浙江大学 ZheJiang University

组合优化

浙江大学数学系　谈之奕

在线问题

Ski-Rental

浙江大学 ZheJiang University

组合优化

- 计划参加滑雪运动，获取装备有两种途径，采用哪种方式可使支出较少
  - 租：租金为 1 ，仅限一次使用
  - 买：价格为 $N(>>1)$ ，可无限次使用
- 若参加滑雪次数 n 已知，最少支出为

$$C^*(n)=\begin{cases} n & n<N, \quad \text{每次滑雪时租用} \\ N & n\geq N \quad \text{首次滑雪时购买} \end{cases}$$

- 若滑雪次数 n 未知，每次滑雪时应如何决策，可使实际支出与最少支出的比值最小
  - 永不购买 ×
  - 多次购买 ×

$\underbrace{\text{租租租……租}}_{k}\text{买}$ $\qquad C(n)=k+N$

浙江大学 ZheJiang University

组合优化

可行策略

- 策略 N
  - 在第 $n, n<N$ 次滑雪时租用装备；若滑雪次数不少于 N，在第 N 次滑雪时购买装备，以后每次滑雪时不再租用
  - 滑雪次数为 n 时采用策略 N 所需支付的总费用

$$C^N(n)=\begin{cases} n & n<N, \\ N-1+N=2N-1 & n\ge N \end{cases} \qquad \frac{C^N(n)}{C^*(n)}=\begin{cases} 1 & n<N \\ 2-\dfrac{1}{N} & n\ge N \end{cases}$$

  - 不论滑雪次数为何，策略 N 的费用不会超过最优费用的 $2-\frac{1}{N}$ 倍

$$C^*(n)=\begin{cases} n & n<N, \\ N & n\ge N \end{cases}$$

是否存在更好的策略？

（最坏情况意义下）

浙江大学
ZheJiang University

组合优化

策略比较

- 策略 k
  - 在第 $n, n<k$ 次滑雪时租用装备；若滑雪次数不少于 k，在第 k 次滑雪时购买装备，以后每次滑雪时不再租用
- 在最坏情况意义下，任意策略不会优于策略 N
  - $k<N$：$\dfrac{C^k(k)}{C^*(k)}=\dfrac{k-1+N}{k}=1+\dfrac{N-1}{k}>1+\dfrac{N-1}{N}=2-\dfrac{1}{N}$
  - $k>N$：$\dfrac{C^k(k)}{C^*(k)}=\dfrac{k-1+N}{N}>\dfrac{2N-1}{N}=2-\dfrac{1}{N}$

$$C^*(n)=\begin{cases} n & n<N, \\ N & n\geq N \end{cases}$$

- 策略 N 是最好（**optimal, best possible** ）策略

不表示采用该策略必在任何情况下都能找到最优解，而是指没有别的策略（在最坏情况意义下）比它更好

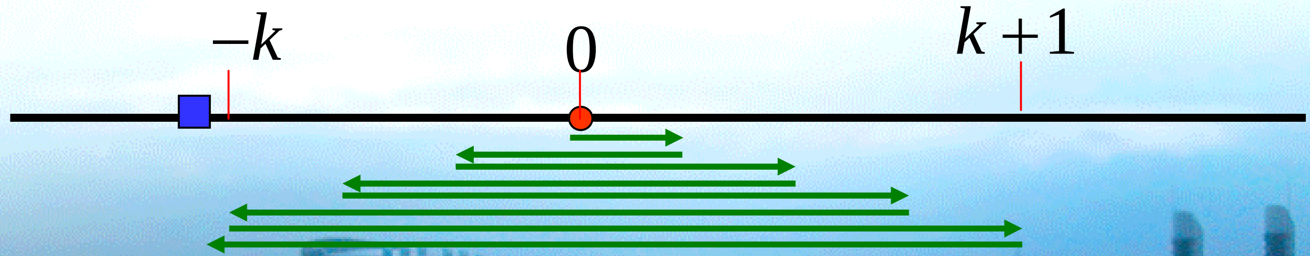
在一次性大额支出与经常性小额支出之间进行选择时，应推迟大额支出直至小额支出累计值与之接近时

