组合优化与算法设计

浙江大学 谈之奕

数学建模

MATH T

组合优化

- 组合优化（Combinatorial Optimization）
  - 应用于离散对象的，从有限多个可行解中找出使某个目标函数达到最优的解的优化问题
  - 组合优化是离散数学（Discrete Mathematics）与最优化的交叉学科分支
- 组合优化问题的求解
  - 相对决策变量为连续变量的连续优化（**Continuous Optimization**）问题，组合优化问题的最优解缺少好的性质，求解缺少好的工具
  - 组合优化问题通常不能通过穷举所有可能的解加以比较来求解，因为可行解的数目可能是一很大的数，以致于当前或相当长的一段时间内人力或计算机不能承受

FIGURE 2

Grand Vizier Sissa Ben Dahir, a skilled mathematician, asks his reward from King Shirham of India.

Gamov G. *One Two Three . . . Infinity: Facts and Speculations of Science*，Viking Press，1947

（中译本：从一到无穷大：科学中的事实与猜想，张卜天译，商务印书馆，**2019**）

离散与连续

- 模型
  - 现有 n 件物品，物品 j 的价值为 p_j ，大小为 w_j 。物品质地均匀，可任意切割
  - 将若干物品的全部或部分放入容量为 C 的背包中，在放入背包物品大小之和不超过背包容量前提下，使放入背包物品价值之和尽可能大
- 求解思路
  - 称 $\frac{p_j}{w_j}$ 为物品 j 的价值密度
  - 将物品按价值密度从大到小的顺序排列，优先放入价值密度大的物品，直至第一个不能完整放入背包的物品。将该物品的一部分放入背包，使得背包没有剩余空间

数学建模

MATH T

$$\max \sum_{j=1}^{n} p_j x_j$$

$$s.t. \quad \sum_{j=1}^{n} w_j x_j \le C$$

$$0 \le x_j \le 1,\ j = 1, \cdots, n$$

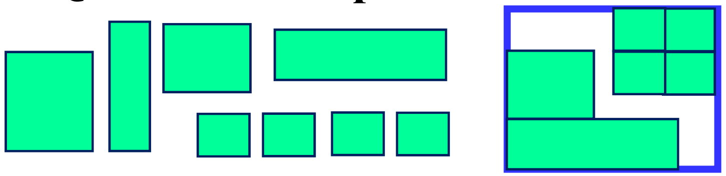
数学建模

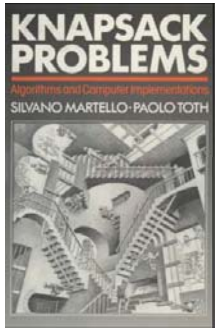
MATH T

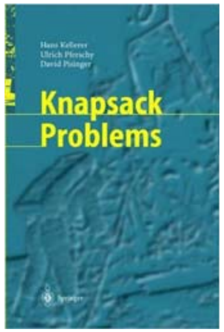
离散与连续

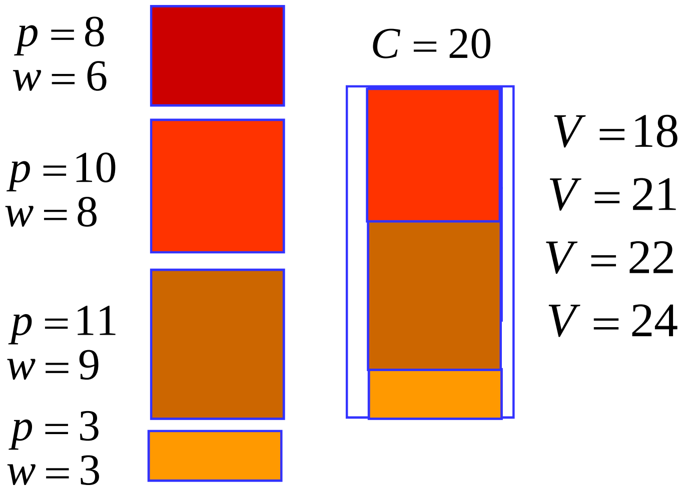
- 背包问题（ **Knapsack** ）
  - 现有 n 件物品，物品 j 的价值为 p_j ，大小为 w_j
  - 将若干物品放入容量为 C 的背包中，在放入背包物品大小之和不超过背包容量前提下，使放入背包物品价值之和 V 尽可能大

