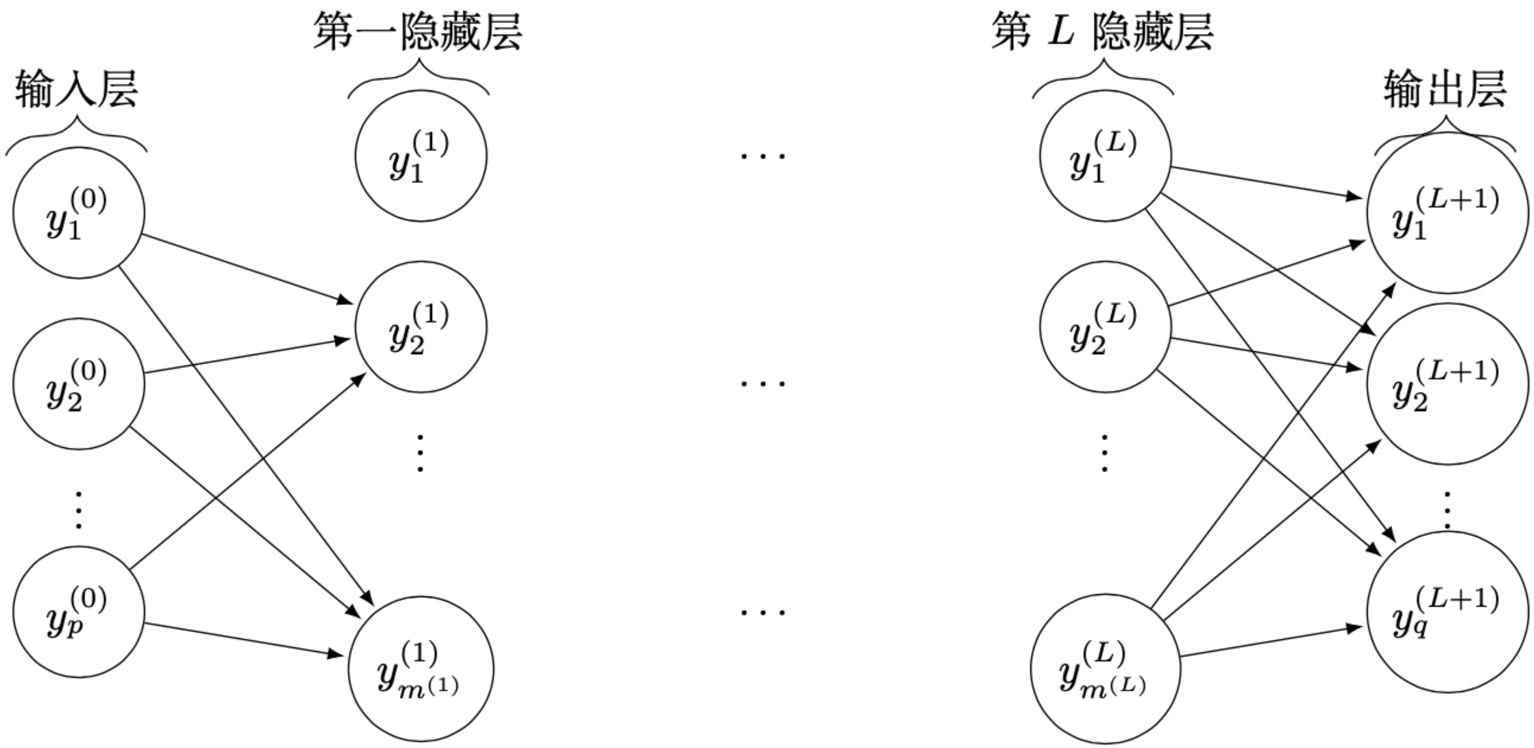
$$r_j(x) = y_j - \phi(t_j; x), \quad j = 1,\ldots,m$$

- 这里$n = 6$, m很大(比如10^5). 它说明即使变量数很小，计算目标函数也可能很昂贵.
- 数据拟合、参数估计、回归分析等许多问题中均涉及此类优化问题.有专用的算法求解.