搜索问题

- 某一物品遗失在两条不同方向无限延伸的道路中某处，但不知道所在道路和确切位置
- 从两条道路的交汇点出发寻找，何种搜索方案最好
  - 以算法走过的路程与最优路程（物品遗失处距交汇点的距离）的比值作为衡量算法优劣的标准
  - 假设物品遗失处与交汇点距离至少为 1
- 可行的方案是在两条道路上不断折返搜索，并且搜索半径逐步扩大
- 以交汇点为原点，每次折返时所处的位置构成一交错数列

算法 1

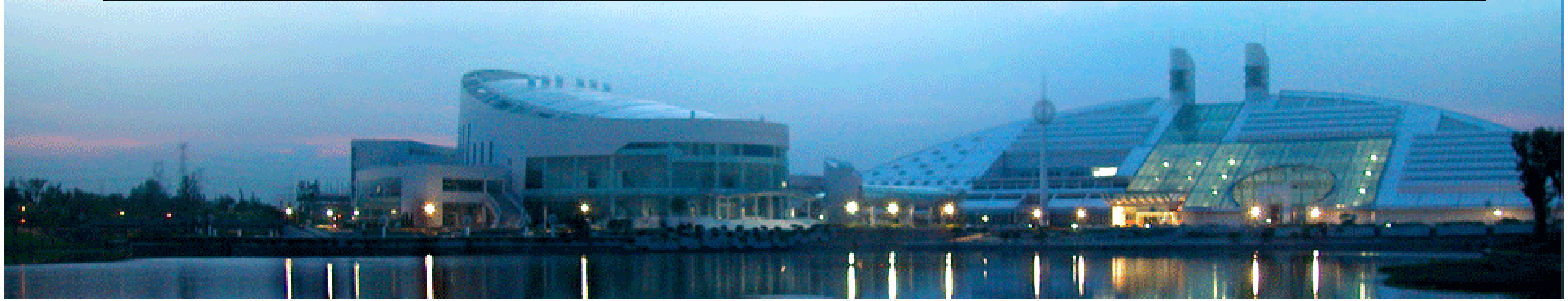
交错数列	**(1, -1, 2, -2, 3, -3,……)** 等差数列
物品位置	$-(k+\varepsilon)$
走过的路程	$4+8+\cdots+4k+2(k+1)+(k+\varepsilon)$ $=2k^2+5k+2+\varepsilon$
比值	$>2k+5$



算法 2

组合优化

交错数列	**(1, -2, 3, -4, 5, -6,……)** 等差数列
物品位置	$-(2k+\varepsilon)$
走过的路程	$(2+4)+(6+8)+\cdots$ $+(2(2k-1)+4k)$ $+2(2k+1)+(2k+\varepsilon)$ $=4k^2+8k+2+\varepsilon$
比值	$>2k+4$



算法 3

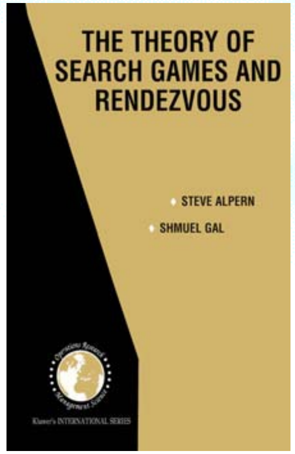
交错数列	**(1, -2, 4, -8, 16, -32,……)**	等比数列
物品位置	$-(2^{2k-1}+\varepsilon)(0<\varepsilon\le 3\cdot 2^{2k-1})$	$2^{2k-2}+\varepsilon(0<\varepsilon\le 3\cdot 2^{2k-2})$
走过的路程	$(2+4)+(2\cdot 2^2+2\cdot 2^3)$ $+\cdots+(2\cdot 2^{2k-2}+2\cdot 2^{2k-1})$ $+2\cdot 2^{2k}+2^{2k-1}+\varepsilon$ $=9\cdot 2^{2k-1}-2+\varepsilon$	$(2+4)+(2\cdot 2^2+2\cdot 2^3)$ $+\cdots+(2\cdot 2^{2k-2}+2\cdot 2^{2k-1})$ $+2^{2k-2}+\varepsilon$ $=9\cdot 2^{2k-2}-2+\varepsilon$
比值	≤ 9	≤ 9

