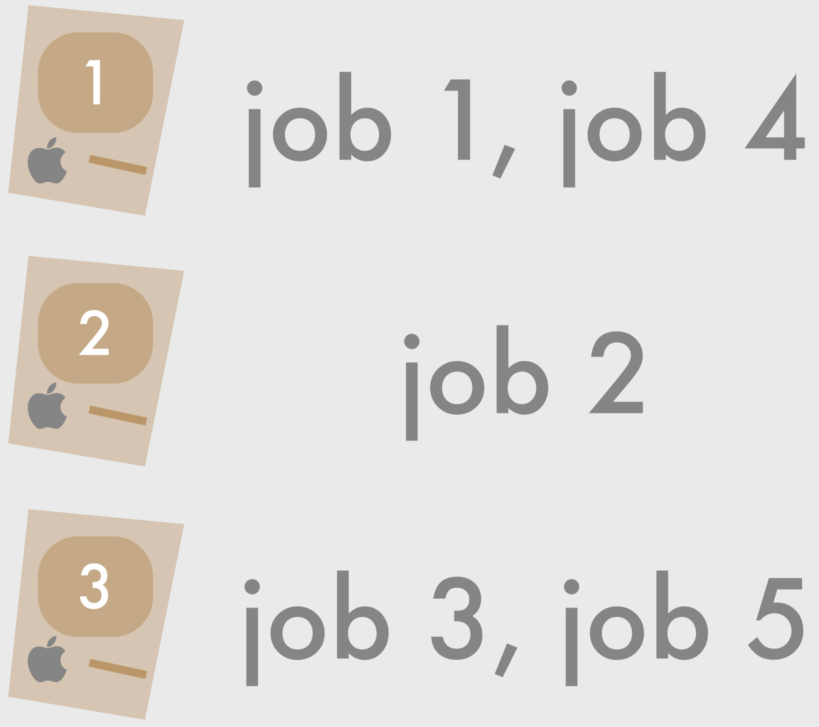
(Non-clairvoyant) scheduling games

joint work with Nguyen Kim Thang

The scenario

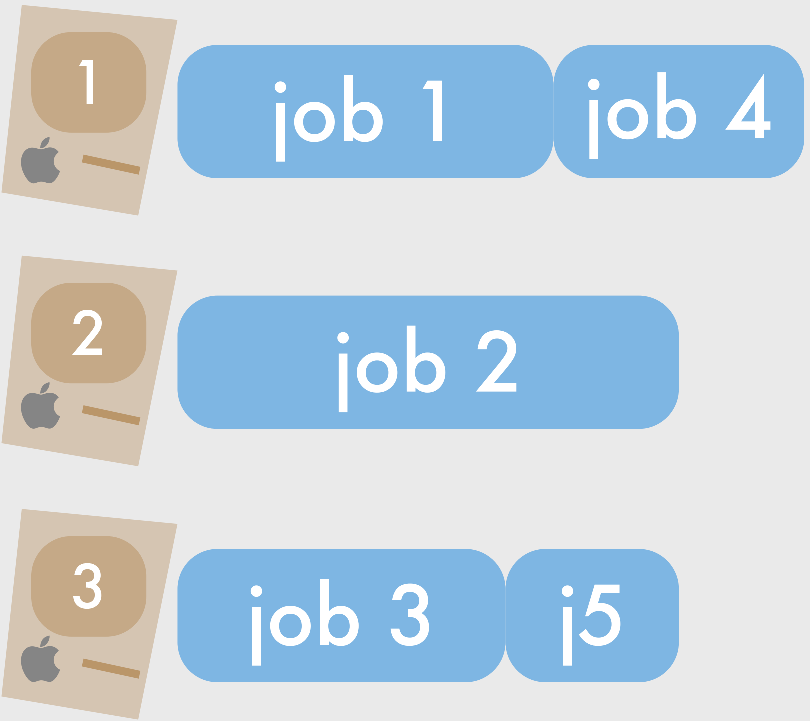
- Every player has a job and chooses a machine where to execute it (strategy).
- Such a job-machine assignment is called a strategy profile.
- There are different machine environments (identical machines, uniform machines ...)