$$\min_{x\in\mathbf{R}^6} \|\boldsymbol{r}(\boldsymbol{x})\|_2$$ 非线性最小二乘问题

$$\min_{x\in\mathbf{R}^6} \|\boldsymbol{r}(\boldsymbol{x})\|_1$$ 或者 $$\min_{x\in\mathbf{R}^6} \|\boldsymbol{r}(\boldsymbol{x})\|_\infty$$

优化实例4：深度学习（DNN）



$$\min_{x} \quad \sum_{i=1}^{m} \|h(a_i;x) - b_i\|_2^2 + \lambda r(x)$$

三、优化问题分类

数学规划问题的分类

连续优化与离散优化 (continuous vs integer)

- 某些或全部变量取整数值才有意义--整数规划 (IP). (田忌赛马)
- 分为整数线性规划和整数非线性规划；整数规划和混合整数规划；一般整数规划和**0-1**整数规划
- 整数规划属**NP**难问题. 常用算法：分支定界法、或其他启发式算法(求解一系列连续优化问题)

约束优化与无约束优化 (constrained vs unconstrained)

$$
\begin{aligned}
\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) \\
\text{subject to} \quad & \boxed{\begin{array}{ll} c_i(x) = 0, & i \in \mathcal{E} \\ c_i(x) \leq 0, & i \in \mathcal{I} \end{array}}
\end{aligned}
$$

$$
\begin{aligned}
\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) \\
\text{subject to} \quad & \boxed{x \in \Omega}
\end{aligned}
$$

restrictions on variables

$$
\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x)
$$

variables are free

- constrained opt. is generally difficult than unconstrained opt.
- e.g. regression problems, DNN

随机优化与确定性优化 (stochastic vs deterministic)

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\ & c_i(x) \le 0, \quad i \in \mathcal{I} \end{aligned}$$

- Stochastic: time-consuming or even impossible to evaluate the function or the gradient
- random approximation: $g(x;\xi) \approx \nabla f(x)$, ξ represents a random sampling
- Very popular in big data scenarios: $f(x) = \sum_{i=1}^{N} f_i(x) := f(x; a_i)$

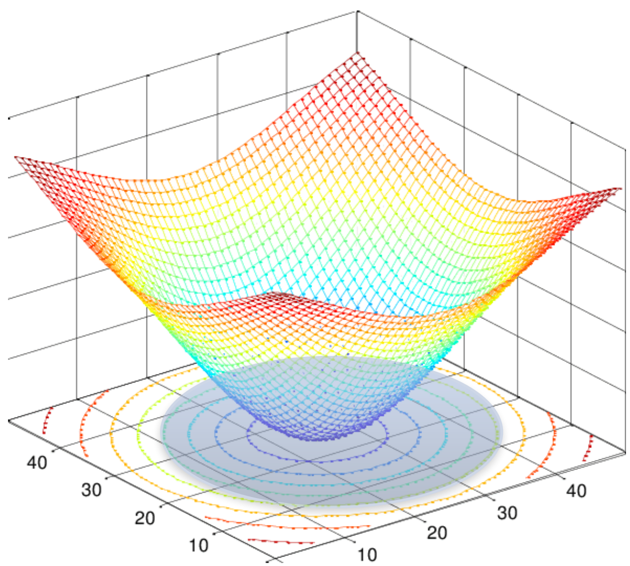
线性规划与非线性规划 (linear vs nonlinear)