可行算法

组合优化

- 搜索问题任意算法的比值不会总小于9
  - 任一合理算法可用数列$\{f_i\}$表示，其中
    $0 < f_i < f_{i+2}, i \geq 1$
  - 设物品位于距交汇点 $f_i + \varepsilon$处
  - 算法走过的路程 $C^A = 2\sum_{j=1}^{i+1} f_j + f_i + \varepsilon$
  - $$\rho \geq \frac{C^A}{C^*} = \frac{2\sum_{j=1}^{i+1} f_j + f_i + \varepsilon}{f_i + \varepsilon}$$
    $$\Rightarrow \rho \geq \frac{2\sum_{j=1}^{i+1} f_j + f_i}{f_i} = 1 + 2\frac{\sum_{j=1}^{i+1} f_j}{f_i}, i \geq 1$$
  - 若 $\rho < 9$， 可导出矛盾

Doubling Method



Gal S. *Search Games*. Academic Press, 1980
Alpern S, Gal S. *The Theory of Search Games and Rendezvous*. Springer, 2006.

下界

- 令 $y_i = \dfrac{\sum_{j=1}^{i+1} f_j}{\sum_{j=1}^{i} f_j}, i \ge 1$，$1 \le y_i = \dfrac{\sum_{j=1}^{i+1} f_j}{\sum_{j=1}^{i} f_j} \le \dfrac{\frac{\rho-1}{2} f_i}{\sum_{j=1}^{i} f_j} = \dfrac{\rho-1}{2}\left(\dfrac{\sum_{j=1}^{i} f_j - \sum_{j=1}^{i-1} f_j}{\sum_{j=1}^{i} f_j}\right) = \dfrac{\rho-1}{2}\left(1-\dfrac{1}{y_{i-1}}\right)$

$$\Rightarrow (y_i - y_{i-1})y_{i-1} \le -y_{i-1}^2 + \frac{\rho-1}{2} y_{i-1} - \frac{\rho-1}{2}$$

- 若$\rho < 9$，则$\left(\dfrac{\rho-1}{2}\right)^2 - 4\dfrac{\rho-1}{2} < 0$，数列$\{y_n\}$单调递减，$\lim\limits_{n\to\infty} y_n$存在
- 两端取极限 $0 = (y-y)y \le -y^2 + \dfrac{\rho-1}{2} y - \dfrac{\rho-1}{2}$

$$\boxed{\rho \ge 1 + 2\frac{\sum_{j=1}^{i+1} f_j}{f_i}, i \ge 1}$$

矛盾

Zhejiang University

组合优化

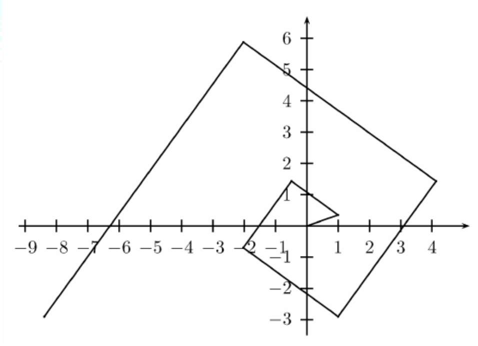
二维搜索

- 在二维平面上搜索一条直线
  - 在二维平面上搜索一条与 x 轴平行的直线 9
  - 在二维平面上搜索一条与坐标轴平行的直线 $\leq 9\sqrt{2}$
- 在二维平面上搜索一点
  - 该点位于连接搜索者当前所在点与原点的线段上