The scenario

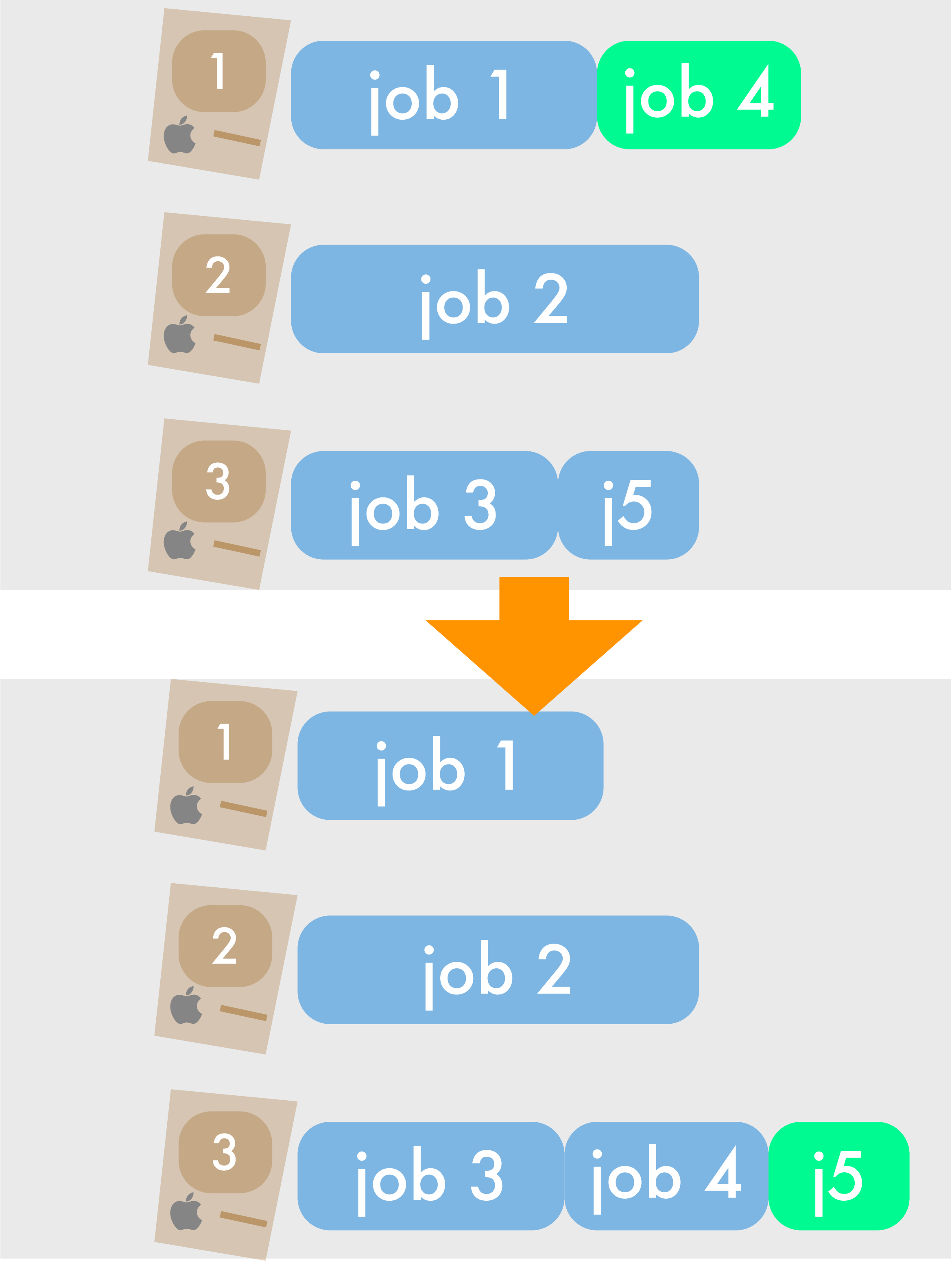
- There is a fixed and known policy that determines how jobs assigned to machine are going to be scheduled on it.
- The player's cost are the completion times of their jobs.
- We assume the player's know the processing times of all jobs, and therefore could compute the cost they would have on another machine.

for example:
LongestFirst



The scenario

- A player is *unhappy* if he can decrease its cost by changing to another machine (best move).
- A strategy profile is a (pure) Nash equilibrium if everyone is happy.
- The Nash dynamics is the graph on strategy profiles where arcs correspond to best moves.



Directions of research

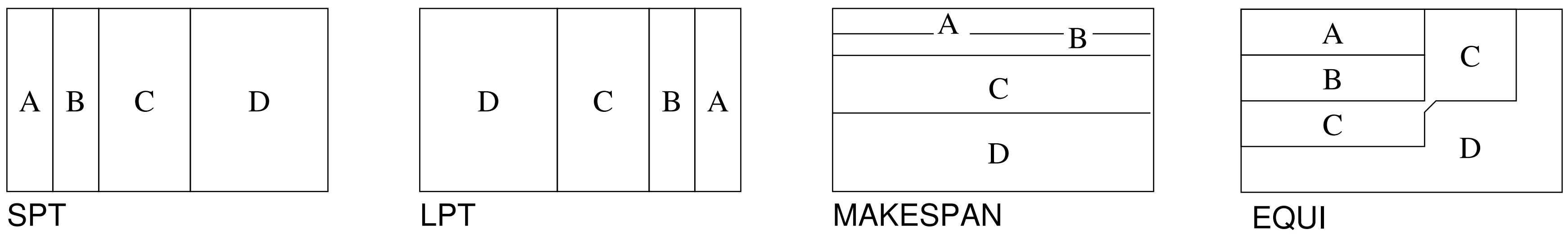
Fix some (machine environment, policy) combination