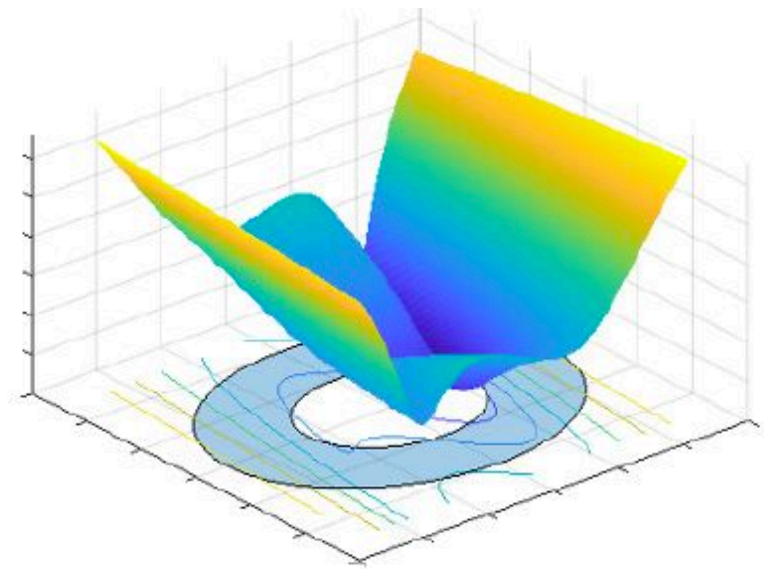
$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\ & c_i(x) \leq 0, \quad i \in \mathcal{I} \end{aligned}$$

- Linear if f and c_is are all linear functions
- Linear optimization is generally way easier than nonlinear

凸优化与非凸优化 (convex vs nonconvex)

$$
\begin{aligned}
\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f(x) \\
\text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\
& c_i(x) \le 0, \quad i \in \mathcal{I}
\end{aligned}
$$





- Convex opt if f and c_is are all convex
- Convex optimization is generally way easier than nonconvex

多目标(multi-objective optimization)与单目标

$$\min_{x \in X} (f_1(x), f_2(x), \ldots, f_k(x))$$

双层优化(bilevel optimization)与单层优化

$$\min_{x \in X, y \in Y} \quad F(x, y)$$

subject to:

$$G_i(x, y) \leq 0, \text{ for } i \in \{1, 2, \ldots, I\}$$
$$y \in \arg\min_{z \in Y} \{f(x, z) : g_j(x, z) \leq 0, j \in \{1, 2, \ldots, J\}\}$$

全局优化(global optimization)与局部优化

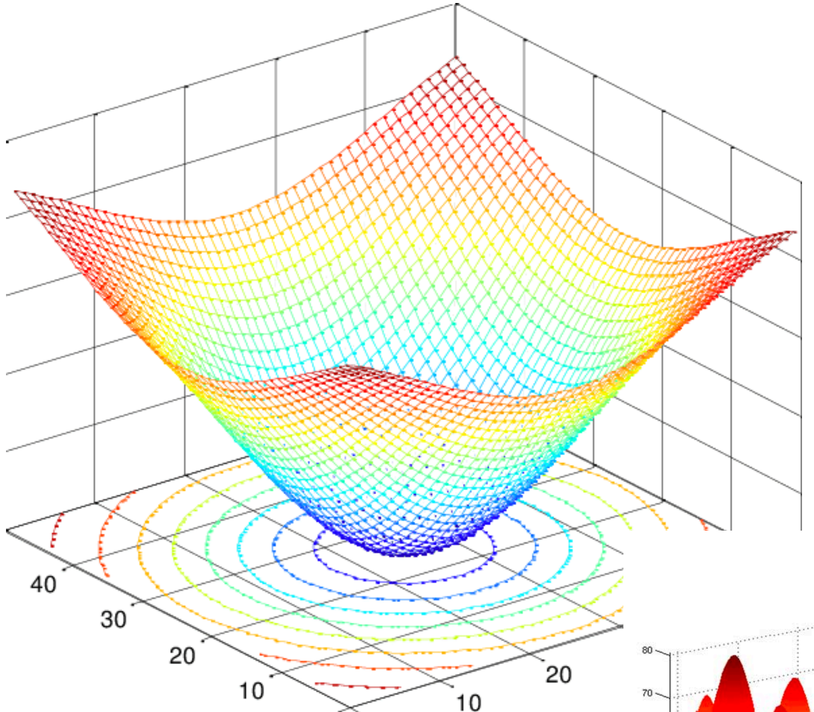
$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & x \in \mathcal{X}, \end{aligned} \tag{1.1.1}$$