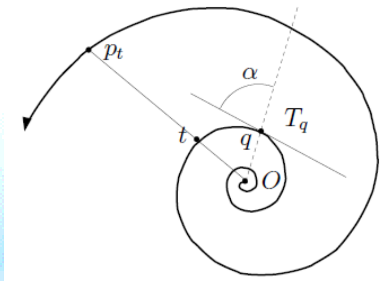
对数螺线

$$\left(e^{\theta\cot(\alpha)}, \theta\right)$$

13.81113



12.5406 12.5385



17.289

Langetepe E. Searching for an axis-parallel shoreline. *Theoretical Computer Science*, 447, 85-99, 2012.

Langetepe E. On the optimality of spiral search. *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, 1-12, 2012.

在线问题

- 在线问题（**online problem**）：决策时未掌握全部实例信息，已做的决策在更多信息呈现后不可更改
  - 在线装箱问题：决定物品 j 所放箱子时不知道物品 $l(l \geq j)$的大小（或是否存在）
- 离线问题（**offline problem**）：实例信息在决策前全部已知的问题

Robert Endre Tarjan
(1948-)
美国计算机科学家
1986年**Turing**奖得主
1982年首届
Nevanlinna奖得主

RESEARCH CONTRIBUTIONS

Programming Techniques and Data Structures

Ellis Horowitz Editor

Amortized Efficiency of List Update and Paging Rules

DANIEL D. SLEATOR and ROBERT E. TARJAN

解决问题的困难并不一定来自于计算资源的缺乏，而源自信息的缺乏

Sleator DD, Tarjan RE, Amortized efficiency of list update and paging rules. ***Communications of the ACM***，**28, 202-208, 1985**

浙江大学
ZheJiang University
组合优化

Paging问题

- **Paging问题**
  - 计算机的存储系统分为缓存（**cache**）和内存，缓存容量为 k
  - 如果程序拟读取的数据恰在缓存中，则可直接读取；若不在缓存中，需将数据从内存读入缓存。如果此时缓存全被占据，则需将某个缓存中的数据转出
  - 给出一缓存数据管理规则，对应用程序发出的一系列请求，读入缓存的次数能尽可能少
- 策略
  - 离线最优策略：**Longest Forward Used**（将来最晚使用的）
  - 在线最好策略：**Least Recently Used**（闲置时间最长），**FIFO** k
  - 其他在线策略：**Least Frequently Used**（使用频率最低），**LIFO** ∞

$k-$server问题

组合优化

- $k-$**server问题**
  - X为度量空间中的点集。有k台服务器，位于X中某些点上，且可在X中的点之间移动
  - 现有一需求序列，序列中每个需求发生于X中某个点上。至少有一台服务器与该需求处于同一位置时方能满足该需求
    - 如果当前需求发生处没有服务器，则需有至少一台服务器移动到此处
  - 希望在满足所有需求的前提下，使服务器移动的总距离尽可能小
- $k-$**server猜想：最好策略下服务器移动的总距离与（离线）最优距离的比值为** k
  - **Paging问题为** $k-$**server问题的特殊情况**

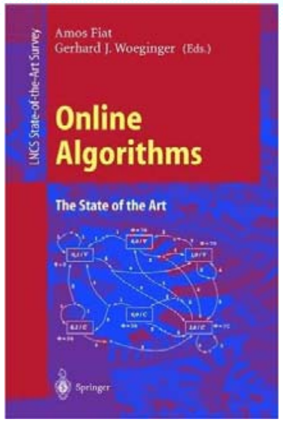
数据集	缓存	读入费用
X	服务器	距离

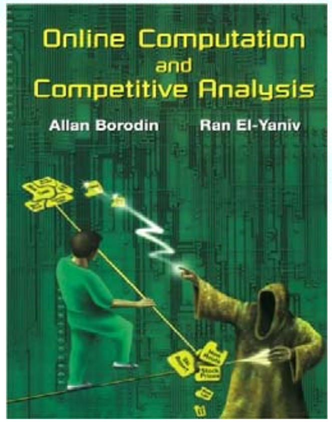
Koutsoupias E. The k-server problem. *Computer Science Review*, 3: 105-118, 2009.

