The curse of rationality in sequential scheduling games

Cong CHEN joint work with **Yinfeng XU**

School of Business Administration
South China University of Technology (SCUT)
China

華南理工大學
South China University of Technology

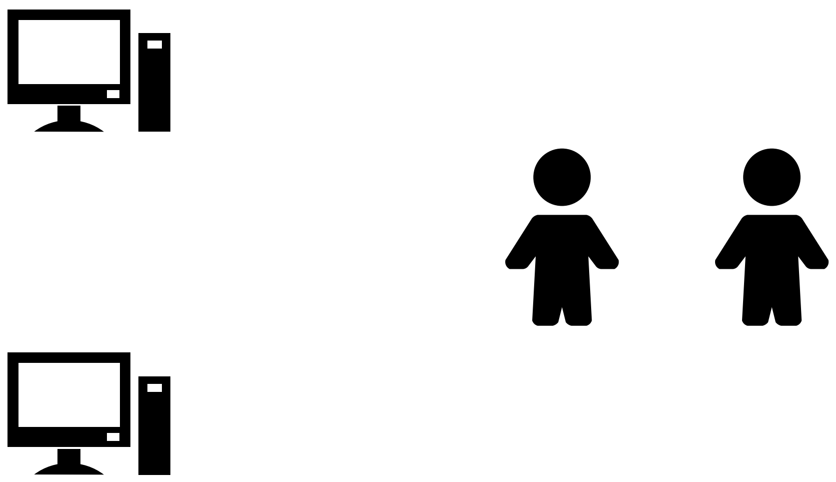
School of Management
Xi'an Jiaotong University (XJTU)
China

西安交通大學
XI'AN JIAOTONG UNIVERSITY

WINE 2020, December

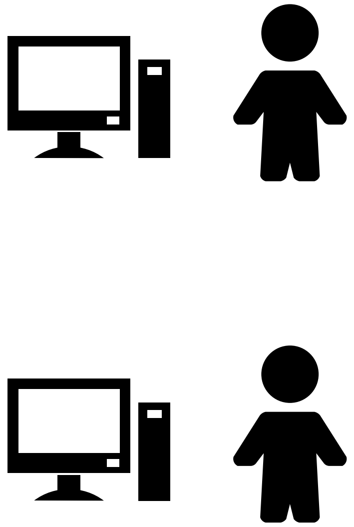
Sequential Scheduling Games

A game of 2 machines and 2 users/jobs



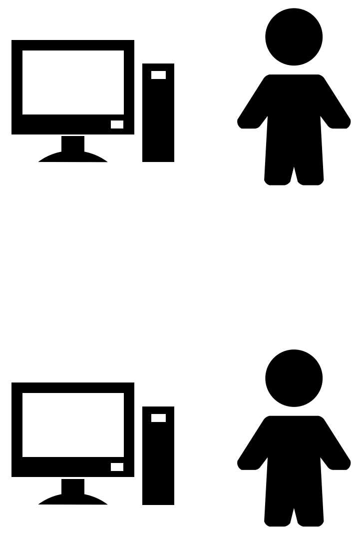
Sequential Scheduling Games

A game of 2 machines and 2 users/jobs

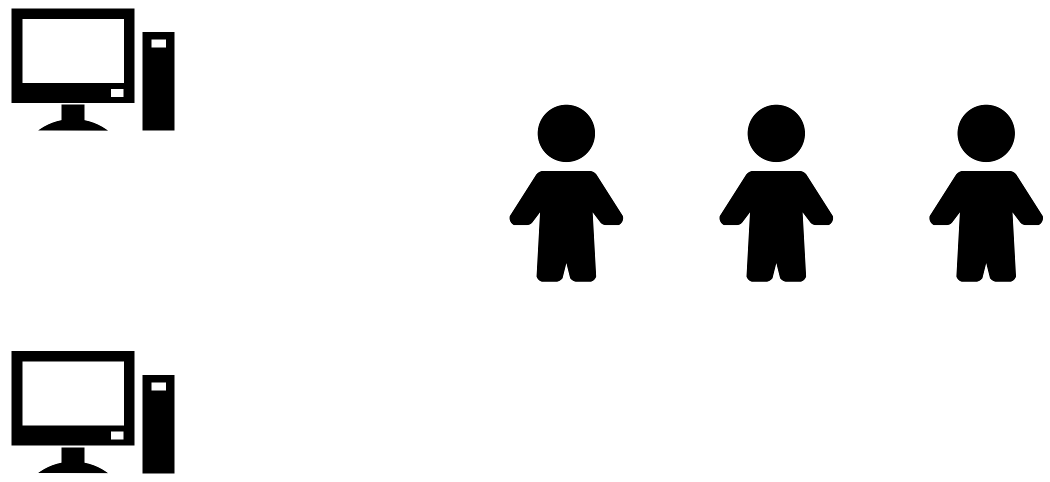


Sequential Scheduling Games

A game of 2 machines and 2 users/jobs

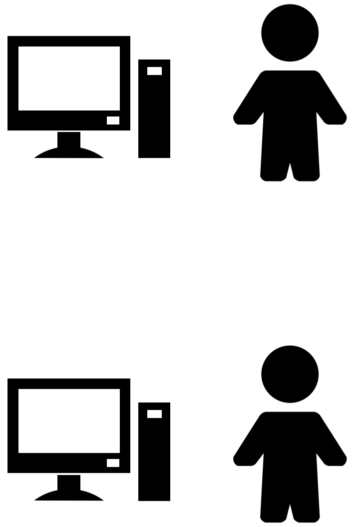


A game of 2 machines and 3 users/jobs

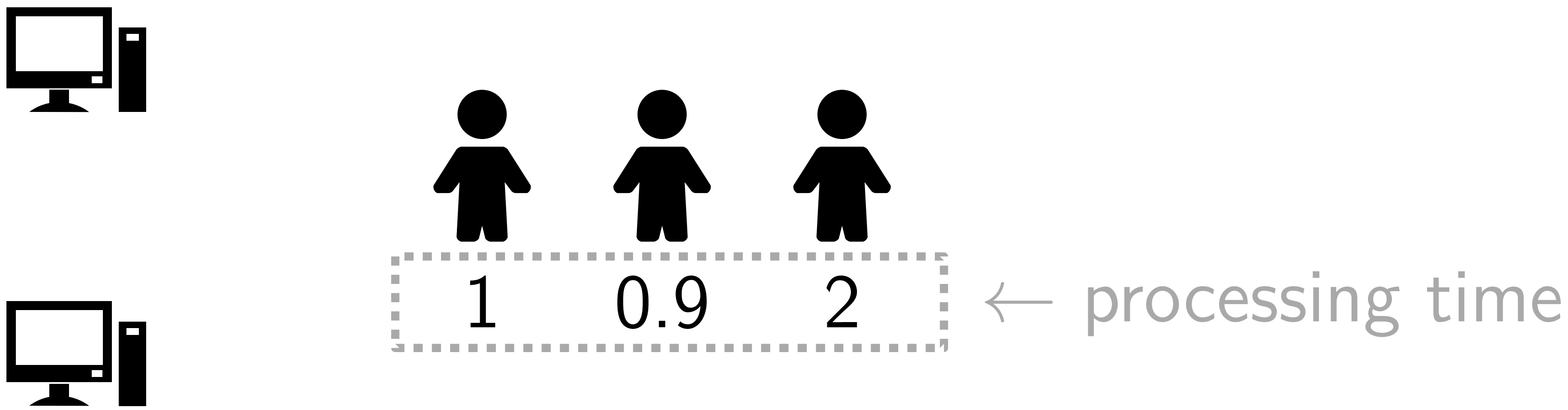


Sequential Scheduling Games

A game of 2 machines and 2 users/jobs

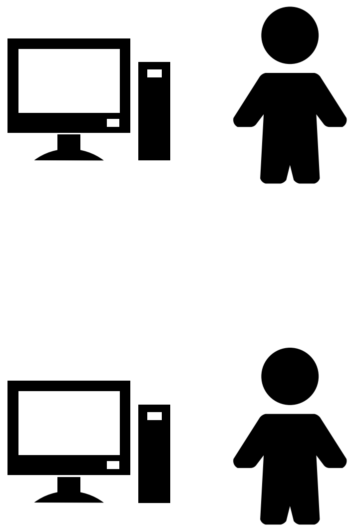


A game of 2 machines and 3 users/jobs

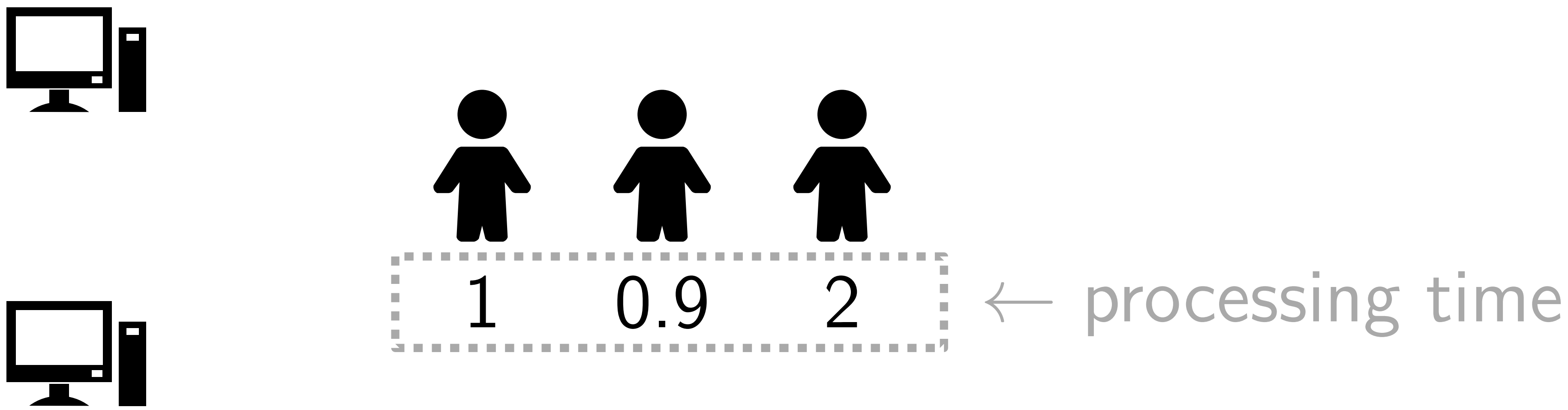


Sequential Scheduling Games

A game of 2 machines and 2 users/jobs



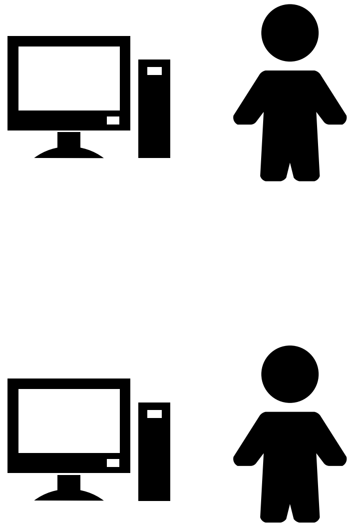
A game of 2 machines and 3 users/jobs



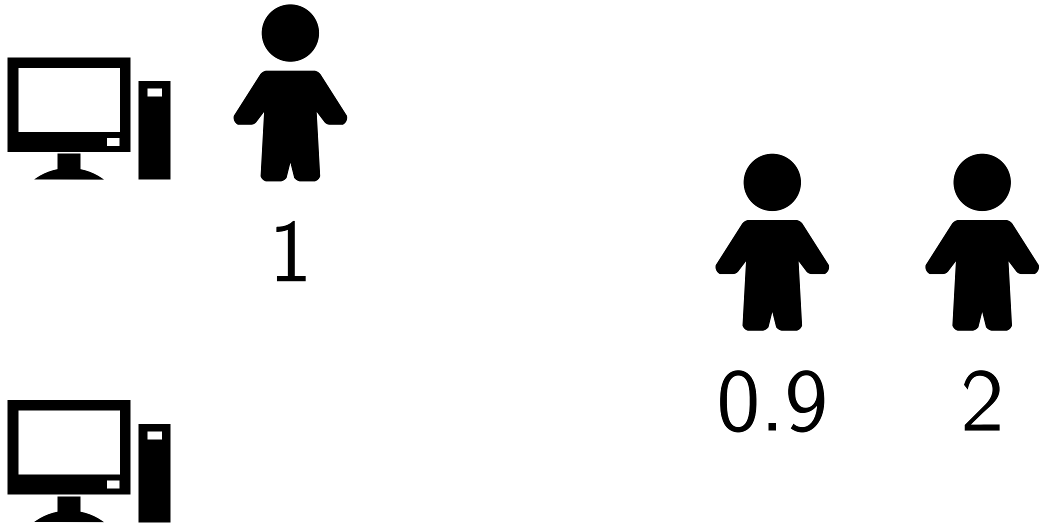
Individual cost: the load of the chosen machine

Sequential Scheduling Games

A game of 2 machines and 2 users/jobs



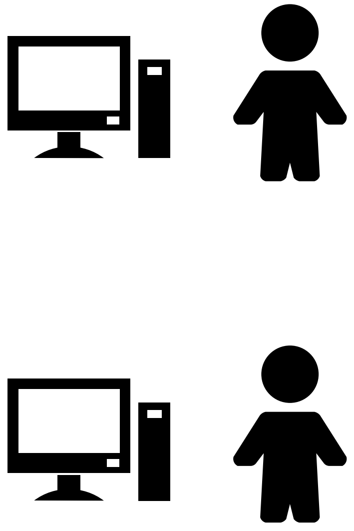
A game of 2 machines and 3 users/jobs

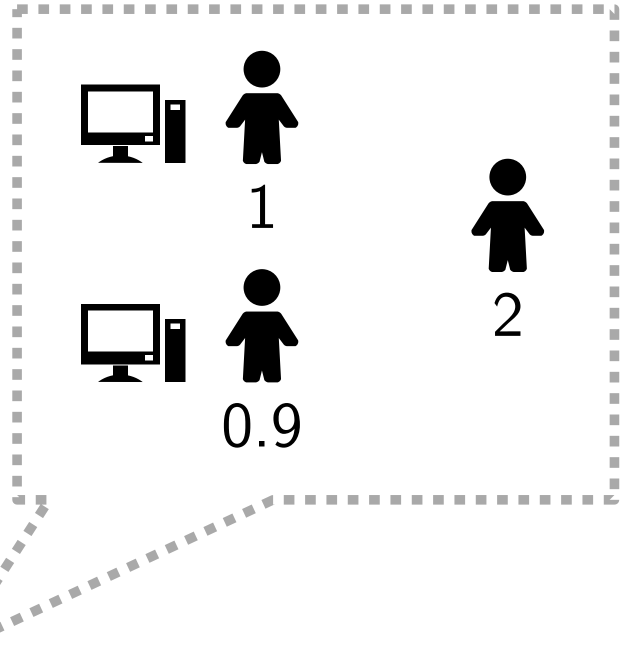


Individual cost: the load of the chosen machine

Sequential Scheduling Games

A game of 2 machines and 2 users/jobs



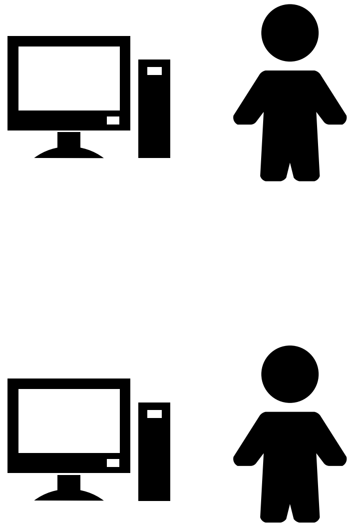


A game of 2 machines and 3 users/jobs

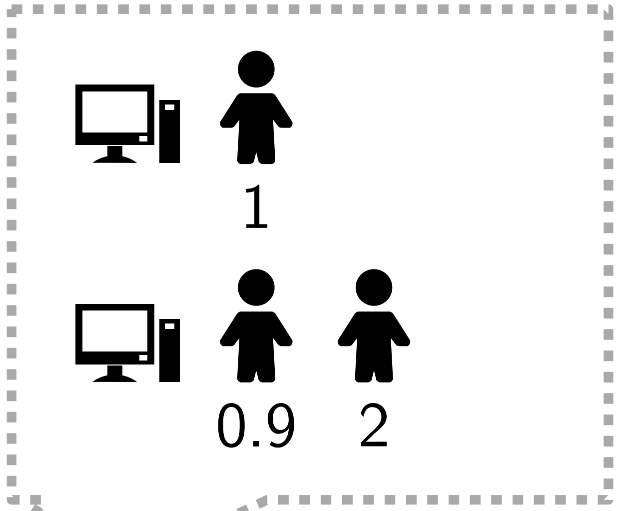
Individual cost: the load of the chosen machine

Sequential Scheduling Games

A game of 2 machines and 2 users/jobs



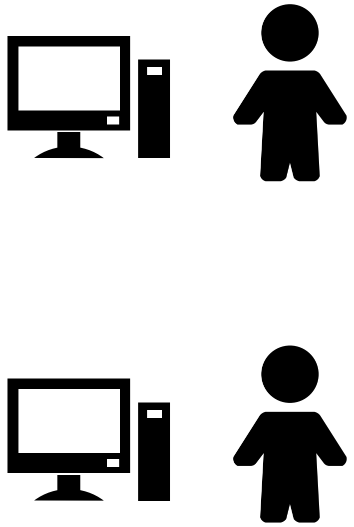
A game of 2 machines and 3 users/jobs



Individual cost: the load of the chosen machine

Sequential Scheduling Games

A game of 2 machines and 2 users/jobs

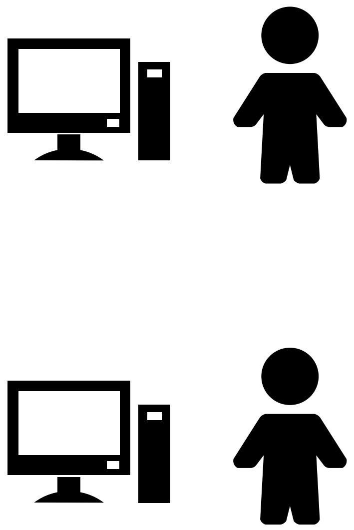


A game of 2 machines and 3 users/jobs

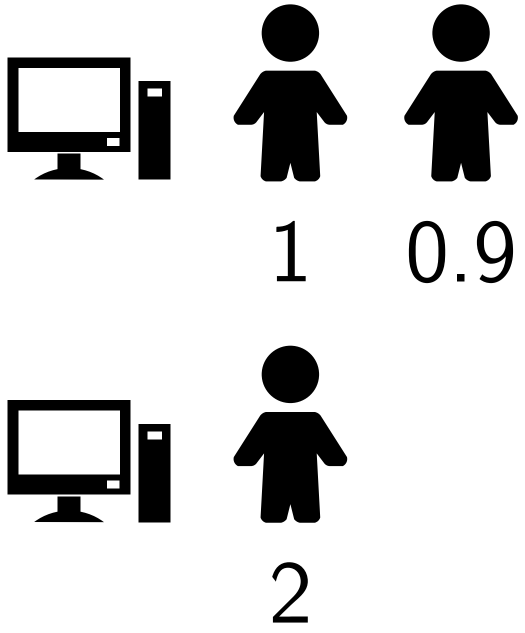
Individual cost: the load of the chosen machine

Sequential Scheduling Games

A game of 2 machines and 2 users/jobs



A game of 2 machines and 3 users/jobs



Individual cost: the load of the chosen machine

Sequential Scheduling Games

A game of 2 machines and 3 jobs: ***unrelated machines setting***

0	1	2
2	1	1

Sequential Scheduling Games

A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Sequential Scheduling Games

