从算法设计到机制设计

张国川

研究组成员：

叶德仕、陈林、程郁琨、余炜、罗文昌

中国运筹学会2012学术年会，沈阳，10月20日

Alan Turing (1912-1954)

- 图灵机
- 可计算性
- 计算的难解性

复杂性分类

- Cook (1971): 3-SAT 是NP-完全的
- Karp (1972): 21个NP-完全问题
- . . .

P ≠ NP

- 计算资源的匮乏
- 困难问题
  - 多项式时间近似算法
  - 指数时间的精确算法
  - 解的质量与时间成本的平衡

多项式时间 | 巨大的间隙 | 指数时间

多项式时间下的计算效能

- 近似比
- 近似方案(PTAS)
- EPTAS
- 完全近似方案(FPTAS)

$P \neq NP$下的效能极限

- 没有FPTAS
- 没有EPTAS
- 没有PTAS
- 近似比的下界

Cardinality Constraints

- We have m identical machines and n jobs
- Machine i can accept up to k_i jobs
- Goal: find a schedule meeting the cardinality constraints so that the makespan is minimized

A quick remark

- You may have in mind the 3-partition problem, from which we move further to k-partition, finally we arrive at k_i-partition that is exactly the problem we just introduced

Previously results

- K-partition

  4/3 [Bable, Kellerer, Kotov, 1998]

  7/6 for k=3, [Bable, Kellerer, 1999]
- K_i-partition

  FPTAS for fixed m [Woeginger 2005]

  3 [Zhang et al. 2009]

  2 [Barna, Aravind 2010] (more general)

  3/2 [Kellerer, Kotov, 2011]

Our contribution

EPTAS

Non-standard ILP formulation+ Best-Fit + Greedy Rounding

精确算法的计算效率

- 伪多项式时间
- 指数时间
- 双指数时间

指数运行时间假设
(Exponential Time Hypothesis)

- Impagliazzo, Paturi, and Zane (2001)
  存在 s>0，使得3-CNF-SAT没有如下时间的精确算法 $2^{sn}(n+m)^{O(1)}$.
- Lokshtanov, Marx, Saurabh(2011)
  Lower bounds based on the ETH

建立从3-SAT的线性规模的归约

另一个影响计算效能的因素

信息

- 不完全信息下的算法设计与分析
  - ✓ 在线算法
  - ✓ 鲁棒优化
  - ✓ 随机优化

算法博弈论
(AGT)

- 纳什均衡的计算及复杂性
- 算法机制设计
- 系统有效性分析

“稳定”装箱

- Input: a set of items $\{a_1, a_2, \ldots, a_n\}$, each of size at most 1, and an infinite number of identical bins
- Output: a feasible packing with a minimum number of bins used

换个角度

- There is one decision maker who does not take care of the own interest of each item
- What happened if every item is handled by an agent?
- What a feasible packing looks like in this case?

分摊机制

- How to design a sharing policy so that any stable packing is not far away from a social optimum?

- Proportional sharing (Bilo 2006)

- Identical sharing (Han et al. 2012)

“厌恶型”选址博弈

问题描述

- A residential area as a network

- The local government plans to build a garbage dump in this area, so that the location is as far away from the residents as possible, i.e., the total distance from the residents is maximized.

博弈元素

- The home address of every resident is private, which is unknown to the government
- The government publicize an "algorithm" to compute the location of the dump
- The residents report their home addresses

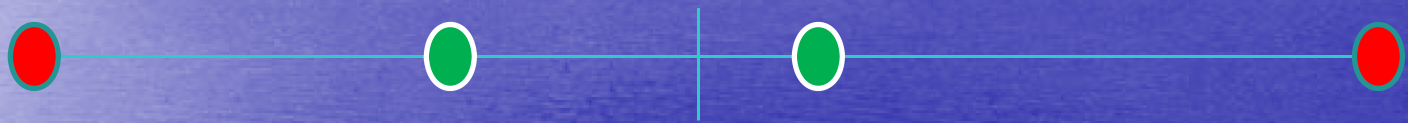
- A resident wants to keep away from the dump
- She has the incentive to lie if the wrong information can help

