Mechanism Design via Correlated Tree Rounding

Yossi Azar

Tel Aviv University

Joint work with:

M. Hoefer, I. Maor, R. Reiffenhauser and B. Vocking

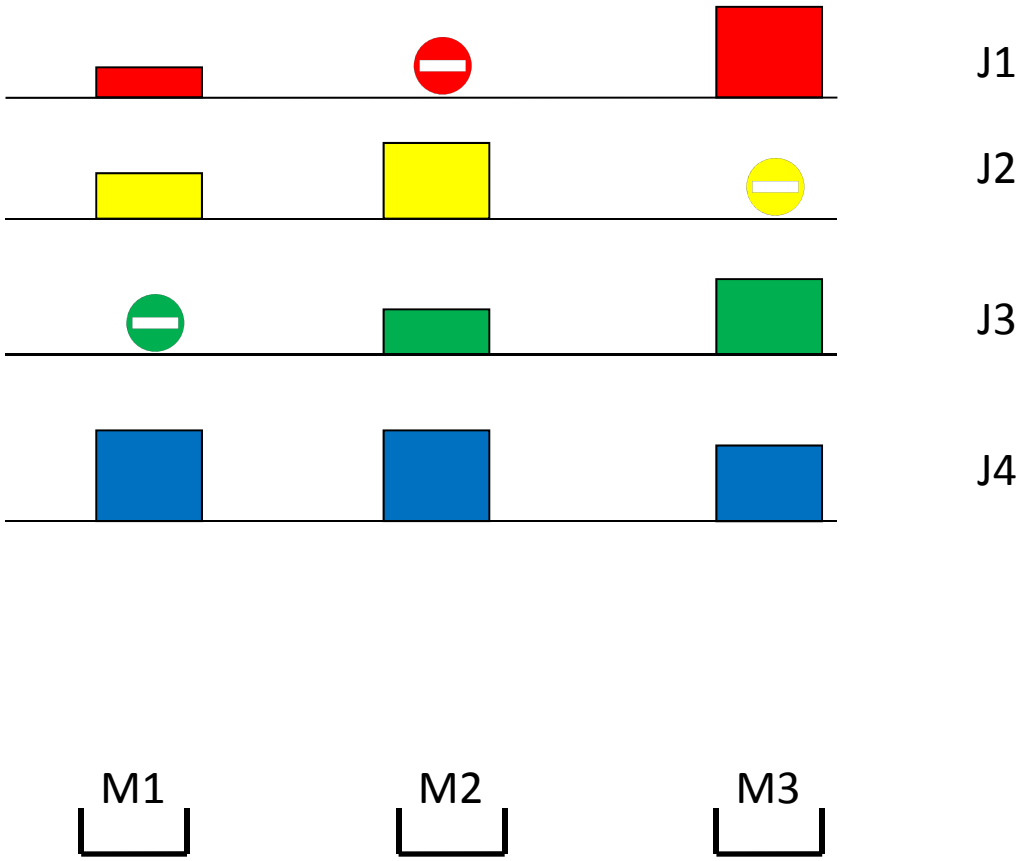
Unrelated Machine Scheduling

Given m machines and n jobs

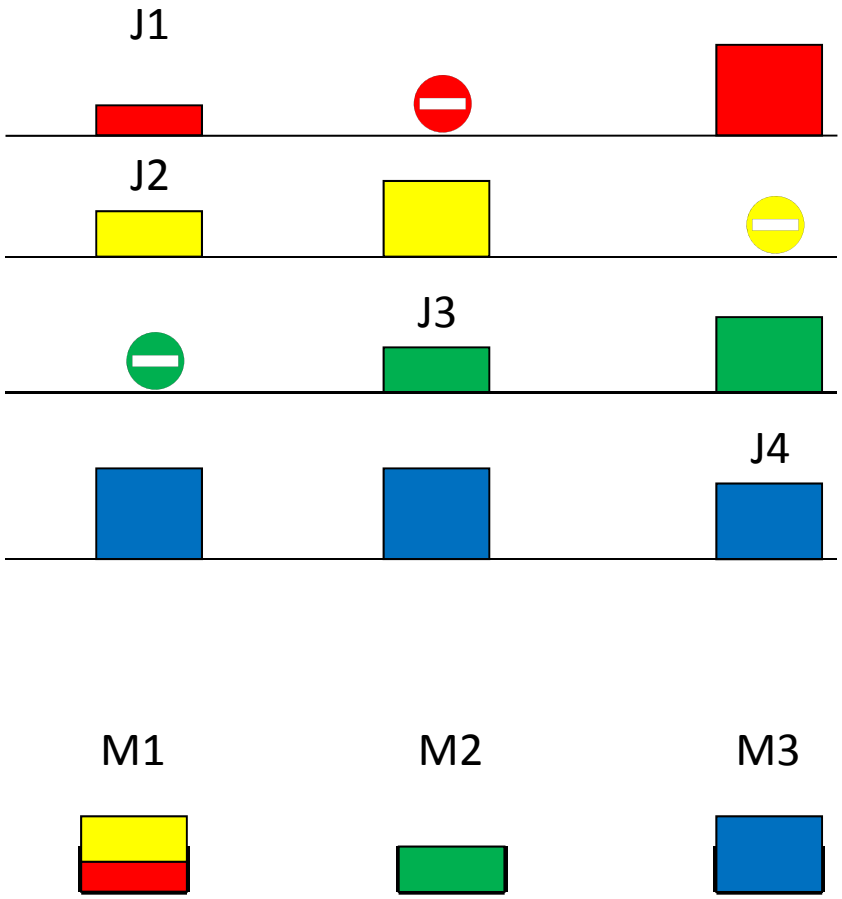
- Job i has processing time (load) $w_i(j)$ if assigned to machine j
- Assign each job to one machine
- Load on machine j: $L_j = \Sigma_{i \mid A(i)=j} w_i(j)$

- Goal: minimize makespan (max load)