Unbounded Knapsack
Multidimensional Knapsack
Multiple Knapsack
Quadratic Knapsack









Martello S, Toth P. ***Knapsack Problems: Algorithms and Computer Implementations*****, Wiley, 1990.**
Kellerer H, Pferschy U, Pisinger D. ***Knapsack Problems*****. Springer, 2004.**

最优解中可能出现价值密度较大的物品未放入背包，而价值密度较小的物品放入背包的情况

算法

数学建模

MATH T

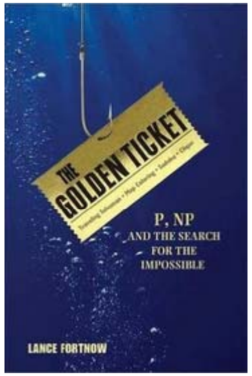
- 算法（algorithm）
  - 在有限步骤内求解某一问题的一组含义明确的可以完全机械执行的规则
  - 一系列将输入转换为输出的计算步骤
- 组合优化问题的算法
  - 设计算法是求解组合优化问题的主要手段
  - 部分问题，如指派问题、最短路问题等，可设计多项式时间算法求得最优解
  - 部分问题，如背包问题、TSP等，还没有找到能在多项式时间内求得最优解的算法

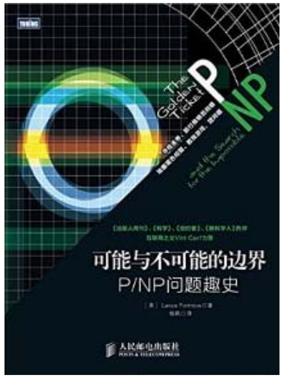
Muḥammad ibn Mūsā al-Khwārizmī
（中）花拉子米，（拉丁）**Algorithmi**
（约780-约850）波斯数学家
主要著作之一 **The Compendious Book on Calculation by Completion and Balancing**（又名**Al-jabr**）

$\mathcal{P}$ vs. $\mathcal{NP}$ 问题

- $\mathcal{P}$ vs. $\mathcal{NP}$ 问题
  - $\mathcal{P}$ 与 $\mathcal{NP}$ 均为问题的集合。$\mathcal{P}$ vs. $\mathcal{NP}$ 问题指 $\mathcal{P} = \mathcal{NP}$ 与 $\mathcal{P} \subset \mathcal{NP}$ 中何者成立
  - $\mathcal{P}$ vs. $\mathcal{NP}$ 问题是数学和计算机科学中的重大未解决难题之一
  - 目前多数人相信 $\mathcal{P} \subset \mathcal{NP}$。此时背包问题、TSP等，不存在多项式时间内求得最优解的算法，但可尝试设计算法求近似解

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.