Goal of the government

An algorithm (truthful mechanism) :

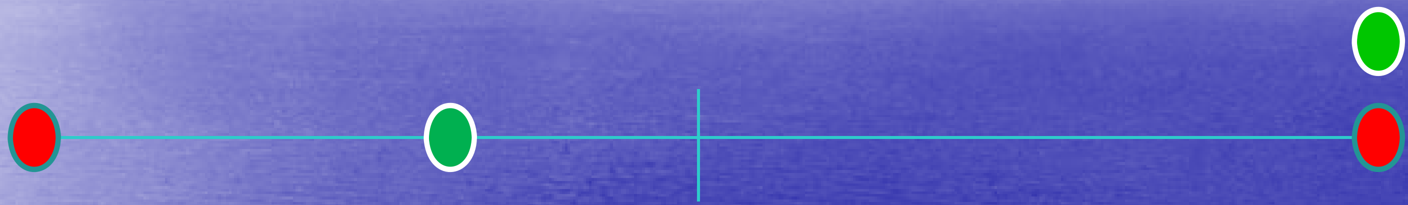
- Decide the facility location
- Every resident has no incentive to lie

Start: 线段



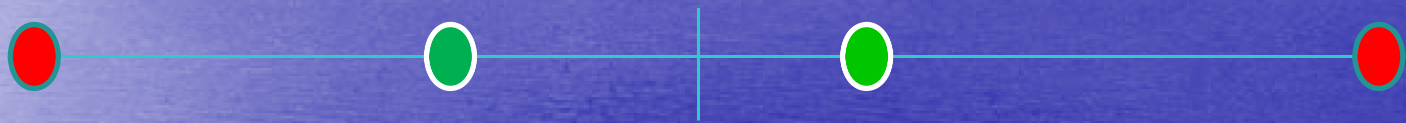
- The optimal algorithm is not truthful!

Start: 线段



- The optimal algorithm is not truthful!

Start: 线段



Left#　　Right#

Majority decides!

Group StrategyProof of 3-approximation

Start: 线段

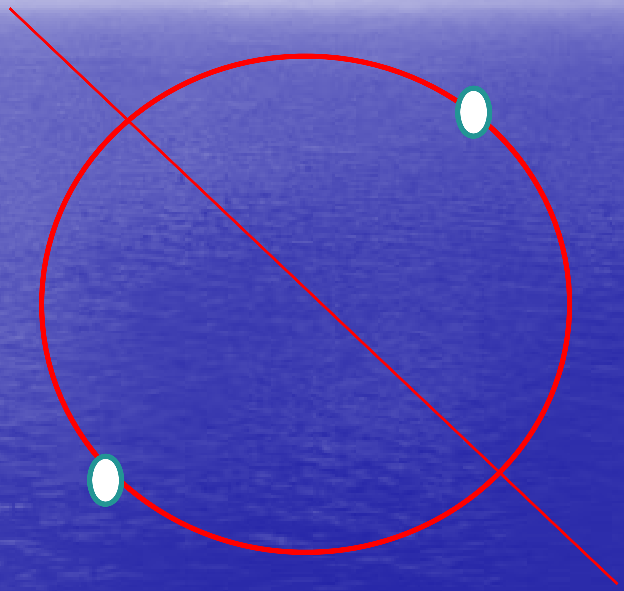
The mechanism is the best possible!

(Han and Du 2012, Ibara and Nagamochi 2012)

Start: 线段

Randomization helps: the ratio can be improved to 3/2.