Unrelated machine scheduling



Unrelated machine scheduling



Identical, Related, Restricted Models

Given m machines and n jobs

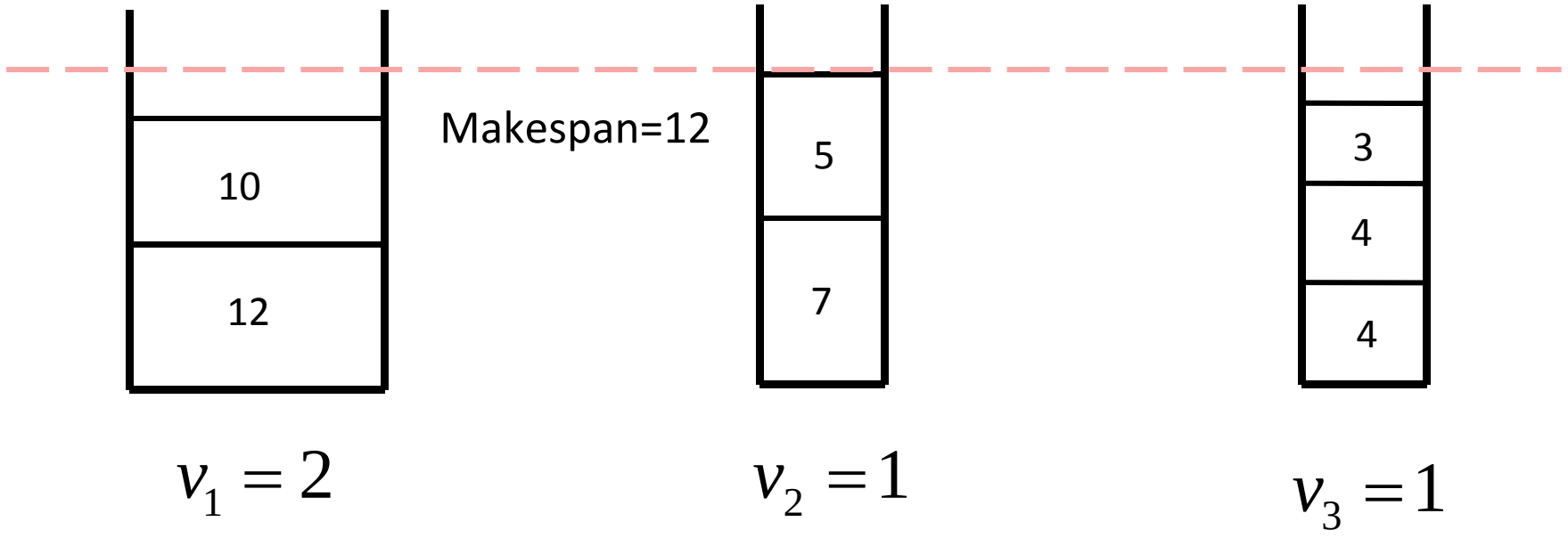
- *Identical*: machines are identical $w_i(j) = w_i$
- *Related:* machine j has speed v_j .

Hence $w_i(j) = w_i / v_j$

- *Restricted*: Job i sizes w_i and $M(i) \subseteq M$

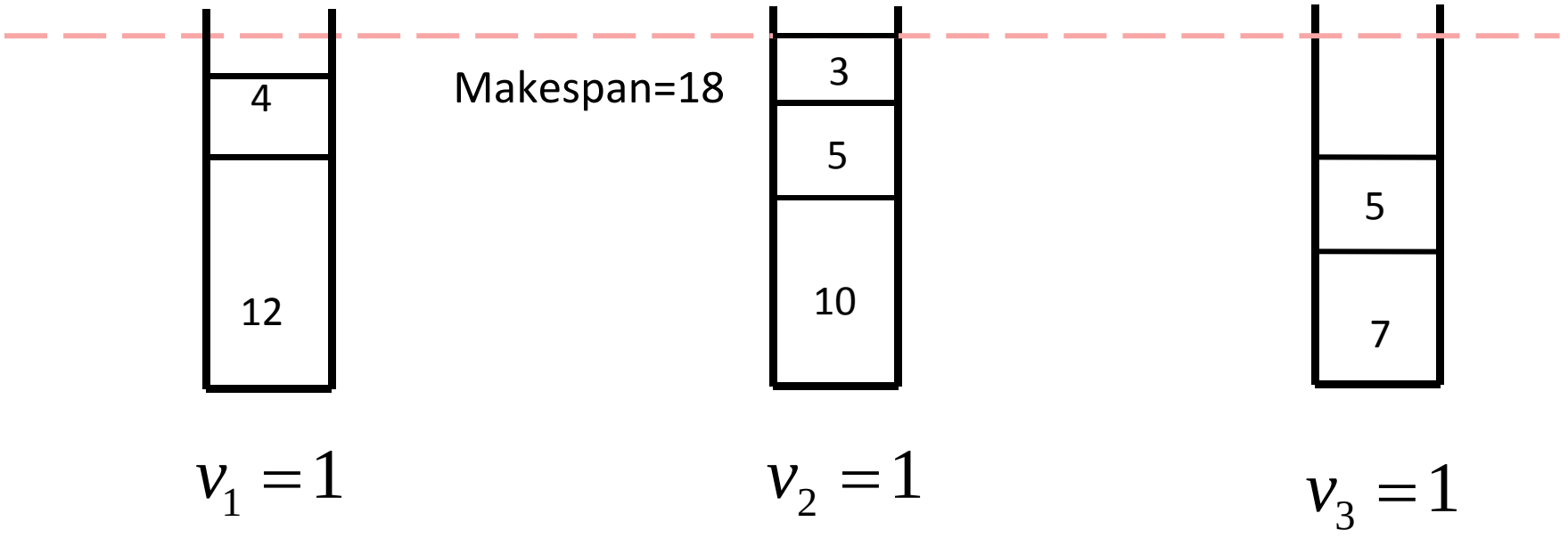
Example (related)

The jobs: 12, 10, 7, 5, 4, 4, 3



Example (restricted)

The jobs: 12 (1,2), 10 (1,2), 7 (2,3), 5 (1,2,3), 5 (2), 4 (1), 3 (1,2)



Offline algorithms

Input is completely known in advance

- Approximation Ratio =

Supremum over all inputs ALG(I)/OPT(I)

- 2 approximation [Lensrtra Shmoys Tardos 87]
- 1.5 computation hardness (NP-hard)
- Identical & Related: PTAS [Hochbaum Shmoys 86]

