在线算法

东南大学计算机学院 方效林

本章内容

- 在线算法基本概念
- 摘麦穗问题
- 牛寻路问题
- 滑雪橇租赁问题
- 页面调度问题

在线算法基本概念

- 前面介绍的算法
  - 在算法执行之前整个输入数据
- 实际应用存在不满足上述条件的情况
  - 磁盘调度问题
  - 操作系统的页面调度问题
  - Data streams

在线算法基本概念

- 竞争比
  - 设在线算法代价为A，离线最优算法代价为OPT，
  - 若存在非负常数 α 和 c，使得$A \le \alpha \cdot OPT + c$，则称 α 为该在线算法的竞争比 (α-competitive)
  - 当在线算法的竞争比不可能再改进时， 称其为最优在线算法

摘麦穗问题

- 苏格拉底带弟子们来到一片麦田，让他们在田间小路走过，每人选摘一支最大的麦穗，不能走回头路，且只能摘一枝。

——《最大的麦穗》苏教版小学语文

哪个企业的工作Offer

进哪位老师实验室

选择和谁深入交往

摘麦穗的真谛

- 时光不会倒流，人生只是单行线。不论是升学、就业，追求爱情、建立婚姻等等，我们眼前都晃动着许多的麦穗，这时需要拥有一双慧眼，从众多的麦穗中择其大者而取之。

摘麦穗问题

- 假设田间小路旁边一共有 *n* 只麦穗可以选择，最大麦穗出现在任何位置的概率均等，问如何摘下最大的麦穗？
- **离线算法**知道所有麦穗的情况，可以准确无误地摘下最大的麦穗，正确率100%
- **在线算法**仅知道走过的路旁的麦穗，该怎么选择？正确概率是多少？

摘麦穗问题-在线算法

- 在线算法A的做法：
  - 边走边观察前一半的麦穗，记下最大的麦穗，过半后，摘下第一个更大的麦穗。
- 定理: 算法A有至少25%的概率可以摘到最大的麦穗。

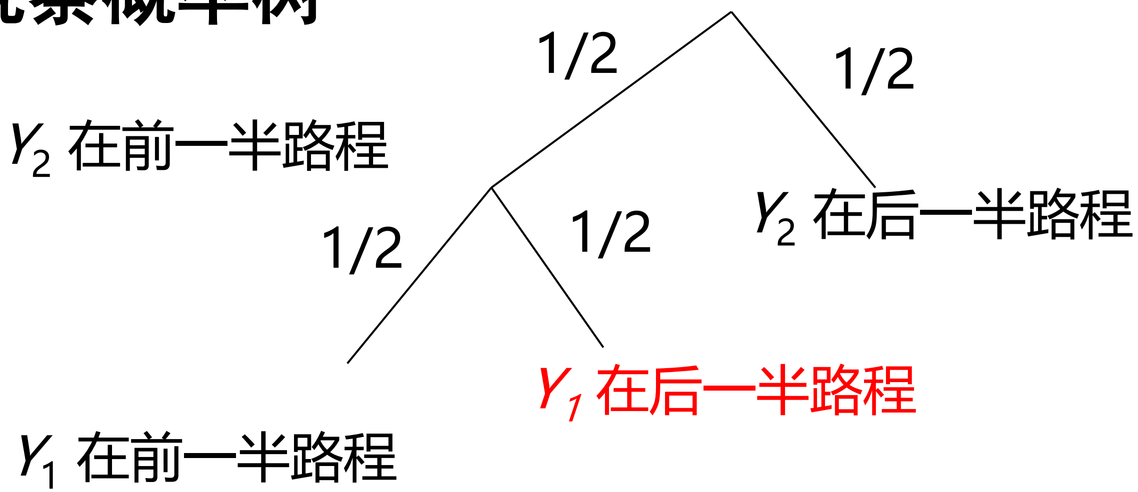
摘麦穗问题-在线算法证明

- **证明：**

  Y_1 – 最大麦穗.

  Y_2 – 第二大麦穗.