圆环



GSP of 3-approximation

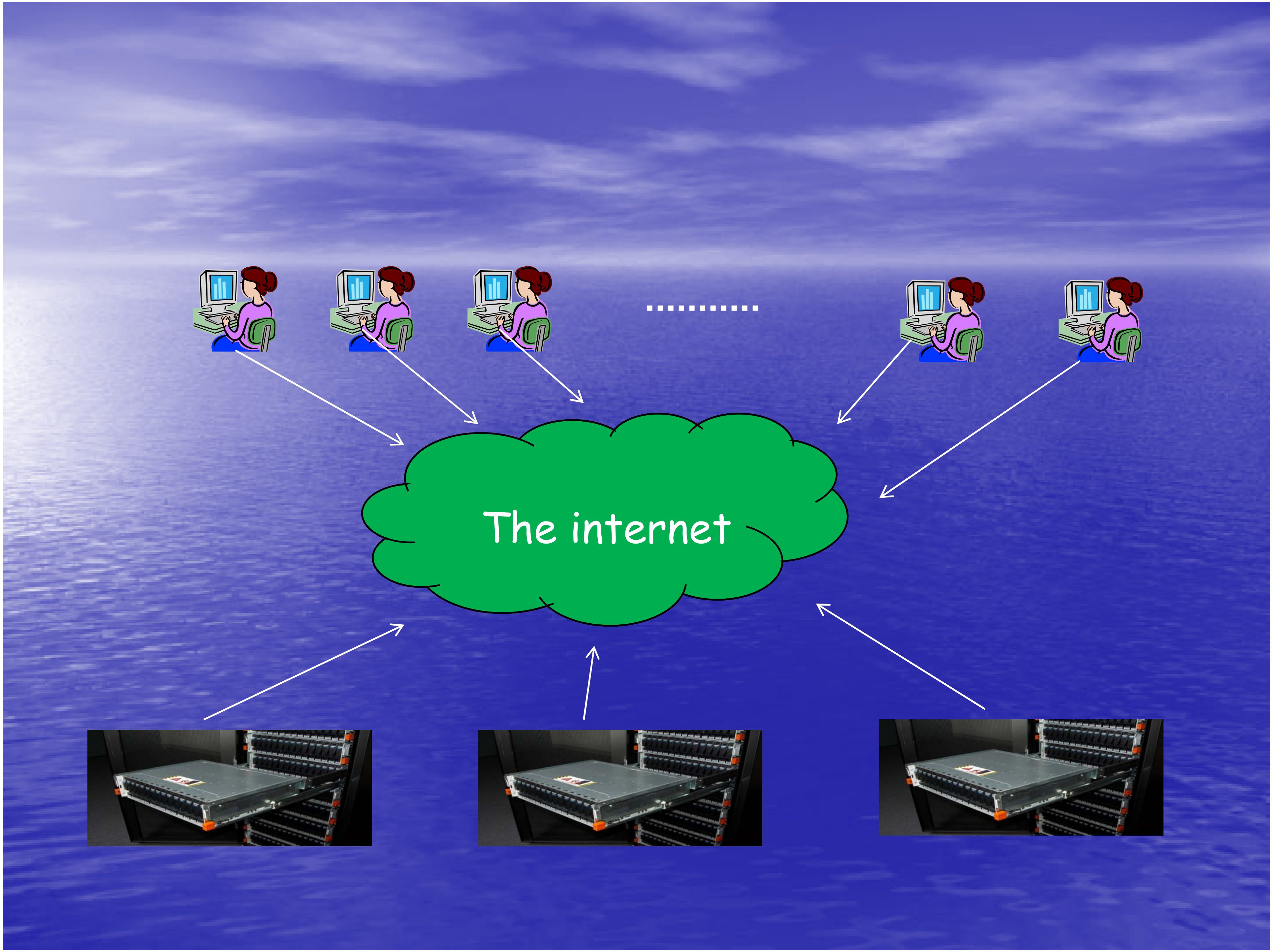
一般网络

GSP of 3-approximation for a tree and 4-approximation for a general network

进一步的研究

- 线段上随机GSP机制的下界
- 一般网络上的随机GSP机制设计
- 设施带服务半径下的机制设计

Efficiencies of Competing Schedulers


The internet

同一个世界，不同的梦想

模型

- Job (Client): minimize its completion time
- Machine (Service Provider): maximize its own revenue by processing jobs
- Game: machines claim the polices, jobs chooses machines