Online algorithms

Jobs arrive one by one and are assigned immediately

- Competitive Ratio =

Supremum over all sequences $ON(\sigma)/OFF(\sigma)$

- $O(\log m)$ competitive algorithm

[A + Aspnes Fiat Plotkin Waarts 93] [A+ Naor Rom 92]

- Tight up to a constant factor
- Identical & Related: constant competitive […]

Strategic jobs

- Jobs are strategic
- Cost of a job = load on machine
- Jobs reach an equilibrium
- Price of Anarchy=

Supremum over all inputs EQ(I)/OPT(I)

deterministic strategies (Pure Nash) or
randomized strategies (Mixed Nash)

Price of Anarchy (Makespan)

- Identical:
  - Pure: 2
  - Mixed: *logm/loglogm* [KoutsoupiasPapadimitriou99]
- Related or Restricted:
  - Pure *logm/loglogm*
  - Mixed: *logm/logloglogm*

  [Czumaj Vocking 02, A+Awerbuch Richter Tsur 03]
- Unrelated: unbounded

Strategic machines

- Machines are strategic
- Mechanism design

  instead of Price of Anarchy
- Machine *j* declares $w_i(j)$ *(bid)* for each *i*
- Auctioneer decides on
  - assignment (of jobs to machines)
  - Payments (to machines)
- Machine`s goal: Maximizing

payment minus load = $P_j - L_j = P_j - \Sigma_{i|A(i)=j} w_i(j)$

- Prior free truthful mechanism = declaring correct $w_i(j)$ *is always dominant strategy*

Truthful Unrelated machines

Minimize the Makespan

- m approx VCG [Nisan Ronen 99]
  - Assign each job to the machine with min bid
  - Pay the machine the second smallest bid
- *m*/2 approx for Fractional solution
  [Christodoulou Koutsoupias Kovacs 07]

- Hardness: 2.61
  [Christodoulou Koutsoupias Vidali 07]
- *m* hardness anonymous
  [Ashlagi Dobzinski Lavi 09]

Truthful Related Machines

Job i sizes w_i (public)

Machine j has speed v_j (private info)

Let $L_j = \Sigma_{i|A(i)=j} w_i / v_j$

- 2 approx (rand) [Archer Tardos 01]
- 5 approx (deter) [Andleman -A- Sorani 05]

  → 3 [Kovacs 05]
- PTAS (randomized)

  [Dhangwatnotai Dobzinski Dughmi Roughgarden 08]
- PTAS (determinstic) [Christodoulou Kovacs 10, Epstein Levin VanStee 13]

Question ?

Unrelated machines mechanism: too hard

Related machines: easy but far from unrelated

Question:

Is their a model “similar” to unrelated machines with “relatively easy” mechanism?

Restricted-Related

Job *i* sizes w_i and $M(i) \subseteq M$ (public)

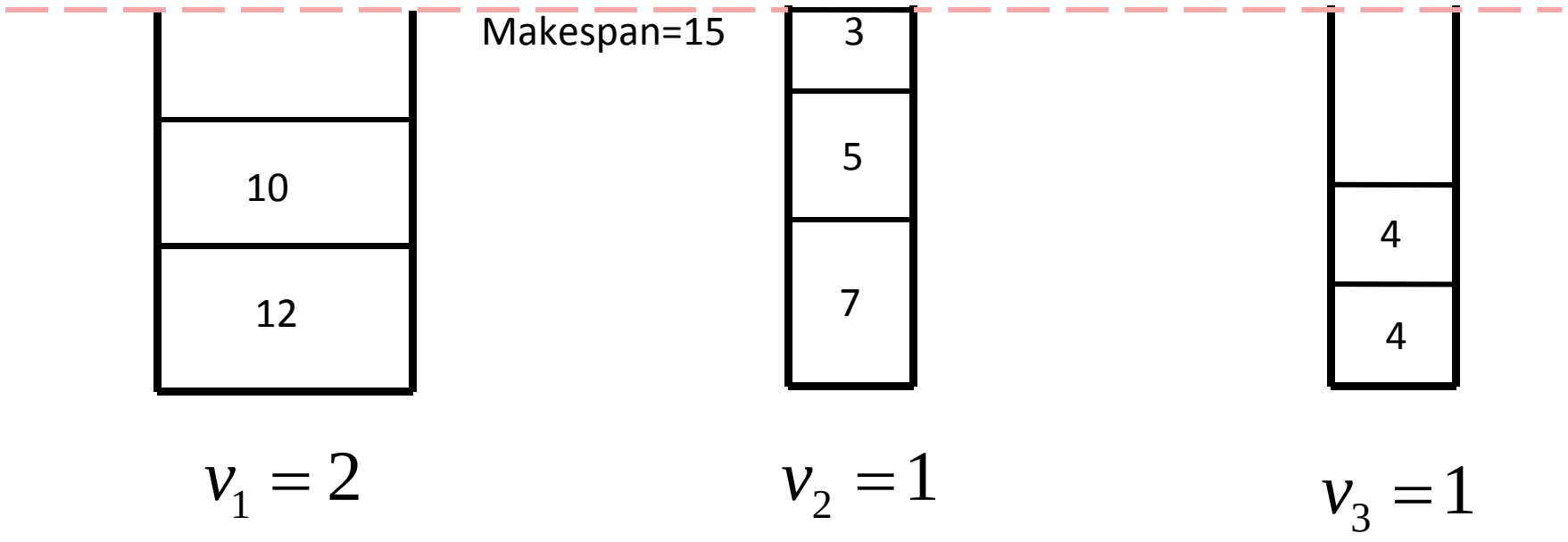
Machine *j* has speed v_j (private info)

$A(i) \in M(i)$ and $L_j = \Sigma_{i|A(i)=j} w_i / v_j$

- Results similar to Unrelated:
- 2 approx vs. 1.5 hardness (offline)
- $O(\log m)$ competitive algorithm (online)
- Tight up to constant

Example (restricted-related)

The jobs: 12 (1,2), 10 (1,3), 7 (2,3), 5(2), 4(1,3), 4(2,3), 3(2)



Simulate Unrelated by Restricted-Related

- Given Algorithm for Restricted-Related →

  Solve unrelated up to $O(\log m)$

- Given input for **unrelated** machines create input for Restricted-Related
- **Solve** Restricted-Related instance
- **Transform** the solution to a solution for unrelated machines

Unrelated → Restricted-Related

Given **input** for **unrelated machines**

- Make *log m* copies for each unrelated machine of speeds *1, ½, ¼, …1/m*
- Let $w_i = \min_j w_i(j)$. *M(i)* includes one copy of machine *j* of efficiency $w_i/w_i(j)$ rounded
- If $w_i/w_i(j) < 1/m$ → no copy of machine *j*
- **Solve** Restricted-Related instance
- **Return for unrelated machines**: Load on machine j is the sum of its copies

Restricted-Related

Conclusion:

Restricted Related is “similar” to Unrelated

Question:

Truthful mechanism for Restricted-Related?