- Does there always exist a Nash eq.?
- Does the Nash dynamics always converges?
- How long does it take?
- How hard is it to find a NE?
- Fix some social cost (typically maximal user cost), how far can a NE be from the social optimum (that might not be a NE)? (the price of anarchy)

The machine models

- **P**: (identical machines) job i has processing time p_i on each machine
- **Q**: (uniform machines) job i has processing time p_i/s_j on machine j.
- **B**: (restricted identical) like P but some jobs are forbidden on some machines
- **R**: (unrelated or specialized machines) Job i ha processing time p_{ij} on machine j

Standard policies



- **ShortestFirst**: is good since it minimizes sum of player costs
- **LongestFirst**: has better price of anarchy
- **Makespan**: for every player to complete at the same time. Good since it strongly related player costs to social cost. Makes sense if machines are lonks in a network.
- **Random**: schedule in random order.
- **EQUI**: distribute CPU time evenly. Jobs complete in same order as for ShortestFirst.

Properties of policies

- for the unrelated model (R) a policy is **(strongly) local** if it depends only on jobs assigned to this machine, (and only on processing times for this machine)

- A policy is preemptive if it does not schedule the jobs in just one piece. It can introduce idle times as well

machines

jobs

1	3	4
7	2	3
6	6	1
9	12	1
4	7	100

jobs assigned to the same machine

Price of anarchy

m=number of machines

	P	Q	B	R
Makespan	O(1)	Θ(log m/loglog m)	Θ(log m/loglog m)	unbounded
ShortestFirst	O(1)	Θ(log m)	Θ(log m)	Θ(m)
LongestFirst	O(1)	O(1)	Θ(log m)	unbounded
Random	O(1)	Θ(log m/loglog m)	Θ(log m/loglog m)	Θ(m)
any local non-preemptive policy				Ω(log m)
any strongly local non-preemptive policy				Ω(m)
AJM1				O(log m)
AJM2				$O(\log^2 m)$
ACOORD				O(log m)
BCOORD				O(log m/loglog m)
EQUI	O(1)	Θ(log m)	Θ(log m)	Θ(m)

[Azar,Jain,Mirrokni,SODA'08]
[Caragiannis,SODA'09]

Existence of NE

	P	Q	B	R
Makespan	yes	yes	yes	yes
ShortestFirst	yes	yes	yes	yes
LongestFirst	yes	yes	yes	open
Random	yes	open (1)	yes	open (2)
AJM1				no
AJM2				yes
ACOORD				yes
BCOORD				open
EQUI	yes	yes	yes	yes

(1) yes when speeds differ by at most 2

(2) yes for 2 machines when processing times differ by at most 2 for fixed job

Random - Q - balanced speeds

- Processing times $p_1 \leq \ldots \leq p_n$
- Machine speeds $s_1 \geq \ldots \geq s_m \geq s_1/2$
- **Lemma**: suppose i makes a best move from a to b, and there is a new unhappy player i'>i. Then $s_a < s_b$.

- **Proof**: case i' was happy on some machine c≠b. Let l_x be the load of machine x before the move.
  - i was unhappy on a
  - i' wants to move to b
  - i' was happy before

$$\ell_a + \frac{p_i}{s_a} > \ell_b + \frac{2p_i}{s_b}$$

$$\ell_c + \frac{p_{i'}}{s_c} > \left(\ell_a - \frac{p_i}{s_a}\right) + \frac{2p_{i'}}{s_a}$$

$$\ell_c + \frac{p_{i'}}{s_c} \leq \ell_b + \frac{2p_{i'}}{s_b}$$

$$\left(\frac{1}{s_b} - \frac{1}{s_a}\right)(p_{i'} - p_i) > 0.$$

the other cases are similar

A potential function

- Given a strategy profile σ : Jobs→machines, let t be the unhappy player with greatest index
- encode t by f_σ : players→{0,1}
  $f_\sigma(i) = 1$ if $i \leq t$, and 0 otherwise

$f_\sigma(i)$	1	1	1	1	1	1	1	0	0
i	1	2	3	4	5	6	7	8	9

- $\phi=(f_\sigma(1), s_{\sigma(1)}, f_\sigma(2), s_{\sigma(2)},\ldots f_\sigma(n), s_{\sigma(n)})$

Refinement of Nash dyn.

- **Claim**: Every time let the unhappy user with greatest index do a best move. Then the potential decreases lexicographically.

- **Proof**:
Let t be the unhappy user with greatest index in σ
Let σ′ be the result of the move
Let t′ be the unhappy user with greatest index in σ′

- case $t' < t$

1	a	1	b	1	a	0	c
1	a	0	b	0	b	0	c

- case $t' > t$
then $b < a$
by previous lemma

1	a	1	b	1	a	0	c
1	a	1	b	1	b	1	c

Conclusion

- Is there always a NE for
  - LongestFirst on unrelated machines
  - Random for unbalanced uniform machines
- Is there a strong relationship between existence of NE and convergence of Nash dynamics?