Scheduling Policies

- Largest (Shortest) Processing Times LPT(SPT)
- Makespan: all jobs on a machine has the same completion time (the total processing time)
- Random Order

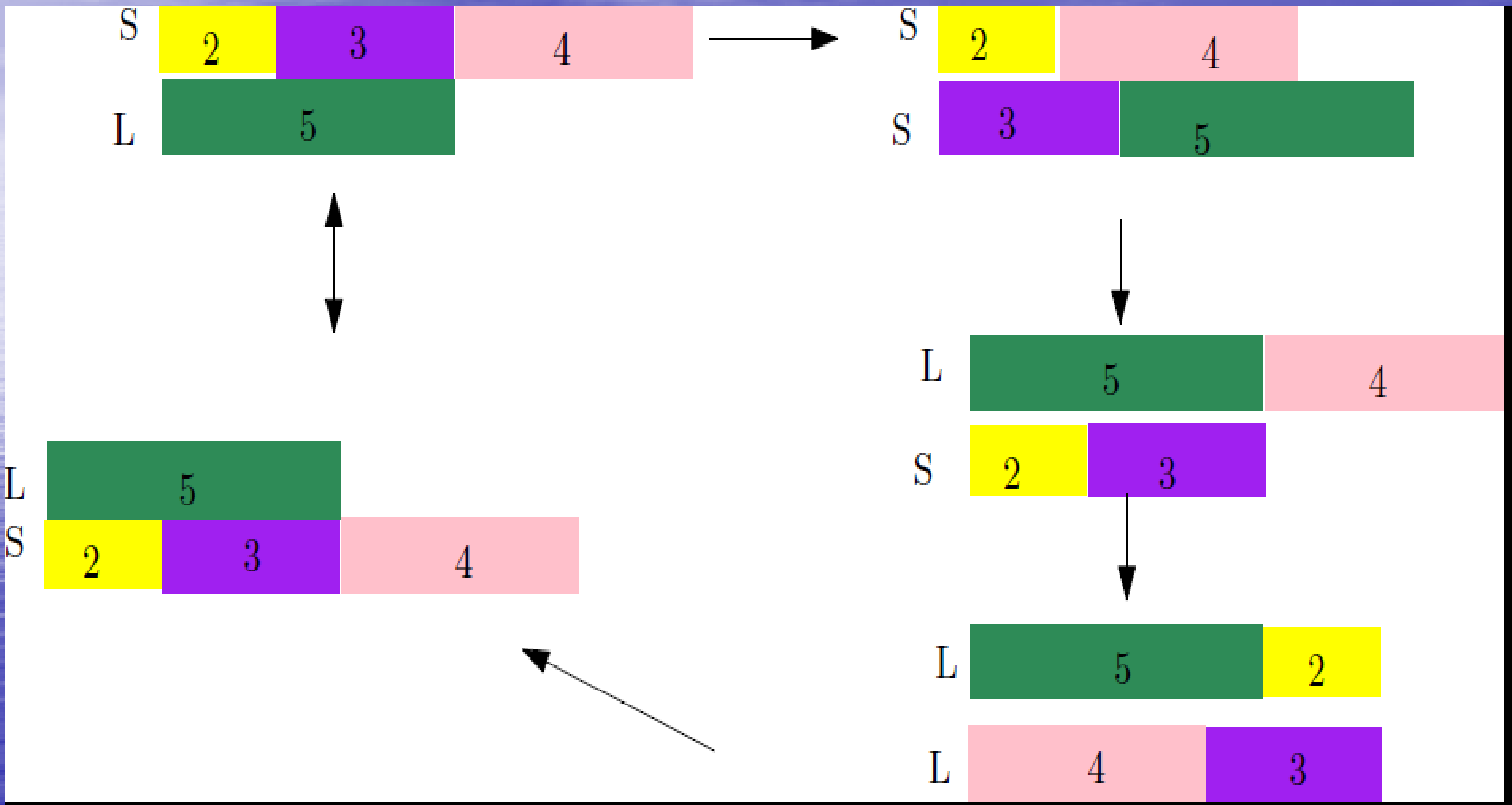
Goal

- Revenue of a machine: load
- Utility of a job: completion time
- Social objective: makespan

Incentive of a machine

- There may exist many equilibria
- A machine is interested in changing its policy if and only if all equilibria are better after the change.