Note: each machine - single parameter

Main result

Main Theorem (Restricted-Related model):

There is a randomized truthful mechanism with 2 approximation for the makespan

Monotone algorithms

Monotone algorithm =

Increase speed of machine → increase its load

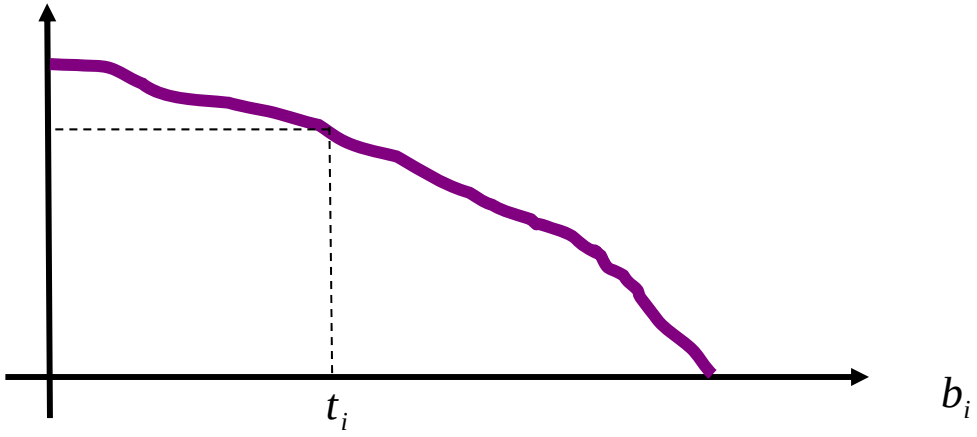
Single parameter truthful mechanism

Monotone algorithm

Why ??

Monotone algorithm

- Monotone: “faster” => more work
- Consider the work assigned to agent *i* as a single-variable function of bid (one over speed)
- Work-curve $w_i(b_{-i}, b_i)$



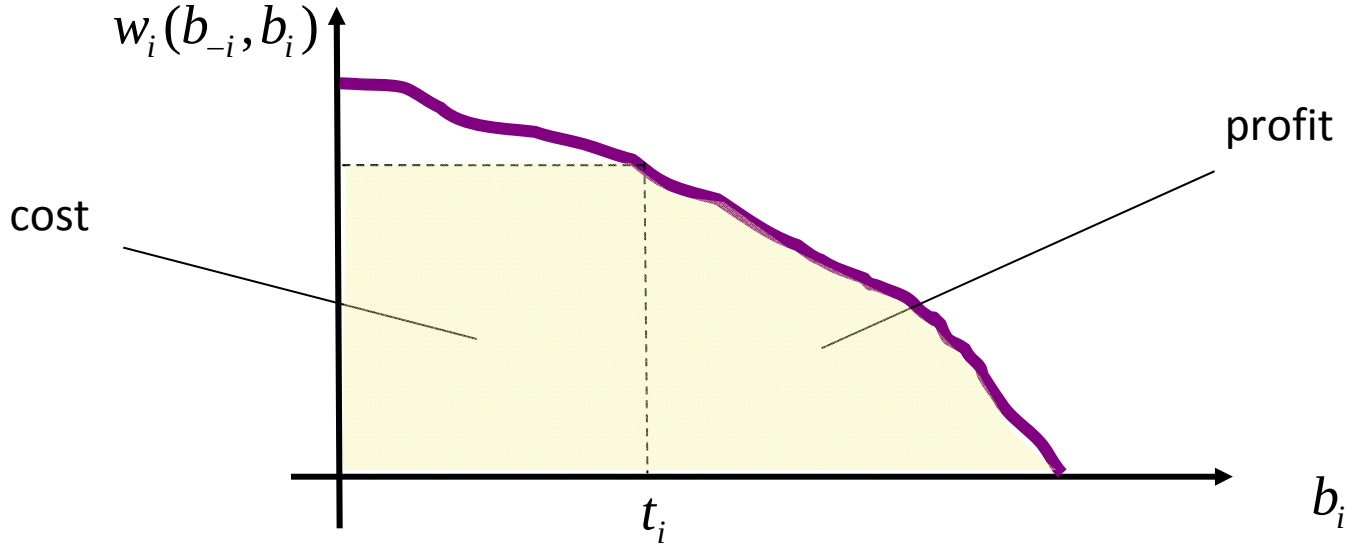
[Archer Tardos 01]
Truthfulness <=> Monotone Algorithm