- **观察概率树**

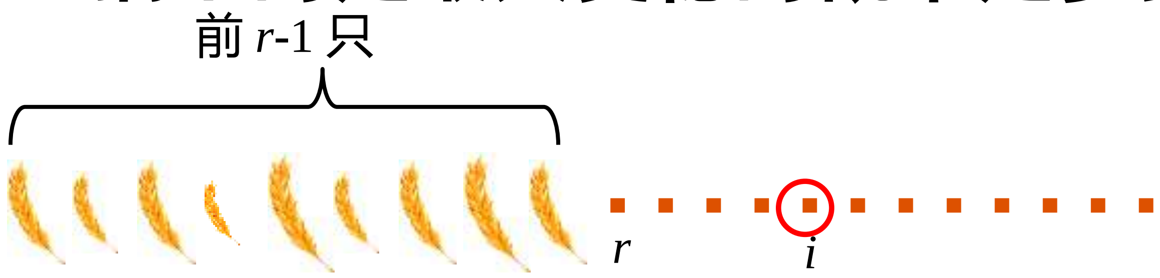


摘麦穗问题

- 摘下最大的麦穗 ←→ 第二大麦穗在前一半路程且最大麦穗在后一半路程
  - 如果这样，我们在前一半路程记下第二大麦穗，在后一半路程选择更大的麦穗，摘下最大麦穗
- 发生的概率：$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4} = 25\%$
- 思考：分析得是否准确，是否还可以提高？

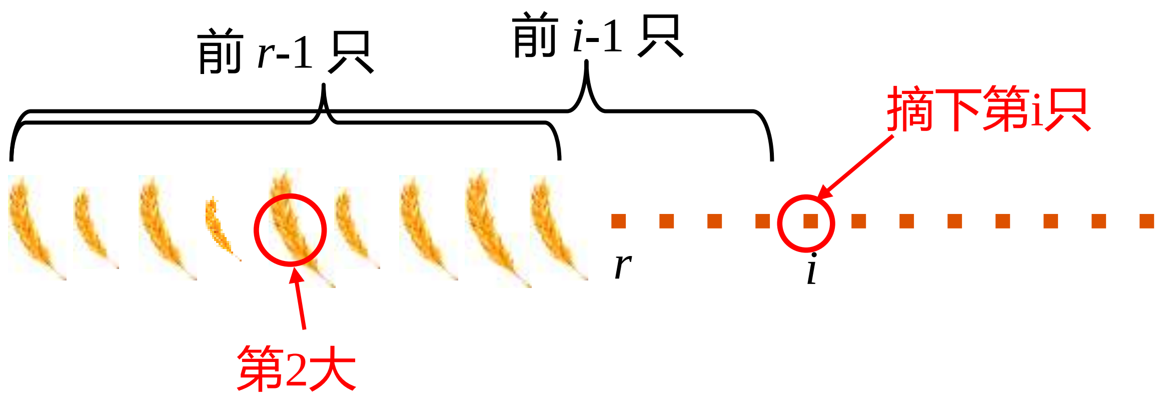
概率是多少

- 边走边观察前 *r*-1 只麦穗，记下最大的，从第 *r* 只麦穗开始，摘下第一个更大的麦穗。
- 问：摘下的是最大麦穗的概率是多少？



概率计算

- 观察前 *r*-1 只麦穗后摘下最大麦穗的概率P_r



$$P_r = \sum_{i=r}^{n} P(i) = \sum_{i=r}^{n} P(\text{选中第}i\text{只}) \times P(\text{第}i\text{只最大})$$

$$= \sum_{i=r}^{n} \frac{r-1}{i-1} \times \frac{1}{n} = \frac{r-1}{n} \sum_{i=r}^{n} \frac{1}{i-1}$$

牛寻路问题

- 有一条很长很长的栅栏，栅栏上只有一个缺口
- 一头牛想从缺口穿过栅栏，但是牛不知道缺口在哪
- 这头牛如何快速找到缺口位置？

牛寻路问题

- 令 *d*=1,side=Right表示牛将向右走*d*=1距离
- Repeat until 找到缺口:
  - 向*side*方向走 *d* 距离
  - 若发现缺口：则终止
  - 若没有发现缺口：回到原点，令*d* =2* d*，*side*设为相反方向

牛寻路问题

- 什么时候最坏情况发生?
- 当差一点点(比如差ε距离)就到达缺口了，但是此时刚调转方向时发生最坏情况。
- 令进行了i次d翻倍，最优解仅仅需要走$2^i+\varepsilon$ 就找到缺口

牛寻路问题

- **但是在线算法需要走过的距离为：**

$$
\begin{aligned}
& 2(1+2+\cdots+2^{i+1})+2^i+\varepsilon \\
= \quad & 2(2^{i+2}-1)+2^i+\varepsilon \\
< \quad & 8\cdot 2^i+2^i+\varepsilon \\
< \quad & 9\cdot \mathsf{OPT} \qquad\qquad \mathsf{OPT}=2^i+\varepsilon
\end{aligned}
$$

滑雪橇租赁问题

- 寒假，班长带我们全班同学去雪山滑雪，可以一直玩到冰雪融化。山上有个划雪橇店，**租雪橇一天1块钱**，**买要T块钱**。你不知道什么时候雪融化，每天早晨你决定租还是买。
- 问：如何决策支出最小？

