



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 $A \leq \alpha \cdot OPT + c$ ，
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。

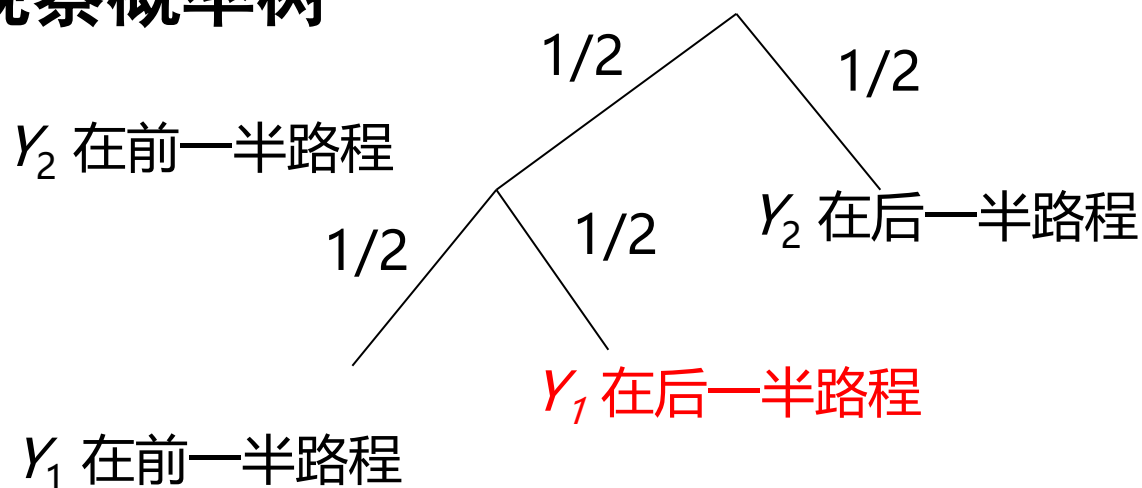
摘麦穗问题-在线算法证明

- 证明:

Y_1 – 最大麦穗.

Y_2 – 第二大麦穗.

- 观察概率树



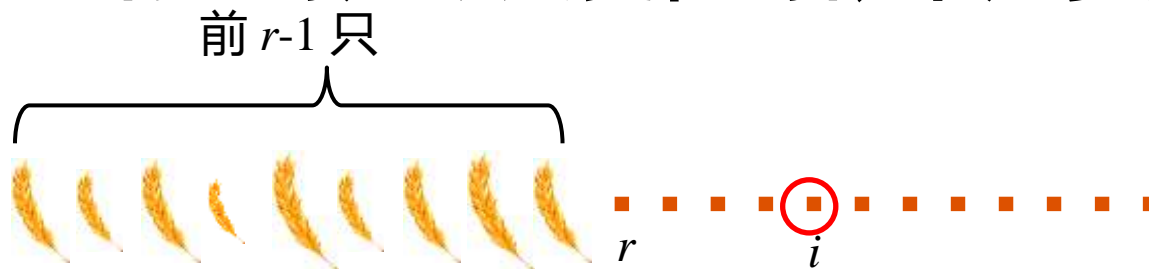
摘麦穗问题

- 摘下最大的麦穗 \leftrightarrow 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？



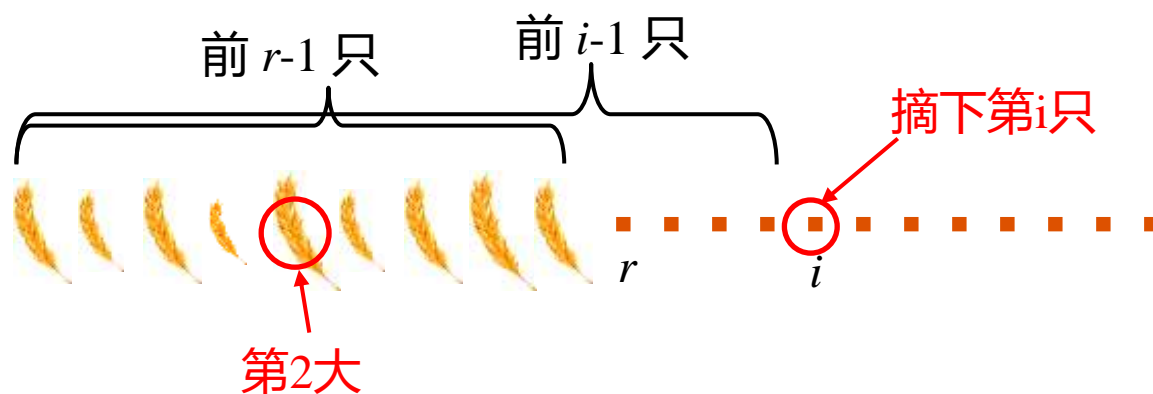
概率是多少

- 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



$$\begin{aligned} P_r &= \sum_{i=r}^n P(i) = \sum_{i=r}^n P(\text{选中第 } i \text{ 只}) \times P(\text{第 } i \text{ 只最大}) \\ &= \sum_{i=r}^n \frac{r-1}{i-1} \times \frac{1}{n} = \frac{r-1}{n} \sum_{i=r}^n \frac{1}{i-1} \end{aligned}$$