在线算法

- 在线算法（**online algorithm**）：可用于求解在线问题的算法
  - **FF**，**LS**算法等是在线算法，**FFD**，**LPT**等算法不是在线算法
  - 在线问题决策和信息显现是逐步进行的，可以形象地将此视作算法设计者和一个对手（**adversary**）不断竞争的过程
- 记 $C^A(I)$ 为算法 A 求解实例 I 所得的目标函数值，$C^*(I)$ 为实例 I 在信息完全已知情况下（即相应离线问题）的最优目标值，称
  - （极小化问题）$c_A = \inf\{r \geq 1 | C^*(I) \leq rC^A(I), \forall I\}$
  - （极大化问题）$c_A = \inf\{r \geq 1 | C^A(I) \leq rC^*(I), \forall I\}$

为在线算法 A 的竞争比（**competitive ratio**）





Fiat, A., Woeginger, G., (Eds.) ***Online Algorithms: The State of the Art*****, Springer, 1998.**

Borodin A, El-Yaniv R. ***Online Computation and Competitive Analysis*****. Cambridge University Press, 2005.**

竞争比分析

- 由于在线问题决策时信息的不完全性，可能出现任何（多项式时间或指数时间）算法均不能找到所有实例的最优解的情况
- 若任一在线算法求解某问题的竞争比至少为 ρ，则称该在线问题的下界（**lower bound**）为 ρ
- 若一在线问题的下界为 ρ, A是该问题竞争比为 ρ的在线算法，则称该算法为最好（**optimal**，**best possible**）算法
  - 设计最好算法是一在线问题研究的最终目标，它必须与问题下界的证明同时进行
- 通过算法设计和下界证明寻求最好算法的在线问题研究方法称为竞争比分析（**competitive analysis**）

下界

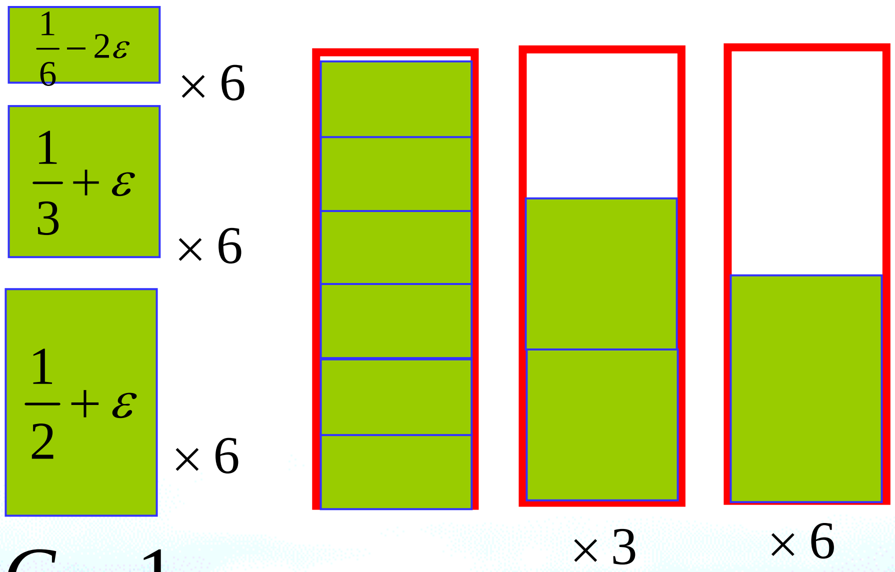
下界类型	产生原因	证明方法	举例
离线（在线）算法最坏情况界（竞争比）的下界	算法自身的局限性	给出某个实例	**FF**算法的绝对性能比至少为$\frac{17}{10}$
（部分）离线问题近似算法的下界	计算时间的限制	归约（**Gap**技术，**PCP**定理）	任一装箱问题算法的绝对性能比至少为$\frac{3}{2}$
在线问题竞争比的下界	信息的不完全性	构造一系列的实例，穷举所有可能的在线算法	任一在线装箱问题算法的绝对性能比至少为$\frac{5}{3}$