A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

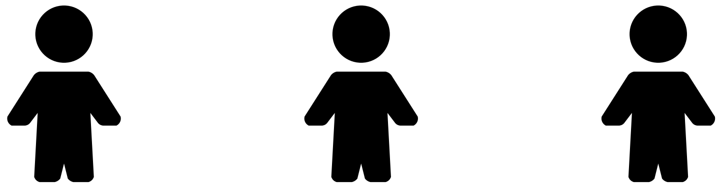
Game tree

↓

Backward induction

Sequential Scheduling Games

A game of 2 machines and 3 jobs: ***unrelated machines setting***

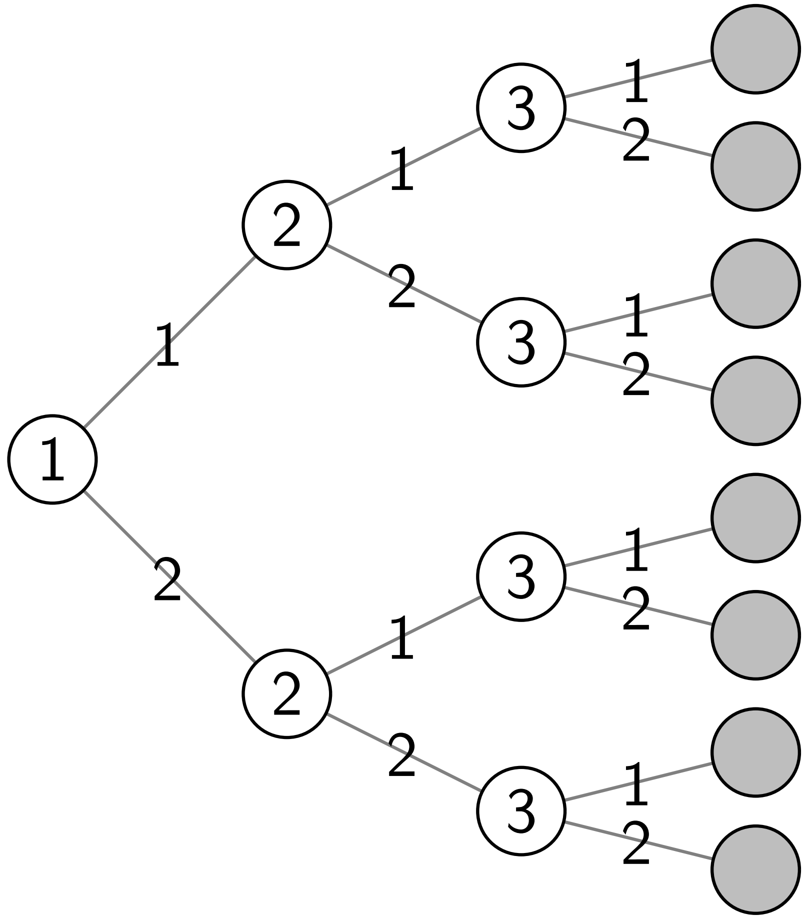


$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree

↓

Backward induction



Sequential Scheduling Games

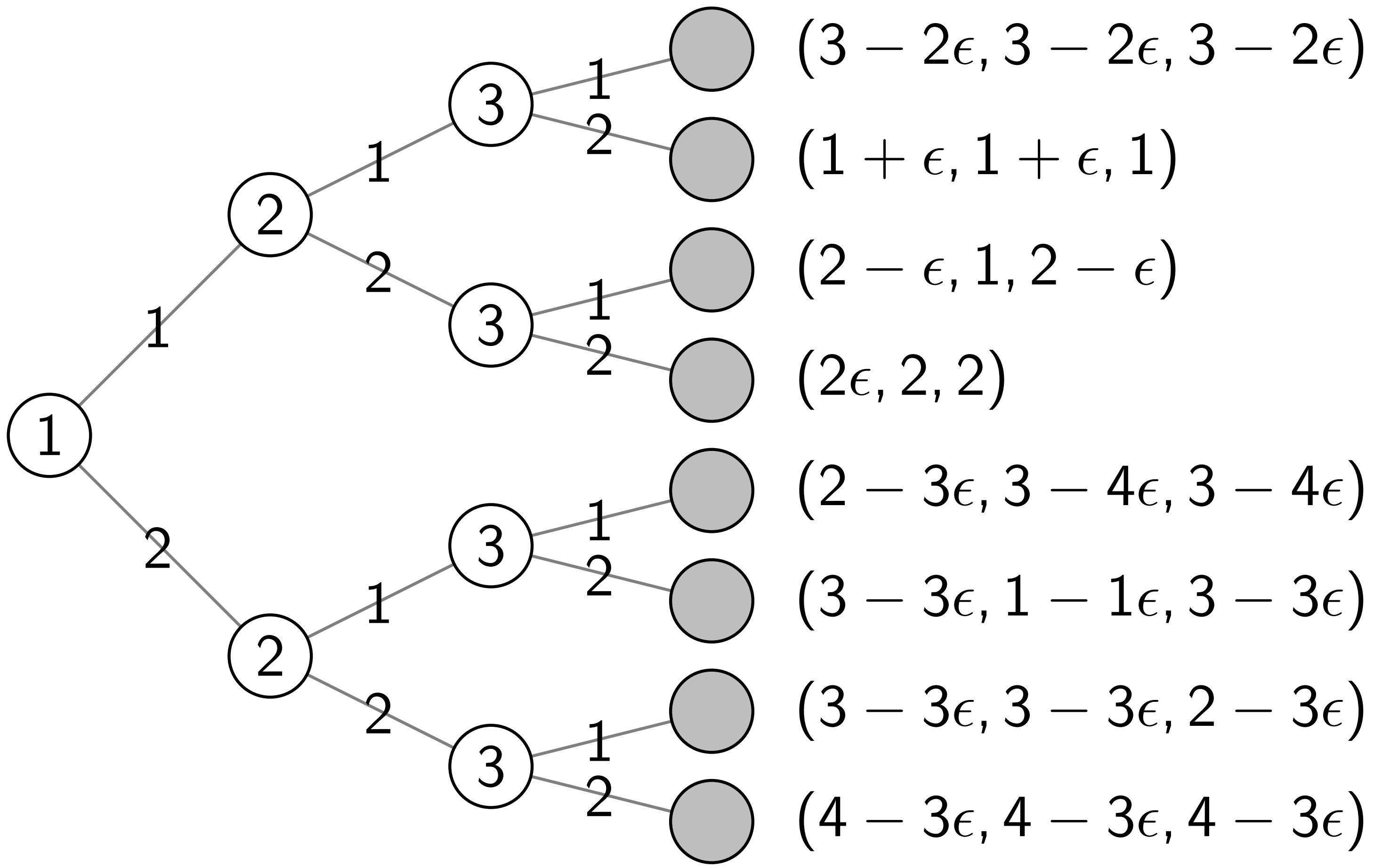
A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree

↓

Backward induction



Sequential Scheduling Games

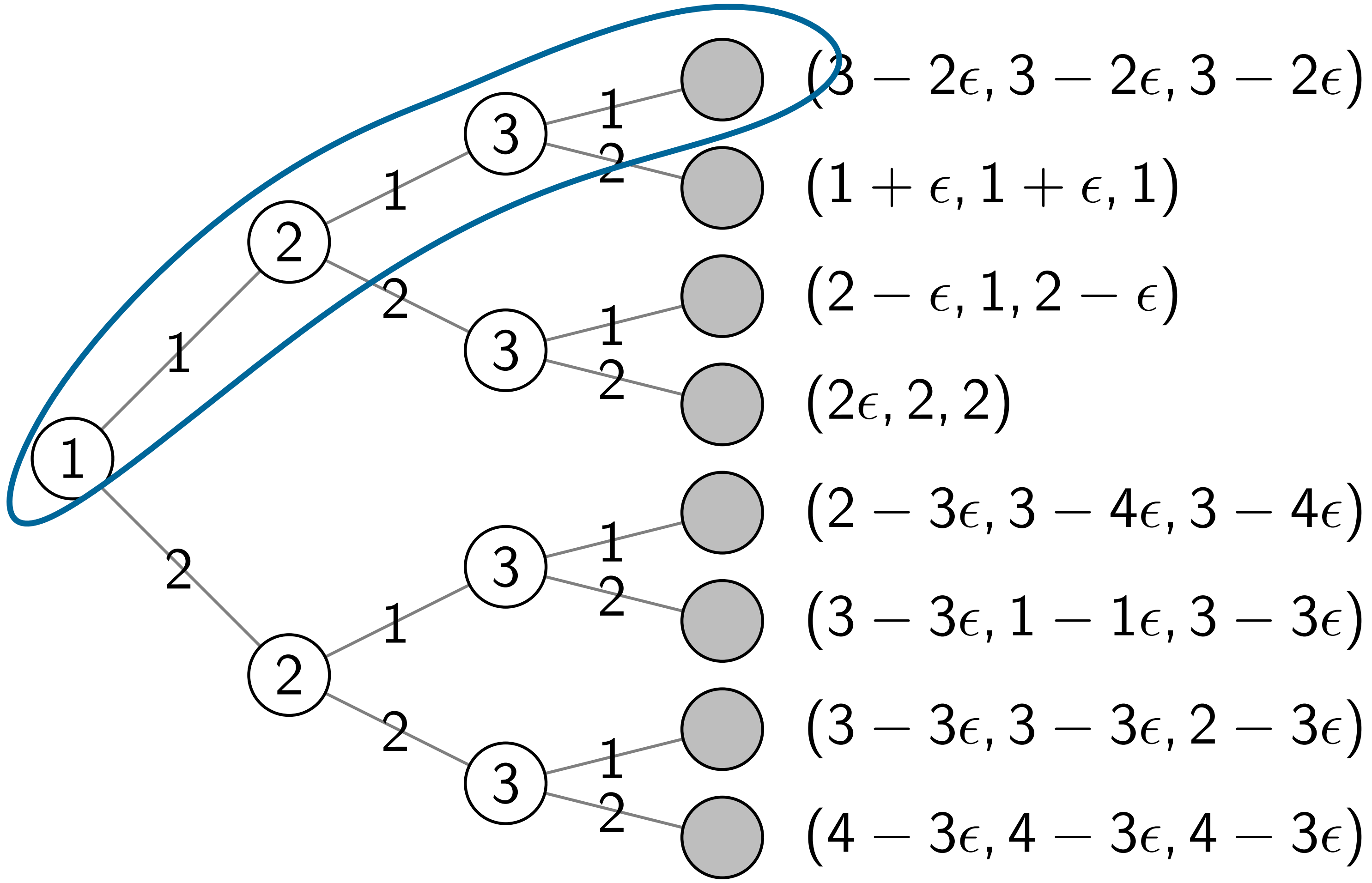
A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree

↓

Backward induction



Sequential Scheduling Games

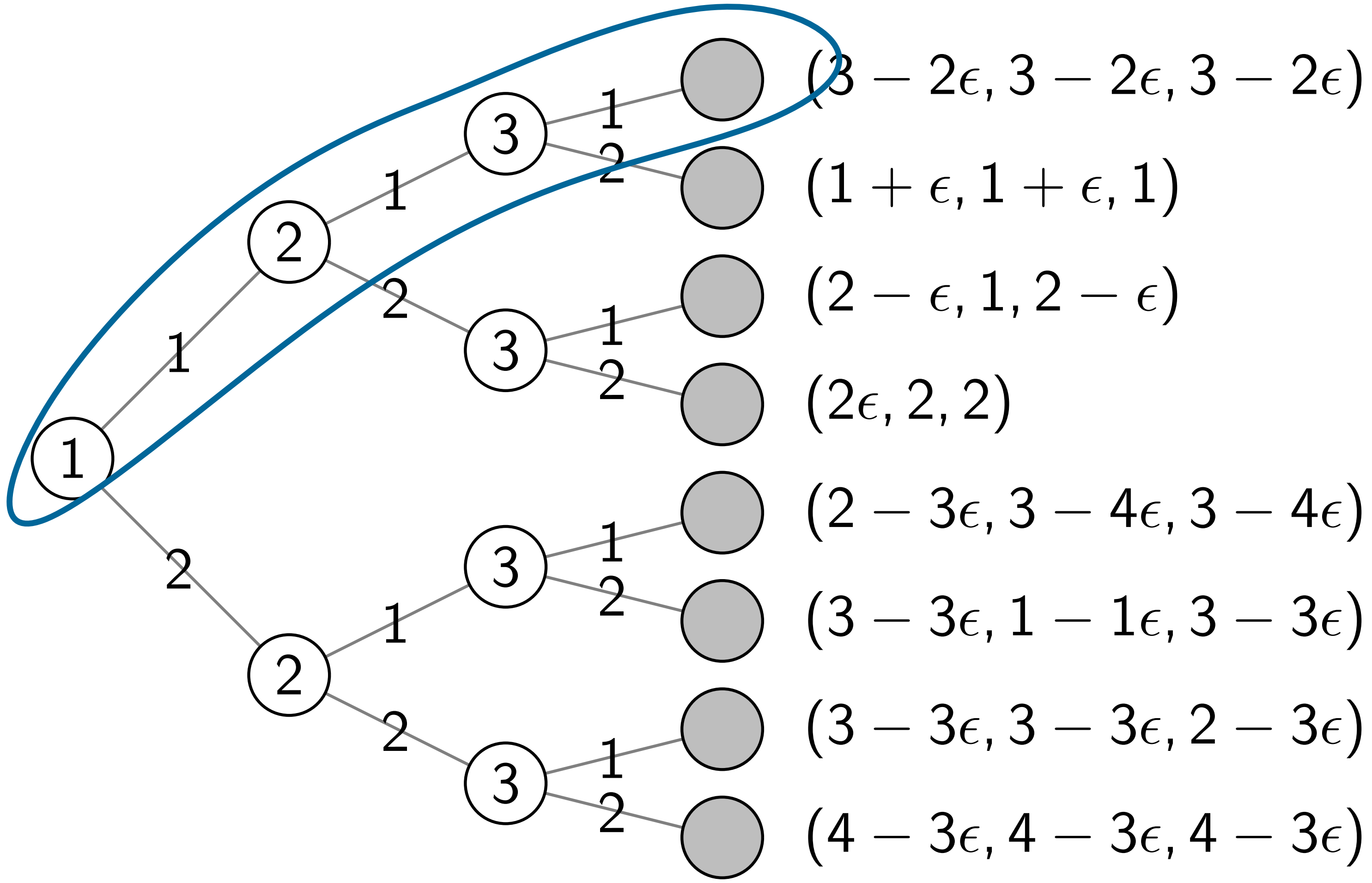
A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree

↓

Backward induction



Sequential Scheduling Games

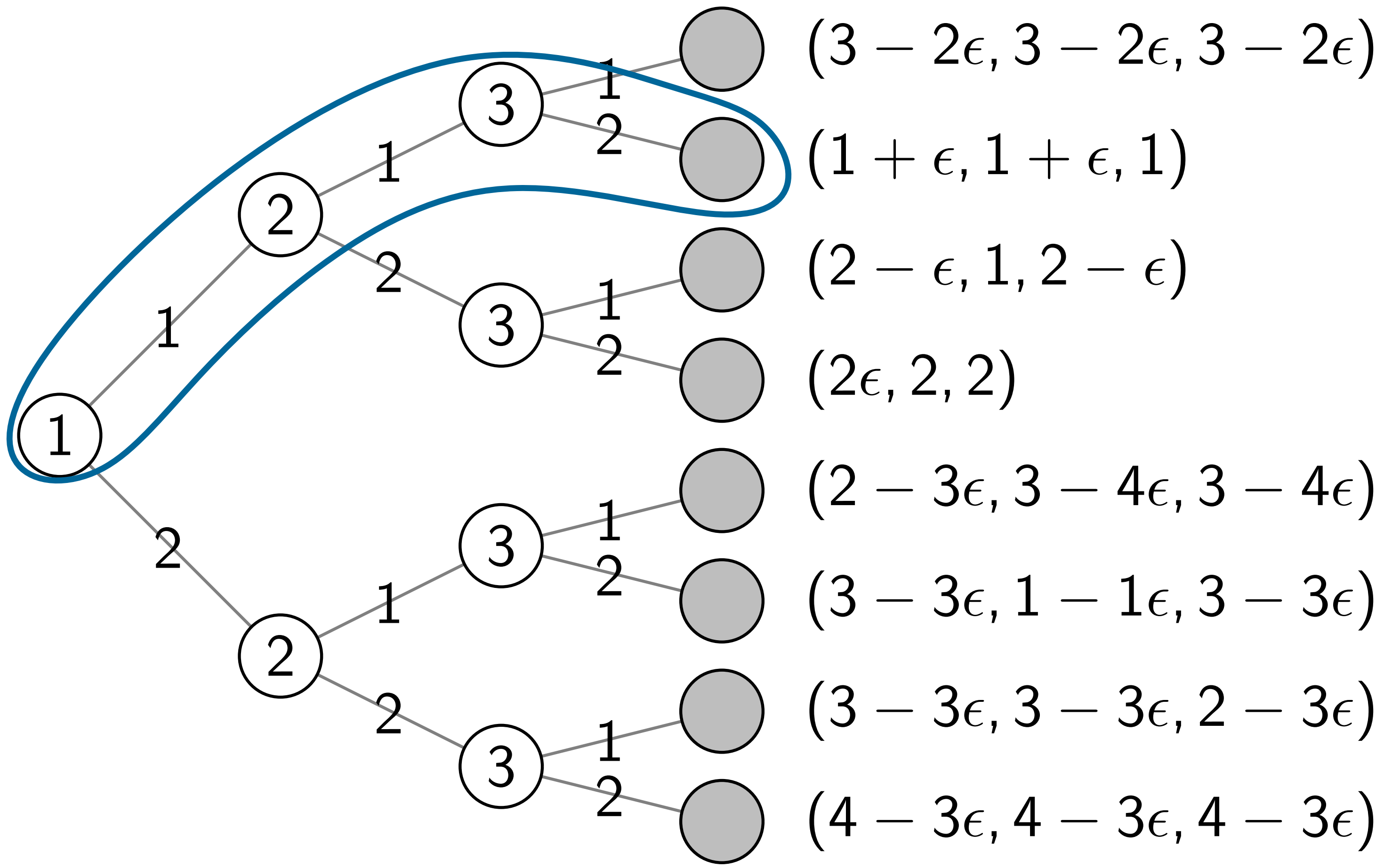
A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree

↓

Backward induction



Sequential Scheduling Games

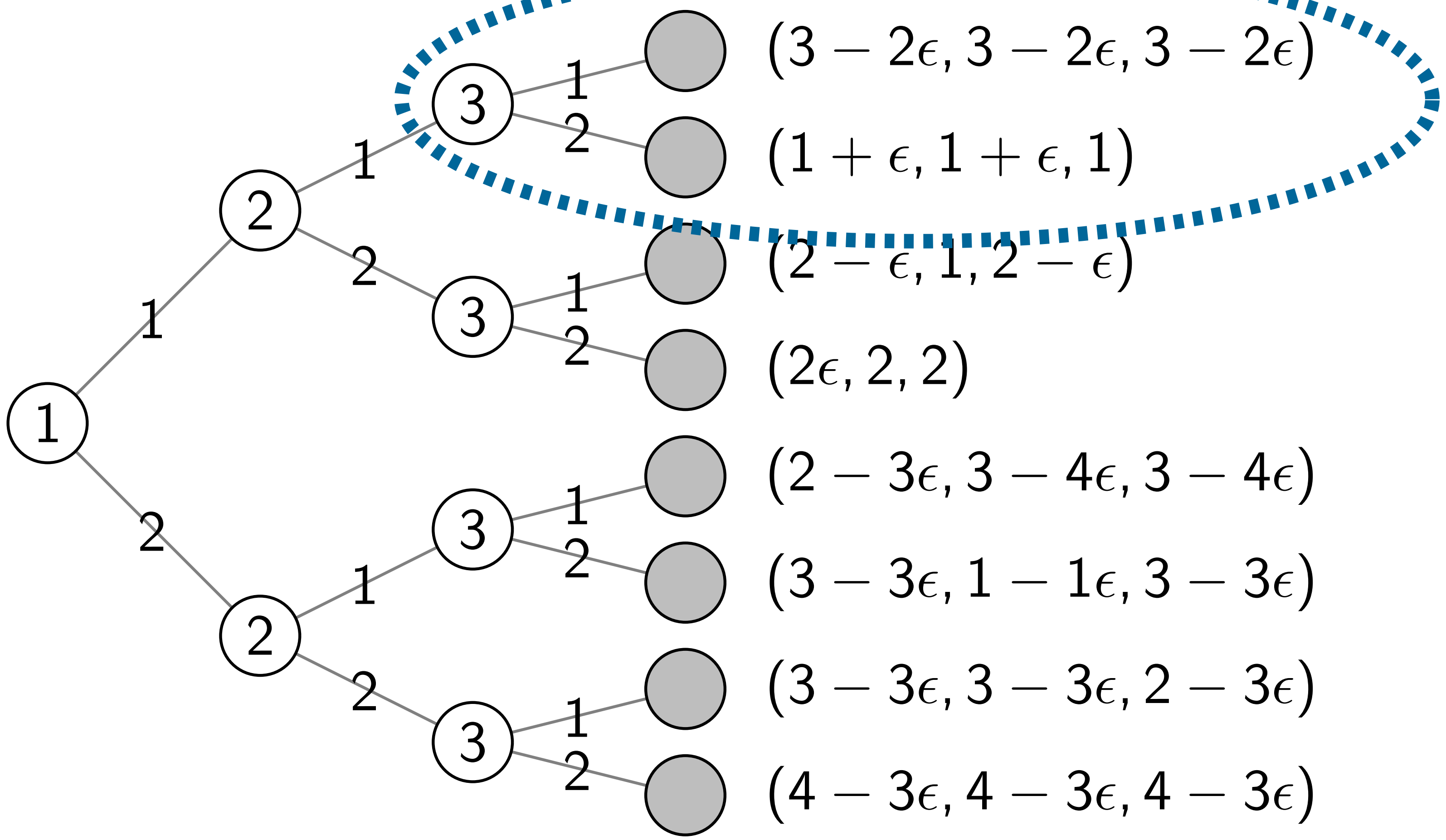
A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree

↓

Backward induction



Sequential Scheduling Games

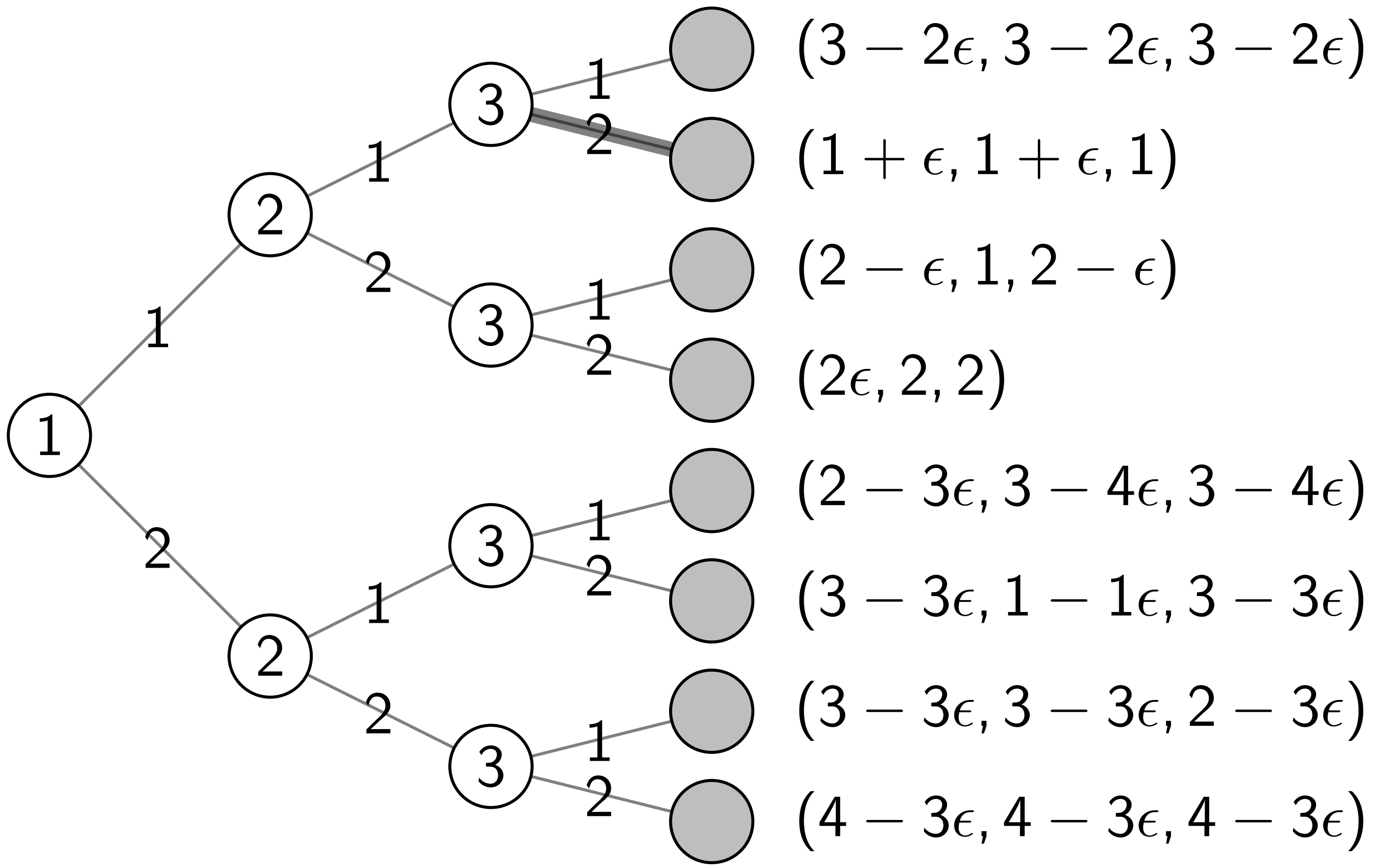
A game of 2 machines and 3 jobs: ***unrelated machines setting***

Machine 1	$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
Machine 2	$2-3\epsilon$	1	1

Game tree

↓

Backward induction



Sequential Scheduling Games

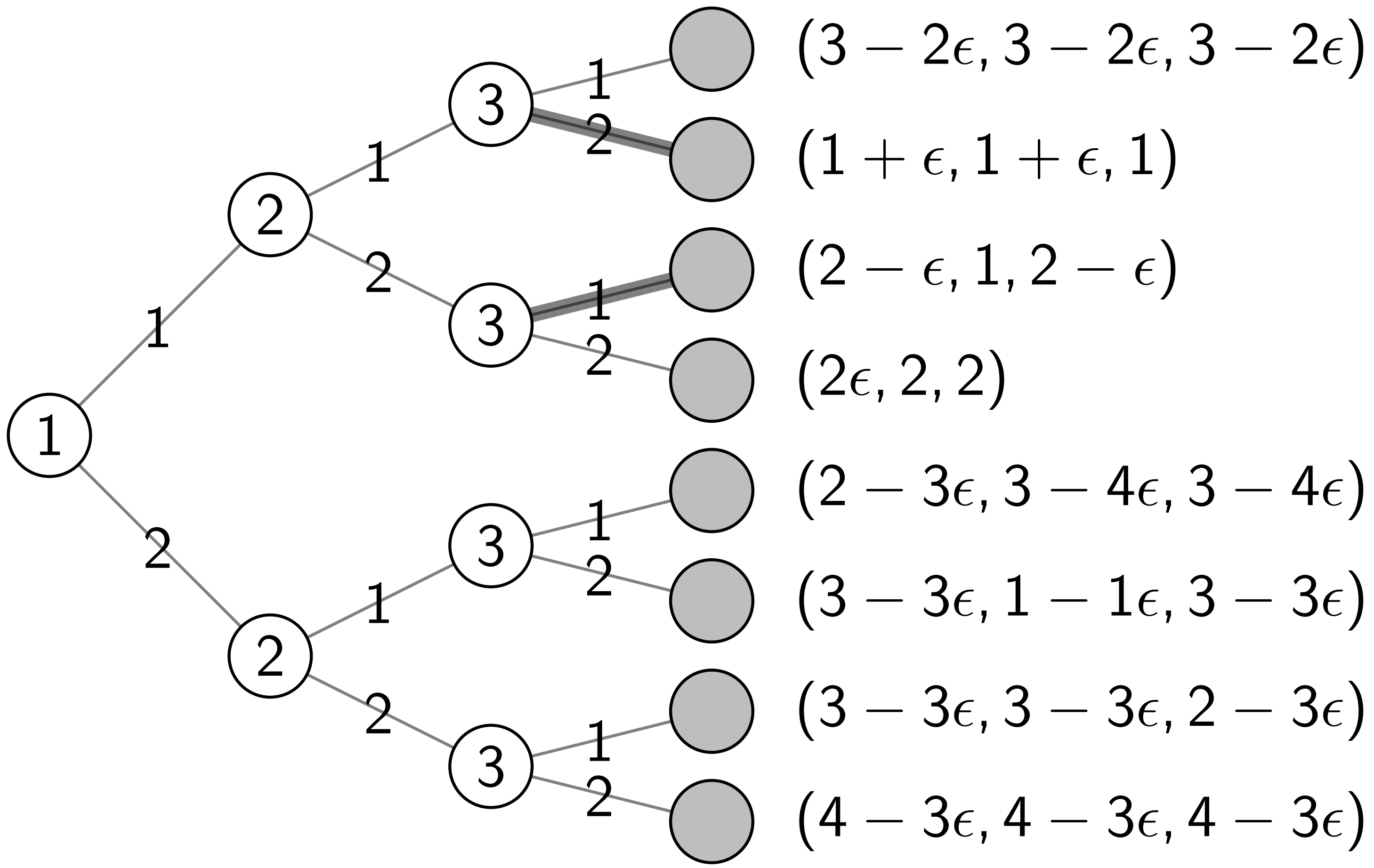
A game of 2 machines and 3 jobs: ***unrelated machines setting***

Machine 1	$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
Machine 2	$2-3\epsilon$	1	1

Game tree

↓

Backward induction



Sequential Scheduling Games

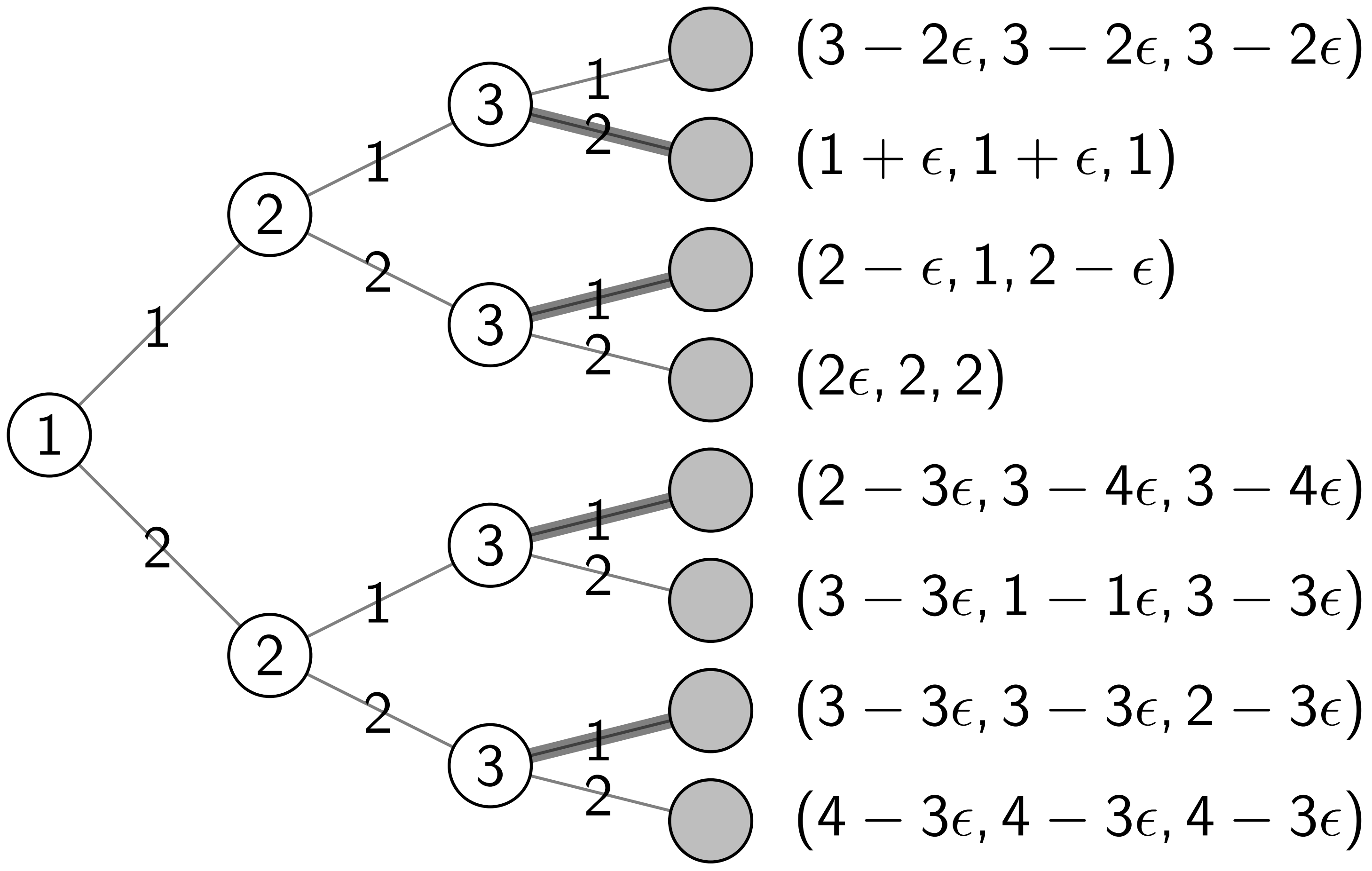
A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree

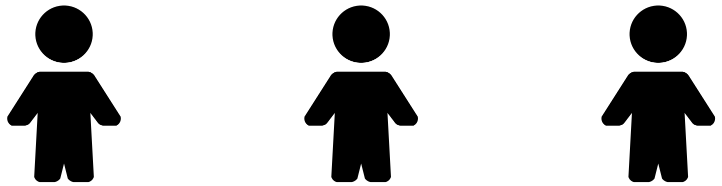
↓

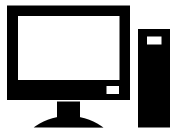
Backward induction

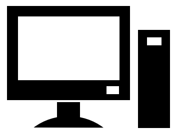


Sequential Scheduling Games

A game of 2 machines and 3 jobs: ***unrelated machines setting***



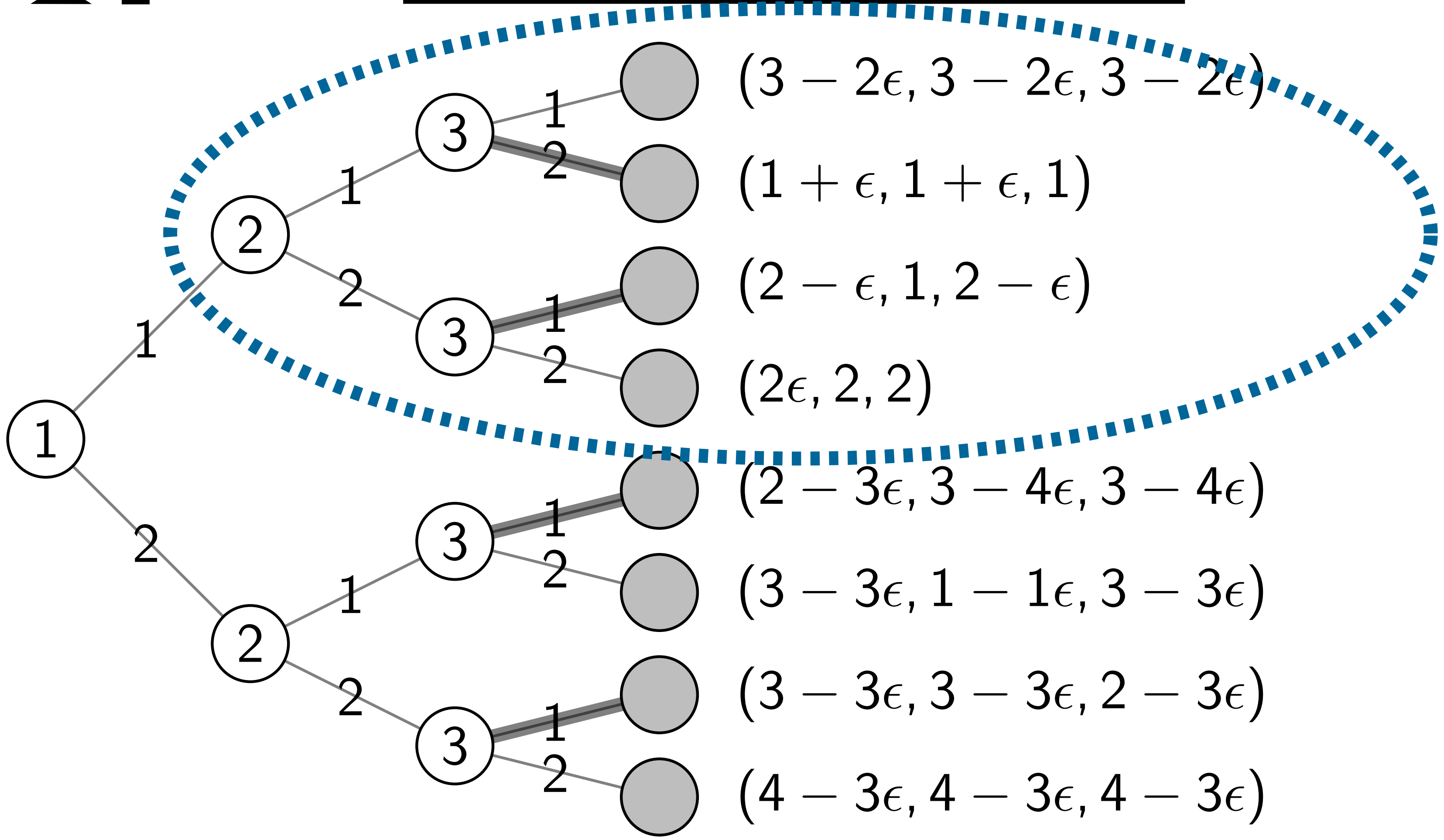




$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

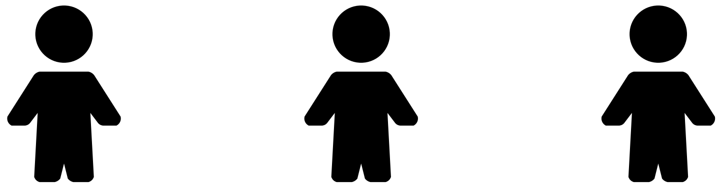
Game tree

Backward induction



Sequential Scheduling Games

A game of 2 machines and 3 jobs: ***unrelated machines setting***

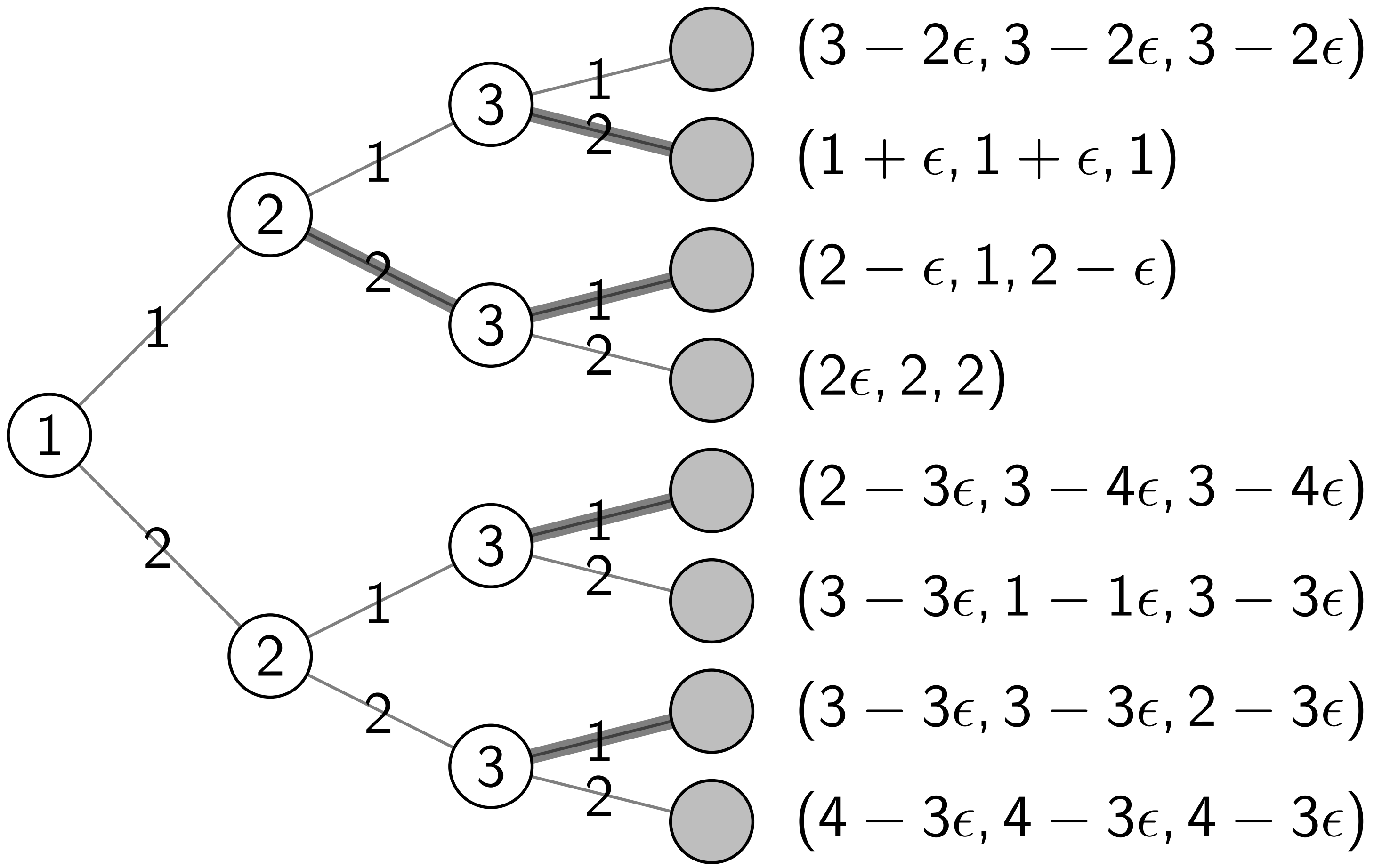


$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree

↓

Backward induction

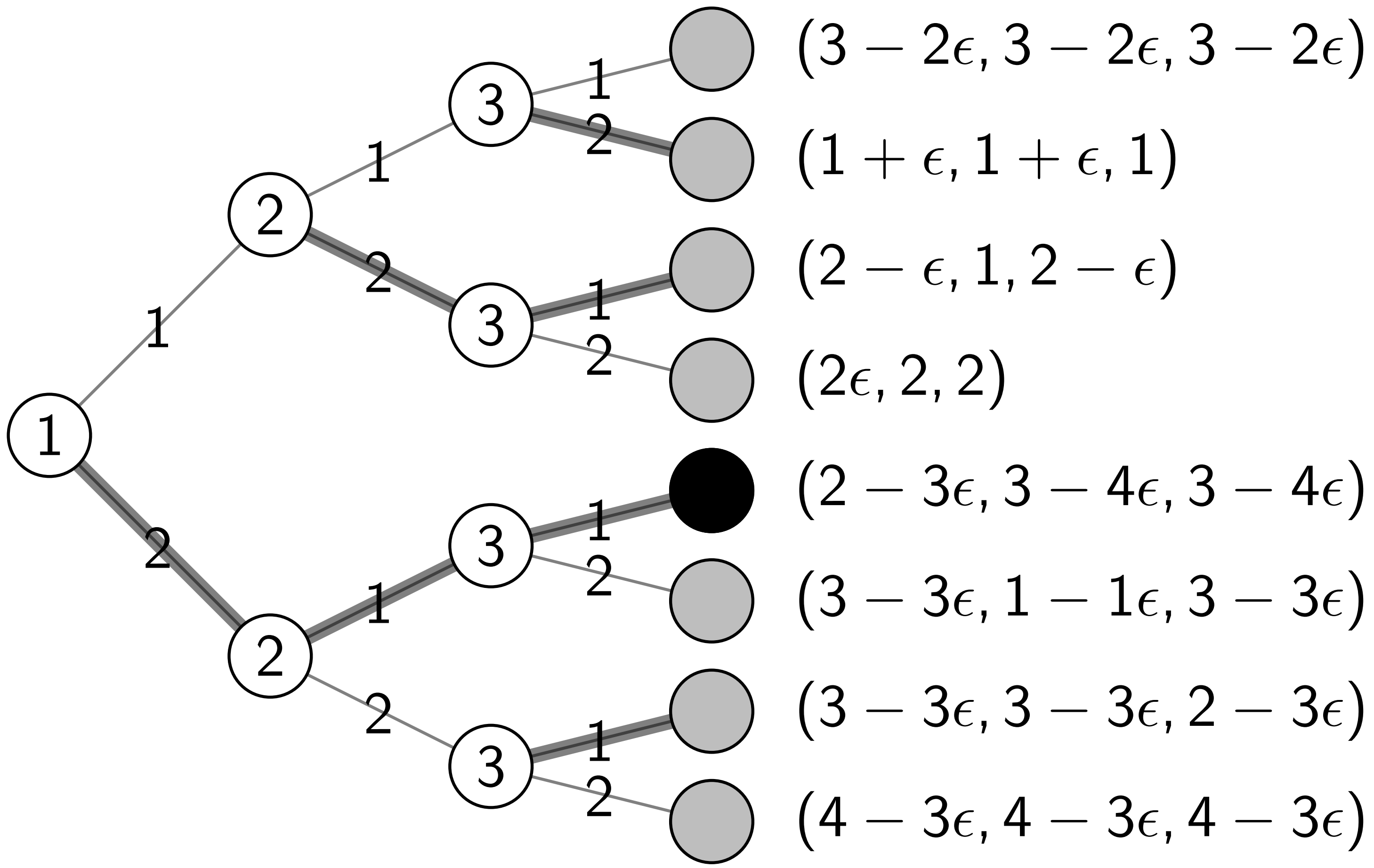


Sequential Scheduling Games

A game of 2 machines and 3 jobs: ***unrelated machines setting***

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

Game tree
↓
Backward induction



Sequential Scheduling Games

Sequential scheduling game on unrelated machines

- Players: jobs $N = \{1, 2, \dots, n\}$
- Strategies: machines $M = \{1, 2, \dots, m\}$

Sequential Scheduling Games

Sequential scheduling game on unrelated machines

- Players: jobs $N = \{1, 2, \dots, n\}$
- Strategies: machines $M = \{1, 2, \dots, m\}$
- Processing time: each job j will take $p_{i,j}$ units of time if processed by machine i

Sequential Scheduling Games

Sequential scheduling game on unrelated machines

- Players: jobs $N = \{1, 2, \dots, n\}$
- Strategies: machines $M = \{1, 2, \dots, m\}$
- Processing time: each job j will take $p_{i,j}$ units of time if processed by machine i
- The self-interested jobs **sequentially** choose one of the machines for processing
- Individual cost of each job: the **load** of the choisen machine, where the load of a machine is the total processing times of the jobs processed on it.
- Farsighted players: when a job makes decision, he knows the choices made by his predecessors as well as the processing times of his successors.

Sequential Scheduling Games

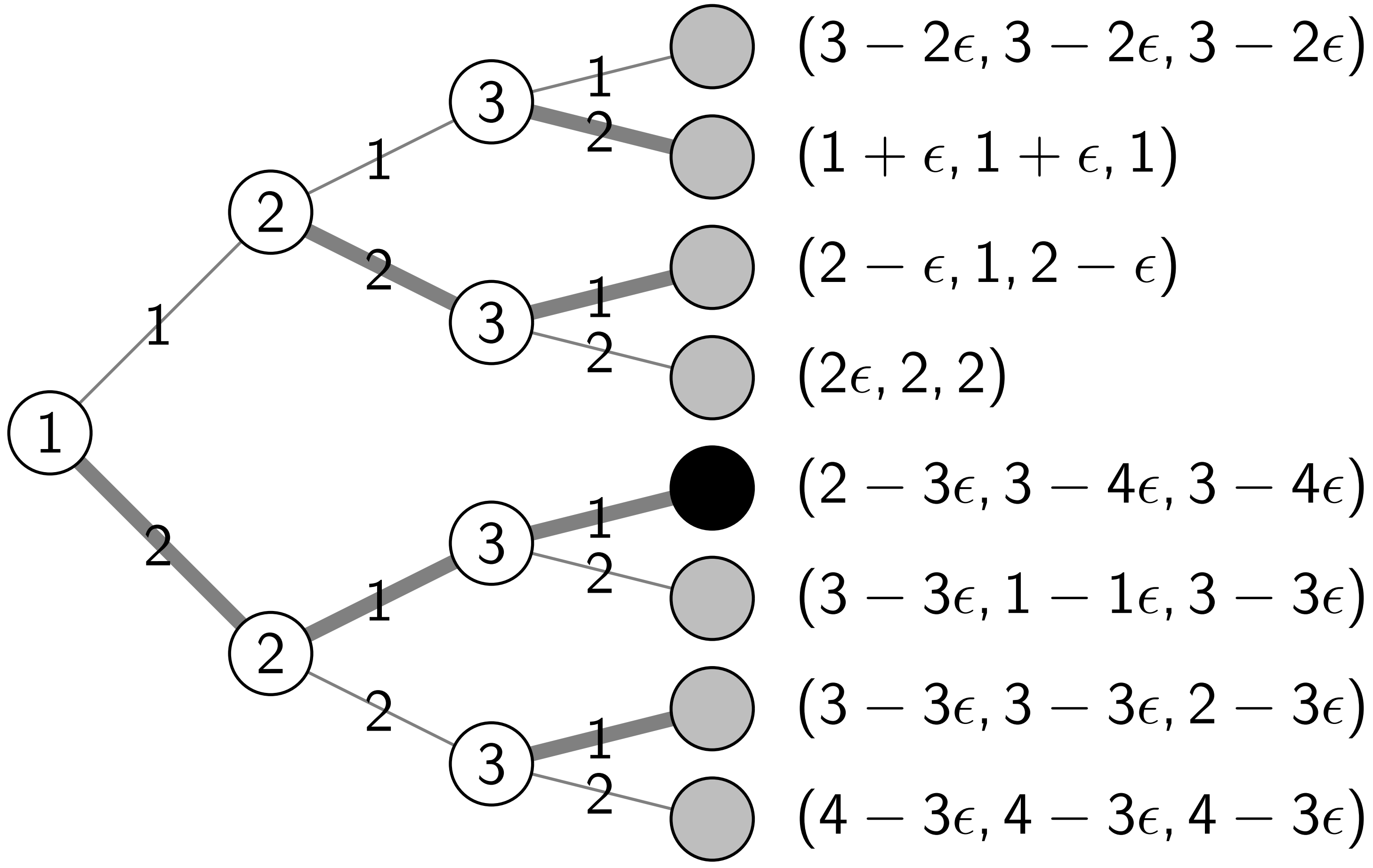
Sequential scheduling game on unrelated machines

- Players: jobs $N = \{1, 2, \dots, n\}$
- Strategies: machines $M = \{1, 2, \dots, m\}$
- Processing time: each job j will take $p_{i,j}$ units of time if processed by machine i
- The self-interested jobs **sequentially** choose one of the machines for processing
- Individual cost of each job: the **load** of the choisen machine, where the load of a machine is the total processing times of the jobs processed on it.
- Farsighted players: when a job makes decision, he knows the choices made by his predecessors as well as the processing times of his successors.
- The game always possesses **subgame-perfect equilibria (SPE)**, which can be calculated by backward induction.

Motivations

Computational ablility of players

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

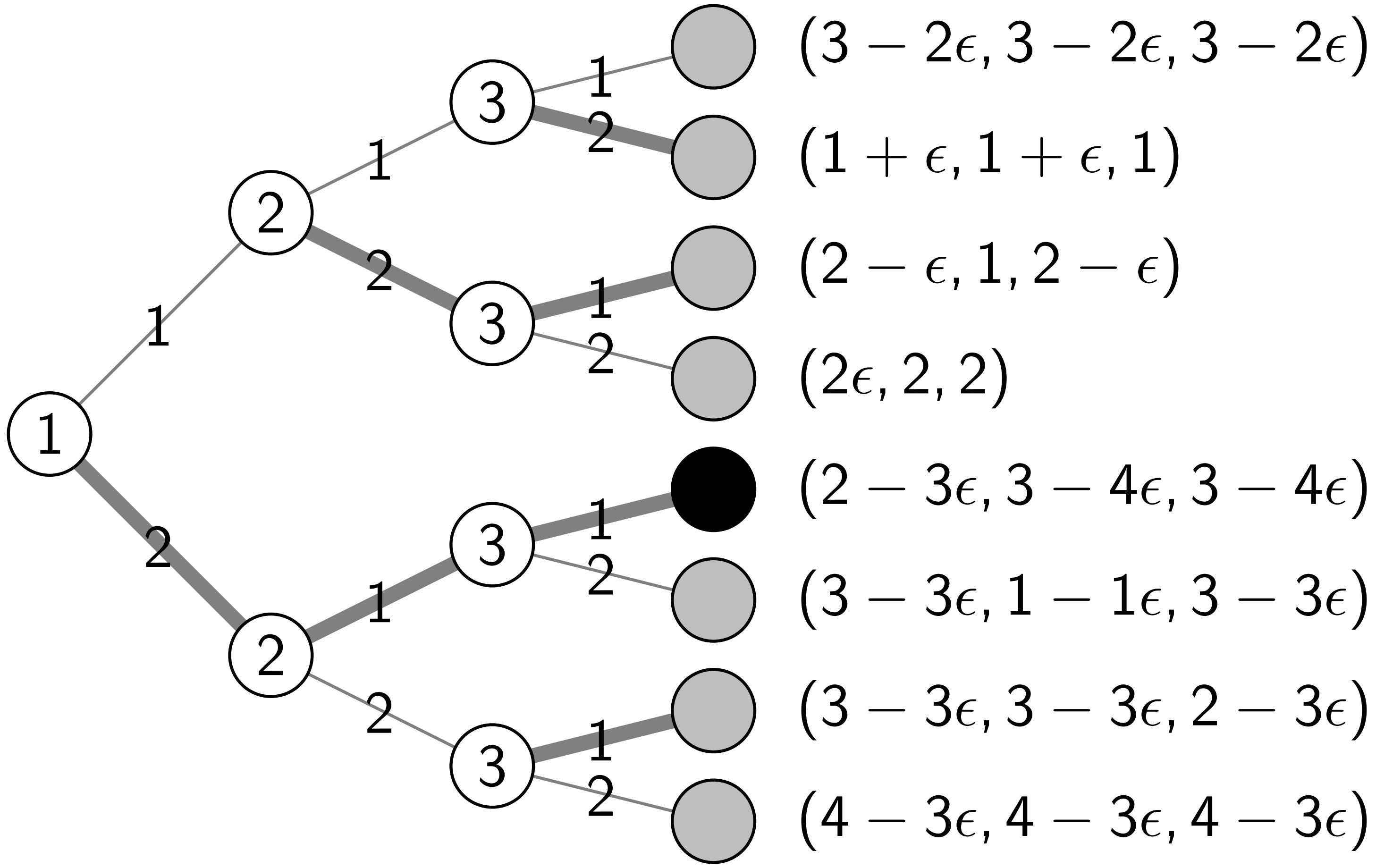


Motivations

Computational ablility of players

- $2^3 = 8$ leaves for 2 machines and 3 jobs
- m^n leaves for m machines and n jobs

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1

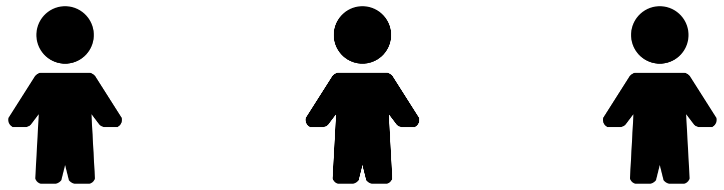


Motivations

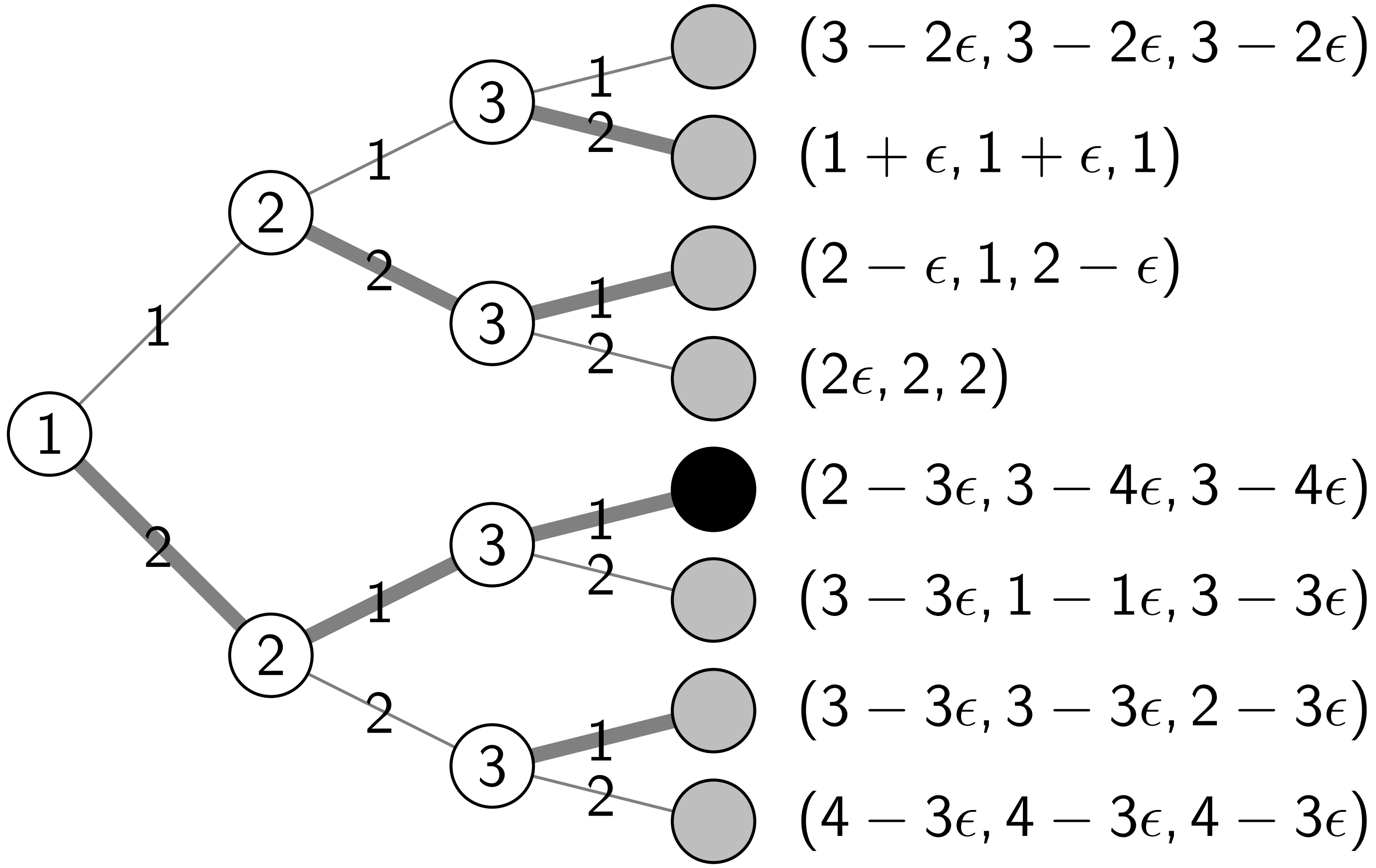
Computational ablility of players

- $2^3 = 8$ leaves for 2 machines and 3 jobs
- m^n leaves for m machines and n jobs

> - [Leme, Syrgkanis,Tardos, 2012]:
>
> Computing a SPE for the unrelated machine scheduling game is PSPACE-complete.



