数学建模

浙江大学数学科学学院　谈之奕

tanzy@zju.edu.cn

运筹与统计

组合优化

经典组合优化问题

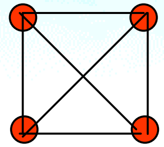
数学建模

- 背包（**Knapsack**）
- 指派（**Assignment Problem**）
- 调度
  - 经典排序问题（**Scheduling**）
  - 资源分配（**Resource Allocation**）
  - 项目排序（**Project Scheduling**）
  - 时间表（**Timetabling**）
  - 轮值表（**rostering**）
- 布局型问题
  - 装箱问题（**Bin-packing**）
  - 装填问题
  - 填充问题
- 图中优化问题
  - 特殊顶点集（独立集、顶点覆盖、支配集）
  - 特殊边集（匹配、割）
  - 特殊子图（最小生成树、最小**Steiner**树）
  - 着色（**Coloring** ）
- 路由问题
  - 最短路
  - **TSP**
  - **VRP**
  - 边路由（**Arc Routing**）
- 选址（**Facility Location**）
- 网络流（**Network Flow**）

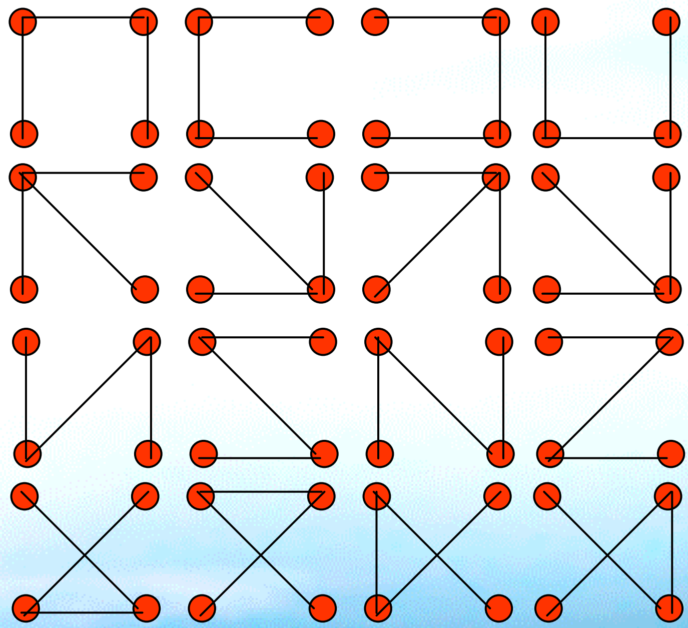
最小生成树

- 在某地区内修建连接若干城镇的公路系统，假设公路造价与长度成正比，如何设计造价最低的建造方案
- 最小生成树（**minimum spanning tree，MST**）
  - 连通的无圈图称为树（**tree**）
  - 树 T 称为图 G 的生成树（**spanning tree**），若 T 是 G 的子图、且和 G 有相同的顶点集
  - 赋权图 G 的所有生成树中总权和最少的生成树称为最小生成树

最小生成树



K_4



- 将城镇视作图的顶点，城镇之间的距离视作连接两个顶点的边的长度。生成树可以把所有城镇都连接起来；最小生成树具有最小的总长度
- 完全图 K_n 有 n^{n-2} 颗不同的生成树

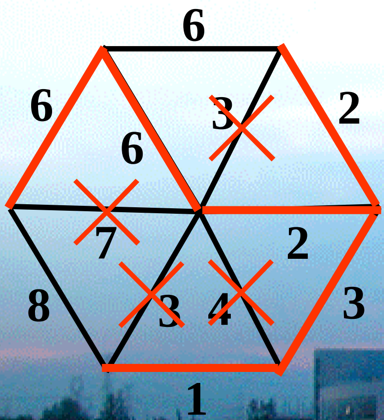
最小生成树

浙江大学
ZheJiang University
数学建模

- **Kruskal 算法**
  - 将连通图 $G=(V,E)$ 的所有边按权非降顺序排列
  - $F=\varnothing$ ， $j=1$
  - 若图 $T=(V,F\cup\{e_j\})$ 不含圈，则令 $F=F\cup\{e_j\}$
  - 若 $|F|=|V|-1$ ，终止，$T=(V,F)$ 即为最小生成树。否则，返回上一步

Otakar Borůvka
（1899－1995）
捷克数学家

Joseph Bernard Kruskal
(1928 –2010)
美国数学家、计算机科学家



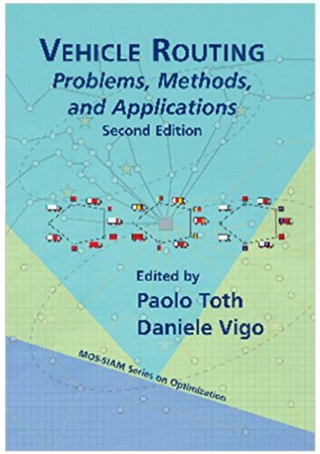
Borůka O, O jistém problému minimálním (About a certain minimal problem), *Práce mor. Přírodověd. spol. v Brně III*, 3, 37-58, 1926

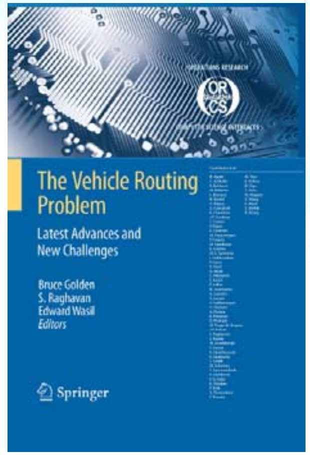
Kruskal JB. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7, 48-50, 1956

车辆路径问题

数学建模

- 车辆路径问题（**Vehicle Routing Problem, VRP**）
  - n个顾客，顾客 i 位于地点 i，需求为 $q_i, i=0,1,\cdots,n$
  - K 辆相同的车，容量均为Q
  - 仓库位于地点 0。从地点 i 到地点 j 的距离为 $c_{ij}, i, j=0,1,\cdots,n$
  - 将顾客分配给各车，每车配送的顾客需求之和不超过 Q，确定每辆车自仓库出发，完成分配给该车的所有顾客的配送，最后回到仓库的路线，使得所有车行驶的总路程最短