数学建模

MATH T

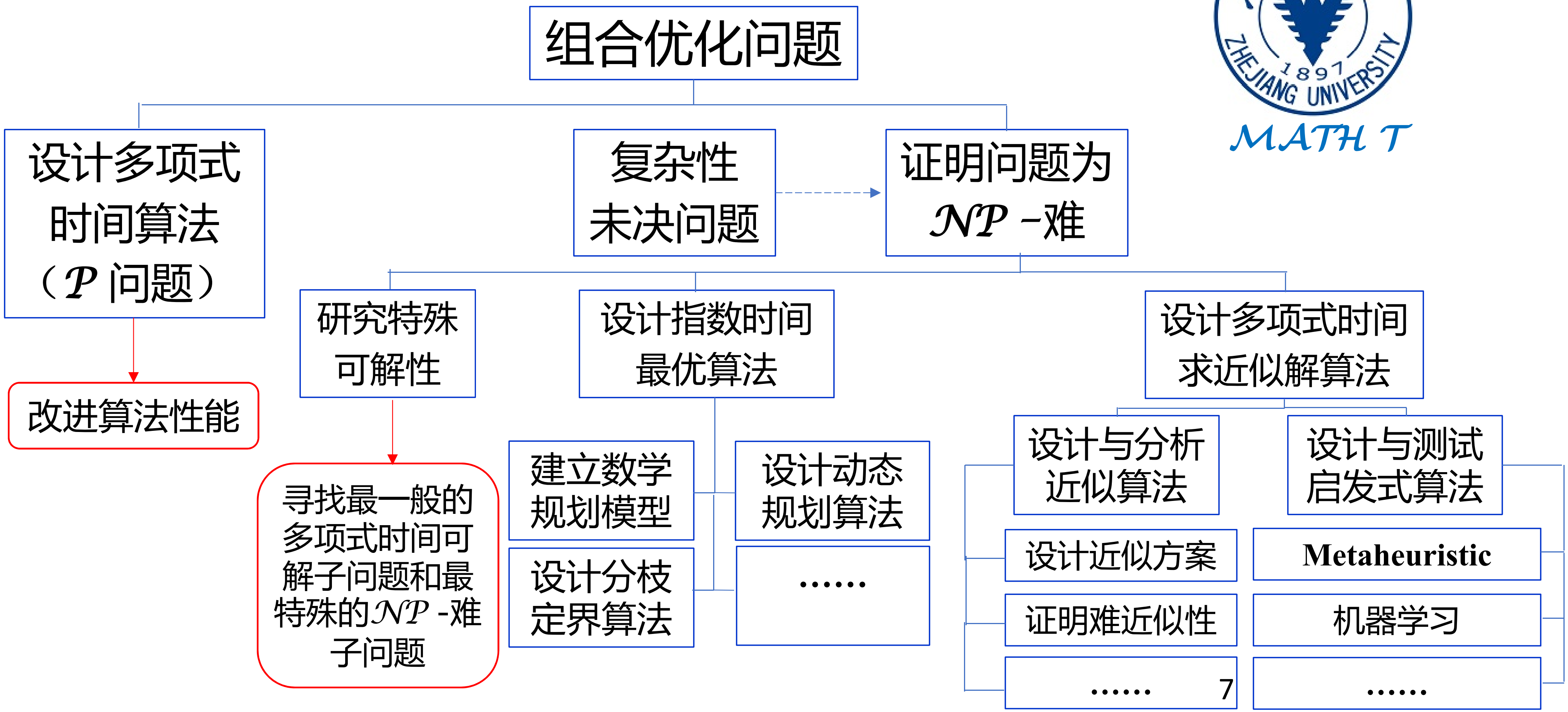
Clay Mathematics Institute

Millennium Problems

1 Yang–Mills and Mass Gap
2 Riemann Hypothesis
3 P vs NP Problem
4 Navier–Stokes Equation
5 Hodge Conjecture
6 Poincaré Conjecture √
7 Birch and Swinnerton-Dyer Conjecture

数学建模

组合优化问题求解方法

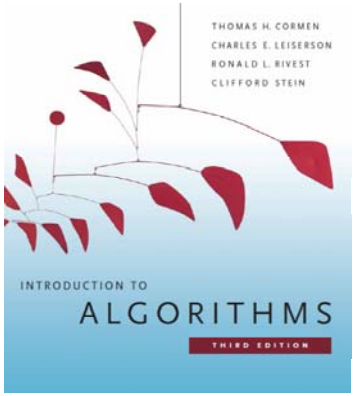


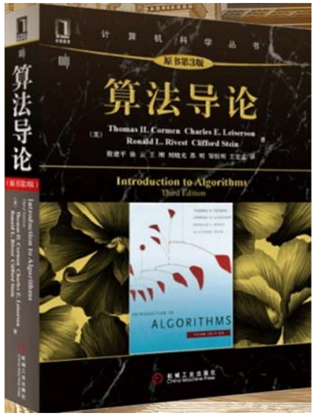
MATH T

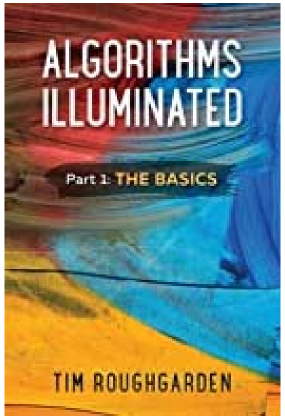
算法设计

数学建模

MATH T







Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms*. MIT press, 2009（中译本：算法导论，殷建平等译，机械工业出版社，2012**.**