Otherwise an NE may not exist!


S
2
3
4
L
5
S
2
4
S
3
5
L
5
4
S
2
3
L
5
2
L
4
3
L
5
S
2
3
4

Previous results

Two-sided game (Ashlagi, AAAI 2010)

- With any two deterministic polices, there always exists a pure NE
- (S, R) and (L, R) also admit pure NE
- There exists a set of three polices, if a machine can use any of these polices, no pure equilibria exist

Our results

Model	(S, L)	(S, M)	(L, M)
Makespan $m = 2$	$9/7$	$3/2$	7/6
Makespan $m \geq 3$	$\frac{2m}{m+1}$	$2 - \frac{1}{m}$	$\frac{2m}{m+1}$
Max-Min	$[2 - \frac{2}{m-1}, 2 - \frac{1}{m}]$	m	[1.691, 1.7]

Remarks

科学发展观

- 眼前利益与长远利益的平衡（Online Optimization，Robust Optimization）
- 局部利益与全局利益的平衡（Mathematical Programming）
- 个人利益和团体利益的平衡（Game Theory）

Final Remarks

- Start from a good optimizer, knowing what best you can achieve.
- Become a strong mechanism designer, monitoring a game most players are happy with.

谢谢诸位捧场！

下面是广告时间

Results based on ETH

Lower bounds on exact algorithms

3-Coloring	$2^{o(n)}$
Dominating set	$2^{o(n)}$
Hamiltonian path	$2^{o(n)}$
Independent set	$2^{o(n)}$
Vertex cover	$2^{o(n)}$
Hamiltonian cycle in planar graph	$2^{o(\sqrt{n})}$

n : number of vertices in a graph

Lower bounds on approximation schemes

Maximum independent set on planar graphs	$2^{O((1/\varepsilon)^{1-\delta})} n^{O(1)}$	$2^{O(1/\varepsilon)} n$
Minimum vertex cover on planar graphs	$2^{O((1/\varepsilon)^{1-\delta})} n^{O(1)}$	$2^{O(1/\varepsilon)} n$
Minimum dominating set on planar graphs	$2^{O((1/\varepsilon)^{1-\delta})} n^{O(1)}$	$2^{O(1/\varepsilon)} n$
TSP with a metric defined by an unweighted planar graph	$2^{O((1/\varepsilon)^{1-\delta})} n^{O(1)}$	$2^{O(1/\varepsilon)} n$

By Daniel Marx (FOCS , 2007)

Our Contributions

Problems	Lower bounds (based on ETH)	Upper bounds
$P \parallel C_{\max}$ #jobs n	$2^{o(n)}$	$2^{O(n)}$ (Horowitz, Sahni , 1976)
$P \parallel C_{\max}$ Input size $\lvert I \rvert$	$2^{O((1/\varepsilon)^{1-\delta})} poly(\lvert I \rvert)$	$2^{O(1/\varepsilon^2 \log^3(1/\varepsilon))} poly(\lvert I \rvert)$ (Jansen, 2010)
$P_m \parallel C_{\max}$ #jobs n	$(1/\varepsilon)^{O(m^{1-\delta})}$	$n(m/\varepsilon)^{O(m)}$ (Jansen, Porkolab, 2001)
$P_m \parallel C_{\max}$ Input size $\lvert I \rvert$	$2^{O((\sqrt{m\lvert I \rvert})^{1-\delta})}$	$2^{O(\sqrt{m\lvert I \rvert}+m\log\lvert I \rvert)}$