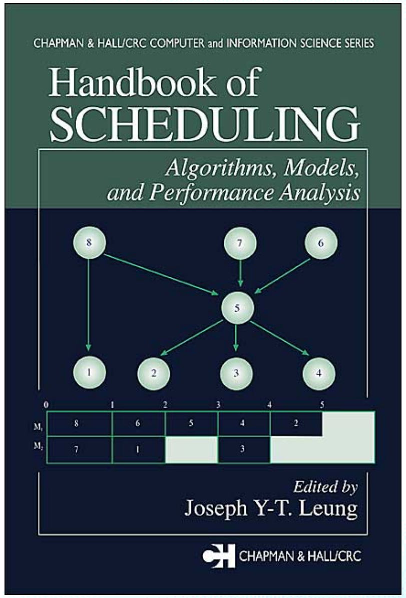
Toth P, Vigo D (Eds.) ***Vehicle Routing: Problems, Methods, and Applications*****. SIAM, 2014.**
Golden BL, Subramanian R, Edward AW(Eds.) ***The Vehicle Routing Problem: Latest Advances and New Challenges*****. Springer, 2008.**

排序问题

浙江大学 ZheJiang University

数学建模

- 排序（**scheduling**）主要研究如何利用有限资源，在给定的限制条件下，将一批任务安排在某些时间段内完成，并使效益最大
  - 早期研究的排序问题背景源自工业生产，习惯上把可用的资源称为机器（**machine**），需要完成的任务称为工件（**job**）
  - 对部分排序问题，可行解由工件加工的顺序决定，这类问题最早也被称作**sequencing**
  - 排序是组合优化中模型最丰富、应用最广泛的问题之一

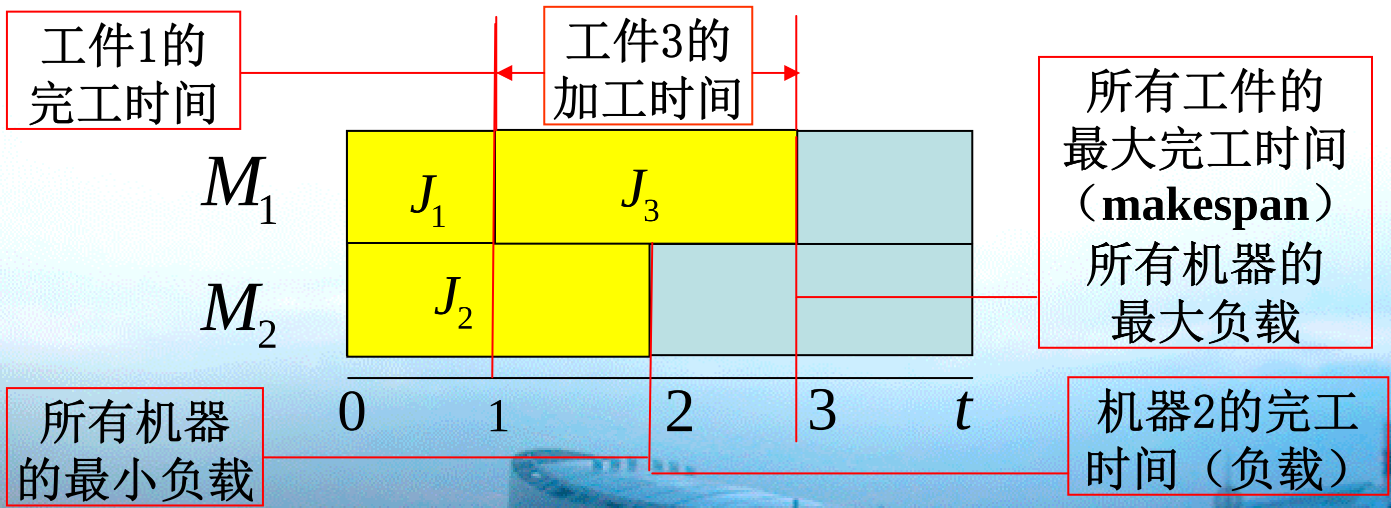


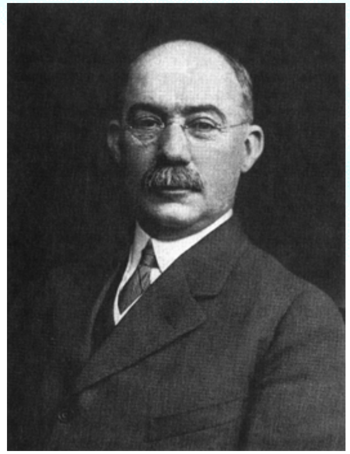
Journal of Scheduling

Leung, JY. (Eds.) *Handbook of scheduling: algorithms, models, and performance analysis*. CRC, 2004

排序问题

- 根据工件加工的要求和机器性质的不同，可以将经典排序问题分为单台机、平行机、车间作业等类型
- **Gantt 图**是描述排序问题及其可行解的常用工具





Henry Gantt
(1861-1919)
美国管理学家

单台机问题

浙江大学 ZheJiang University

数学建模

- 现有 n 个工件，工件 j 的加工时间为p_j，预定交工期（duedate）为 d_j，$j=1,\cdots,n$ 。机器在同一时刻只能加工一个工件，工件加工不可中断。如何确定工件的加工顺序，可使得工件尽量按时完工
  - 误工（完工时间大于预定交工期）的工件数最少 $\mathcal{P}$ 问题
  - 各工件延误时间（完工时间与预定交工期之差）的最大值最小

    EDD规则：将工件按预定交工期的非减序排列
  - 所有误工工件的总延误时间最小 $\mathcal{NP}$ －难问题

Moore JM. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, 15, 102-109, 1968

Du J, Leung JYT. Minimizing total tardiness on one machine is NP-hard. *Mathematics of Operations Research*, 15, 483-495, 1990.

Santa Claus问题

数学建模

- 圣诞老人欲将n件礼物分给m位小朋友。一件礼物只能分给一位小朋友，但一位小朋友可以拿到多件礼物。第i位小朋友拿到礼物j时他的满意度为p_{ij}，$i=1,\cdots,m$，$j=1,\cdots,n$，一位小朋友拿到多件礼物时的满意度为各礼物相应满意度之和。希望给出一礼物分配方案，使满意度最小的小朋友的满意度尽可能大

礼　物——工　件　　　满意度 ——— 加工时间

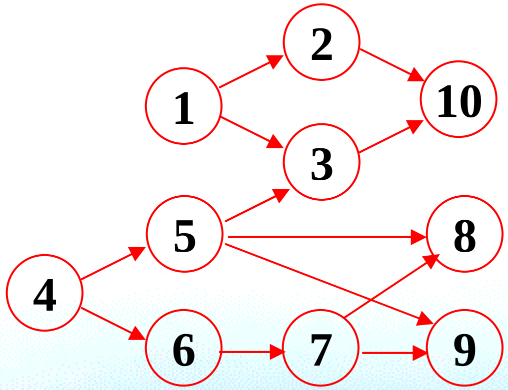
小朋友——机　器　　小朋友的满意度—机器完工时间

目标函数———极大化最小机器完工时间

Bansal, N., Sviridenko, M., The Santa Claus problem，*Proceedings of the 38th Annual ACM Symposium on Theory of computing*, 31-40, 2006