Roughgarden T. *Algorithms Illuminated* (I-IV). Soundlikeyourself Publishing, 2017.（中译本：算法详解，徐波译，人民邮电出版社，2019.）

Donald Ervin Knuth（高德纳）

（**1938-** ）

美国计算机学家

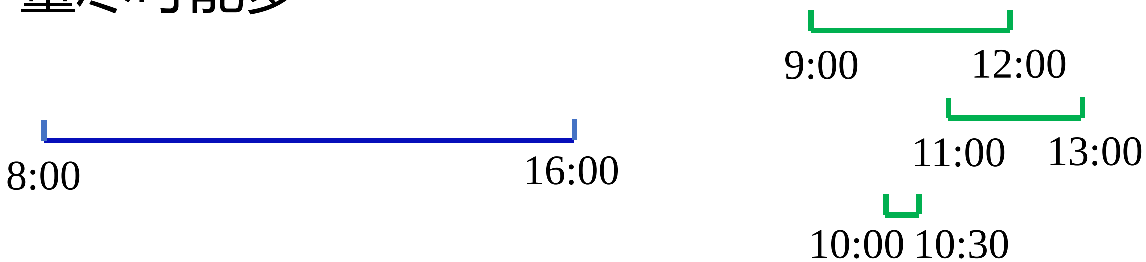
1974年图灵奖得主

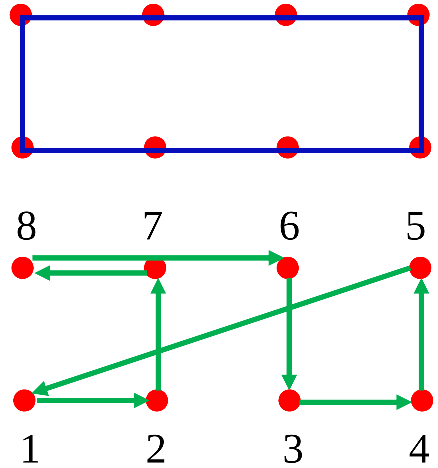
Knuth DE. *The Art of Computer Programming*. Addison-Wesley, 1968-2011.

贪心

MATH T

- 贪心（greedy）
  - 在每一次决策时，选择当前可行且最有利的决策
- 场馆安排问题
  - 某场馆收到 n 项借用申请，第 i 项申请的活动开始时间为 s_i ，结束时间为 t_i ，持续时间为 $d_i = t_i - s_i$
  - 场馆在同一时刻只能进行一项活动，一项活动开始后必须连续进行直至结束
  - 希望选择接受部分申请，使得场馆能开展的活动数量尽可能多





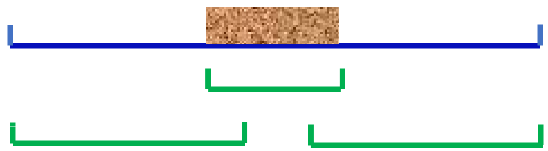
局部最优解未必是全局最优解

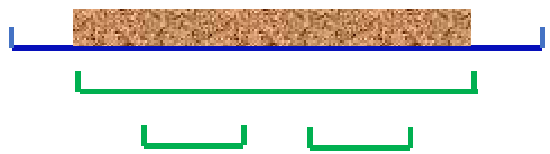
贪心算法

- 场馆安排问题的贪心算法
  - 将所有申请按某种顺序排列，依次考虑各项申请
  - 若当前申请涉及的活动所需时段未被已接受的申请涉及的活动占用，则接受该申请，否则拒绝该申请
- 申请排列的顺序