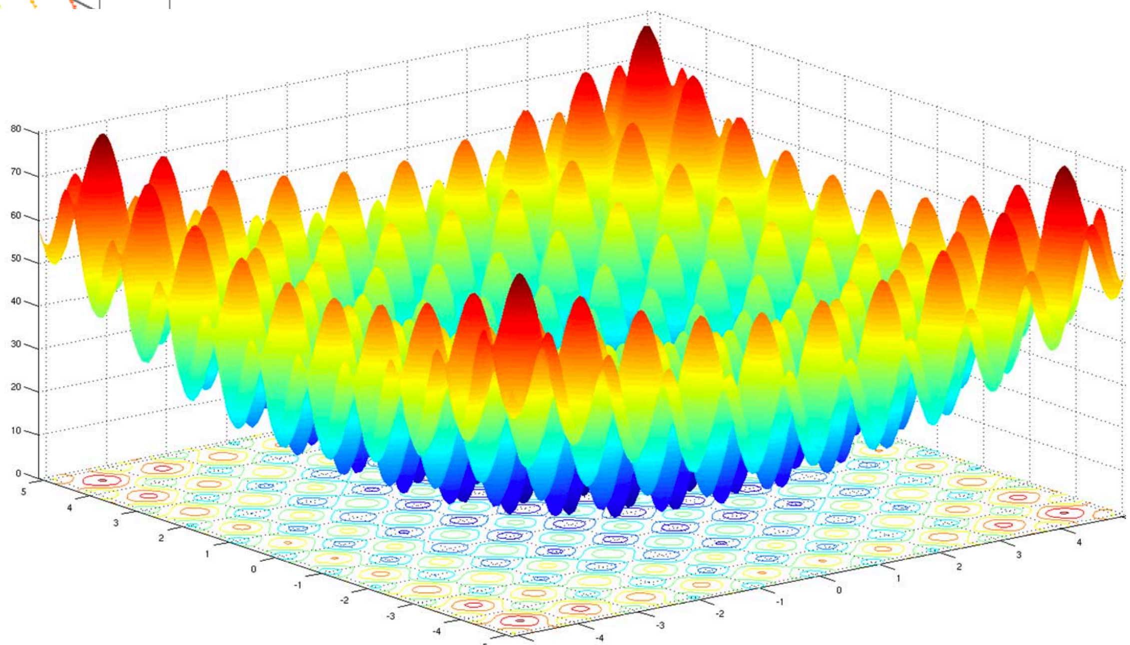
定义 1.1 (最优解) 对于可行点 $\bar{x}$ (即 $\bar{x} \in \mathcal{X}$)，定义如下概念：