一维在线装箱

浙江大学 ZheJiang University

组合优化

- 任一在线算法求解一维在线装箱问题的最坏情况界至少为 $\frac{5}{3}$



$C=1$

$C^*=1 \quad C^A \geq 2$

$\frac{C^A}{C^*} \geq 2 > \frac{5}{3}$ ×

$C^*=3 \quad C^A \geq 5$

$\frac{C^A}{C^*} \geq \frac{5}{3}$ ×

$C^*=6 \quad C^A=10$

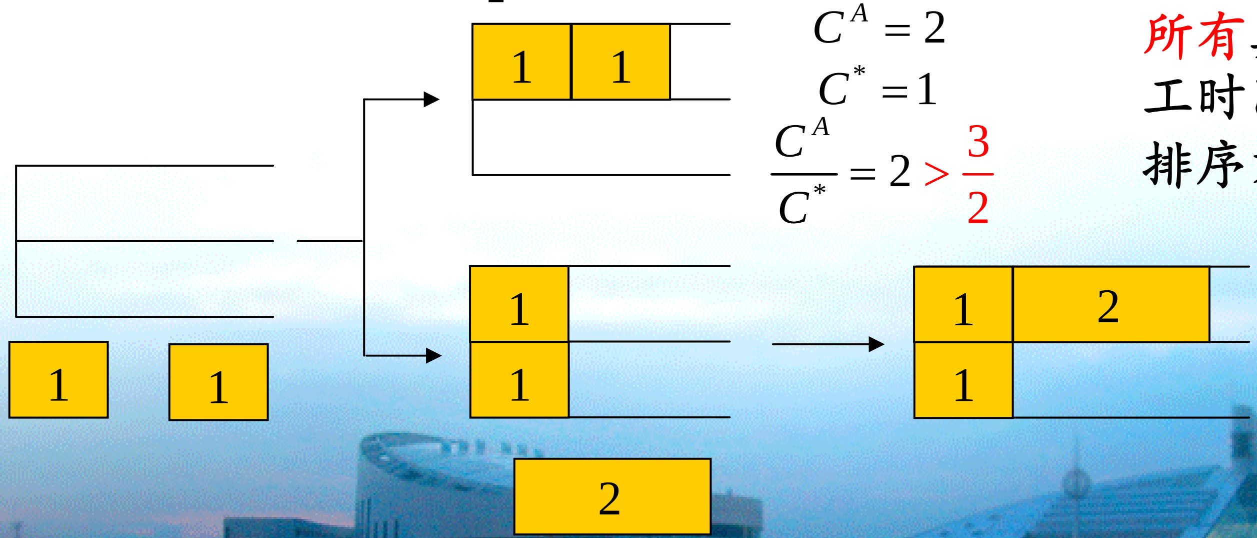
$\frac{C^A}{C^*} = \frac{5}{3}$

Balogh J, Békési J, Dósa G, Sgall J, van Stee R. The optimal absolute ratio for online bin packing. *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1425-1438, 2015.