Monotone => Truthful

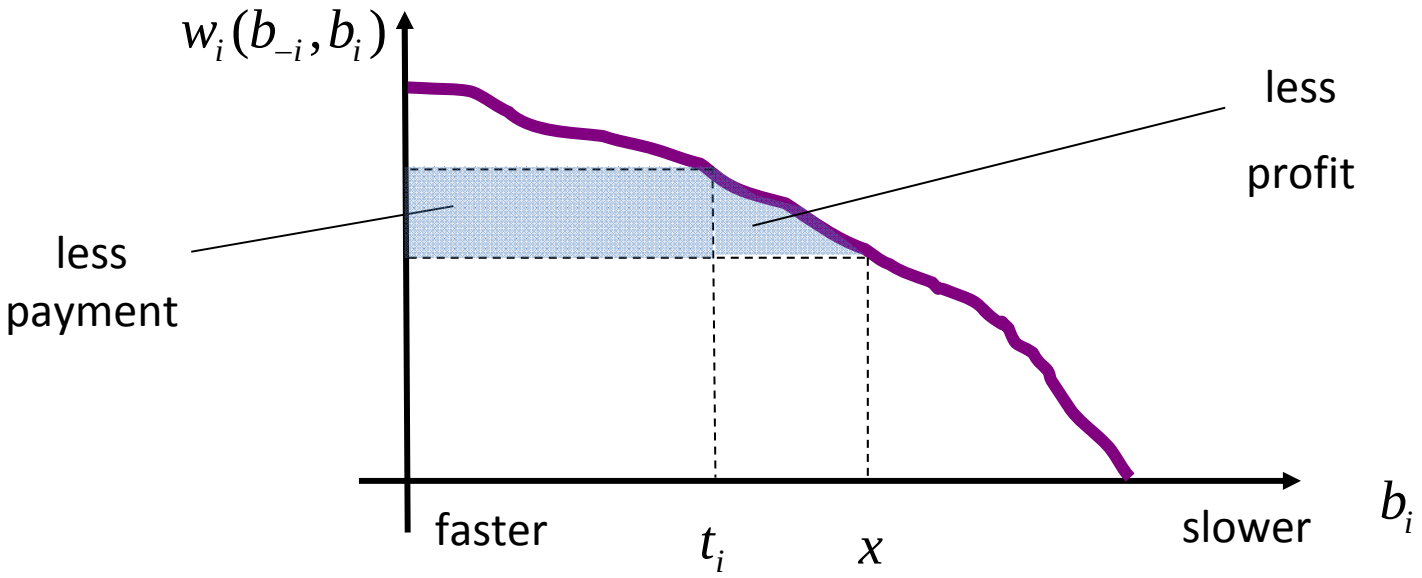
- Truthful Mechanism:
  - Assignment: monotone algorithm
  - Payment scheme

$$P_i(b_{-i}, b_i) = b_i w_i(b_{-i}, b_i) + \int_{b_i}^{\infty} w_i(b_{-i}, u)du$$

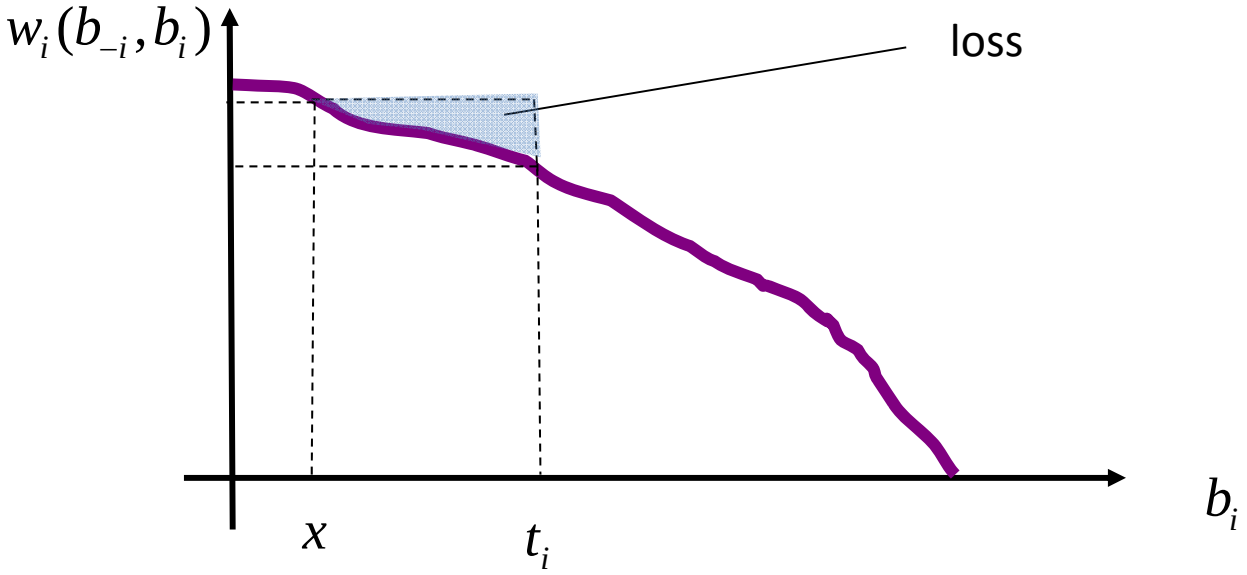
- The work-curve



Overbidding



Underbidding



Monotone algorithms

Conclusions:

Forget about truthful mechanisms

Think about monotone algorithms

Mechanism (=monotone Algorithm)

Monotone algorithm has 3 steps:

1. Fraction Assignment (+small fix)
2. Open Cycles
3. Dependent Rounding

Monotone Algorithm (Step 1)