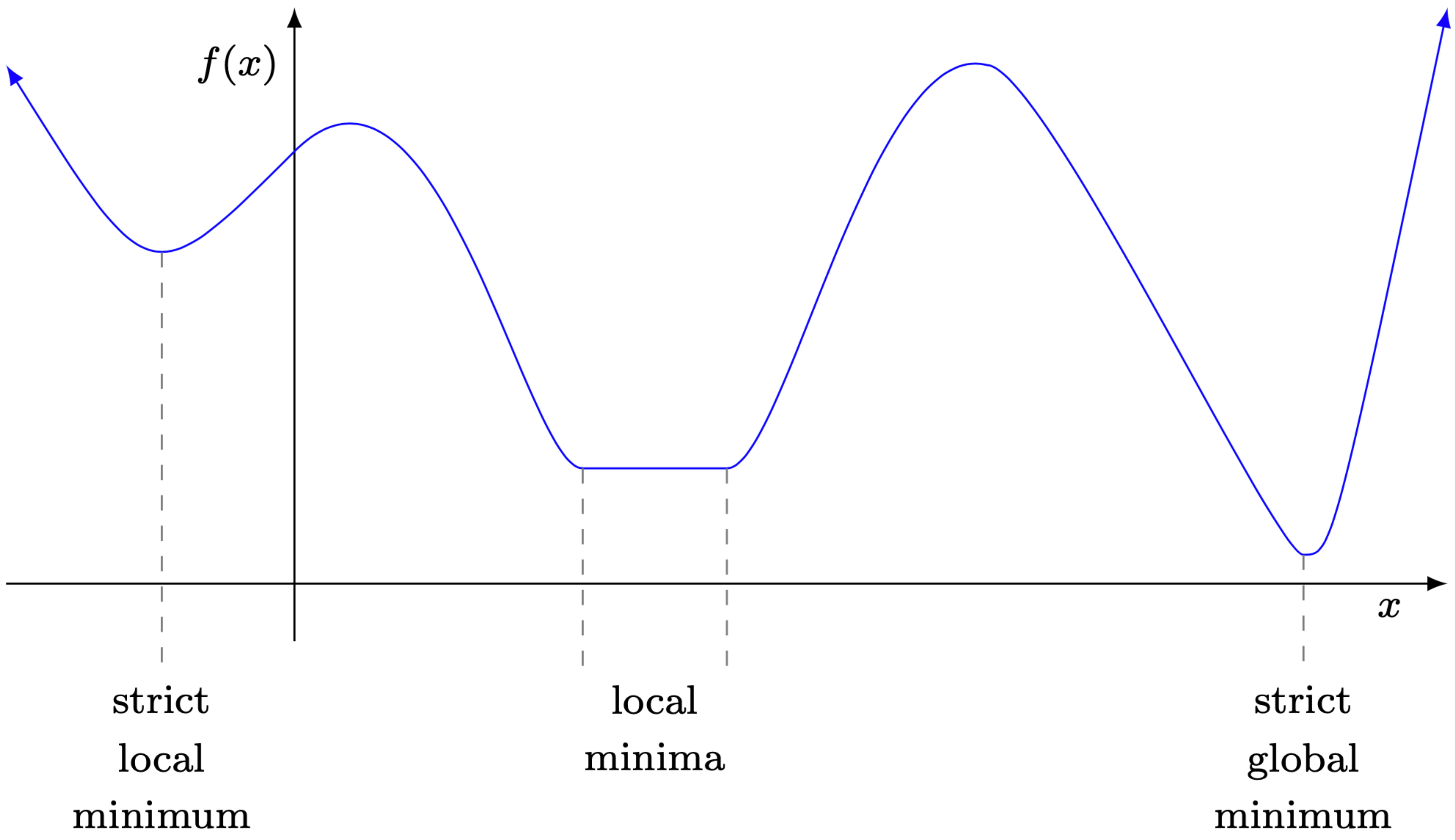
(1) 如果 $f(\bar{x}) \leqslant f(x), \forall x \in \mathcal{X}$，那么称 $\bar{x}$ 为问题 (1.1.1) 的**全局极小解（点）**，有时也称为（全局）最优解或最小值点；

(2) 如果存在 $\bar{x}$ 的一个 ε 邻域 $N_\varepsilon(\bar{x})$ 使得 $f(\bar{x}) \leqslant f(x), \forall x \in N_\varepsilon(\bar{x}) \cap \mathcal{X}$，那么称 $\bar{x}$ 为问题 (1.1.1) 的**局部极小解（点）**，有时也称为局部最优解；

(3) 进一步地，如果有 $f(\bar{x}) < f(x), \forall x \in N_\varepsilon(\bar{x}) \cap \mathcal{X}$，$x \neq \bar{x}$ 成立，则称 $\bar{x}$ 为问题 (1.1.1) 的**严格局部极小解（点）**.


40
30
20
10
20
10




$f(x)$
x
strict
local
minimum
local
minima
strict
global
minimum

最优化问题种类

Linear Programming
Nonlinear Programming
Integer Programming
Stochastic Optimization
Dynamic Programming
Conic Optimization
Polynomial Optimization
Sparse Optimization
Large-scale Optimization
Distributed Optimization
Derivative-Free Optimization
Multi-objective Optimization
Bi-level Optimization
Robust Optimization
Parallel Computing
Sensitivity Analysis
…
…

这门课的主要内容

先修课程：线性代数，多变量微积分，一种高级编程语言

四、Solvers（优化求解器简介）

优化算法和优化软件

- 迭代法
- 从最优解的某个初始猜测出发，产生一个依次提高的估计序列，得到精确解或者逼近解.
- 大部分利用目标函数和约束，可能还有这些函数的一阶和二阶导数.
- 通常收敛到

目标函数的驻点(无约束问题)或者

KKT点(约束问题的极大点、极小点或鞍点).