排序悖论

工件	1	2	3	4	5	6	7	8	9	10
加工时间	8	7	7	2	3	2	2	8	8	15
	7	6	6	1	2	1	1	7	7	14



目标：尽早完工

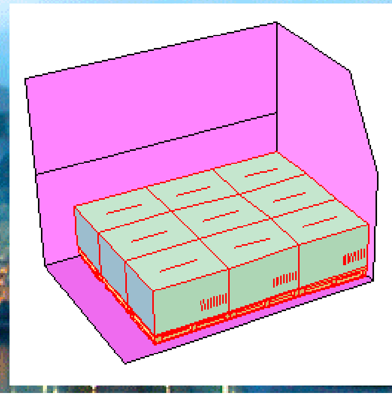
贪婪思想：可以加工的工件尽早开工

装箱问题

浙江大学 ZheJiang University

数学建模

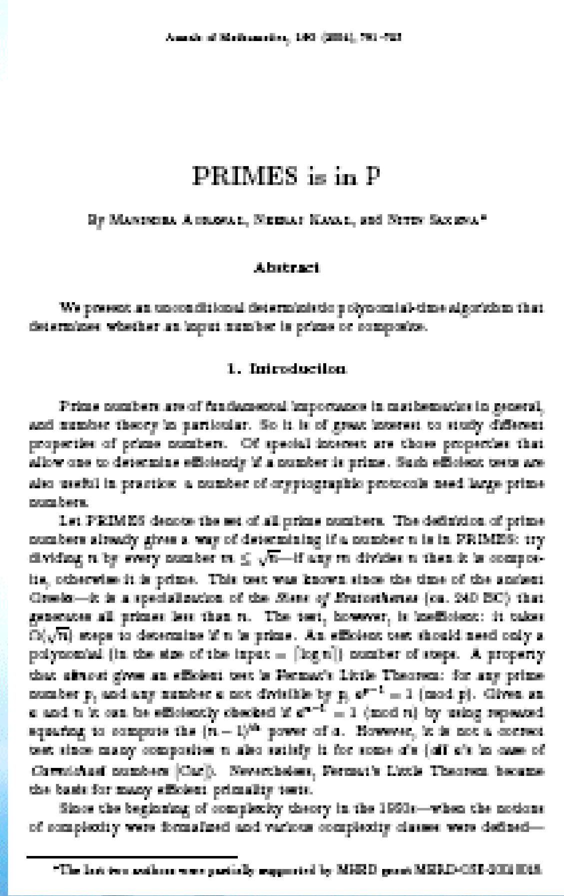
- 装箱问题（**bin-packing problem**）：将一系列物品放入容量一定的若干箱子中，放入每个箱子中的物品大小之和不超过箱子容量，目标为所用箱子数尽可能少
  - 下料问题（**cutting-stock problem**）：给定生产一批产品所需的某种材料的大小与数量列表，如何从一定规格的原料中下料，使所用的原料最少
  - 一维装箱：箱子容量和物品大小均为有理数
  - 二维装箱：箱子和物品均为长、宽一定的矩形
  - 二维条状装箱（**strip packing**）：将若干长、宽一定的矩形物品无重叠地放入长无限、宽固定的矩形区域，目标为物品占用长度最小
  - 三维装箱：箱子和物品均为长、宽、高一定的长方体
  - 物品可以/不可以旋转、倒置，考虑/不考虑稳定性等



$\mathcal{P}$ 类

- $\mathcal{P}$ 类
  - 若一问题已找到多项式时间算法，称这样的问题属于多项式时间可解问题类（**polynomial solvable problem class**），记为 $\mathcal{P}$
  - 证明一问题属于$\mathcal{P}$ 类只需设计出求解该问题的多项式时间算法
  - 最小生成树、指派问题均为$\mathcal{P}$ 类中的问题；判断一个整数是否为质数也属于$\mathcal{P}$ 类

Agrawal, M., Kayal, N., Saxena, N., PRIMES is in P. *Annals of Mathematics*, 160, 2, 781-793, 2004



Annals of Mathematics, 160 (2004), 781–793

PRIMES is in P

By Manindra Agrawal, Neeraj Kayal, and Nitin Saxena*

Abstract

We present an unconditional deterministic polynomial-time algorithm that determines whether an input number is prime or composite.

1. Introduction

Prime numbers are of fundamental importance in mathematics in general, and number theory in particular. So it is of great interest to study different properties of prime numbers. Of special interest are those properties that allow one to determine efficiently if a number is prime. Such efficient tests are also useful in practice: a number of cryptographic protocols need large prime numbers.

Let PRIMES denote the set of all prime numbers. The definition of prime numbers already gives a way of determining if a number n is in PRIMES: try dividing n by every number $m \leq \sqrt{n}$—if any m divides n then it is composite, otherwise it is prime. This test was known since the time of the ancient Greeks—it is a specialization of the *Sieve of Eratosthenes* (ca. 240 BC) that generates all primes less than n. The test, however, is inefficient: it takes $\Omega(\sqrt{n})$ steps to determine if n is prime. An efficient test should need only a polynomial (in the size of the input $= \lceil \log n \rceil$) number of steps. A property that almost gives an efficient test is Fermat's Little Theorem: for any prime number p, and any number a not divisible by p, $a^{p-1} = 1 \pmod{p}$. Given an a and n it can be efficiently checked if $a^{n-1} = 1 \pmod{n}$ by using repeated squaring to compute the $(n-1)^{th}$ power of a. However, it is not a correct test since many composites n also satisfy it for some a's (all a's in case of Carmichael numbers [Car]). Nevertheless, Fermat's Little Theorem became the basis for many efficient primality tests.

Since the beginning of complexity theory in the 1960s—when the notions of complexity were formalized and various complexity classes were defined—

*The last two authors were partially supported by MHRD grant MHRD-CSE-20010018.