Fractional Assignment = job may be split among more than one machine

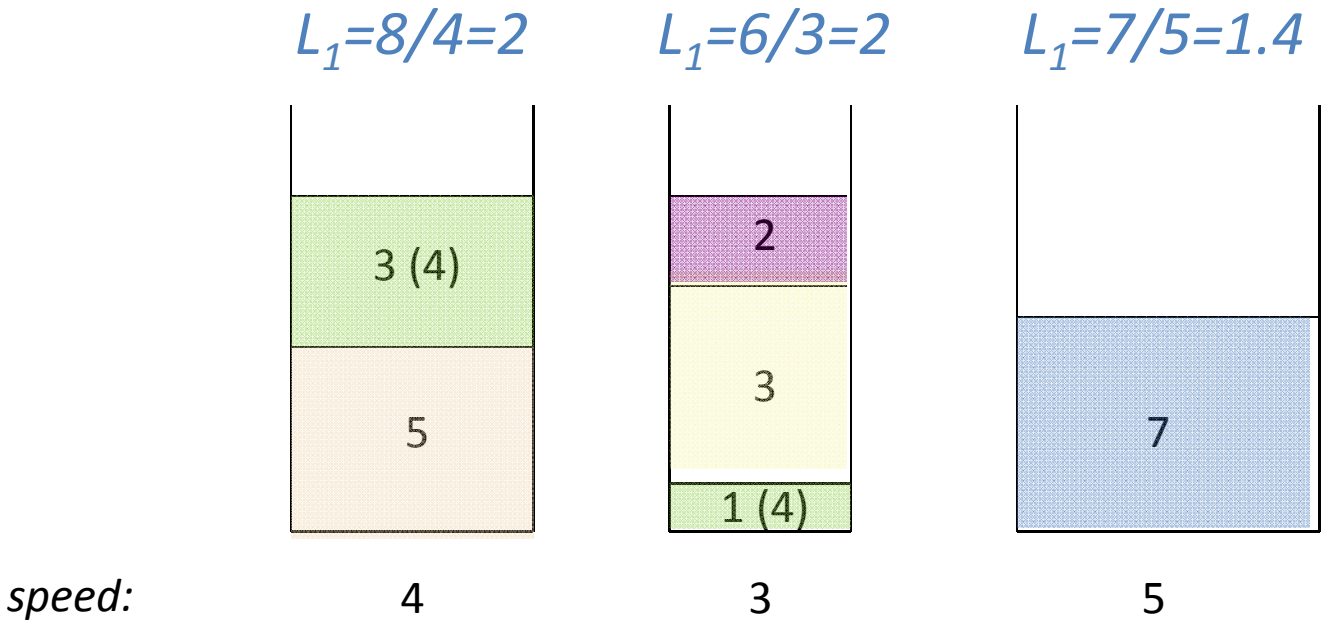
Find the

- Min makespan T
- Max lexicographic load vector subject to T

$(L_1, L_2, L_3, \ldots, L_m)$

Valid Fractional Assignment

- Example
  - Jobs: 7 (1,3),5(1,2),4(1,2),3(1,2),2 (2)
  - Speeds: 4, 3, 5



Monotone Algorithm (Step 1)

- Load on each machine is unique
- Assignment is not unique
- Can be found in poly-time by iterative flow or iterative LP

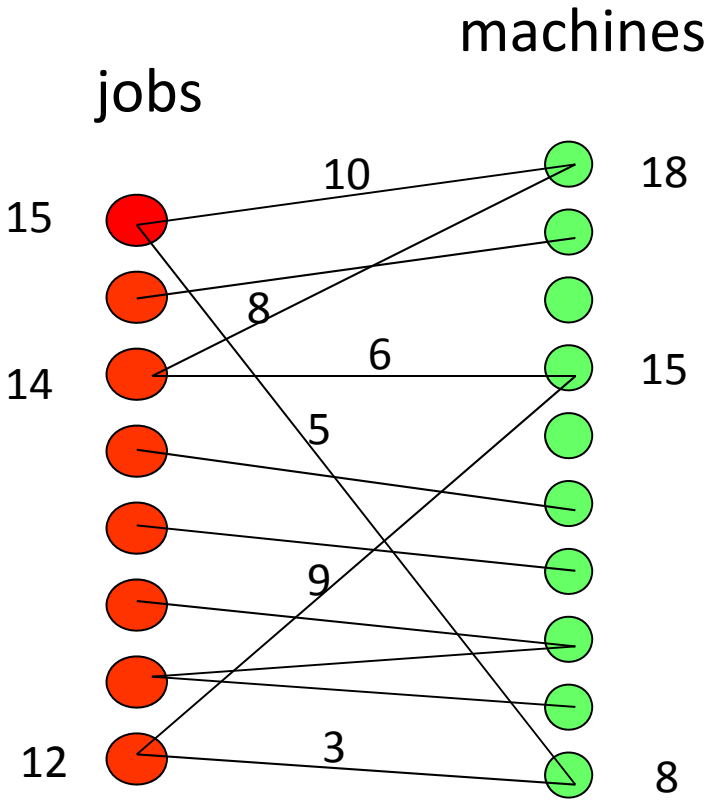
  → Volume is monotone in speed

Monotone Algorithm (Step 1 + fix)

Fix the Max-lexicographic assignment

- Find the min makespan T s.t. if $w_i/v_j > T$ → job i cannot be assigned to machine j at all
- Find the Max-lexicographic assignment with the new restrictions
- Can be found in poly-time & monotone

Resulting fractional assignment



Monotone Algorithm (Step 2)

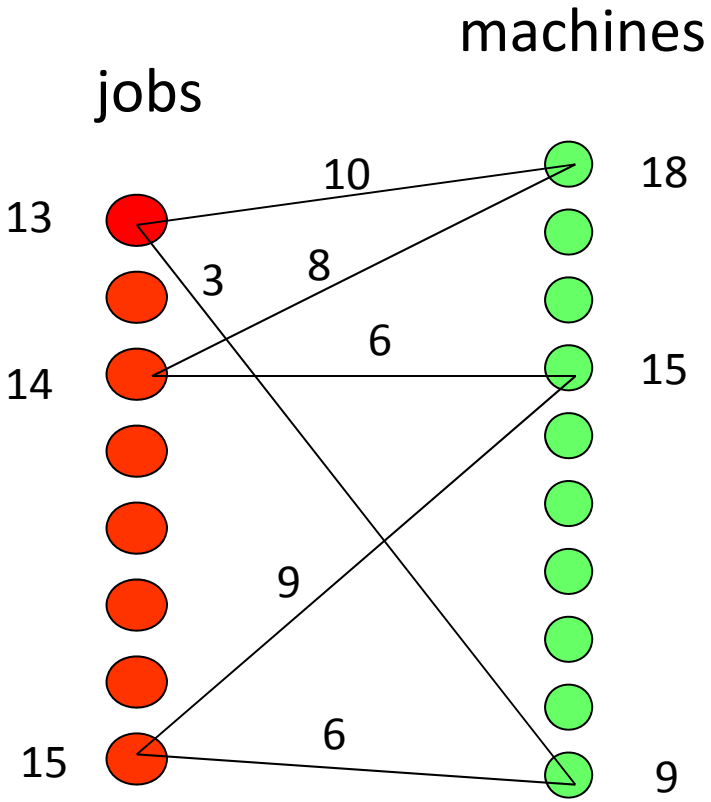
Open cycles (iteratively):