如果问题是凸规划，则可确保算法收敛到全局极小点.

Convergent sequences

Recall the following definition.

(Convergent sequence)

A sequence $\{x_k\}$ is said to converge to x_* if

$$\lim_{k \to \infty} \|x_k - x_*\|_2 = 0.$$

The following theorem is also extremely useful when analyzing algorithms.

(Bolzano-Weierstrauss Theorem)

Every bounded sequence in $\mathbb{R}^n$ *has a convergent subsequence.*

- Naturally, when applying Newton's method to solve $F(x) = 0$, we want $\{x_k\}$ (or a subsequence of it) to converge to x_* satisfying $F(x_*) = 0$.
- However, we have seen that the iterate sequence generated by Newton's method may not converge, and it may be unbounded.

Global convergence of an algorithm

(Globally convergent algorithm)

If, from any initial iterate x_0 (and under presumed conditions), the iterate sequence $\{x_k\}$ generated by an algorithm for solving a problem converges to a solution of the problem, then the algorithm is said to be globally convergent (under the presumed conditions).

- Note that, in the context of optimization, this definition does not presume or require convergence to a globally optimal solution. (Even optimization experts are often confused by this!)
- This definition is not saying that a given iterate sequence is "globally" convergent. After all, if such an algorithm produces a sequence $\{x_k\}$, the sequence itself is simply convergent.
- Rather, this definition refers to a property of an algorithm and the fact that it produces a convergent sequence from any starting point.
- We may also say that an algorithm is globally convergent if, e.g., a stationarity measure vanishes, even if the iterate sequence itself does not converge; e.g., if $\nabla f(x_k) \to 0$.
- Newton's method is not globally convergent in general.

Local convergence of an algorithm

(Locally convergent algorithm)

If, from any initial iterate x_0 in a neighborhood of a solution x_* (and under presumed conditions), the iterate sequence $\{x_k\}$ generated by an algorithm converges to x_*, then the algorithm is said to be locally convergent to x_* (under the presumed conditions).

- A locally convergent algorithm is not necessarily globally convergent.
- A globally convergent algorithm is locally convergent.
- However, an important characterization of local convergence is the corresponding rate of convergence. This distinguishes "how quickly" the iterates converge; e.g., 0.99999999999999^k converges to 0, but I don't want to have to wait for it to get very close to 0...
- (One can also talk about rates of global convergence, as is commonly done in convex optimization, but this will not be a focus in this course.)
- Newton's method is locally convergent (under some assumptions).

Convergence rates

Suppose $\|x_k - x_*\|_2 \to 0$; i.e., the sequence $\{x_k\}$ converges to x_*.

(Q-linear convergence)

If there exists a constant $c \in [0, 1)$ and $\hat{k} \geq 0$ such that

$$\|x_{k+1} - x_*\|_2 \leq c\|x_k - x_*\|_2 \text{ for all } k \geq \hat{k},$$

then $\{x_k\}$ converges Q-linearly to x_*.

Convergence rates

Suppose $\|x_k - x_*\|_2 \to 0$; i.e., the sequence $\{x_k\}$ converges to x_*.

(Q-superlinear convergence)

If there exists a sequence $\{c_k\} \to 0$ such that

$$\|x_{k+1} - x_*\|_2 \leq c_k \|x_k - x_*\|_2,$$

then $\{x_k\}$ converges Q-superlinearly to x_*.

(Q-quadratic convergence)

If there exists a constant $c \geq 0$ and $\hat{k} \geq 0$ such that

$$\|x_{k+1} - x_*\|_2 \leq c\|x_k - x_*\|_2^2 \text{ for all } k \geq \hat{k},$$

then $\{x_k\}$ converges Q-quadratically to x_*.

There is also "R-" convergence, but we'll skip that; so we'll drop the "Q-".

Examples

The constants c (and $\{c_k\}$) can make a difference, but generally, linear convergence is "slow" and superlinear/quadratic convergence is eventually very "fast".