$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1



Motivations

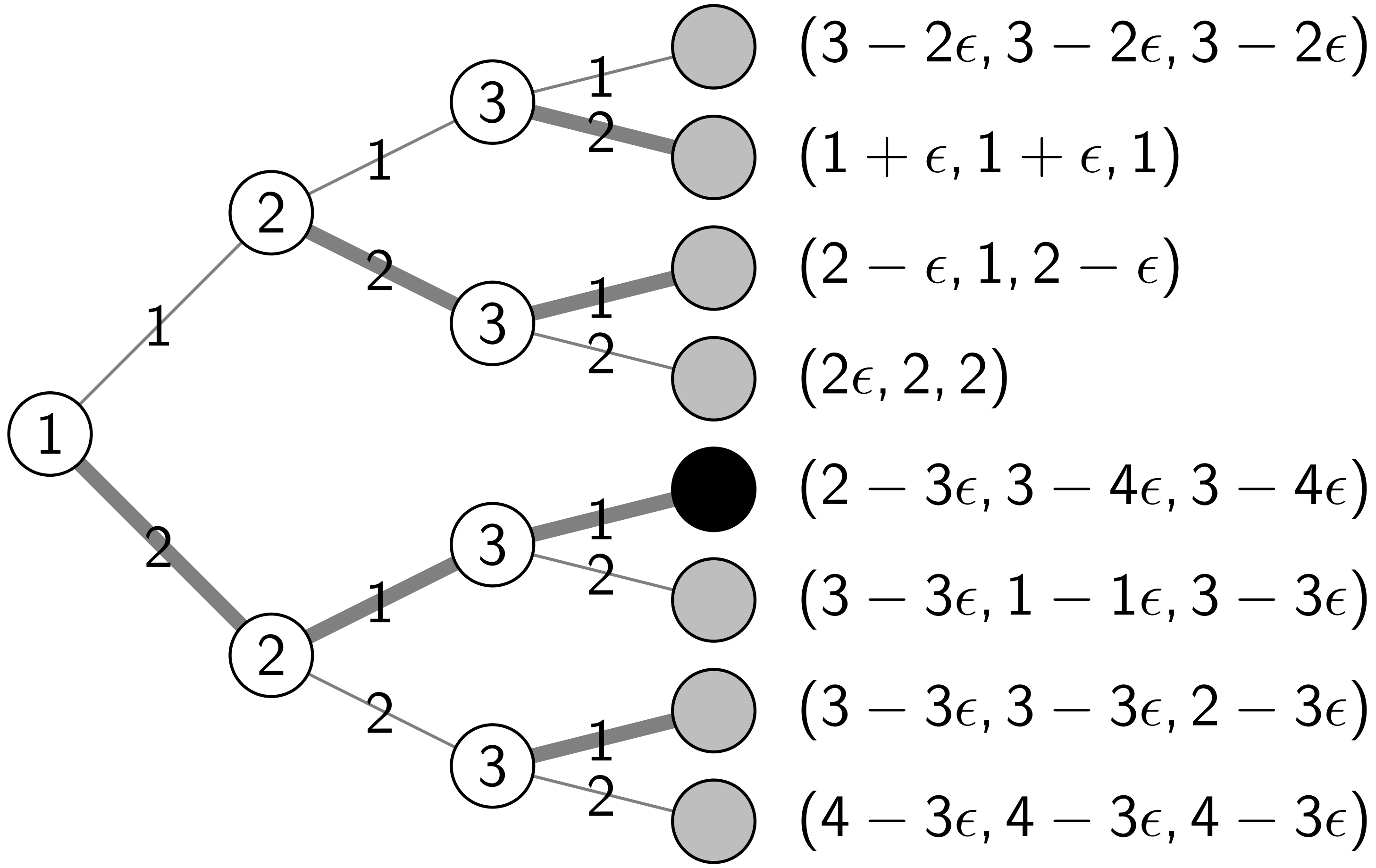
Computational ablility of players

- $2^3 = 8$ leaves for 2 machines and 3 jobs
- m^n leaves for m machines and n jobs

> - [Leme, Syrgkanis, Tardos, 2012]:
>
> Computing a SPE for the unrelated machine scheduling game is PSPACE-complete.

- It is hard for players to make an optimal decision

$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1



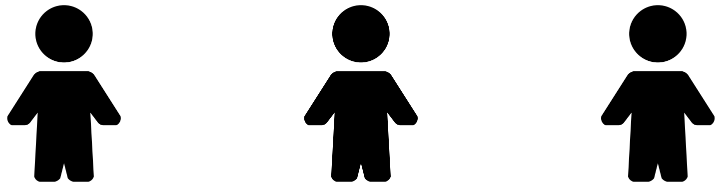
Motivations

Computational ablility of players

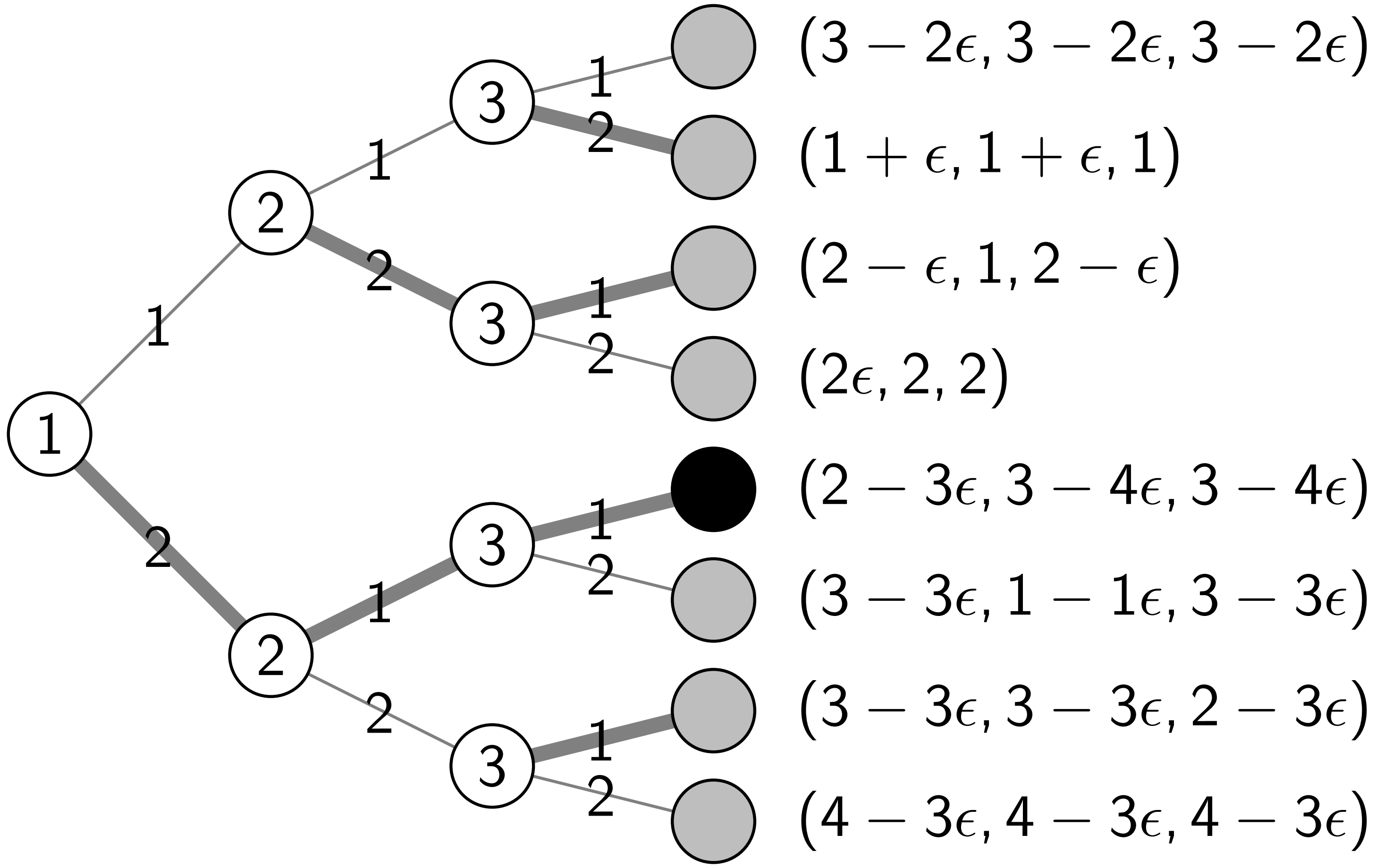
- $2^3 = 8$ leaves for 2 machines and 3 jobs
- m^n leaves for m machines and n jobs

> - [Leme, Syrgkanis, Tardos, 2012]:
>
> Computing a SPE for the unrelated machine scheduling game is PSPACE-complete.

- It is hard for players to make an optimal decision
- It is very likely for some player to deviate from the SPE solution



$0+2\epsilon$	$1-\epsilon$	$2-3\epsilon$
$2-3\epsilon$	1	1



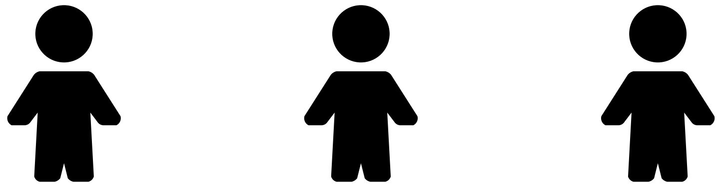
Motivations

Computational ablility of players

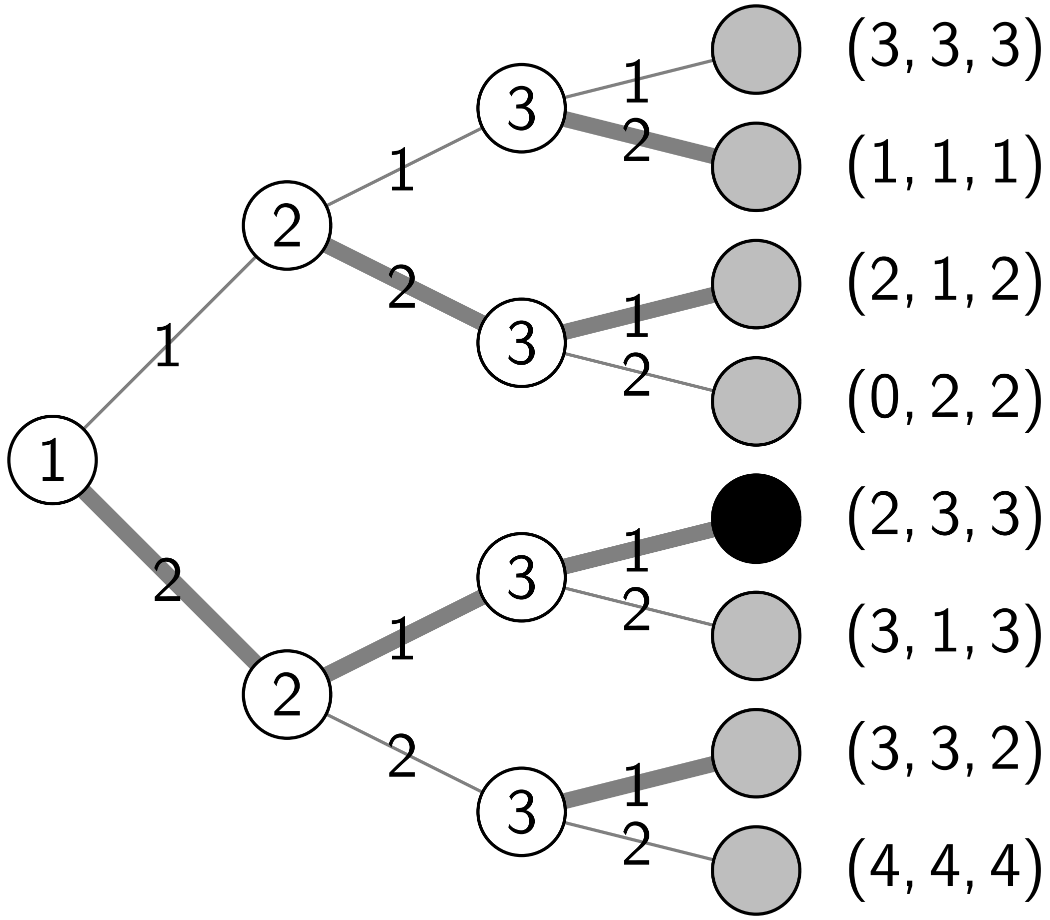
- $2^3 = 8$ leaves for 2 machines and 3 jobs
- m^n leaves for m machines and n jobs

> - [Leme, Syrgkanis,Tardos, 2012]:
>
> Computing a SPE for the unrelated machine scheduling game is PSPACE-complete.

- It is hard for players to make an optimal decision
- It is very likely for some player to deviate from the SPE solution



0	1	2
2	1	1



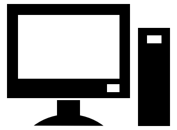
Motivations

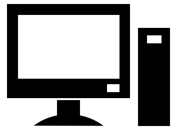
Computational ablility of players

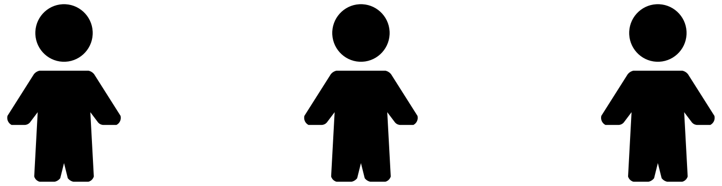
- $2^3 = 8$ leaves for 2 machines and 3 jobs
- m^n leaves for m machines and n jobs

> - [Leme, Syrgkanis,Tardos, 2012]:
>
> Computing a SPE for the unrelated machine scheduling game is PSPACE-complete.

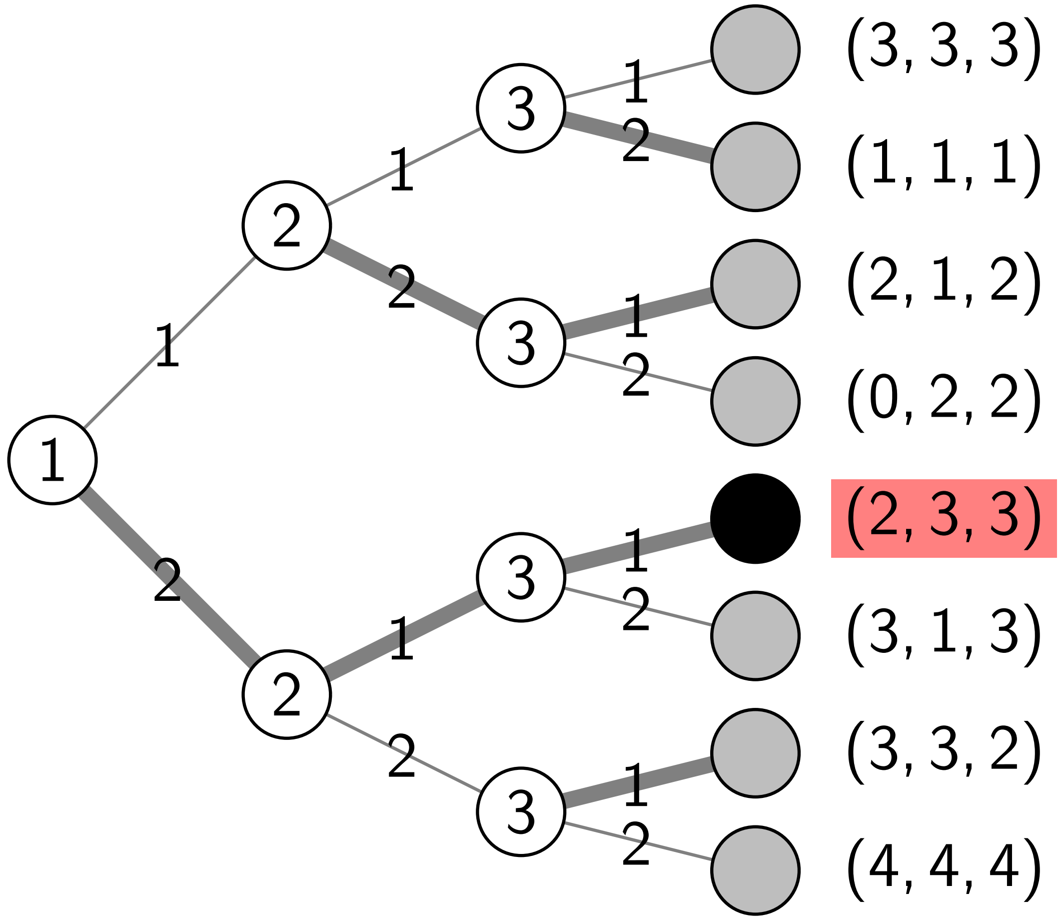
- It is hard for players to make an optimal decision
- It is very likely for some player to deviate from the SPE solution







0	1	2
2	1	1



Motivations

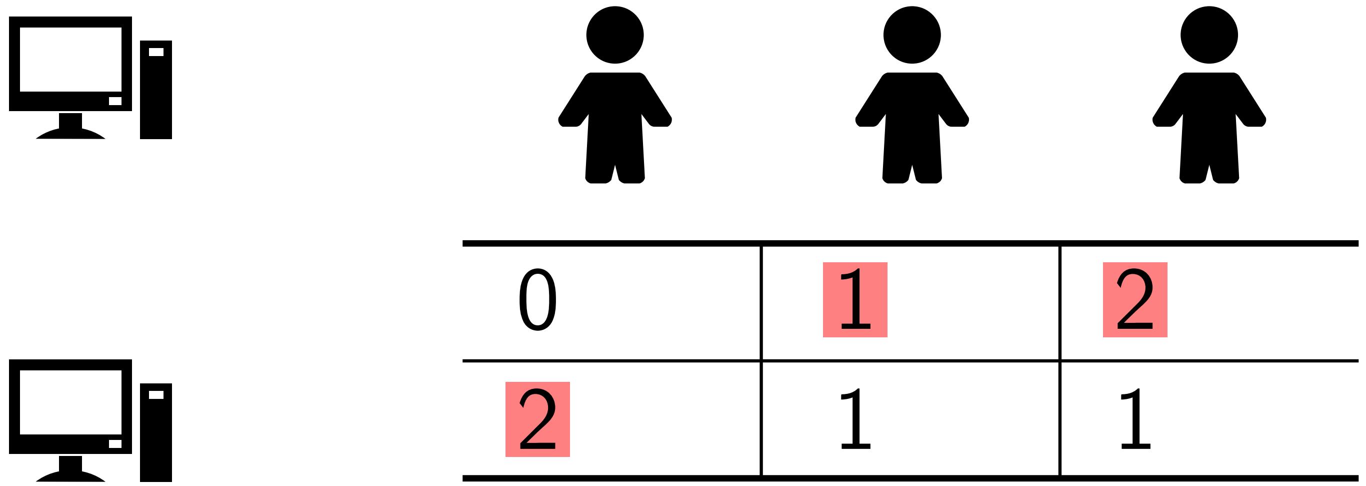
Computational ablility of players

- $2^3 = 8$ leaves for 2 machines and 3 jobs
- m^n leaves for m machines and n jobs

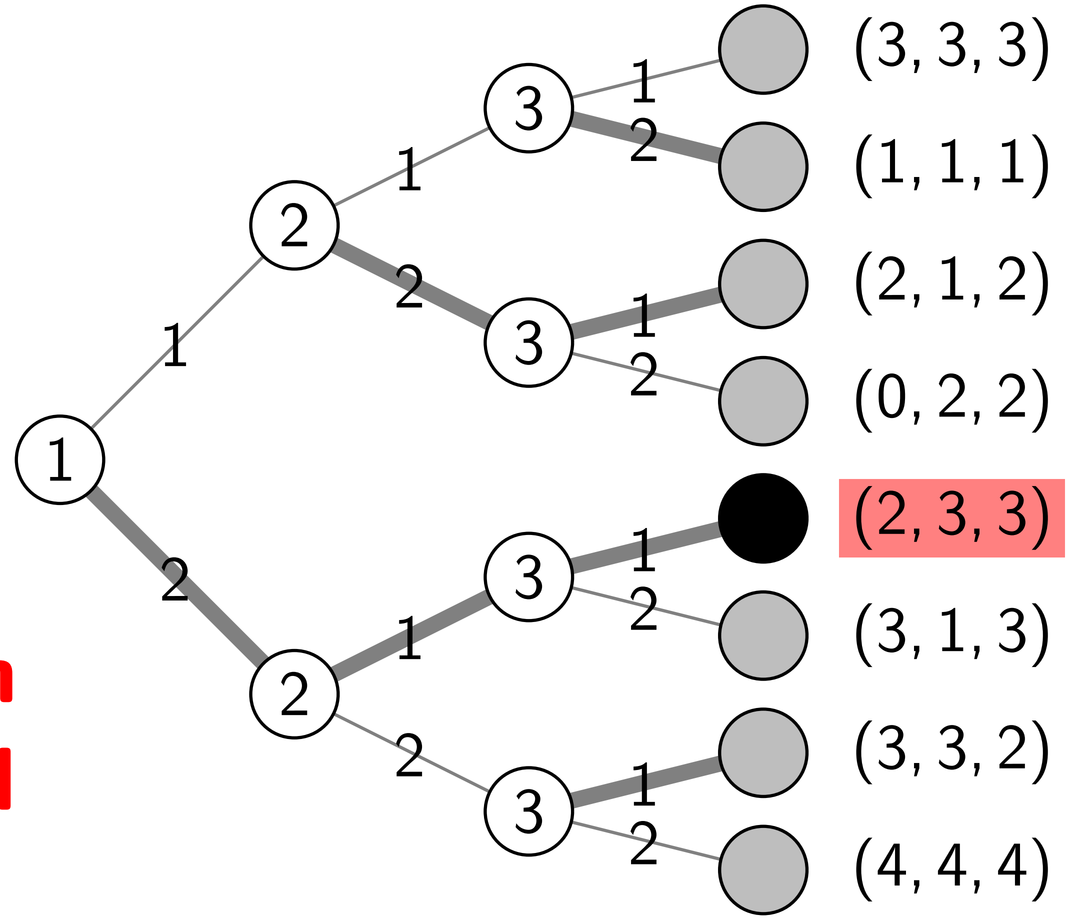
> - [Leme, Syrgkanis,Tardos, 2012]:
>
> Computing a SPE for the unrelated machine scheduling game is PSPACE-complete.