shift weight over cycle in the assignment graph

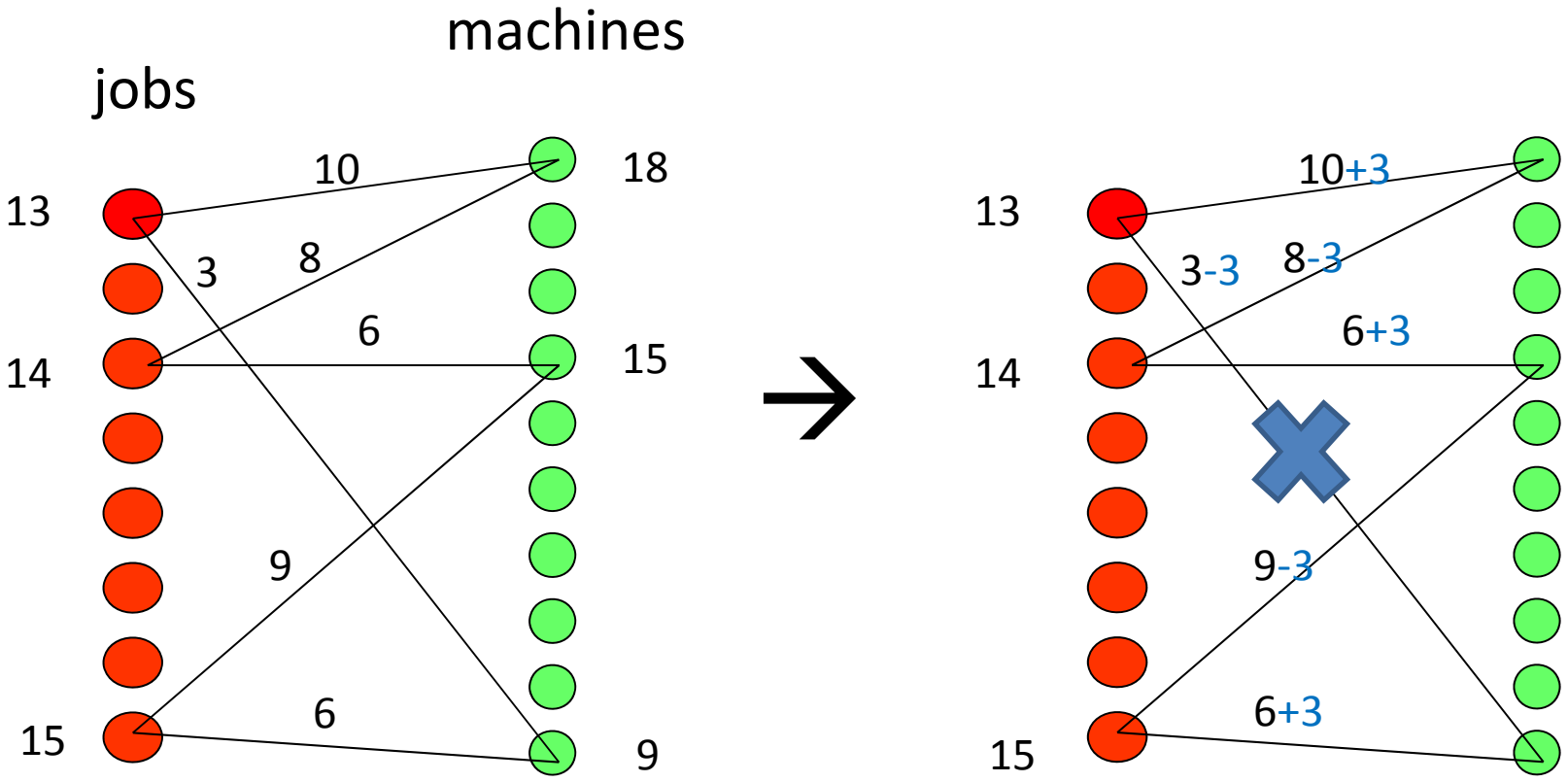
→ Tree

- Load on each machine remains the same

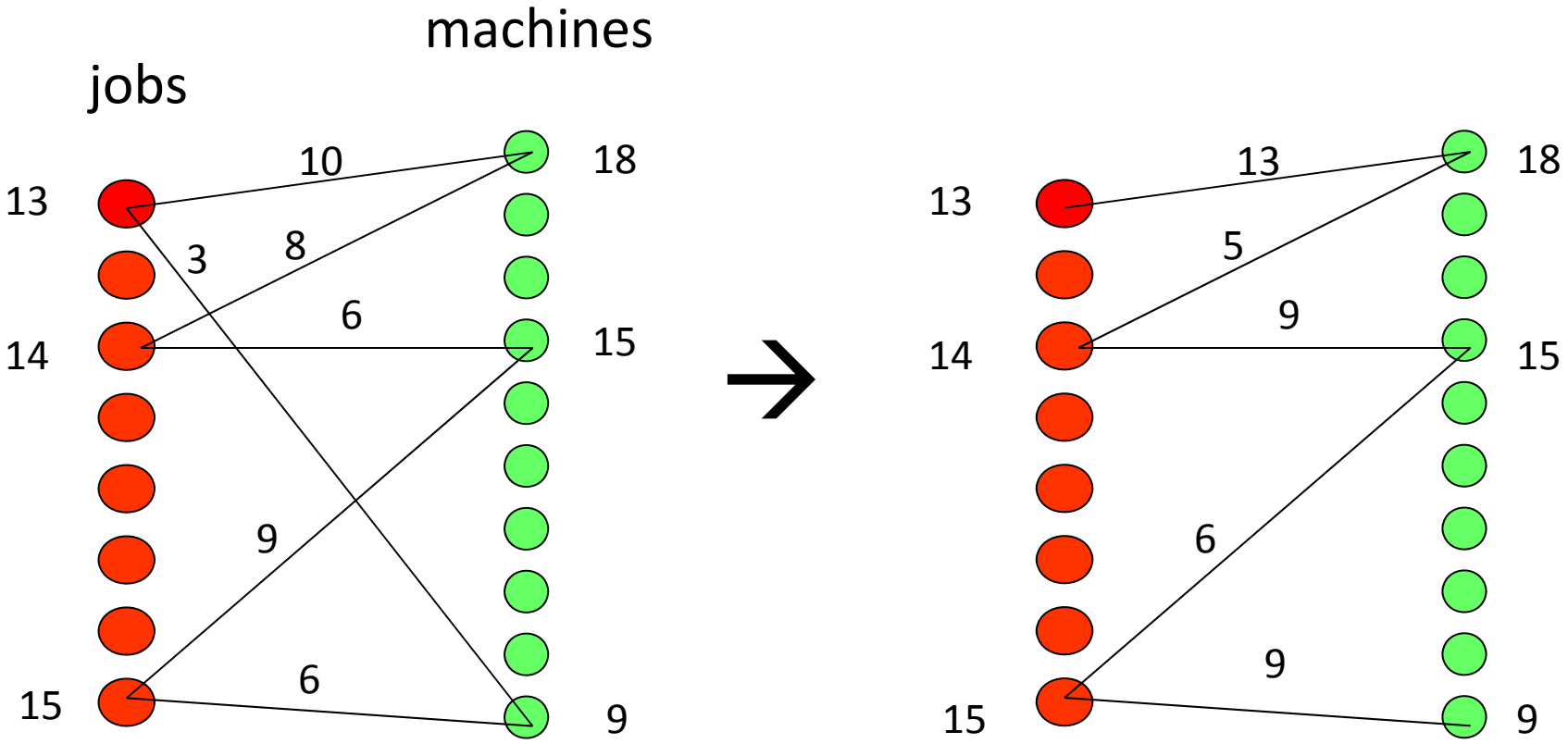
Monotone Algorithm (Step 2)