  可行性

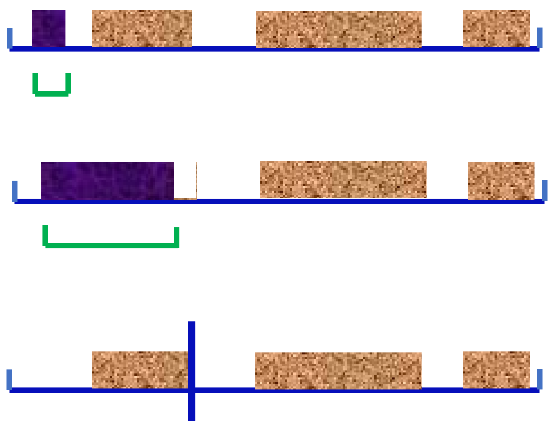
  - 按持续时间从小到大的顺序排列
  - 按开始时间从小到大的顺序排列
  - 按结束时间从小到大的顺序排列 √





贪心算法

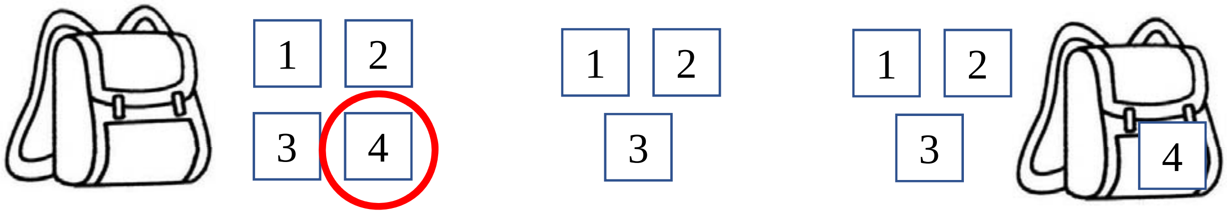
- 最优性证明
  - 存在一个最优解 σ ，接受结束时间最早的申请 i_0
    - 反证法。假设 σ 中 i_0 被拒绝
    - 若在 i_0 的持续时间内，场馆空闲
      - 增加接受 i_0 ，接受申请数增加，与 σ 是最优解的假设矛盾
    - 若在 i_0 的持续时间内，场馆在一段时间内被其他接受的申请 i 占用
      - 由于 i_0 是结束时间最早的申请， i 的结束时间晚于 i_0
      - 接受 i_0 ，拒绝 i ，接受申请数不变。场馆提早空闲，不会影响其他已接受申请，仍是最优解
  - 考虑开始时间晚于 i_0 的结束时间的所有申请，继续运用上述性质



数学建模

动态规划

- 动态规划（**dynamic programming，DP**）
  - 动态规划是求解多阶段决策优化问题的一种数学方法和算法思想
  - 动态规划求解组合优化问题
    - 将需求解的实例转化为一系列互有联系、规模较小的实例，并导出不同实例最优解之间的关系，从而可由初始条件出发逐步递推求得最优解



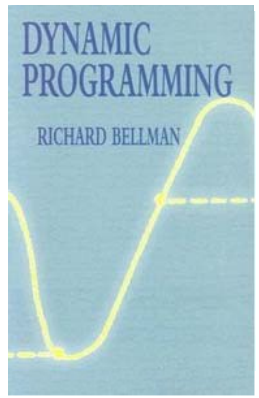
从4件物品中选择哪些物品放入背包

3件物品，容量不变背包的最优解

3件物品，容量减小背包的最优解

未放入背包　　放入背包

最优解是否把第4件物品放入背包



Richard Ernest Bellman
(1920-1984)
美国运筹学家

Bellman RE, *Dynamic Programming*, Princeton University Press, 1957

动态规划

- 背包问题的动态规划
  - 由物品集中的前 $k(k \le n)$ 个物品和容量为 $w(w \le C)$ 的背包组成的实例，记为 $I(k,w)$ ，其最优值为 $V(k,w)$
  - $I(k,w)$ 的最优解
    - 若物品 k 未放入背包，背包中放入的物品均为前 $k-1$ 个物品中的部分物品，价值之和的最大值为 $V(k-1,w)$
    - 若物品 k 放入背包，背包剩余容量为 $w-w_k$ ，可放入前 $k-1$ 个物品中的部分物品，价值之和的最大值为 $V(k-1,w-w_k)$
  - $V(k,w)=\max\left\{V(k-1,w),V(k-1,w-w_k)+p_k\right\}$
    - 若 $w_k > w$ ，则物品 k 不能放入容量为 w 的背包中，$V(k,w)=V(k-1,w)$
  - 初始条件 $V(0,w)=0,\quad w=0,\cdots,C$ ，最优值 $V(n,C)$