- It is hard for players to make an optimal decision
- It is very likely for some player to deviate from the SPE solution

It is reasonable to reconsider the common assumption that all players have unlimited computational capacity (full rationality)



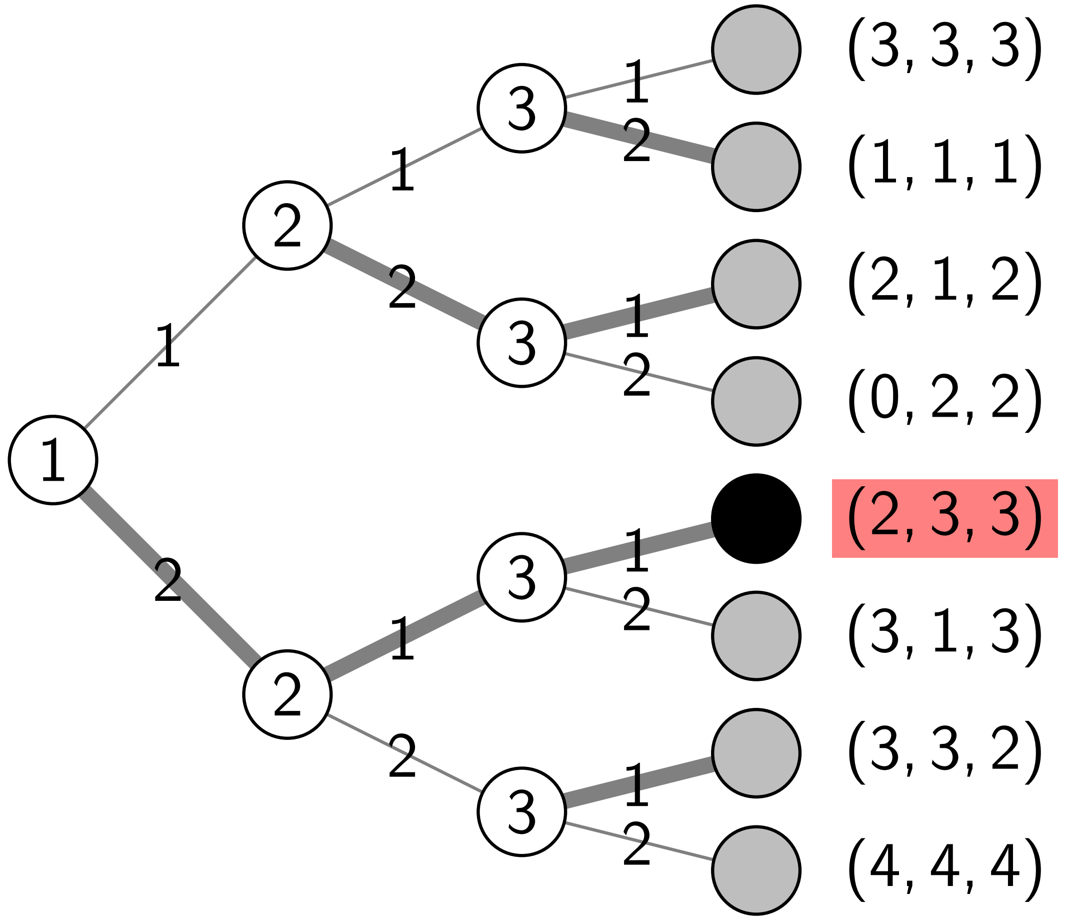
0	1	2
2	1	1



Motivations

Efficiency of equilibrium

0	1	2
2	1	1

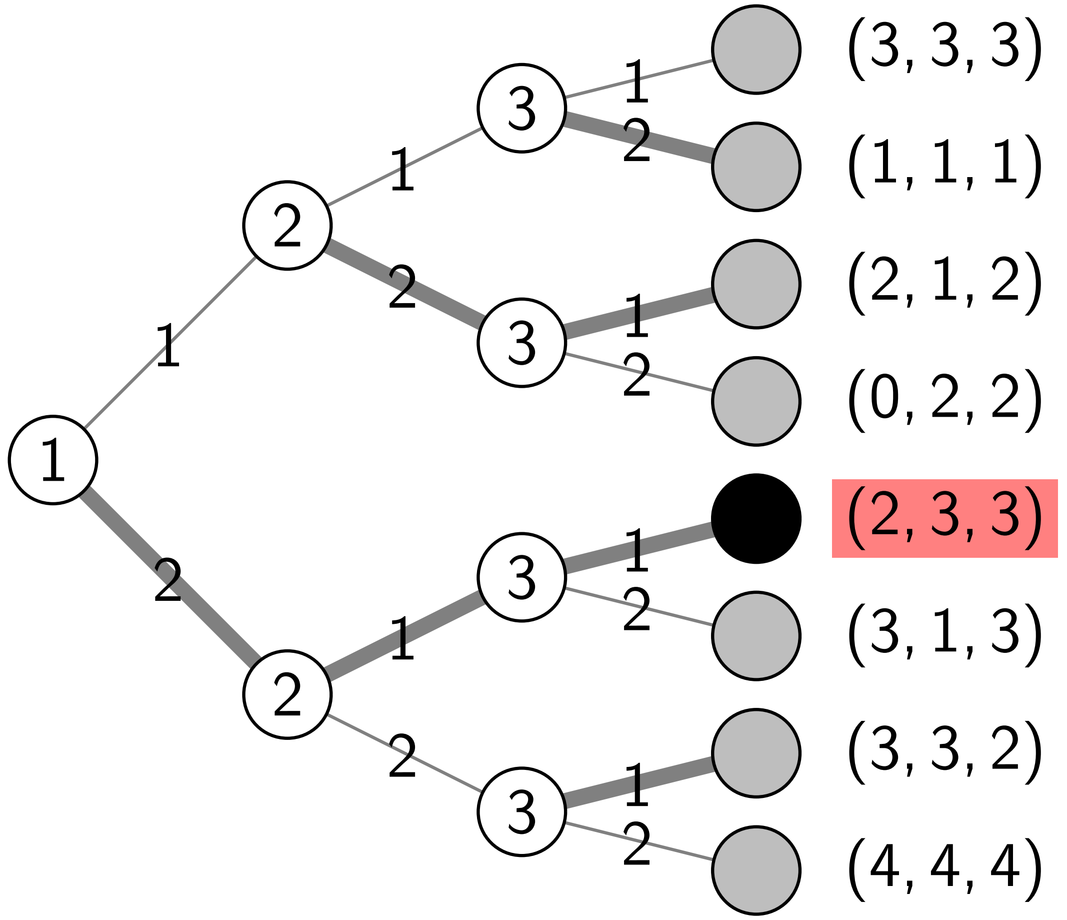


Motivations

Efficiency of equilibrium

- **SPE:** $L_1 = 3,\ L_2 = 2 \quad \Rightarrow \quad L_{\max} = 3$

0	1	2
2	1	1

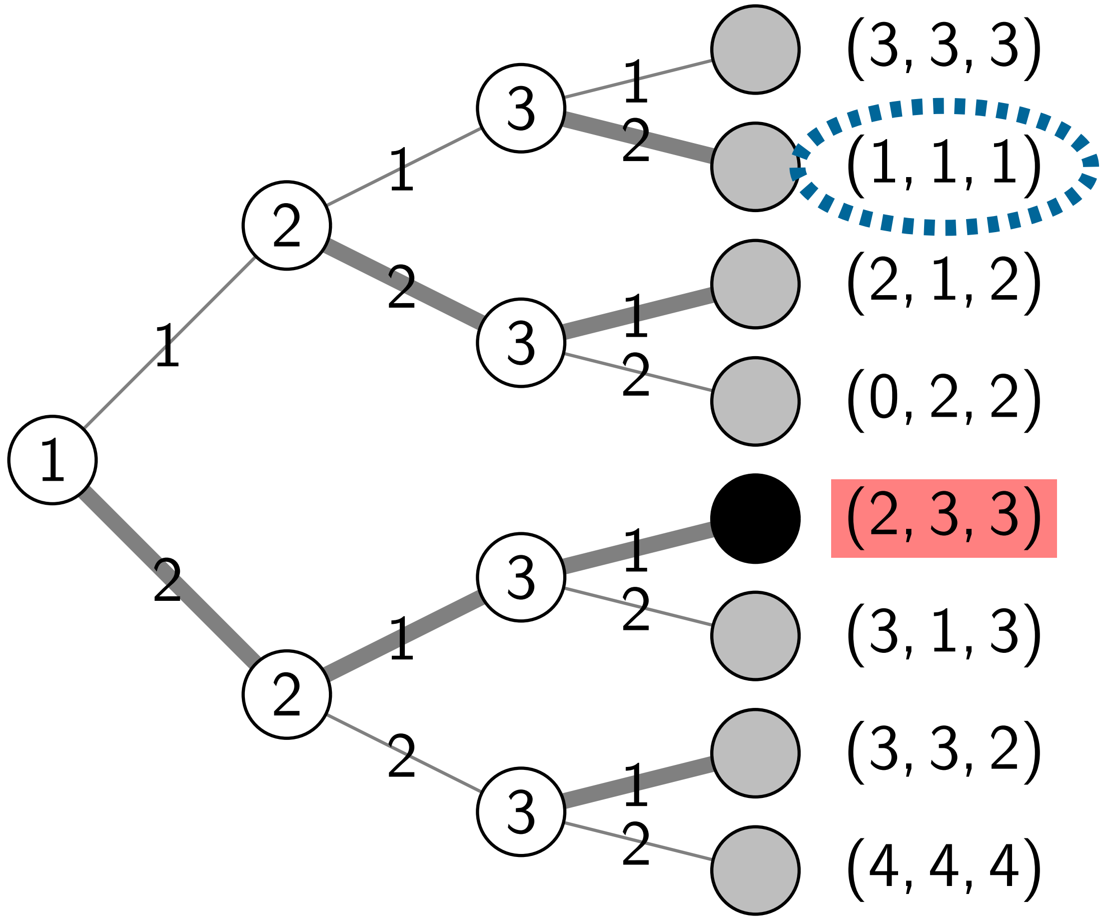


Motivations

Efficiency of equilibrium

- **SPE:** $L_1 = 3,\ L_2 = 2 \quad \Rightarrow \quad L_{\max} = 3$
- **OPT:** $L_1 = 1,\ L_2 = 1 \quad \Rightarrow \quad L_{\max} = 1$

0	1	2
2	1	1



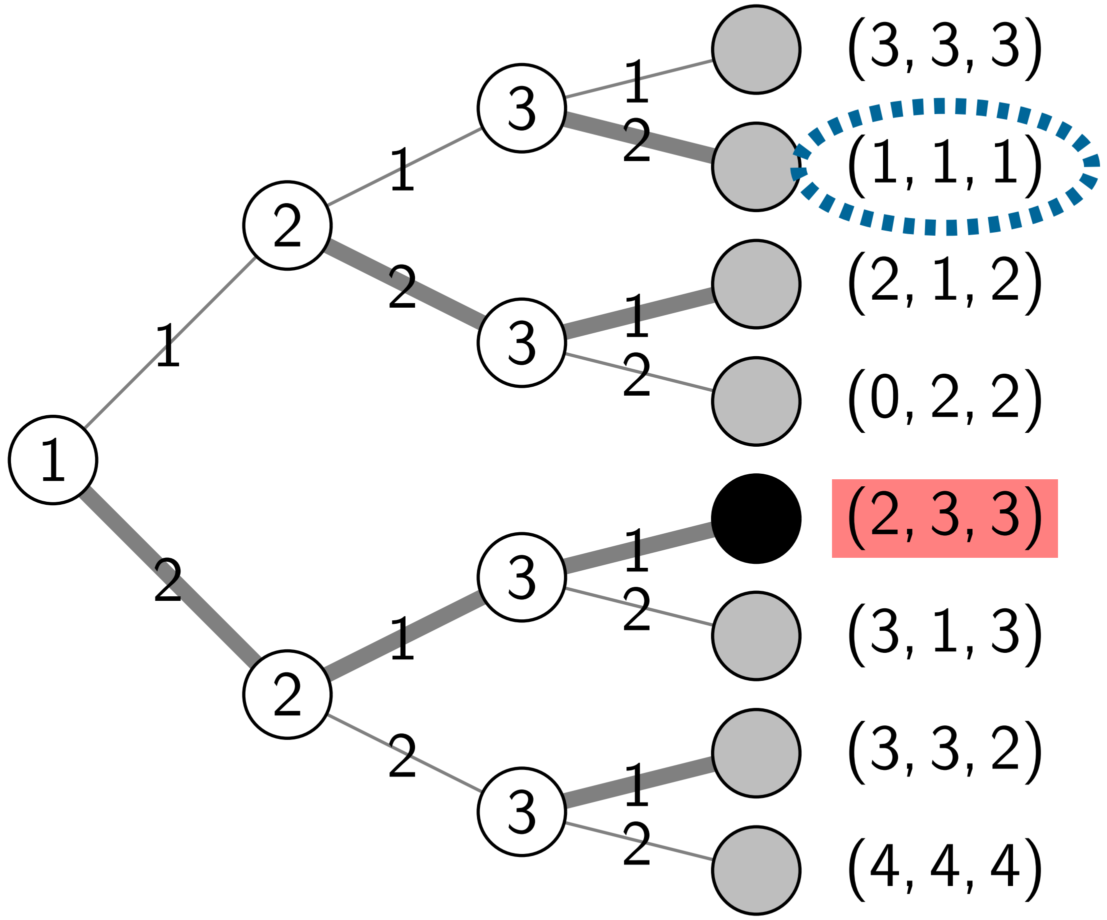
Motivations

Efficiency of equilibrium

- **SPE:** $L_1 = 3,\ L_2 = 2 \quad \Rightarrow \quad L_{\max} = 3$
- **OPT:** $L_1 = 1,\ L_2 = 1 \quad \Rightarrow \quad L_{\max} = 1$

  SPE is bad in the view of **social cost (i.e. makespan)**

0	1	2
2	1	1



Motivations

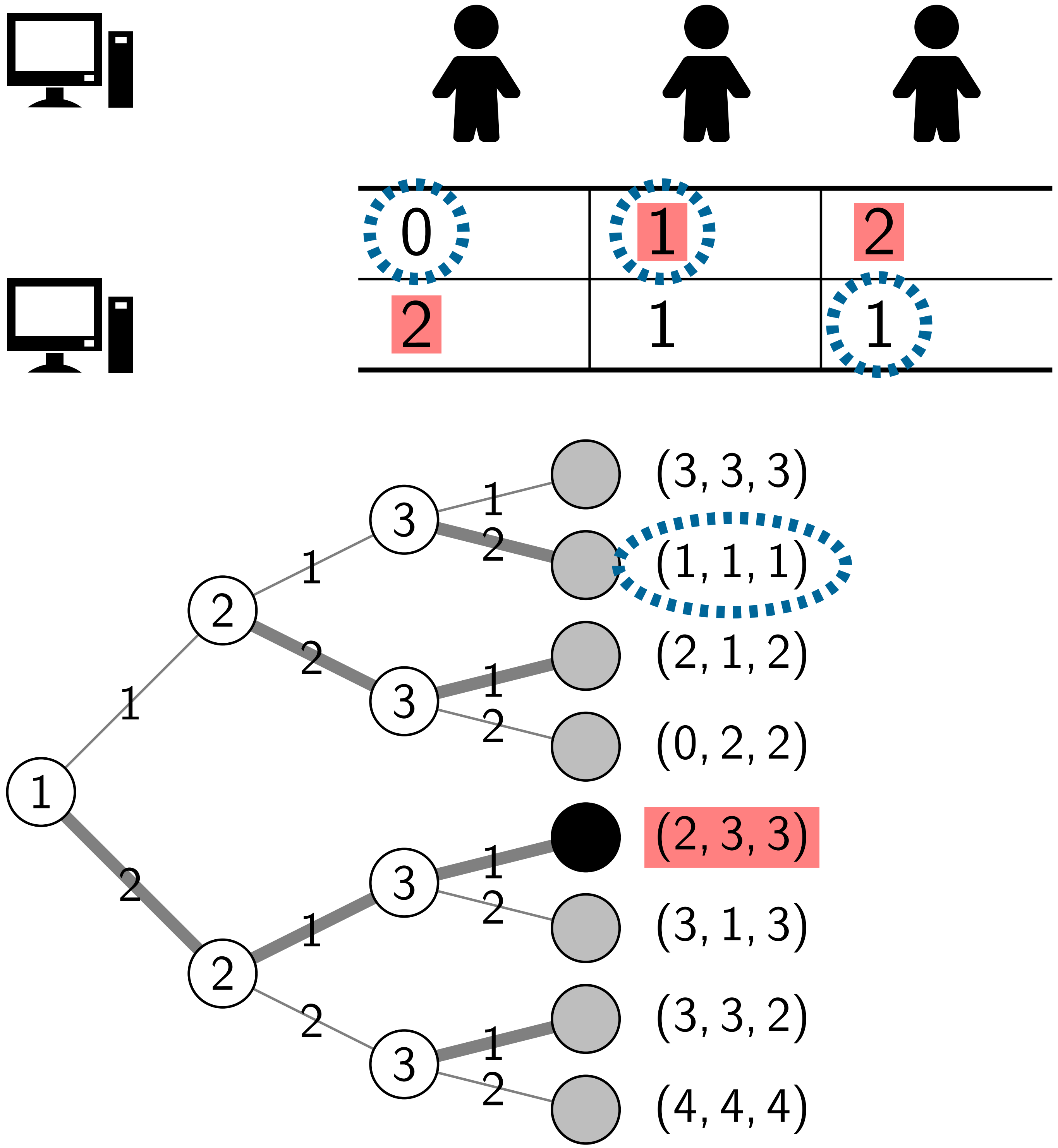
Efficiency of equilibrium

- **SPE:** $L_1 = 3,\ L_2 = 2 \quad \Rightarrow \quad L_{\max} = 3$
- **OPT:** $L_1 = 1,\ L_2 = 1 \quad \Rightarrow \quad L_{\max} = 1$

