



第七讲 近似算法

主讲教师：朱建明

Email: jmzhu@ucas.ac.cn

Homepage: <http://people.gucas.ac.cn/~jianming>



- ✓ 精确算法
- ✓ 启发式算法
- ✓ 近似算法

博学笃志
格物明德

洛甫样



精确算法

- ✓ P类问题
- ✓ 枚举算法

博学笃志
格物明德

洛甫样

heuristic algorithm(启发式算法)

- ✓ In computer science, a **heuristic algorithm**, or simply a **heuristic**, is an algorithm that is able to produce an **acceptable solution** to a problem in many practical scenarios, in the fashion of a general heuristic, but for which there is no formal proof of its correctness. Alternatively, it may be correct, but may not be proven to produce an optimal solution, or to use reasonable resources. Heuristics are typically used when there is no known method to find an optimal solution, under the given constraints (of time, space *etc.*) or at all.

启发式算法的发展

- ✓ 40年代：由于实际需要，提出了启发式算法（快速有效）。
- ✓ 50年代：逐步繁荣，贪婪算法和局部搜索得到关注。
- ✓ 60年代：反思，发现以前提出的启发式算法速度很快，但是解得质量不能保证，而且对大规模的问题仍然无能为力（收敛速度慢）。
- ✓ 70年代：计算复杂性理论的提出，NP问题。许多实际问题不可能在合理的时间范围内找到全局最优解。发现贪婪算法和局部搜索算法速度快，但解不好的原因主要是他们只是在局部的区域内找解，解没有全局最优性。

启发式算法的发展

- ✓ 1975年holland提出遗传算法 (Genetic Algorithm)
- ✓ 1977年Glouer提出禁忌搜索算法 (Tabu Search)
- ✓ 1982年Kirkpatrick提出模拟退火算法 (Simulated Annealing)
- ✓ 1995年Dorigo提出蚁群算法 (Ant Colony Optimization)

启发式算法的优点与缺点

✓ 启发式算法的优点

1. 模型误差、数据不精确性、参数估计误差等可能造成最优算法的解比启发式算法的解更差；
2. 复杂问题无法求得最优算法或最优算法太复杂；
3. 简单易行，直观，程序简单。

✓ 启发式算法的缺点

1. 不能保证最优；
2. 不稳定；
3. 依赖于实际问题、设计者经验。

启发式算法的分类

◆ 简单直观的算法

- ❧ 一步算法：不在两个可行解之间比较，在未终止的迭代过程中，得到的中间解有可能不是可行解；
- ❧ 改进算法：迭代过程是从一个可行解到另一个可行解变换，通过两个解的比较而选择好的解，直到满足一定的要求为止；

启发式算法的分类

- ◆ 数学规划算法

用连续优化（如线性规划）的方法求解组合优化问题（如整数线性规划模型），其中包括一些启发式规则。

基于数学规划的理论。

博学笃志
格物明德

洛甫样

启发式算法的分类

◆ 现代优化算法

禁忌搜索算法

模拟退火算法

遗传算法

人工神经网络

蚁群算法

粒子群算法

混合算法

特点:

- 基于客观世界中的一些自然现象;
- 建立在计算机迭代计算的基础上;
- 具有普适性, 可解决实际应用问题。

启发式算法的性能分析

- ◆ 评价算法优劣的指标

 - 算法的复杂性（计算效率）

 - 解的偏离程度（计算效果）

 - 算法的稳健性（不同实例、不同时间、不同起点的差异）

- ◆ 评价算法优劣的手段

 - 最坏情况分析（纯理论）

 - 概率分析（理论分析）

 - 计算模拟分析（统计特性）

近似算法

- ✓ 给出满意解，而并不一定是最优解
- ✓ 追求算法时间复杂度为多项式
- ✓ 性能分析：

近似比

博学笃志
格物明德

洛甫样



背包问题的贪婪算法 (Greedy algorithm)