素性测试

Eratosthenes筛法（Sieve of Eratosthenes）

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Prime numbers

Eratosthenes of Cyrene
（约公元前**276-**
约公元前**194**）
古希腊科学家

14

素性测试

数学建模

- 素性测试问题：给定整数 n，判断 n 是否为素数
  - 实例规模为 $\log_2 n$，**Eratosthenes**筛法是指数时间算法
  - **AKS素性测试**算法可在 $O(\log_2^{7.5} n \cdot \text{poly}(\log\log n))$ 时间内完成

Manindra Agrawal：印度理工学院坎普尔学院（**Indian Institute of Technology Kanpur**）计算机科学与工程系教授

Neeraj Kayal与**Nitin Saxena2002**年时均为该系本科生，**"Towards a Deterministic Polynomial-Time Primality Test"**是他们的一项本科生科研项目

EATCS与ACM学会Godel Prize（2006）
outstanding papers in the area of theoretical computer science

MOS与AMS学会Fulkerson Prize（2006）
outstanding papers in the area of discrete mathematics

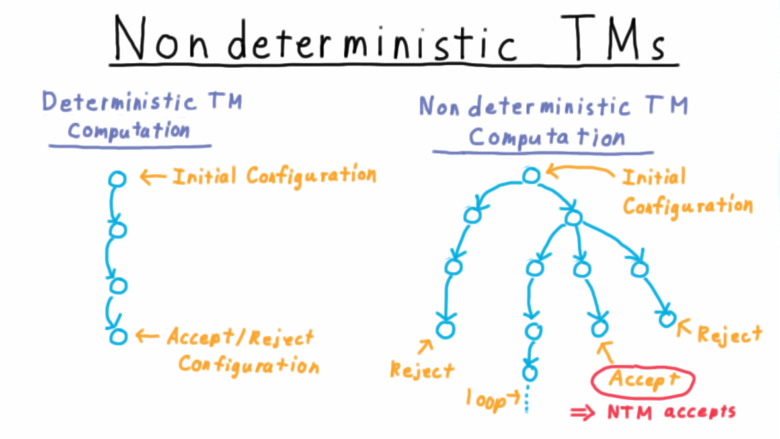
15

$\mathcal{NP}$ 类

浙江大学 ZheJiang University

数学建模

- 非确定性图灵机多项式时间可解问题类（**nondeterministic polynomial solvable problem class**）称为$\mathcal{NP}$ 类，简记为$\mathcal{NP}$
  - 确定性图灵机的状态转移函数是单值的，非确定性图灵机的状态转移函数可能是多值的
  - 确定性图灵机是一种特殊的非确定性图灵机 $\mathcal{P} \subseteq \mathcal{NP}$
  - 任一非确定性图灵机可用一确定性图灵机进行模拟，完成同样计算，但需花费更多时间 $\mathcal{P} = \mathcal{NP}$?



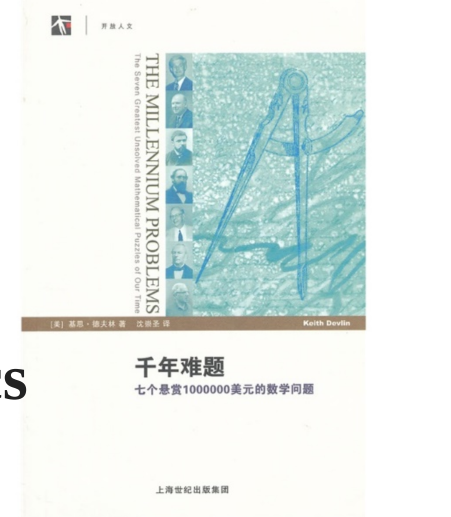
Brubaker C, Fortnow L. Computability, Complexity & Algorithms, Online Course(CS 6505), Georgia Institute of Technology

16

千年难题

- **Millennium Problems**
  - **Yang–Mills and Mass Gap**
  - **Riemann Hypothesis**
  - **P vs NP Problem**
  - **Navier–Stokes Equation**
  - **Hodge Conjecture**
  - **Poincaré Conjecture**
  - **Birch and Swinnerton-Dyer Conjecture**

Clay Mathematics Institute（CWI）



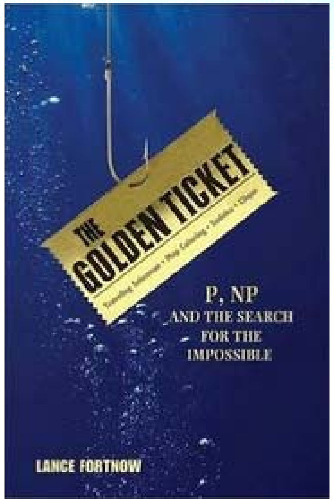
Devlin, K. J., *The Millennium Problems: The Seven Greatest Unsolved Mathematical Puzzles of Our Time*, Basic Books , 2003.（中译本：千年难题——七个悬赏1000000美元的数学问题，沈崇圣译，上海科技教育出版社，2012）

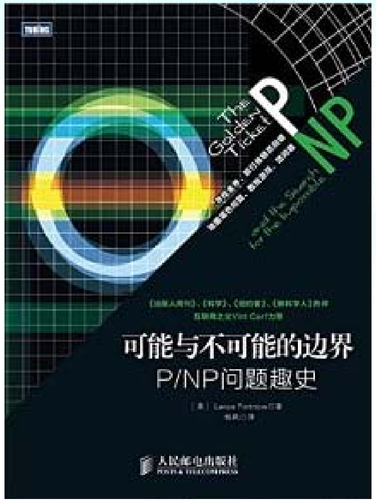
$\mathcal{P}=\mathcal{NP}$ 猜想

数学建模

- 尽管 $\mathcal{P}=\mathcal{NP}$ 猜想是未决问题，但是目前普遍相信 $\mathcal{P}\neq\mathcal{NP}$ 成立，并在此假设下进一步研究 $\mathcal{NP}$ 类内部的结构

	$\mathcal{P}\neq\mathcal{NP}$	$\mathcal{P}=\mathcal{NP}$	02-09	10-19	20-29
2002	61%	9%	5%	12%	13%
2012	83%	9%		1%	11%
2019	88%	12%			22%





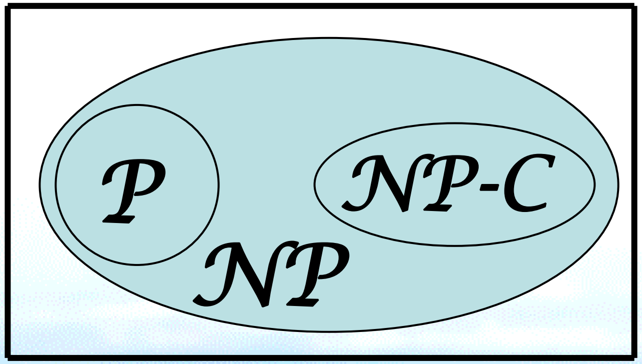
Gasarch WI . The Third P=? NP poll. *ACM SIGACT News*, 50(1):38-59, 2019.