动态规划

数学建模

MATH T

- 背包问题的动态规划
  - $V(k,w)=\begin{cases}\max\{V(k-1,w),V(k-1,w-w_k)+p_k\} & 若w_k\le w\\ V(k-1,w) & 若w_k>w\end{cases}$
  - 实例

    $n=4, C=5$

    $p_1=3, p_2=4, p_3=5, p_4=6$

    $w_1=2, w_2=3, w_3=4, w_4=5$

    - 最优值：7
    - 最优解：物品1,2放入背包

			$p_1=3$	$p_2=4$	$p_3=5$	$p_4=6$
	$w \backslash k$	0	1	2	3	4
	0	0	0	0	0	0
	1	0	0	0	0	0
$w_1=2$	2	0	3	3	3	3
$w_2=3$	3	0	3	4	4	4
$w_3=4$	4	0	3	4	5	5
$w_4=5$	5	0	3	7	7	7

启发式算法

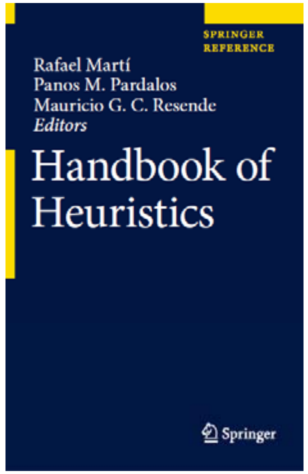
- 启发式算法（heuristic）
  - 基于某种直观想法、合理假定，或者借助物理、化学、生命科学中的一些原理而设计的算法
  - 体现了在求解的最优性、精确性与求解资源之间的权衡
- **Metaheuristic**
  - **A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms**

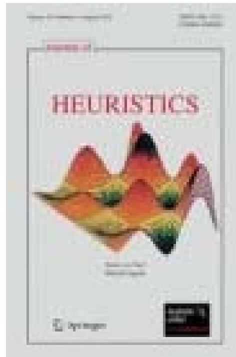
Martí Rafael, Pardalos PM, Resende MGC (Eds) *Handbook of Heuristics*, Springer, 2018

数学建模

MATH T

μετά **εὑρίσκω**
Meta- **heuristic**
(beyond in the sense of high-level) **(search)**



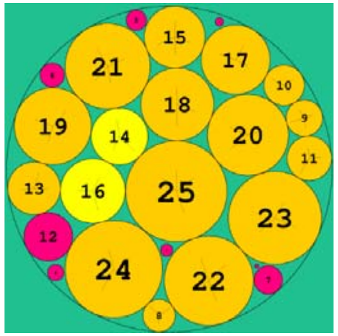


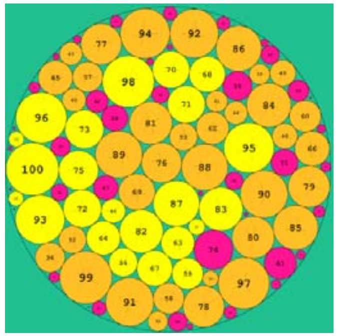
Journal of Heuristics

启发式算法

- 启发式算法
  - 启发式算法的有效性一般需通过计算机模拟验证
  - 实例来源
    - 真实数据
    - 实例库
    - 随机生成实例
  - 算法性能
    - 算法解与最优解的接近程度
    - 模拟环境与算法所用时间
    - 算法的稳健性

$$1 \le \frac{\text{最优解}}{\text{算法解}} \le \frac{\text{最优解的上界}}{\text{算法解}} \approx r$$





TSPLIB

TSPLIB is a library of sample instances for the TSP (and related problems) from various sources and of various types.

最优解的估计

连续形式背包问题的最优解是背包问题最优解的一个上界

MATH T

遗传算法
（genetic algorithm）
模拟退火算法
（simulated annealing）
禁忌搜索
（tabu search）
蚁群算法
（ant colony optimization）
粒子群优化算法
（partial swarm optimization）


谢 谢