- $\{1 + 2^{-k}\} = \{2, \frac{3}{2}, \frac{5}{4}, \frac{9}{8}, \ldots\}$ converges linearly to $x_* = 1$.
  Proof:

- $\{1 + 2^{-2^k}\} = \{\frac{3}{2}, \frac{5}{4}, \frac{17}{16}, \frac{257}{256}, \ldots\}$ converges quadratically to $x_* = 1$.
  Proof:

Arithmetic Convergence

Linear convergence is also called geometric convergence. There is another (slower) type of convergence:

Definition. If the sequence $\{r_k\}$ converges to r^* in such a way that

$$|r_k - r^*| \le C\frac{|r_0 - r^*|}{k^p},\ k \ge 1,\ 0 < p < \infty$$

where C is a fixed positive number, the sequence is said to converge *arithmetically* to r^* with order p.

When $p = 1$, it is referred as arithmetic convergence. The greater of p the faster of the convergence.

Example 4. The sequence $r_k = 1/k$ converges to zero arithmetically. The convergence is of order one but it is not linear, since $\lim\limits_{k\to\infty}(r_{k+1}/r_k) = 1$, that is, β is not strictly less than one.

相关的语言和软件

- 编程语言：Matlab，Python
- 最优化求解工具：excel，matlab，python，cplex，lingo，lindo
- 最前沿的最优化求解器：NEOS optimization solvers
- 模型语言：AMPL，GAMS，AIMMS，lingo，lindo

NEOS optimization solvers：https://neos-server.org/neos/solvers/index.html

NEOS optimization solvers： https://neos-server.org/neos/solvers/index.html

NEOS   Contact   Help

Listed below are the available solvers organized by Problem Type. An addit

If you need help in selecting a solver, consult the Optimization Tree of the N
problem.

Each solver has sample problems and background information on the solve
you encounter problems, consult the NEOS Server FAQ, or contact us by c

Problem Type   Solver

- Job Queue Tools
- Application
- Bound Constrained Optimization
- Combinatorial Optimization and Integer Programming
- Complementarity Problems
- Extended Mathematical Programming
- Global Optimization
- Linear Network Programming
- Linear Programming
- Mathematical Programs with Equilibrium Constraints
- Mixed Integer Linear Programming
- Mixed Integer Nonlinearly Constrained Optimization
- Mixed-Integer Optimal Control Problems

Problem Type   Solver

Job Queue Tools

- View Job Queue
- View Job Results / Kill a Job

Application

- CONVERT [GAMS Input]
- Domino [jpeg Input]
- ECM [csv Input][single_text Input][zip Input]
- Fishwerks [csv Input]

Bound Constrained Optimization

- L-BFGS-B [AMPL Input]

Combinatorial Optimization and Integer Programming

- BiqMac [SPARSE Input]
- concorde [TSP Input]

Complementarity Problems

- Knitro [AMPL Input]
- MILES [GAMS Input]
- NLPEC [GAMS Input]
- PATH [AMPL Input][GAMS Input]

Extended Mathematical Programming

- DE [GAMS Input]
- JAMS [GAMS Input]

Global Optimization

- ANTIGONE [GAMS Input]
- ASA [AMPL Input]
- BARON [AMPL Input][GAMS Input]
- Couenne [AMPL Input][GAMS Input]
- icos [AMPL Input]
- LGO [AMPL Input]
- LINDOGlobal [GAMS Input]
- PGAPack [AMPL Input]
- PSwarm [AMPL Input]
- RAPOSa [AMPL Input]
- scip [AMPL Input][CPLEX Input][GAMS Input][MPS Input][OSIL Input][Python Input][ZIMPL Input]

需要掌握

- 利用编程语言（matlab、python）实现一些最基本的算法
- 了解并且懂得调用一些常见软件中的优化工具包：excel，matlab，python，ampl
- 了解现存的模型语言：AMPL，AIMMS，GAMS等。并会用AMPL建模简单的最优化问题和求解
- 能解读常用软件的最优化求解报告：求解信息、输出信息、模型解释