平行机排序

组合优化

- 平行机排序的在线形式
  - 工件按顺序一个个到达，在确定工件 J_j 的加工机器时只知道 $J_1,\cdots,J_j$ 的加工时间，而不知道未到达工件 $J_{j+1},\cdots,J_n$ 的加工时间
  - 工件的加工机器一旦指定就不能改变
- $P2\|C_{\max}$ 的下界为 $\frac{3}{2}$



$C^A=2$

$C^*=1$

$\frac{C^A}{C^*}=2>\frac{3}{2}$

所有算法对两个加工时间为 1 的工件排序方式只有两类

$C^A=3$

$C^*=2$

$\frac{C^A}{C^*}=\frac{3}{2}$

浙江大学 ZheJiang University

组合优化

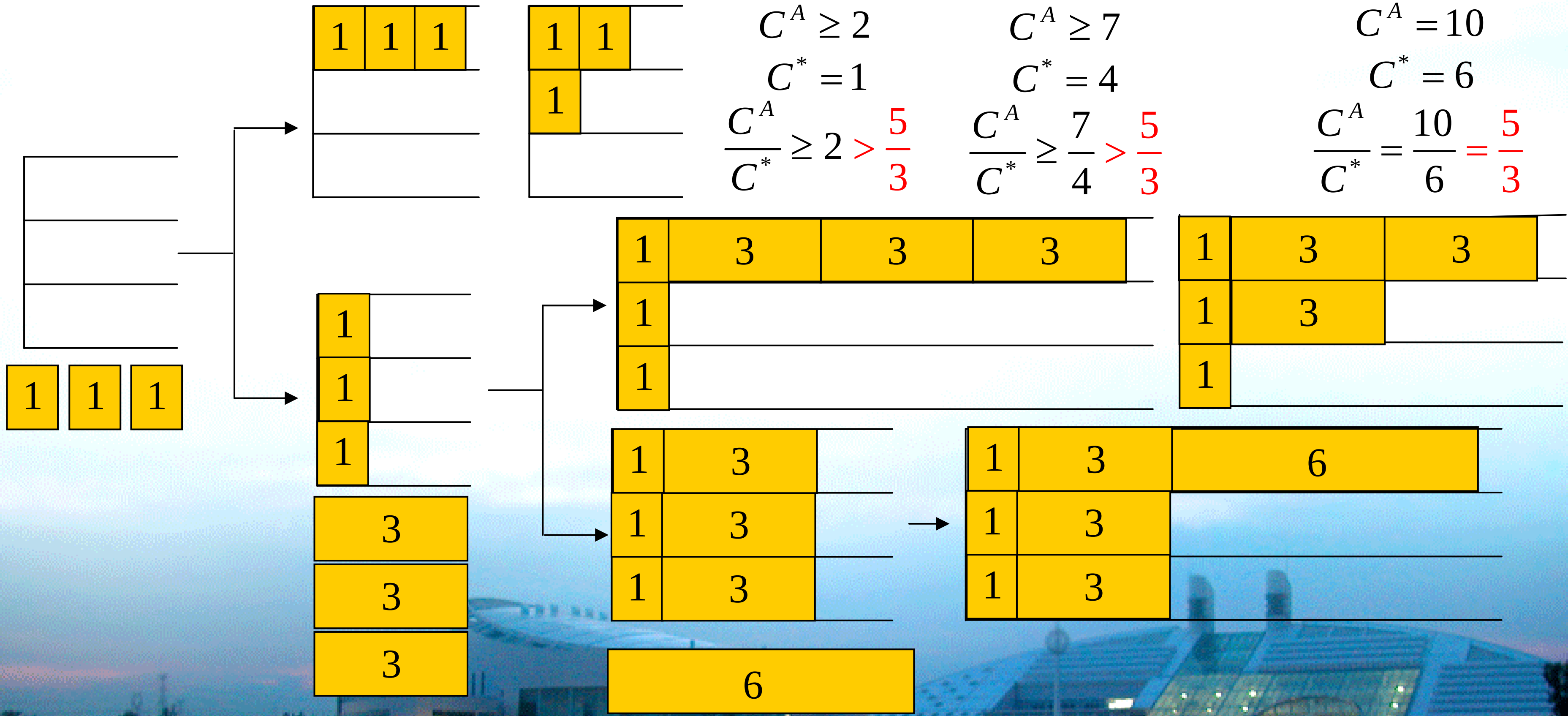
平行机排序

- $P3||C_{\max}$ 的下界为 $\frac{5}{3}$

$$C^A \geq 2,\quad C^* = 1,\quad \frac{C^A}{C^*} \geq 2 > \frac{5}{3}$$

$$C^A \geq 7,\quad C^* = 4,\quad \frac{C^A}{C^*} \geq \frac{7}{4} > \frac{5}{3}$$

$$C^A = 10,\quad C^* = 6,\quad \frac{C^A}{C^*} = \frac{10}{6} = \frac{5}{3}$$



平行机排序

浙江大学 ZheJiang University

组合优化

- **LS算法是** $P2||C_{\max}$，$P3||C_{\max}$ **的最好算法**

LS算法是否是 $Pm||C_{\max}, m \ge 4$ **的最好算法**

$\frac{2+\sqrt{2}}{2} \approx 1.707$	1.837	1.852	1.85358	$1+\sqrt{\frac{1+\ln 2}{2}} \approx 1.9201$	1.923	1.945	$2-\frac{1}{70} \approx 1.985$	$2-\frac{1}{m}-\varepsilon_m$	$2-\frac{1}{m}$
1989	**1995**	**1999**	**2000**	**2000**	**1999**	**1996**	**1995**	**1993**	**1966**

Graham RL. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45，1563-1581.

Faigle U, Kern W, Turan G, On the performance of online algorithms for partition problems, *Acta Cybernetica*, 9, 107-119, 1989.

Galambos G, Woeginger G, An on-line scheduling heuristic with better worst case ratio than Graham's list scheduling, *SIAM Journal on Computing*, 22, 349-355, 1993.