Monotone Algorithm (Step 2)



Monotone Algorithm (Step 2)

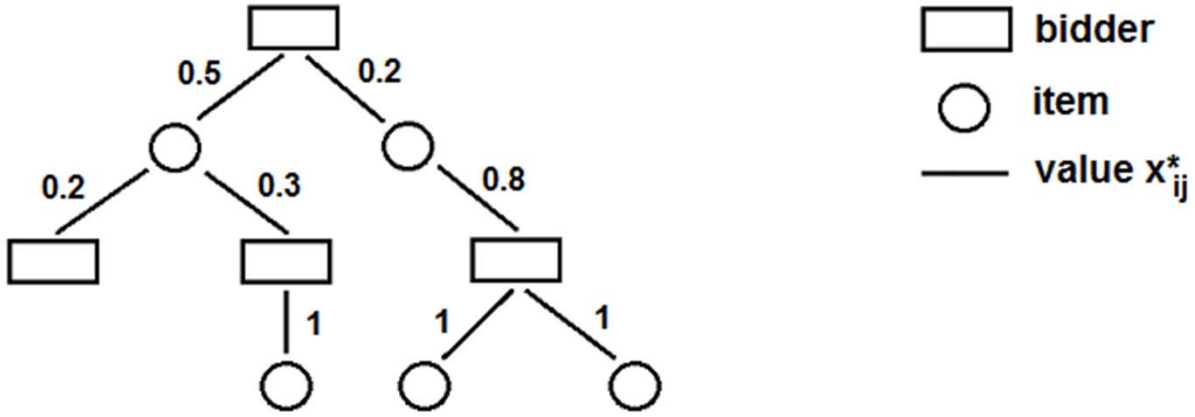


Monotone Algorithm (Step 3)

Round the assignment

- Rounding of [LST87] does not maintain truthfulness
- Randomized rounding maintains truthfulness but the load can grow by a factor of *O(log m)*
- We apply dependent rounding from root to leaves (top down)

Monotone Algorithm (Step 3)



Monotone Algorithm (Step 3)

Round the assignment

- Consider Jobs assigned to a specific machine j:

  jobs of size w_i and fraction $p_i(j)$

  – Sort jobs assigned to it from large to small w_i

  – Let $S_k = \Sigma_{i \le k} p_i(j)$

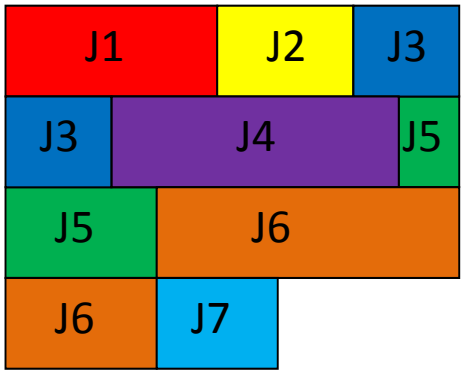
  – Choose random $0 \le x < 1$

  – Assign to machine j all jobs *i* such that

  $[S_{k-1} \pmod 1, S_k \pmod 1]$ contains x

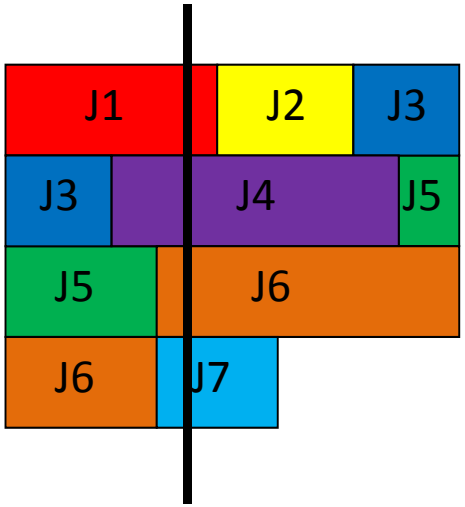
- All machines are coordinated (top down) s.t. each jobs is assigned to precisely one machine

Rounding for each machine



- Jobs sorted by sizes **J1 ≤ j2≤ j3 ≤ j4 ≤ j5 ≤ j6 ≤ j7**
- Width := Fraction = Probability

Rounding for each machine



- Jobs sorted by sizes **J1 ≤ j2≤ j3 ≤ j4 ≤ j5 ≤ j6 ≤ j7**
- Width = Probability
- **J1, j4, j6, j7** are assigned to machine

Properties of rounding

- Valid:
  - All jobs are assigned
  - Each job is assigned to one machine
- Prob. that a job *i* is assigned to machine j = fraction $p_i(j)$
- Expected load = fractional load (monotone)
- Load on each machine at most fractional load plus one job (implies 2 approximation)

→ monotone algorithm & 2 approx

Conclusions+ Open problems

- Randomized truthful mechanism for Restricted-Related model that approximates makespan
- The approximation ratio is 2

- Q?: Find deterministic truthful mechanism with constant approx

- Q???: Mechanism for unrelated machines

Thanks !!! Questions?