贪婪算法：采用逐步构造最优解的方法。

在每个阶段，都作出一个看上去最优的决策（在一定的标准下）。决策一旦作出，就不可再更改。作出贪婪决策的依据称为贪婪准则（greedy criterion）。

博学笃志
格物明德

洛甫样

背包问题的贪婪算法 (Greedy algorithm)

STEP 1 对物品以 $\frac{c_i}{a_i}$ 从大到小排列，不妨把排列记成 $\{1, 2, \dots, n\}$, $k := 1$;

STEP 2 若 $\sum_{i=1}^{k-1} a_i x_i + a_k \leq b$, 则 $x_k = 1$; 否则 $x_k = 0$,
 $k := k + 1$; 当 $k = n + 1$ 时, 停止; 否则, 重复STEP2。

时间复杂性分析

算法性能分析

格博
物志
明德

点覆盖 (vertex covering set)



- ✓ 设 $F \subset V(G)$, 若 G 的每条边至少有一个端点属于 F , 则称 F 是 G 的一个点覆盖集。若对 $\forall v \in F, F - \{v\}$ 都不再是 G 的点覆盖, 则称点覆盖 F 是一个极小点覆盖。图 G 的含点数最小的点覆盖称为最小点覆盖, 其点数称为 G 的点覆盖数, 记为 β 或 $\beta(G)$



点覆盖贪婪算法

- ✓ Input: A graph $G=(V,E)$
- ✓ Output: A node cover C .

Begin

$C=\text{empty};$

while $E \neq \text{empty}$ do

 choose the node in V that has the largest degree,
 remove it from G and add it to C .

End

算法分析

性能分析

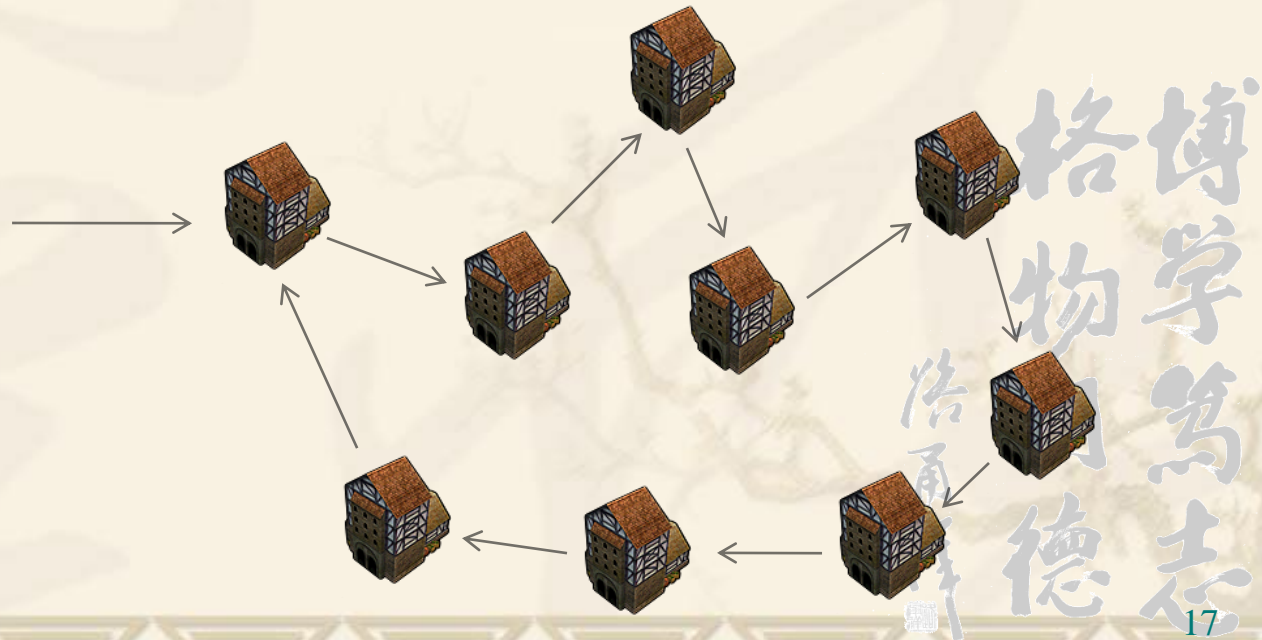
改进算法

格物致知
明德行

博学笃志



- ✓ The **Travelling Salesman Problem (TSP)** is a problem in combinatorial optimization studied in operations research and theoretical computer science. Given a list of cities and their pair-wise distances, the task is to find a shortest possible tour that visits each city exactly once.