Bartal Y, Fiat A, Karloff H, Vohra R, New algorithms for an ancient scheduling problem, *Journal of Computer and System Science*, 51, 359-366, 1995.

Karger DR, Phillips SJ, Torng E. A better algorithm for an ancient scheduling problem. *Journal of Algorithms*, 2, 400-430, 1996.

Albers S, Better bounds for online scheduling, *SIAM Journal on Computing*, 29, 459-473, 1999.

Fleischer R, Wahl M, On-line scheduling revisited, *Journal of Scheduling*, 3, 343-353, 2000.

Gormley T, Reingold N, Torng E, Westbrook J, Generating adversaries for request-answer games, *Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms*, 564-565, 2000.

$P4 \| C_{\max}$

- m 较小时，存在竞争比为 $\max\left\{\frac{4m^2-3m}{2m^2-2}, \frac{2(m-1)^2+\sqrt{1+2m(m-1)}-1}{(m-1)^2+\sqrt{1+2m(m-1)}-1}\right\}$ 的更好算法
- $P4 \| C_{\max}$ 的下界至少为 $\sqrt{3}$　　　　**Gap=0.001!**

	4	**5**	**6**	**7**	**8**
竞争比	$\frac{26}{15}\approx 1.7333$	$\frac{85}{48}\approx 1.7705$	$\frac{9}{5}=1.8000$	$\frac{175}{96}\approx 1.8229$	$\frac{116}{63}\approx 1.8413$
下界	**1.7320**	**1.7462**	**1.7730**	**1.7910**	**1.8007**

Chen B, van Vliet A, Woeginger G, New lower and upper bounds for on-line scheduling, *Operations Research Letters*, 16, 221-230, 1994.
Rudin III F, Chandrasekaran R, Improved bounds for the online scheduling problem, *SIAM Journal on Computing*, 32, 717-735, 2003.

谢 谢