

研究生算法课课堂笔记

上课日期: 12.29

第(3)节课

组长: 杨蕴伦

组员: 官平胤

组员: 白艺冲

组员: 孙豪泽

注意: 请提交 Word 格式文档。

NP-hard 问题:

什么是 NP-hard? 和 NPC 有何区别?

NP-hard 可能比 NPC 问题更难, 有可能无法在多项式时间内验证。

作业: 动态规划第二题

答案:

递推公式:

$$opt(v, i) = \max\{opt(u, i-1) * p(u, v)\} \quad (u, v) \in E, \sigma(u, v) = \sigma_i, 1 < i \leq k$$

$opt(v, i)$ 代表从 v_0 开始, 路径覆盖了 σ_1 到 σ_i

初始条件:

$$opt(v, 1) = \begin{cases} p(v_0, v) & \text{if } (v_0, v) \in E \text{ and } \sigma(v_0, v) = \sigma_1 \\ 0 & \text{otherwise} \end{cases}$$

返回值:

$$\max\{opt(v, k)\}$$

时空复杂度:

$$\theta(mk)$$

另外一种思路:

$opt(v, i)$ 表示从 v 开始, 路径覆盖 σ_i 到 σ_k

递推公式:

$$opt(v, i) = \max\{opt(u, i+1) * p(v, u)\} \quad (v, u) \in E, \sigma(v, u) = \sigma_i, 1 \leq i \leq k$$

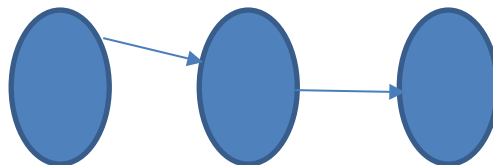
初始条件:

$$opt(v, k+1) = 1$$

两种作业中见到的做法:

① 利用最短路径快速算法。将每条边上的权值取 \log 并取相反数, 再利用 Dijkstra 算法求从 v_0 开始的最短路径。但是必须满足覆盖 σ_1 到 σ_k 。

② 有一种方法是复制 $k+1$ 个图, 将 k 步中每一步可能到达的顶点连起来, 再去寻找新图中概率最大的路径。



空间复杂度: $\theta(kn + km)$

SPFA 改进:

用队列的方法: 顶点 v_0 进队列, 弹出顶点, 沿出边压入新的顶点。

SPFA 任意时刻队列最大长度为 n 。这道题中为 $2n$ 。

空间复杂度: $\theta(n)$, 但无法反向追踪。

如何进行反向追踪:

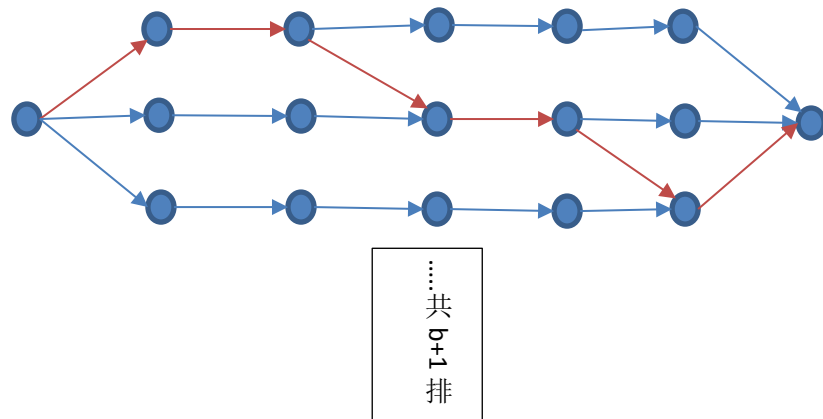
1. 保留二维表
2. 顶点不出队列, 用元素加指针指向此顶点

网络流作业:

第一题: 可以找出变换前最小割唯一的图

第二题: 第二问命题用 BFS 是假的

反例如下: 每边流量都为 1



如图, 红线即为一条 BFS 路径, 假设共有 $b+1$ 行, 则流量并不是 $1/b$ 。