Fortnow, L., *The Golden Ticket: P, NP, and the Search for the Impossible*, Princeton University Press , 2013.（中译本：可能与不可能的边界：P/NP问题趣史，杨帆译，人民邮电出版社，2014.

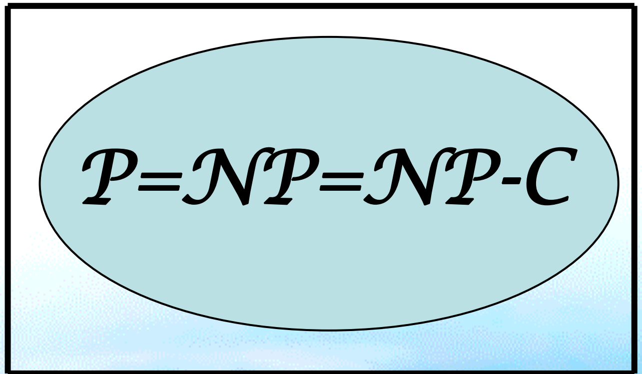
Fortnow L. The status of the P versus NP problem. *Communications of the ACM*, 2009, 52(9): 78-86.

$\mathcal{NP}$–完全问题

- $\mathcal{NP}$ 类中最“难”的问题子集称为$\mathcal{NP}$ －完全类，记为$\mathcal{NP}$-C。$\mathcal{NP}$-C 类中的问题称为$\mathcal{NP}$ －完全问题（$\mathcal{NP}$ -**complete problem**）
  - 若$\mathcal{NP}$-C 类中有一个问题有多项式时间算法，则$\mathcal{NP}$ 类中所有问题都有多项式时间算法



$\mathcal{P} \neq \mathcal{NP}\ (\mathcal{P} \cup \mathcal{NP}\text{-}C \neq \mathcal{NP})$



$\mathcal{P} = \mathcal{NP}$

P，NP，与NP—完全

TEACHER'S BOOK 普通高级中学实验教科书 数学3（必修）教师教学用书

4. 计算的复杂性

计算的复杂性测度函数有三类：一类是指数型的，常写为 c^n 的形式（c 是常数）；

的，常写为 n^k 的形式（k 为非负整数）；另一类是对数型的，常写为 $\log^k n$ 的形式．具

问题分别称为指数复杂性，多项式复杂性和对数复杂性．

人们习惯于把理论上可计算的问题类称为能行可计算的，而把具有多项式复杂性

效可计算的，通常称为 P 问题．NP 问题是指还未找到多项式复杂性算法的问题．

研究和实验表明，单纯靠提高计算机速度并不能解决 NP 问题．例如，根据某些

录，对于复杂性为 2^n 的问题，即使计算机速度提高 1 000 倍，也只能是多算约 10 道题

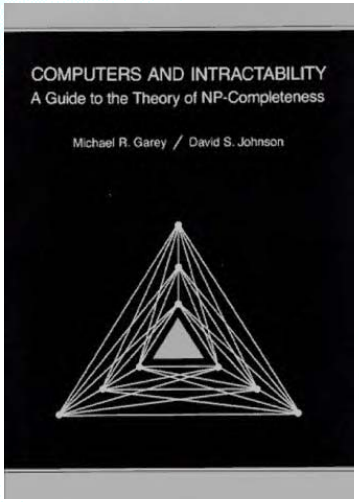
为，解决 NP 问题的关键是要从数学上找出好的算法．事实上，数学家们也找到了各种各样的好办法，

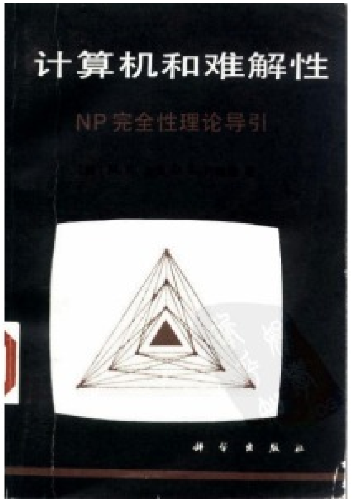
大大简化了计算．例如，用计算机对卫星照片进行处理，如果在一张 10 cm² 的照片上以一微米为间隙打上格子，则处理一张照片需要进行 10^{16} 次运算，即使用每秒百亿次的计算机也要连续算上十多个昼夜．

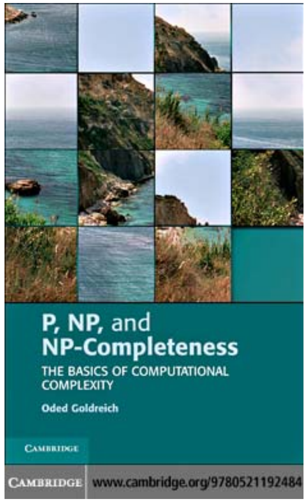
后来，有人发明了一种好的算法，大大降低了计算的复杂性，使得用同样的计算机计算只需要 $\frac{1}{3}$ 秒．

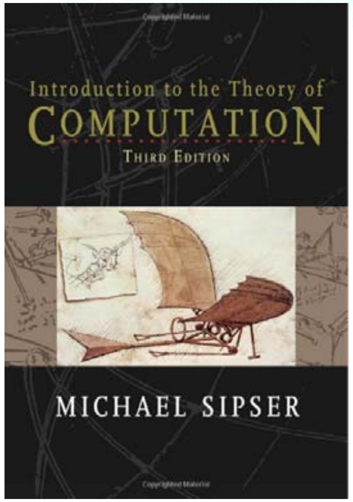
参考资料

浙江大学
ZheJiang University
数学建模









Garey MR, Johnson DS. ***Computers and intractability: a guide to the theory of NP-completeness***. **Freeman, 1979.** （中译本：计算机和难解性：NP 完全性理论导引. 张立昂，沈泓译，科学出版社，1990.）

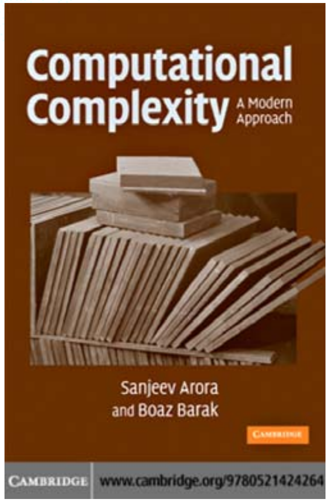
Goldreich O. ***P, NP, and NP-Completeness: The basics of Computational Complexity***. **Cambridge University Press, 2010.**