网络优化的其他应用

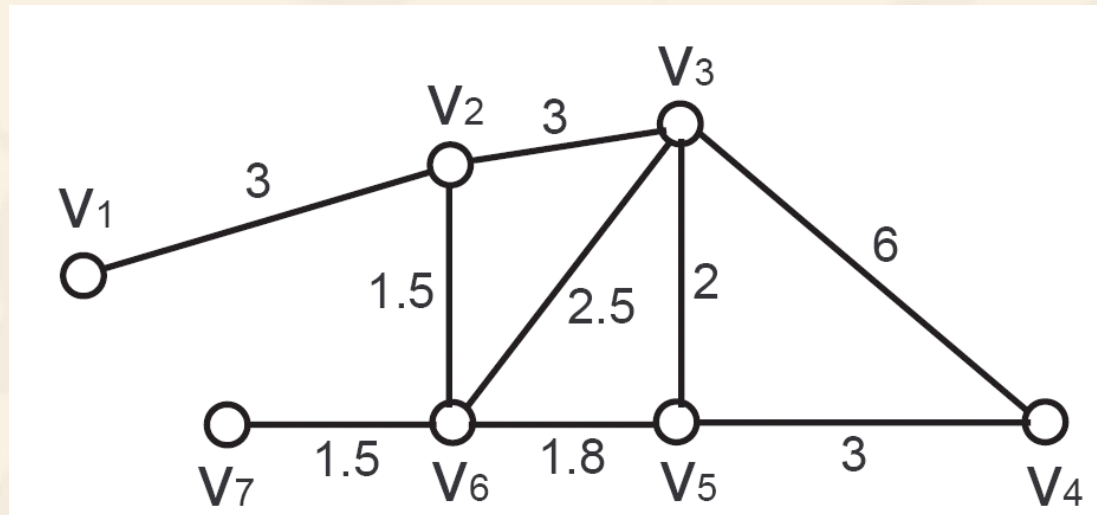
- ✓ 图的中心
- ✓ 支配集
- ✓ 覆盖集

博学笃志
格物明德

洛甫样

应用实例-消防队选址

- ✓ 设某市有7个化工厂，已知这些工厂间的公路以及（单位：km）如图。现拟建一个消防队负责这些工厂的消防工作。问消防队应设在哪个工厂，以便任一个工厂失火时都能及时赶去救火？



图的中心、中位点



- ✓ Given a weighted graph $G = (V, A)$ with n vertices and an $n \times n$ weight matrix $C = [c_{ij}]$.
- ✓ $d(u, v)$ 表示顶点 u 到 v 的最小距离。
- ✓ 图 G 的半径(radius): $\text{rad } G = \min_{u \in V(G)} \max_{v \in V(G)} d(u, v)$
- ✓ 图 G 的直径(diameter):

$$\text{diam } G = \max_{u \in V(G)} \max_{v \in V(G)} d(u, v) = \max\{d(u, v) \mid \forall u, v \in V(G)\}$$

- ✓ 图 G 中顶点 u 的离心率(eccentricity): $e(u) = \max_{v \in V(G)} d(u, v)$
- ✓ 图 G 的中心(center): 图 G 中具有最小离心率的顶点
- ✓ 图 G 的中位点(median): 设 $g(u) = \sum_{v \in V(G)} d(u, v)$, 使 $g(u)$ 达到最小的顶点称为图 G 的中位点。

应用实例-应急救援中心的选址

- ✓ 欲在 n 个地点设置若干个应急中心，使得每个地点都至少一个应急中心相邻（有直达通路）。同时为了减少造价，应急中心的数目要尽可能少。问应设多少个？设置在哪儿？