  SPE is bad in the view of **social cost (i.e. makespan)**

- [Koutsoupias, Papadimitriou, 2009]:
  - **Price of Anarchy (PoA)**

$$\text{PoA} = \frac{\text{worst Nash Equilibrium}}{\text{OPT}}$$



Motivations

Efficiency of equilibrium

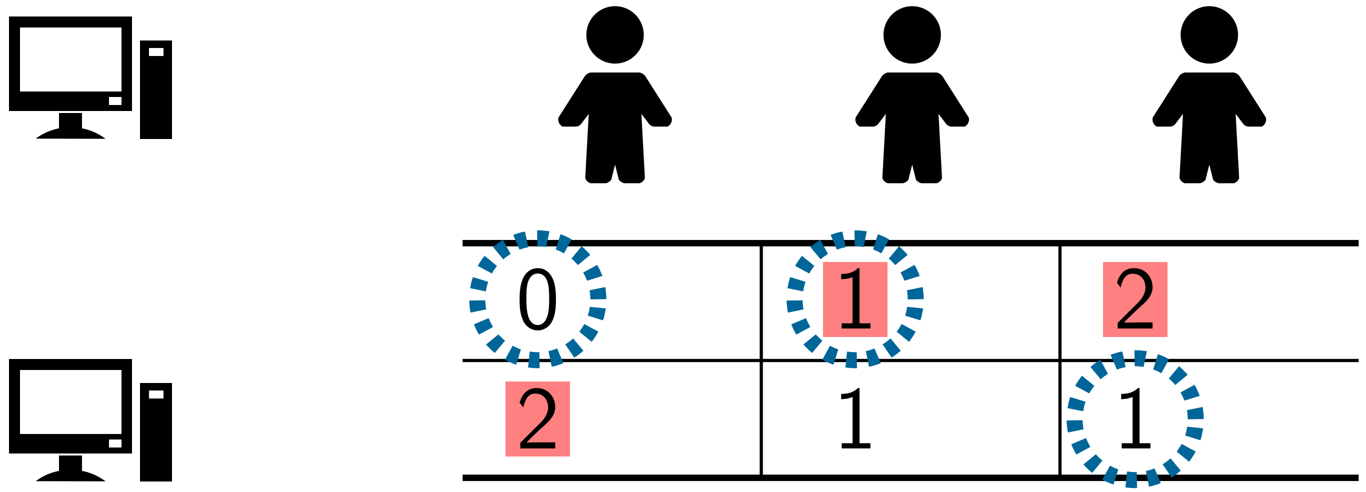
- **SPE:** $L_1 = 3,\ L_2 = 2 \quad \Rightarrow \quad L_{\max} = 3$
- **OPT:** $L_1 = 1,\ L_2 = 1 \quad \Rightarrow \quad L_{\max} = 1$

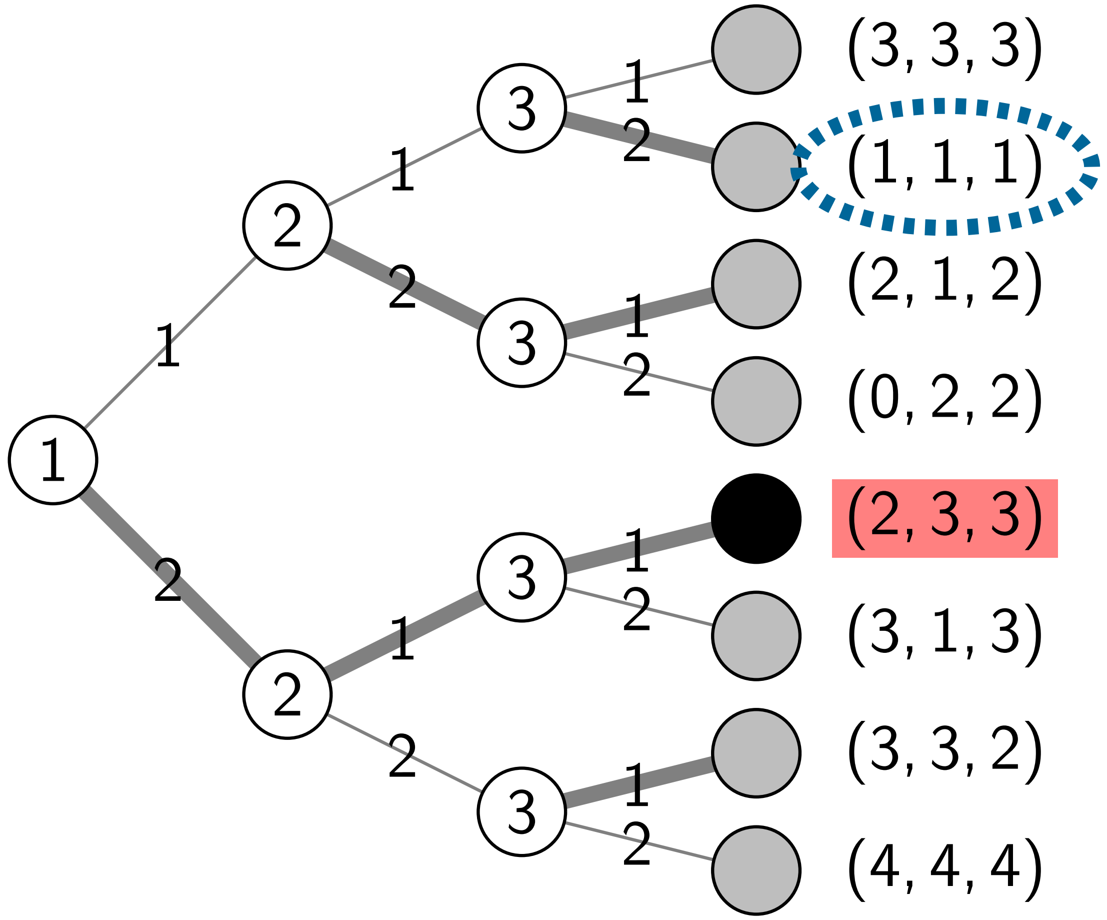
SPE is bad in the view of **social cost (i.e. makespan)**

- [Koutsoupias, Papadimitriou, 2009]:
  - **Price of Anarchy (PoA)**

$$\text{PoA} = \frac{\text{worst Nash Equilibrium}}{\text{OPT}}$$

PoA: How bad could it be if allowing the jobs to play strategically?





Motivations

Efficiency of equilibrium

- **SPE:** $L_1 = 3,\ L_2 = 2 \quad \Rightarrow \quad L_{\max} = 3$
- **OPT:** $L_1 = 1,\ L_2 = 1 \quad \Rightarrow \quad L_{\max} = 1$

SPE is bad in the view of **social cost (i.e. makespan)**

- [Koutsoupias, Papadimitriou, 2009]:

  - **Price of Anarchy (PoA)**

$$\text{PoA} = \frac{\text{worst Nash Equilibrium}}{\text{OPT}}$$

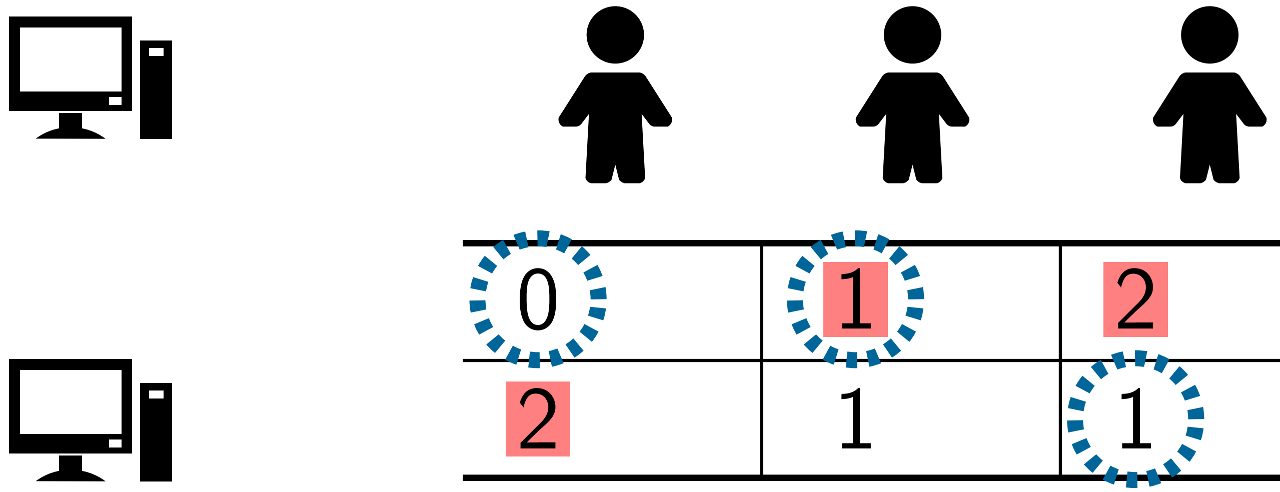
PoA: How bad could it be if allowing the jobs to play strategically?

- [Leme, Syrgkanis,Tardos, 2012]:

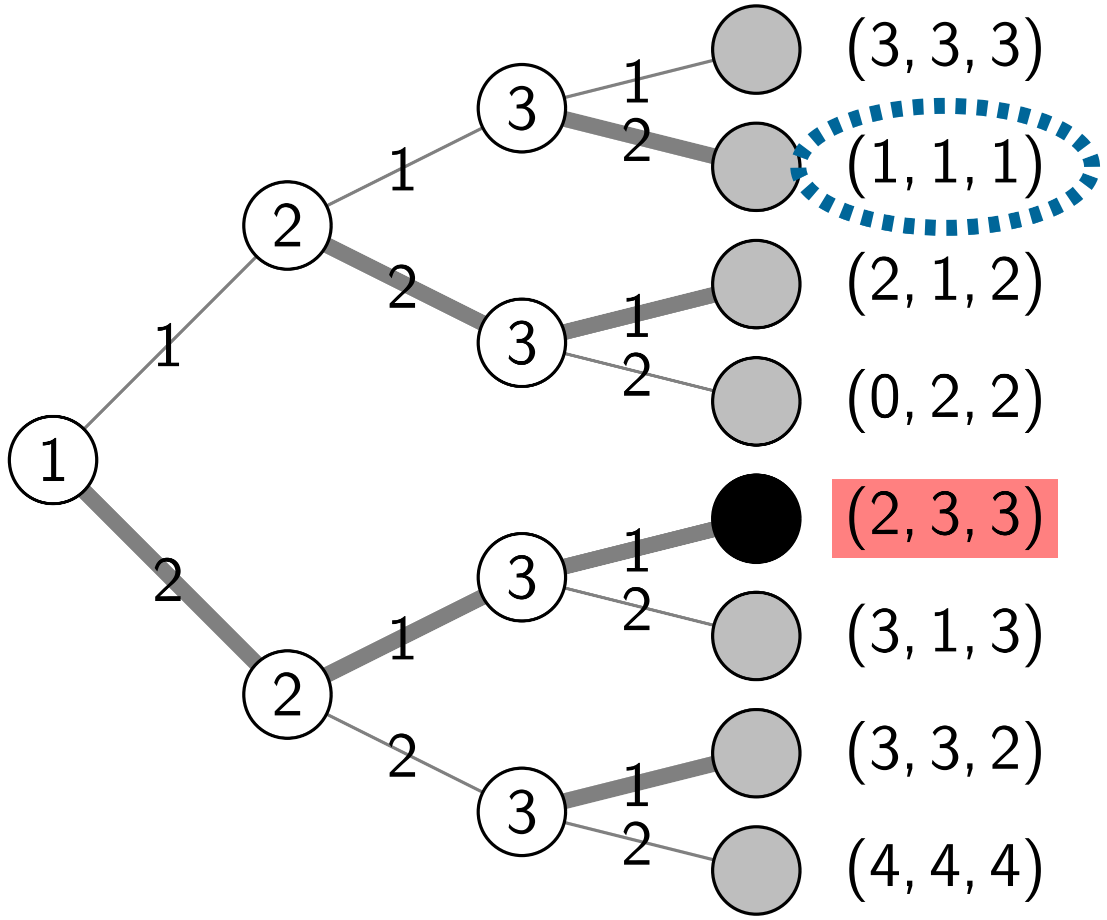
  - **Sequential Price of Anarchy (SPoA)**

$$\text{SPoA} = \frac{\text{worst SPE}}{\text{OPT}}$$

e.g., $\text{SPoA} = \frac{3}{1} = 3$



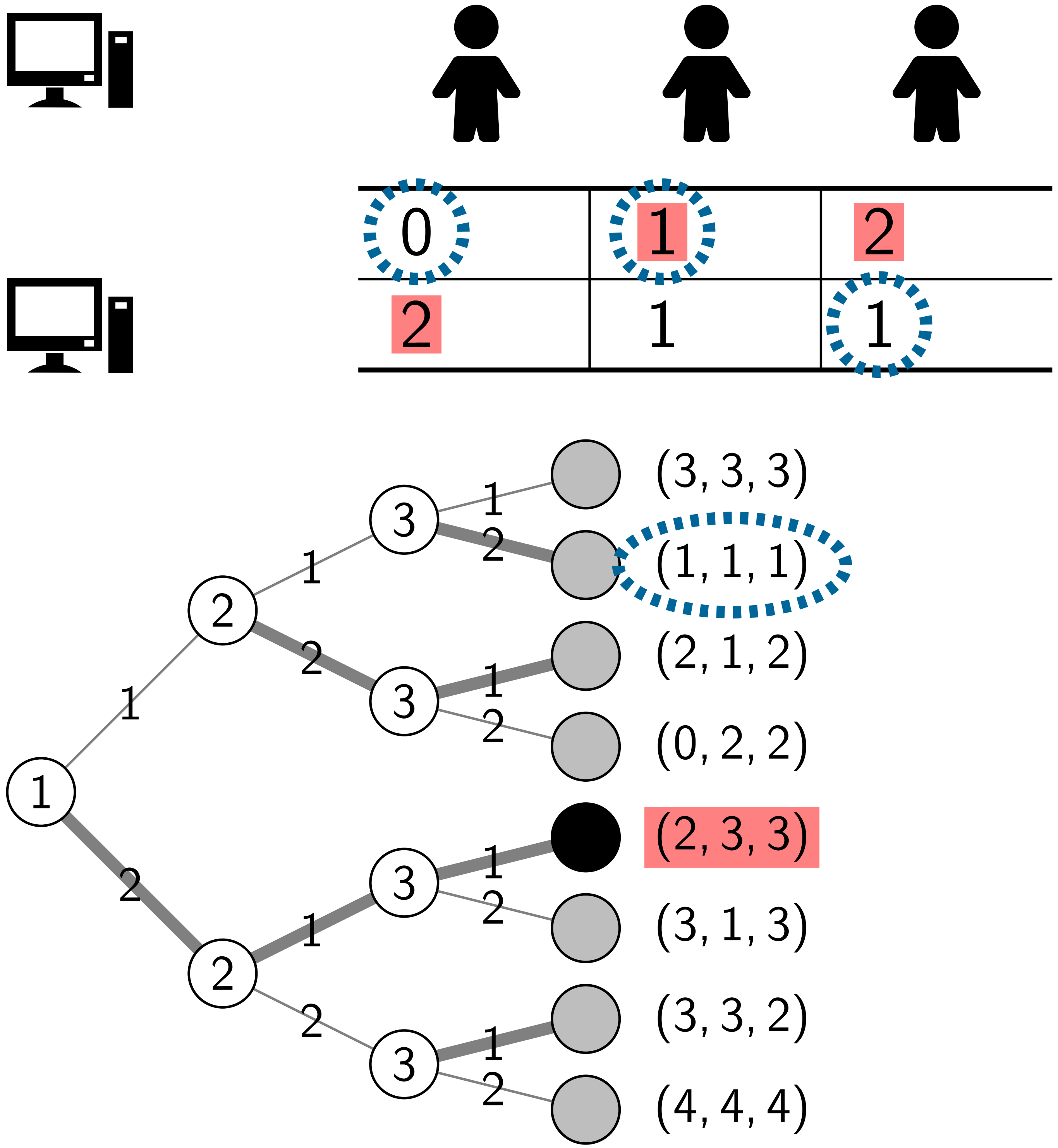
0	1	2
2	1	1



Motivations

General version: m machines, n jobs, and arbitrary p_{ij}

- [Leme, Syrgkanis, Tardos, 2012]:
  - $n \leq \text{SPoA} \leq m2^n$
- [Bilò, Flammini, Monaco, Moscardelli, 2015]:
  - $2^{\Omega(\sqrt{n})} \leq \text{SPoA} \leq 2^n$



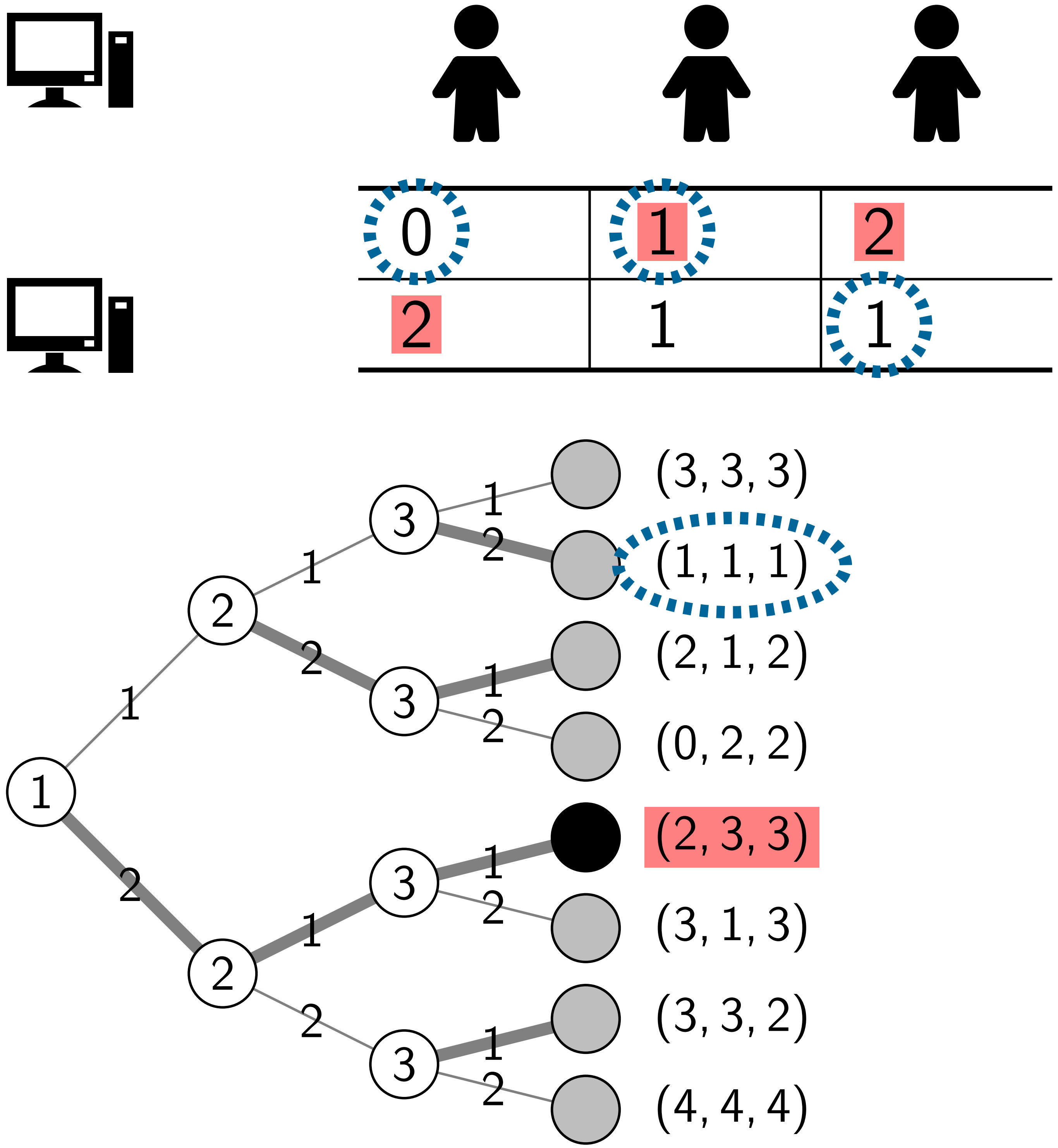
Motivations

General version: m machines, n jobs, and arbitrary p_{ij}

- [Leme, Syrgkanis,Tardos, 2012]:
  - $n \leq \text{SPoA} \leq m2^n$

- [Bilò, Flammini, Monaco, Moscardelli, 2015]:
  - $2^{\Omega(\sqrt{n})} \leq \text{SPoA} \leq 2^n$