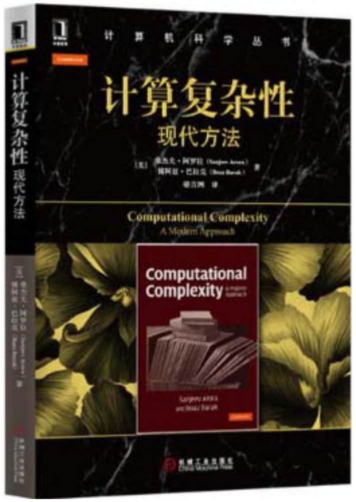
Sipser M. ***Introduction to the Theory of Computation*** **(3rd). Cengage Learning, 2012.** (中译本：计算理论导引. 段磊、唐常杰译，机械工业出版社，2015）

参考资料

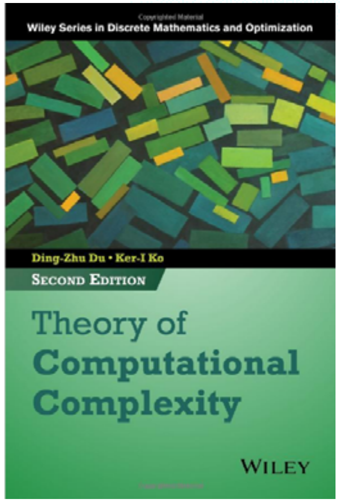
浙江大学
ZheJiang University

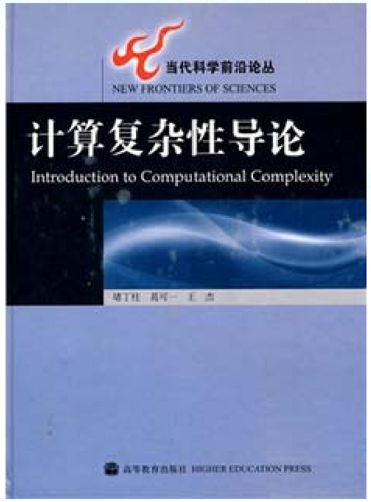
数学建模





Arora S, Barak B. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.（中译本：计算复杂性：现代方法. 骆吉洲译，机械工业出版社，2015.）





Du DZ, Ko KI. *Theory of Computational Complexity*. Wiley, 2011.
堵丁柱，葛可一，王杰，计算复杂性导论，高等教育出版社，**2002.**

浙江大学
ZheJiang University
数学建模

$\mathcal{NP}$ –难问题汇编

A compendium of NP optimization problems

Editors:
[Pierluigi Crescenzi], and [Viggo Kann]
Subeditors:
Magnús Halldórsson (retired)
Graph Theory: Covering and Partitioning, Subgraphs and Supergraphs, *Sets and Partitions*.
Marek Karpinski
Graph Theory: Vertex Ordering, *Network Design*: Cuts and Connectivity.
Gerhard Woeginger
Sequencing and Scheduling.

This is a continuously updated catalog of approximability results for NP optimization problems. The compendium is also a part of the book Complexity and Approximation. The compendium has not been updated for a while, so there might exist recent results that are not mentioned in the compendium. If you happen to notice such a missing result, please report it to us using the web forms.

You can use web forms to report new problems, new results on existing problems, updates of references or errors.

http://www.nada.kth.se/~viggo/problemlist/compendium.html

23

$\mathcal{NP}$ 优化问题汇编

MAXIMUM KNAPSACK

- INSTANCE: Finite set U, for each $u \in U$ a size $s(u) \in Z^+$ and a value $v(u) \in Z^+$, a positive integer $B \in Z^+$.
- SOLUTION: A subset $U' \subseteq U$ such that $\sum_{u \in U'} s(u) \leq B$.
- MEASURE: Total weight of the chosen elements, i.e., $\sum_{u \in U'} v(u)$.
- *Good News*: Admits an FPTAS [266].
- *Comment*: The special case when *s(u)=v(u)* for all $u \in U$ is called MAXIMUM SUBSET SUM. The corresponding minimization problem where $\sum_{u \in U'} s(u) \geq B$ also admits an FPTAS, as well as several other variations of the knapsack problem [191].
- *Garey and Johnson*: MP9

浙江大学 ZheJiang University

数学建模

指派问题

- 设有 n 项任务需分配给 n 位员工，每人完成其中一项，员工 i 完成任务 j 所需时间为 c_{ij}，如何分配可使完成所有任务所用总时间最少
  - 决策变量 $x_{ij}=\begin{cases}1, & \text{员工}\ i\ \text{被分配完成工作}\ j\\ 0, & \text{其他}\end{cases}\quad i,j=1,2,\cdots,n$
  - 整数规划

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$ 完成所有任务所用总时间

$$s.t.\ \sum_{j=1}^{n} x_{ij}=1,\ i=1,\cdots,n$$ 每位员工完成一项工作

$$\sum_{i=1}^{n} x_{ij}=1,\ j=1,\cdots,n$$ 每项工作由一位员工完成

$$x_{ij}=0\ \text{或}\ 1,\ i,\ j=1,\cdots,n$$

整数规划 $\in \mathcal{NP}\text{-}C$

指派问题 $\in \mathcal{P}$?

指派问题

- 记决策变量向量为 $\mathbf{x}=(x_{11},x_{12},\cdots,x_{1n},x_{21},\cdots,x_{2n},\cdots,x_{n1},\cdots,x_{nn})^T$ ，整数规划 $\min\{\mathbf{cx}\mid\mathbf{Ax}=\mathbf{b},x_{ij}\in\{0,1\}\}$ 的系数矩阵为

$$\mathbf{A}=\begin{pmatrix} 1\ 1\ \cdots\ 1 & 0\ 0\ \cdots\ 0 & & 0\ 0\ \cdots\ 0 \\ 0\ 0\ \cdots\ 0 & 1\ 1\ \cdots\ 1 & \cdots & 0\ 0\ \cdots\ 0 \\ \vdots & \vdots & & \vdots \\ 0\ 0\ \cdots\ 0 & 0\ 0\ \cdots\ 0 & & 1\ 1\ \cdots\ 1 \\ \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \end{pmatrix}$$

$$\begin{aligned} \min\ & \sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij} \\ s.t.\ & \sum_{j=1}^{n}x_{ij}=1,\ i=1,\cdots,n \\ & \sum_{i=1}^{n}x_{ij}=1,\ j=1,\cdots,n \\ & 0\le x_{ij}\le 1\quad i,j=1,\cdots,n \end{aligned}$$

- 若矩阵的所有子式均为**0**或 ±1，则称该矩阵为全幺模（**totally unimodular**）矩阵
- 系数矩阵为全幺模矩阵的整数规划的松弛线性规划的最优解必为整数解

TSP问题

数学建模

- 决策变量 $x_{ij}=\begin{cases}1, & \text{离开城市}\ i\ \text{后到达城市}\ j\\ 0, & \text{其他}\end{cases}$ $\quad i,j=1,2,\cdots,n$
- 整数规划

决策变量能表示环游，
同时能计算环游长度

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$s.t.\quad \sum_{j=1}^{n} x_{ij}=1,\quad i=1,\cdots,n$$ 离开城市 i 后到达另一个城市

$$\sum_{i=1}^{n} x_{ij}=1,\quad j=1,\cdots,n$$ 从一个城市来到城市 j

$$x_{ij}=0,1,\ i,j=1,\cdots,n$$

指派问题 $\in \mathcal{P}$

TSP $\in \mathcal{NP}\text{-}\mathcal{C}$?