滑雪橇租赁问题-算法

- 离线算法：一开始就知道哪一天雪融化
  - 如果剩余天数<T，每天租；
  - 否则，一开始就买

- 在线算法
  - 在前k天租雪橇，第k+1天买下雪橇
  - 需要确定k的大小，使得开支最小

滑雪橇租赁问题-在线算法

- 土豪在线算法
  - 第一天就买划雪橇
- 结论：该策略具有*T-Competitive*

滑雪橇租赁问题-在线算法

- 假设L天后雪融化
- 当L=1时
  - 最优离线算法$C_{OPT}(1)=1$
  - 土豪在线算法$C_{ON}(1)=T$
- 这是最坏情况，因为L>1时
  - $C_{OPT}(k)>1$ 而 $C_{ON}(k)=T$
- 所以，我们有

$$\forall k, C_{ON}(k) \leq T \cdot C_{OPT}(k)$$

滑雪橇租赁问题-在线策略

- 租滑雪橇直到第T+1天，买下雪橇
- 结论：该策略具有2-*Competitive*

滑雪橇租赁问题-在线策略

- **在第T+1天买下雪橇具有2倍竞争比**
- **证明：**

$L < T$: $C_{ON} = C_{OPT}$.

$L \geq T$: $C_{ON} = 2T$

$C_{OPT} = T$

$\Downarrow$

$$\forall k, \quad C_{ON}(k) \big/ C_{OPT}(k) \leq 2$$

最坏的情况发生在
第T+1天是最后一天

■

开拓思维

- 如果有多家滑雪橇商店，每一家的租赁价格和购买价格各不相同。
- 决策何时从哪家买？

SIGMETRICS 2014

公交还是走路？

- 东大九龙湖地铁站到学校内梅园宿舍，可以坐接驳车也可以走路，坐接驳车需要B=10分钟，走路需要F=30分钟。假设等接驳车的时间不可预测。
- 问：如何回学校用时最短？

离线策略

- 如已知接驳车F-B=20分钟内到达，最好等车
- 如已知接驳车F-B=20分钟后到达，最好走路

- 但是，如果完全不知道呢？

在线策略

- 你决定：先等车F-B=20分钟，如果没等到，你就走回学校。
- 分析你的策略：
  - 情况1：车F-B=20分钟内到，乘车回，策略最优
  - 情况2：等F-B=20分钟后，走回学校，你用2F-B=50分钟，最优解OPT用F=30分钟。比例为$2-\frac{B}{F}$

页面调度问题

- 问题：
  - 高速缓存可放k个页面；低速内存有多个页面
  - 给定一个页面请求序列$< p_1, p_2, \cdots, p_n >$，当高速缓存占满的情况下，在高速缓存出现页面缺失时，选择哪个页面与低速内存页面交换，使得遇到缺失的次数最少

页面调度问题

- LIFO(Last In First Out,后进先出法)
  - 将最后放入高速缓存的页面交换出去
  - 该方法不是 α-competitive算法
  - 例如页面请求序列为<a,b,a,b,a,b>,
    - 假设高速缓存中页面$x_1, x_2, \cdots, x_k$不含a,b
    - LIFO中，每次a被交换进马上又被交换出，b也一样
    - 最优解OPT里只需将a,b与高速缓存中2页面交换

页面调度问题

- LFU (Least Frequently Used,最少访问法)
  - 将高速缓存中访问次数最少的页面交换出去
  - 该方法不是 α-competitive算法
  - 例如页面请求序列为<a,b,a,b,a,b>,
    - 假设高速缓存中页面$x_1, x_2, \cdots, x_k$访问次数均大于1,且不含a,b
    - LFU中，每次a被交换进马上又被交换出，b也一样
    - 最优解OPT里只需将a,b与高速缓存中2页面交换

页面调度问题

- LRU (Least Recently Used,最旧访问法)
  - 将高速缓存中最早访问的页面交换出去
  - k=3，页面请求序列<4,3,4,2,3,1,4,2>，LRU过程
  - 4
  - 4 3
  - 3 4
  - 3 4 2
  - 4 2 3
  - 2 3 1
  - 3 1 4
  - 1 4 2

页面调度问题

- LRU (Least Recently Used,最旧访问法)
  - 将高速缓存中最早访问的页面交换出去
  - 该方法是 k-competitive算法
  - 证明：
    - 将LRU算法调度过程分为多个阶段$< s_1, s_2, \cdots, s_m >$，第1个阶段页面缺失数不多于k，后面所有阶段页面缺失数等于k
    - 只要证明OPT中每一阶段至少需要交换1次，就可得证明k-competitive
      - 否则的话，第i阶段，OPT所有页面都在cache中，则LRU不可能缺失数等于k次