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 $2^i + \epsilon$ 就找到缺口

牛寻路问题

- 但是在线算法需要走过的距离为：

$$2(1 + 2 + \cdots + 2^{i+1}) + 2^i + \varepsilon$$

$$= 2(2^{i+2} - 1) + 2^i + \varepsilon$$

$$< 8 \cdot 2^i + 2^i + \varepsilon$$

$$< 9 \cdot \text{OPT} \quad \text{OPT} = 2^i + \varepsilon$$

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

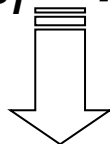
- 租滑雪橇直到第 $T+1$ 天，买下雪橇
- 结论：该策略具有2-Competitive

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:

$$\begin{array}{ll} L < T: & C_{ON} = C_{OPT} \\ L \geq T: & C_{ON} = 2T \\ & C_{OPT} = T \end{array}$$



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

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

The Multi-shop Ski Rental Problem

Lingqiang Ai
HIS, Tsinghua University
ailingqiang@126.com

Xian Wu
HIS, Tsinghua University
wuxian12@mao.tsinghua.edu.cn

Lingxiao Huang
HIS, Tsinghua University
huanglingxiao1990@126.com

Longbo Huang
HIS, Tsinghua University
longbohuang@tsinghua.edu.cn

Pingzhong Tang
HIS, Tsinghua University
keshin@tsinghua.edu.cn

Jian Li
HIS, Tsinghua University
lijian83@tsinghua.edu.cn

ABSTRACT

We consider the multi-shop ski rental problem. This problem generalizes the classic ski rental problem to a multi-shop setting, in which each shop has different prices for renting and purchasing a pair of skis, and a consumer has to make decisions on when and where to buy. We are interested in the optimal online (nonclairvoyant) algorithmic strategy from the consumer's perspective. For our problem, in its basic form, we obtain existing closed-form solutions and a linear time algorithm for computing them. We further demonstrate the generality of our approach by investigating three extensions of our basic problem, namely ones that consider costs incurred by entering a shop or switching to another shop. Our solutions to these problems suggest that the consumer need design policies proactively to control one shop at any buying time. Our results apply to many real-world applications, ranging from route management in taxi dispatch to scheduling in distributed computing.

Categories and Subject Descriptors

G.3 [Probability and Statistics]: Distribution functions; F.2 [Analysis of Algorithms and Problem Complexity]: Miscellaneous

General Terms

Algorithms, Performance, Theory

Keywords

multi-shop ski rental, ski rental, optimal strategy, dynamic programming, online algorithm

1. INTRODUCTION

The ski rental problem (SR) is a dilemma faced by a consumer, who is uncertain about how many days she will ski

and has to trade off between buying and renting skis: once she buys the skis, she will enjoy the remaining days rent-free, but before that she must pay the daily renting cost. The literature is interested in investigating the online optimal strategy of the consumer. That is, a strategy that yields the lowest competitive ratio without having any information of the future (as is standard in the literature, competitive ratio is defined as the ratio between the cost yielded by the consumer's strategy and the cost yielded by the optimal strategy of a prophet, who knows how many days the trip will last and designs the optimal strategy accordingly). The ski rental problem and its variants constitute an important part of the online algorithm design literature from both theoretical and applied perspectives [8, 12, 28, 17, 16, 32].

In this paper, we consider the multi-shop ski rental problem (MSR), in which the consumer faces multiple shops that offer different renting and buying prices. The most challenging step immediately after she arrives at the ski field and must rent or buy the skis is that particular shop since then. In other words, once she has chosen a shop, the only decision variable is when to buy the skis. Beyond the basic setting, we also propose three important extensions of MSR as below:

- MSR with switching cost (MSR-S): The consumer is allowed to switch from one shop to another and such switching costs her some constant amount of money.
- MSR with entry fee (MSR-E): Each shop requires some entry fee and the consumer must enter each shop.
- MSR with entry fee and switching (MSR-SE): The consumer is able to switch from one shop to another, and she pays the entry fee as long as she enters any shop.¹

In all the settings above, the consumer's objective is to minimize the competitive ratio. In MSR and MSR-E, she has to consider two questions at the very beginning: (1) where should she rent or buy the skis (once), and (2) when should she buy the skis (during)? While MSR-S and MSR-SE allow the consumer to switch shops and use that more fine-grained than the previous two, in the sense that she is able to decide where to rent or buy the skis at any time. For example, if it is among her options to rent in shop 1 on day 1, switch to shop 2 from day 2, and finally switch to shop 3 and then buy the skis.

¹For example, if she switches from shop 1 to shop 2, and then switches back to shop 1, she pays the entry fee of shop 1 twice and the entry fee of shop 2 once.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. No copying, storing, or distributing in any form, or by any means, is permitted. This article is intended solely for the personal use of the individual user and is not to be disseminated broadly. SIGMETRICS '14, June 16-20, 2014, Austin, Texas, USA. Copyright 2014 ACM 978-1-4558-1071-1/14/06...\$5.00. http://dx.doi.org/10.1145/2591071.2591084

公交还是走路？

- 东大九龙湖地铁站到校内梅园宿舍，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 $\langle p_1, p_2, \dots, p_n \rangle$ ，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 $\langle a, b, a, b, a, b \rangle$,
 - 假设高速缓存中页面 x_1, x_2, \dots, 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 $\langle a, b, a, b, a, b \rangle$,
 - 假设高速缓存中页面 x_1, x_2, \dots, 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 $\langle 4, 3, 4, 2, 3, 1, 4, 2 \rangle$, 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 $\langle s_1, s_2, \dots, s_m \rangle$ ，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