问题可行解与其整数规划可行解之间未一一对应

TSP问题

- **TSP环游不允许出现经过城市数小于** n **的子环游**

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \cdots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, \cdots, n$$

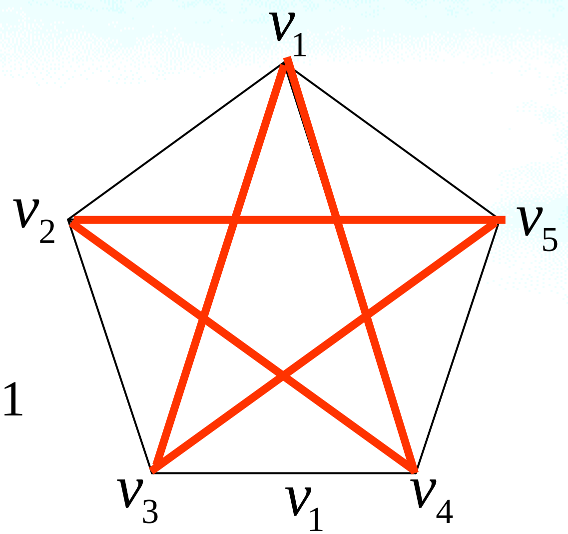
- 若存在一相继经过 k 个城市 $i_1, i_2, \cdots, i_k$ 的子环游，则

$$x_{i_1 i_2} = x_{i_2 i_3} = \cdots = x_{i_{k-1} i_k} = x_{i_k i_1} = 1$$

$$\sum_{i, j \in \{i_1, i_2, \cdots, i_k\}} x_{ij} \ge k$$

$$x_{13} = x_{35} = x_{52} = x_{24} = x_{41} = 1$$

$$x_{ij} = 0, (i, j) \notin \{(1,3), (3,5), (5,2), (2,4), (4,1)\}$$

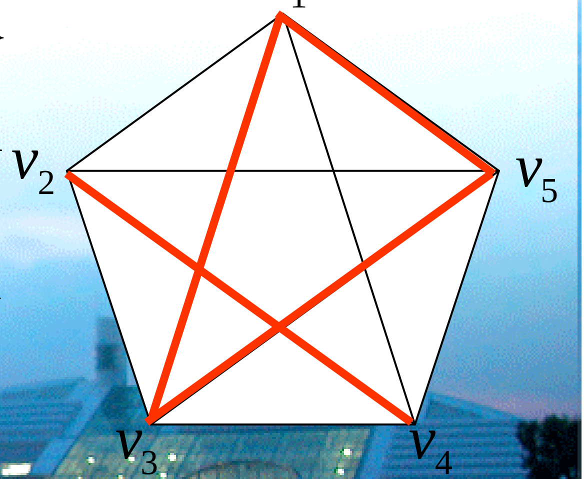


- 若路线不包含经过 k 个城市 $i_1, i_2, \cdots, i_k$ 的子环游

$$\sum_{i, j \in \{i_1, i_2, \cdots, i_k\}} x_{ij} \le k - 1$$

$$x_{13} = x_{35} = x_{51} = x_{24} = x_{42} = 1$$

$$x_{ij} = 0, (i, j) \notin \{(1,3), (3,5), (5,1), (2,4), (4,2)\}$$



TSP问题的整数规划

数学建模

- 为使整数规划的解不存在任意子环游，需增加约束以消除子环游

$$\sum_{i,j\in S} x_{ij} \le |S|-1,\ \ \forall \varnothing \ne S \subseteq \{2,3,\cdots,n\}$$

  - 增加约束后的整数规划约束个数可能为$O(2^n)$
- 增加 $n-1$ 个实决策变量 $u_2,u_3,\cdots,u_n$ 和 $(n-1)^2$ 个约束

$$(n-1)x_{ij}+u_i-u_j \le n-2,\ \ i,j=2,3,\cdots,n$$

也可消除子环游

Miller CE, Tucker AW, Zemlin RA. Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7, 326-329, 1960

MTZ整数规划

浙江大学 ZheJiang University 数学建模

- 若解含有子环游，则约束不被满足
  - 若含有子环游，则必有一子环游 $(i_1\ i_2 \cdots i_k)$ 不经过城市 1
  - 若该子环游仅含一个城市 i，则 $x_{ii}=1$

$$(n-1)x_{ii}+u_i-u_i=n-1>n-2$$

矛盾

  - 若该子环游至少含有两个城市，则 $x_{i_j i_{j+1}}=1\left(i_{k+1}=i_1\right)$，

$$(n-1)x_{i_j i_{j+1}}+u_{i_j}-u_{i_{j+1}}=n-1+u_{i_j}-u_{i_{j+1}},\ j=1,2,\cdots,k$$

$$\sum_{j=1}^{k}\left((n-1)x_{i_j i_{j+1}}+u_{i_j}-u_{i_{j+1}}\right)=\sum_{j=1}^{k}\left(n-1+u_{i_j}-u_{i_{j+1}}\right)=k(n-1)>k(n-2)$$

矛盾

$$\boxed{(n-1)x_{ij}+u_i-u_j\le n-2,\ \ i,j=2,3,\cdots,n}$$

MTZ整数规划

浙江大学 ZheJiang University
数学建模

- 对任意可行环游，存在一组可行解满足约束
  - 对任意可行环游，选定城市 1 为起点，若城市 i 是环游中的第 k 个经过的城市，取 $u_i = k$，$2 \le u_i \le n$，x_{ij} 的取值同前定义
  - 若对某个 i, j 组合，$x_{ij} = 0$，则
$$(n-1)x_{ij} + u_i - u_j = u_i - u_j \le n-2$$
  - 若对某个 i, j 组合，$x_{ij} = 1$，则 $u_i - u_j = -1$，
$$(n-1)x_{ij} + u_i - u_j = (n-1) - 1 = n-2$$