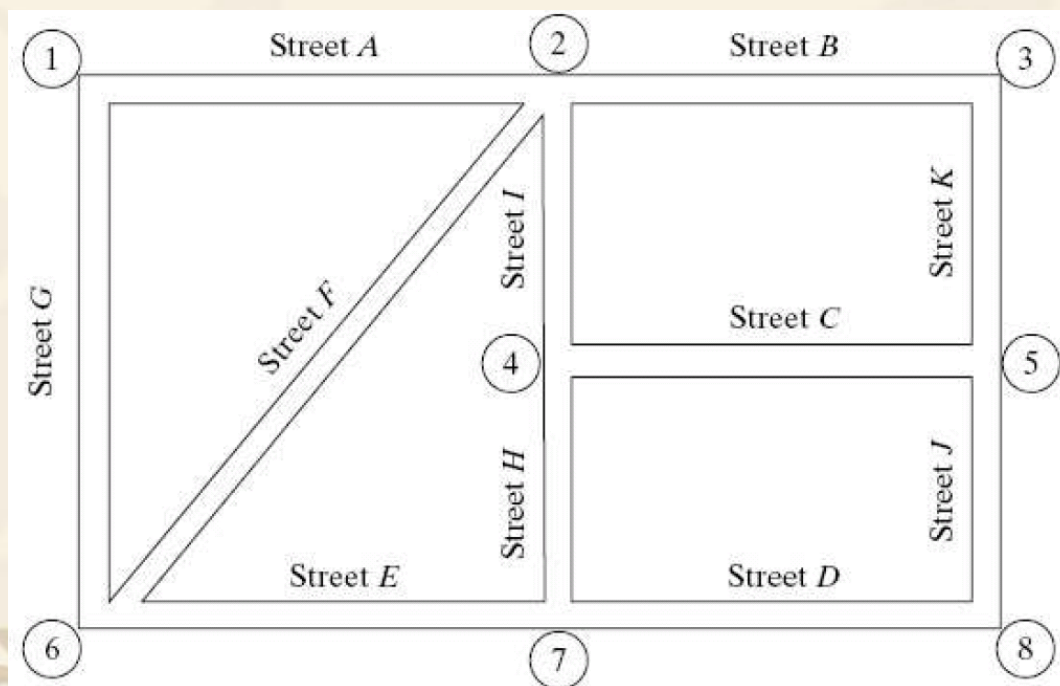
支配集 (domination set)



- ✓ 设 $D \subseteq V(G)$, 若对 $\forall u \in V(G)$, 要么 $u \in D$, 要么 u 与 D 中的某些顶点相邻, 则称 D 为图 G 的一个支配集。
- ✓ 如果一个支配集的任何真子集都不是支配集, 则称它为极小支配集。
- ✓ 图 G 的含顶点最少的支配集称为最小支配集。最小支配集的顶点个数称为 G 的支配数, 记为 $\gamma(G)$ 或 γ 。

应用实例

- ✓ 为了提高校园的安全性，A大学的保安部门决定在校园内部的几个位置安装紧急报警电话。保安部希望在校园的每条主要街道上至少都有一部电话的情况下，使得安装的总电话数目最少。



独立集 (vertex independent set)



- ✓ 设 $I \subseteq V(G)$, 若 I 中任二顶点均不相邻, 则称 I 为图 G 的一个 **点独立集** (简称独立集); 若对 $\forall u \in V(G) \setminus I, I \cup \{u\}$ 都不再是 G 的独立集, 则称独立集 I 为图 G 的一个 **极大点独立集**。 G 的含点数最多的点独立集称为 **最大点独立集**, 它含点的个数称为 G 的独立数, 记为 $\alpha(G)$ 或 α 。



应用实例-信号干扰

- ✓ 信号干扰问题
- ✓ 课表问题
- ✓ 药品存储问题

博学笃志
格物明德

洛甫样

vehicle routing problem

- ✓ The **vehicle routing problem (VRP)** is a combinatorial optimization and integer programming problem seeking to service a number of customers with a fleet of vehicles.