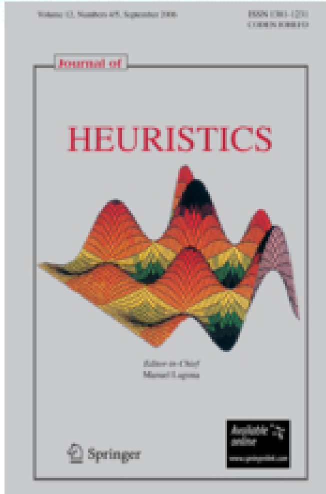
$$\boxed{(n-1)x_{ij} + u_i - u_j \le n-2,\ \ i, j = 2,3,\cdots,n}$$

近似算法

浙江大学 ZheJiang University

数学建模

- 可给出$\mathcal{NP}$－难问题任一实例最优解的算法总需要指数时间。较为现实的思路是"用精度换时间"，在多项式时间或可接受的实际运行时间内得到一个目标值与最优值较为接近的可行解
  - 近似算法（**approximation algorithm**）：算法的时间复杂性可通过分析确定（一般要求多项式时间），算法给出的近似解与最优解目标值之间的差距可通过证明严格估计
  - 启发式算法（**heuristic**）：无法说明算法的时间复杂性，或无法估计算法给出的近似解与最优解目标值之间的差距



Journal of Heuristics

$(\epsilon\upsilon\rho\iota\sigma\kappa\epsilon\iota\nu)$

to find

最坏情况比

浙江大学 ZheJiang University
数学建模

- 设Π是一个极小化优化问题，A是它的一个算法。对Π的任意实例 I，算法 A 给出一个可行解，其目标值为 $C^A(I)$，实例 I 的最优目标值为 $C^*(I)$
- 称 $r_A = \inf\{r \geq 1 \mid C^A(I) \leq rC^*(I), \forall I\}$或 $r_A = \sup_I \left\{ \frac{C^A(I)}{C^*(I)} \right\}$ 为算法 A 的最坏情况比（**worst-case ratio**）
  - 若算法 A的最坏情况比为 r_A，则对该问题的任意实例 I，均有 $C^A(I) \leq r_A C^*(I)$
  - 最坏情况比越接近于1，说明算法给出的可行解目标值越接近于最优值，算法近似性能越好

近似方案

浙江大学 ZheJiang University

数学建模

- 近似方案$\{A_\varepsilon\}$
  - 对任意给定的 ε，算法A_ε的最坏情况比为$1+\varepsilon$
  - 若算法A_ε的时间复杂度为$O(\text{poly}(n))$，称为多项式时间近似方案（**Polynomial Time Approximation Scheme，PTAS**）；若算法A_ε的时间复杂度为$O\left(\text{poly}\left(n,\frac{1}{\varepsilon}\right)\right)$，称为完全多项式时间近似方案（**Fully Polynomial Time Approximation Scheme，FPTAS**） $O\left(\frac{n}{\varepsilon^2}\right)$ $O\left(n^{\frac{1}{\varepsilon}}\right)$
  - 近似方案具有几乎最好的近似性能，但算法实现通常较为复杂，实际运算时间可能很长。多数组合优化问题在 $\mathcal{P}\neq\mathcal{NP}$ 假设下不存在近似方案
  - 背包问题存在时间复杂度为 $O\left(n\min\left\{\ln n,\ln\frac{1}{\varepsilon}\right\}+\frac{1}{\varepsilon^2}\ln\frac{1}{\varepsilon}\min\left\{n,\frac{1}{\varepsilon}\ln\frac{1}{\varepsilon}\right\}\right)$ 的**FPTAS**

Kellerer H, Pferschy U. Improved dynamic programming in connection with an FPTAS for the knapsack problem. *Journal of Combinatorial Optimization*, 8, 5-11，2004.

平均情况比

- 假设实例中的数据服从一定概率分布，$\frac{C^A(I)}{C^*(I)}$ 为一随机变量，其期望 $\left(\frac{C^A(I)}{C^*(I)}\right)$ 即为算法 的**平均情况比**（**average-case ratio**）
  - 平均情况比依赖于所假设的实例中数据所服从的概率分布
    - 在多数情况下，缺乏足够的依据来判断一个实例中的数据来自何种分布
  - 即便已得到某一算法在某个分布下的平均情况比，对来自该分布的一个实例 I，也无法对 $\frac{C^A(I)}{C^*(I)}$ 的值作出估计
  - 平均情况比的证明比最坏情况比更为复杂

太阳能小屋

- **2012B太阳能小屋的设计的子问题**
  - 用不同类型的光伏电池铺设给定长宽的矩形屋顶
  - 类型为 i 的光伏电池是长为 a_i，宽为 b_i 的矩形，每块该类型电池预期效益为 p_i
  - 铺设时光伏电池不能互相覆盖，也不能超出屋顶之外，但不必覆盖屋顶所有区域
  - 采用怎样的铺设方案可使效益最大

$$\max \quad Z = \sum_{i=1}^{24} f_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{24} S_i x_{ij} \leqslant A_j,$$

$$x_{ij} \in N.$$

二维装箱　背包

多背包　多维（大小）背包　多维（区域）背包

在使用某一方案的光伏电池组件组合铺设一个墙面或房顶时，其最大的限制条件是面积限制，即光伏电池组件组合的面积和不得超过墙体的面积．其约束条件的数学公式表达即为 $xyS_i \leq S$，其中 S_i 表示第 i 种电池组件的单个面积，S 表示需要优化的墙体或屋顶的面积．

36

寻路

- 给定一个带权重的有向图 $G=(V,E)$，V 为顶点集，E 为有向边集，每一条有向边均有一个权重。对于给定的顶点 s,t，以及 V 的子集 V'，寻找从 s 到 t 的无环有向路径 P，使得 P 经过 V' 中所有的顶点（对经过 V' 中节点的顺序不做要求）


HUAWEI Code Craft 2016 “未来网络·寻路”初赛题目介绍　　文档密级：公开

HUAWEI Code Craft 2016

未来网络•寻路

初赛题目介绍

2016.03.04

前言

赛题源自“未来网络”业务发放中的路由计算问题。算路问题属于基础算法问题，在图论、网络、交通等各个方面均有着广泛的研究与运用，里面不乏一些经典的算法，例如最短路中的广度优先搜索，Dijkstra算法等。网络算路问题的更优算法实现对于网络资源高效配置具有重要价值。

$\boxed{\text{最短路+TSP}}$

Dreyfus SE. An appraisal of some shortest-path algorithms. *Operations Research*, 17(3): 395-412, 1969.

Ibaraki T. Algorithms for obtaining shortest paths visiting specified nodes. *SIAM Review*, 15(2): 309-317, 1973.

37

谢　谢