- [Chen, Giessler, Mamageishvili, Mihalák, Penna, 2020]:
  - For $m = 2$: $\text{SPoA} = \Theta(n)$



Motivations

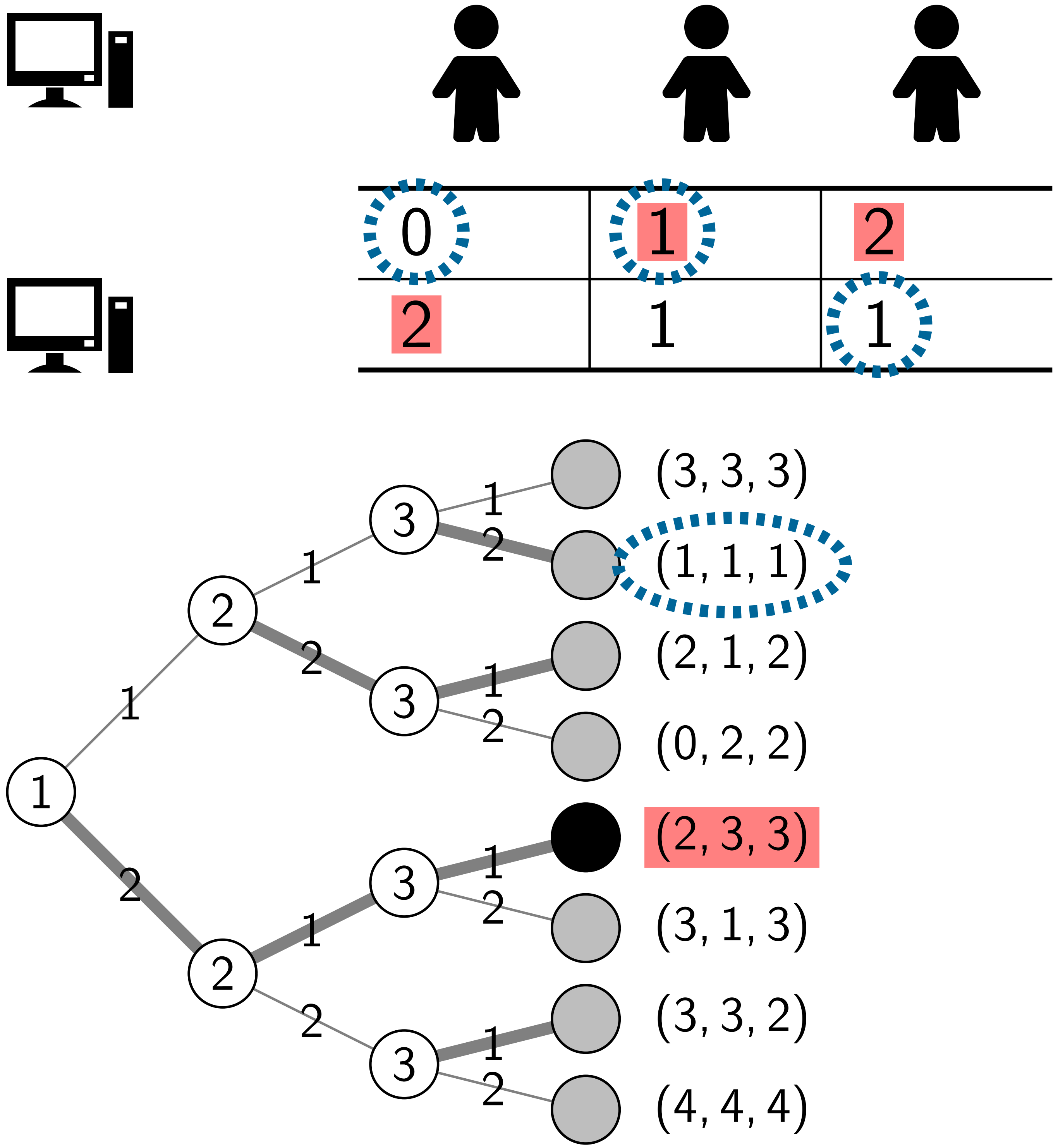
General version: m machines, n jobs, and arbitrary p_{ij}

- [Leme, Syrgkanis, Tardos, 2012]:
  - $n \leq \text{SPoA} \leq m2^n$
- [Bilò, Flammini, Monaco, Moscardelli, 2015]:
  - $2^{\Omega(\sqrt{n})} \leq \text{SPoA} \leq 2^n$
- [Chen, Giessler, Mamageishvili, Mihalák, Penna, 2020]:
  - For $m = 2$: $\text{SPoA} = \Theta(n)$
- Online greedy algorithm for 2 unrelated machines:

  Competitive ratio $= 2$

myopic player

$\text{SPoA} = 2$



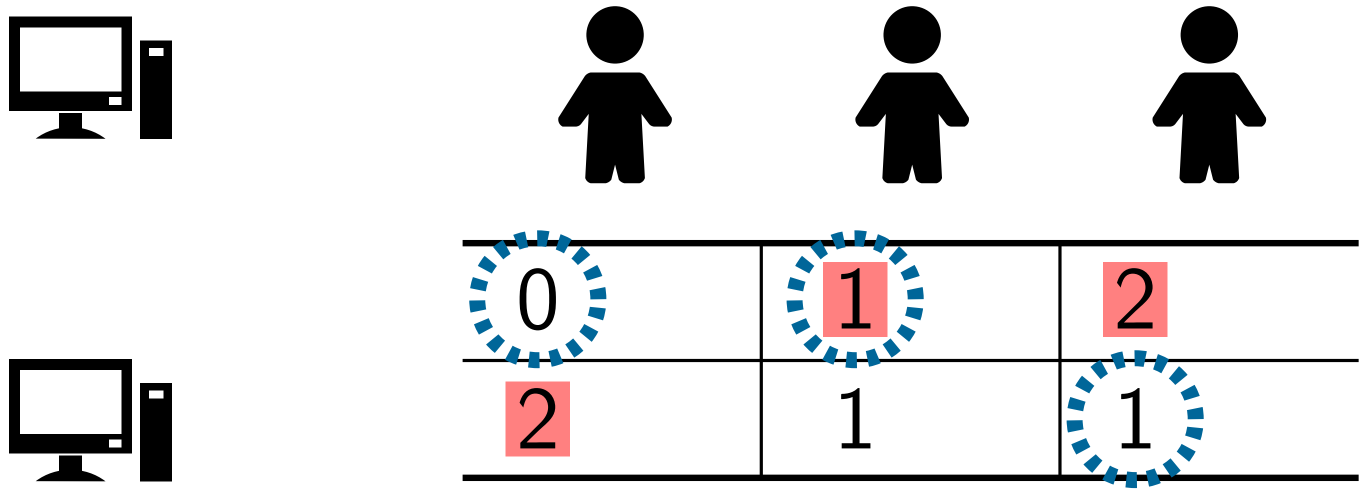
Motivations

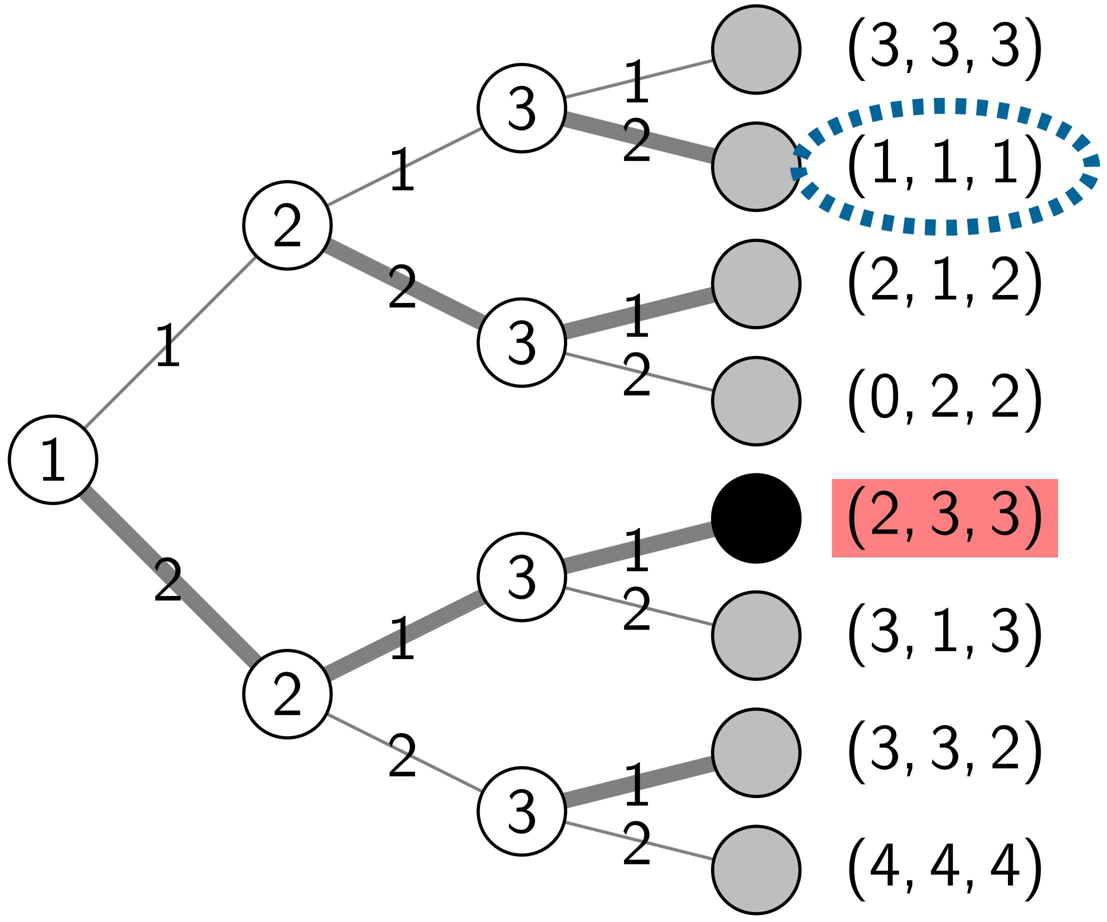
General version: m machines, n jobs, and arbitrary p_{ij}

- [Leme, Syrgkanis, Tardos, 2012]:
  - $n \leq \text{SPoA} \leq m2^n$
- [Bilò, Flammini, Monaco, Moscardelli, 2015]:
  - $2^{\Omega(\sqrt{n})} \leq \text{SPoA} \leq 2^n$
- [Chen, Giessler, Mamageishvili, Mihalák, Penna, 2020]:
  - For $m = 2$: $\text{SPoA} = \Theta(n)$
- Online greedy algorithm for 2 unrelated machines:

Competitive ratio $= 2$

myopic player	full rationality
$\text{SPoA} = 2$	$\text{SPoA} = \Theta(n)$





Motivations

General version: m machines, n jobs, and arbitrary p_{ij}

- [Leme, Syrgkanis, Tardos, 2012]:
  - $n \leq \text{SPoA} \leq m2^n$

- [Bilò, Flammini, Monaco, Moscardelli, 2015]:
  - $2^{\Omega(\sqrt{n})} \leq \text{SPoA} \leq 2^n$

- [Chen, Giessler, Mamageishvili, Mihalák, Penna, 2020]:
  - For $m = 2$: $\text{SPoA} = \Theta(n)$

- Online greedy algorithm for 2 unrelated machines:

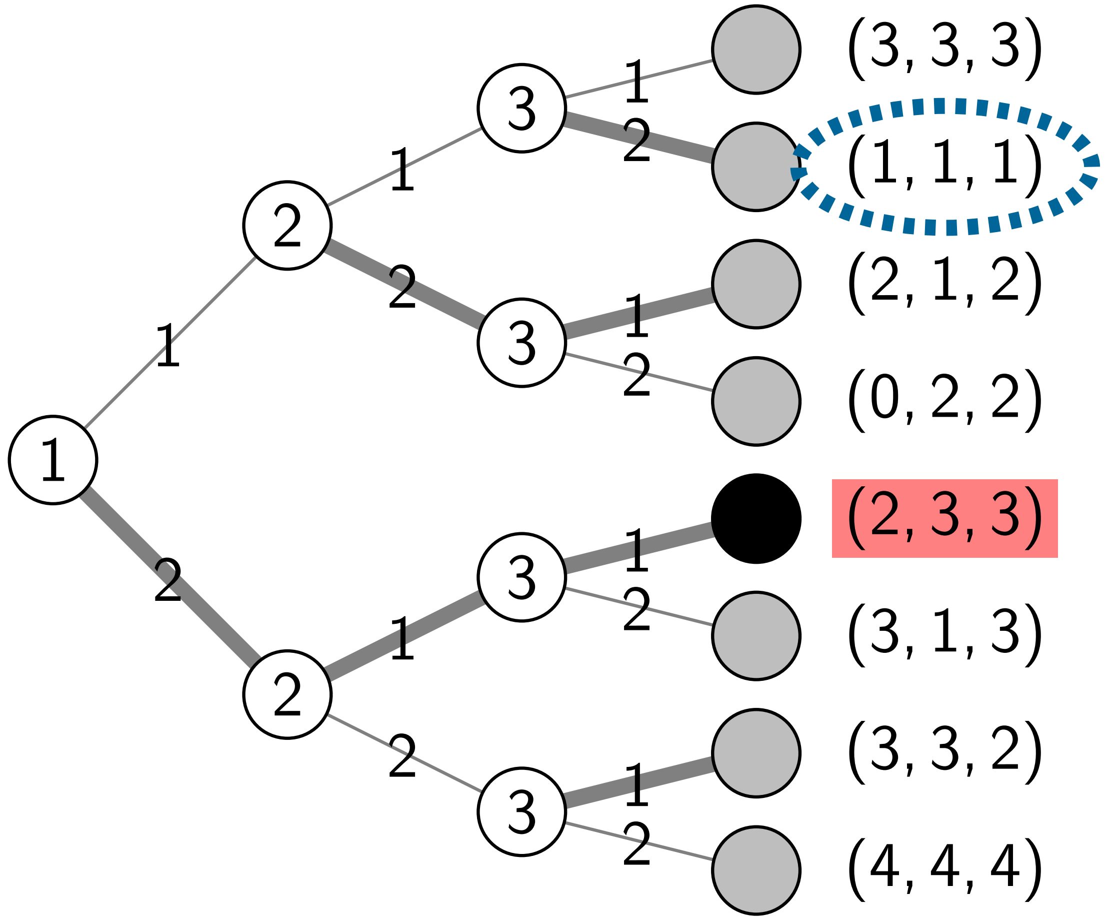
Competitive ratio $= 2$

myopic player — full rationality

$\text{SPoA} = 2 \longleftrightarrow \text{SPoA} = \Theta(n)$

curse of rationality

0	1	2
2	1	1



Modeling the bounded rationality

- [Simon, 1955]:

  Bounded rationality — “rational choice that takes account the cognitive limitations of the decision-maker – limitations of both knowledge and computational capacity”

Modeling the bounded rationality

- [Simon, 1955]:

  Bounded rationality — "rational choice that takes account the cognitive limitations of the decision-maker – limitations of both knowledge and computational capacity"

Frank Hahn remarks that "there is only one way to be perfectly rational, while there are an infinity of ways to be partially rational..."

Modeling the bounded rationality

- [Simon, 1955]:

  Bounded rationality — "rational choice that takes account the cognitive limitations of the decision-maker – limitations of both knowledge and computational capacity"

  Frank Hahn remarks that "there is only one way to be perfectly rational, while there are an infinity of ways to be partially rational..."

(1) Players with k-lookahead

- each player only considers the next k successors' information for computing his decision

Modeling the bounded rationality

- [Simon, 1955]:

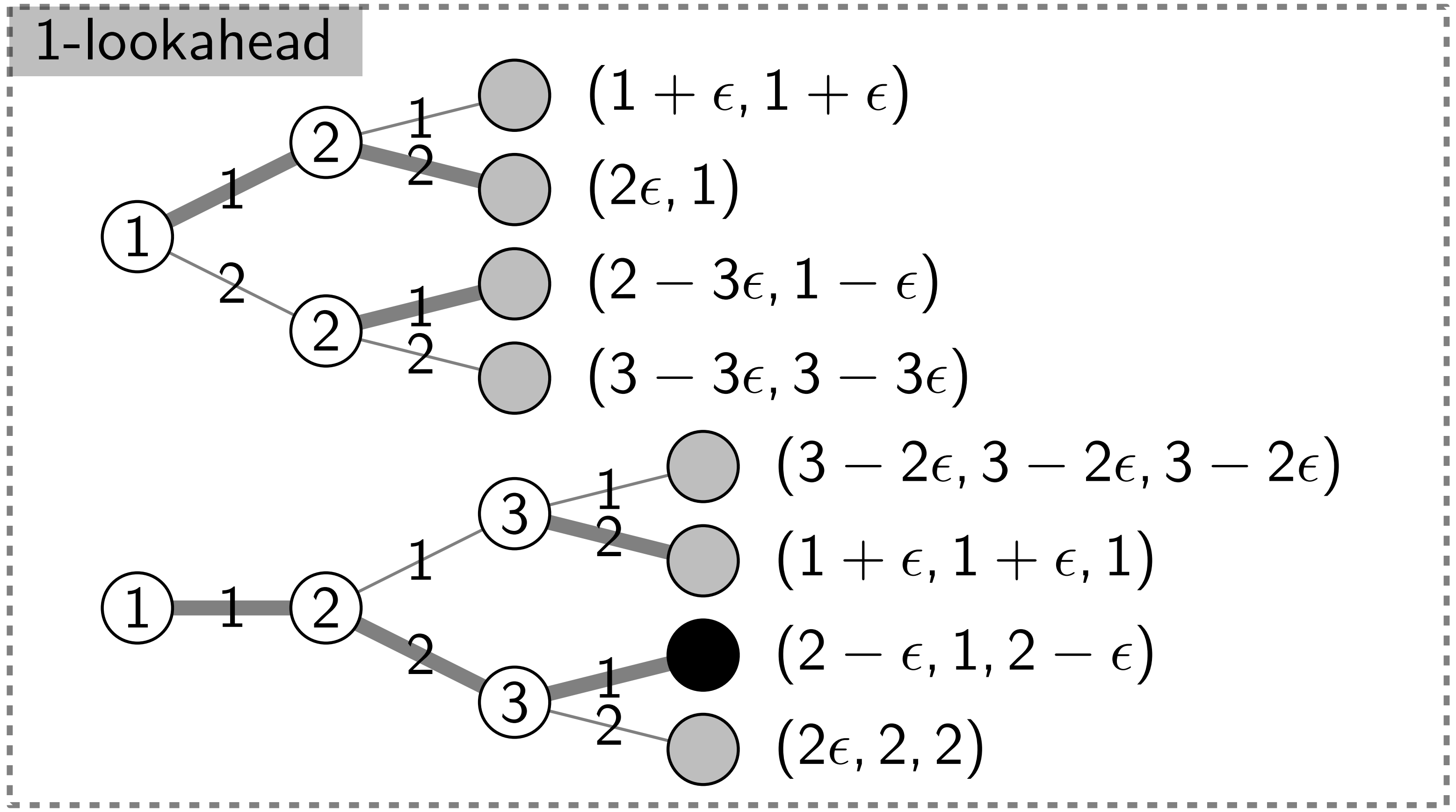
Bounded rationality — "rational choice that takes account the cognitive limitations of the decision-maker – limitations of both knowledge and computational capacity"

Frank Hahn remarks that "there is only one way to be perfectly rational, while there are an infinity of ways to be partially rational..."

(1) Players with *k*-lookahead

- each player only considers the next k successors' information for computing his decision

	job 1	job 2	job 3
machine 1	2ϵ	$1-\epsilon$	$2-3\epsilon$
machine 2	$2-3\epsilon$	1	1



Modeling the bounded rationality

- [Simon, 1955]:

  Bounded rationality — "rational choice that takes account the cognitive limitations of the decision-maker – limitations of both knowledge and computational capacity"

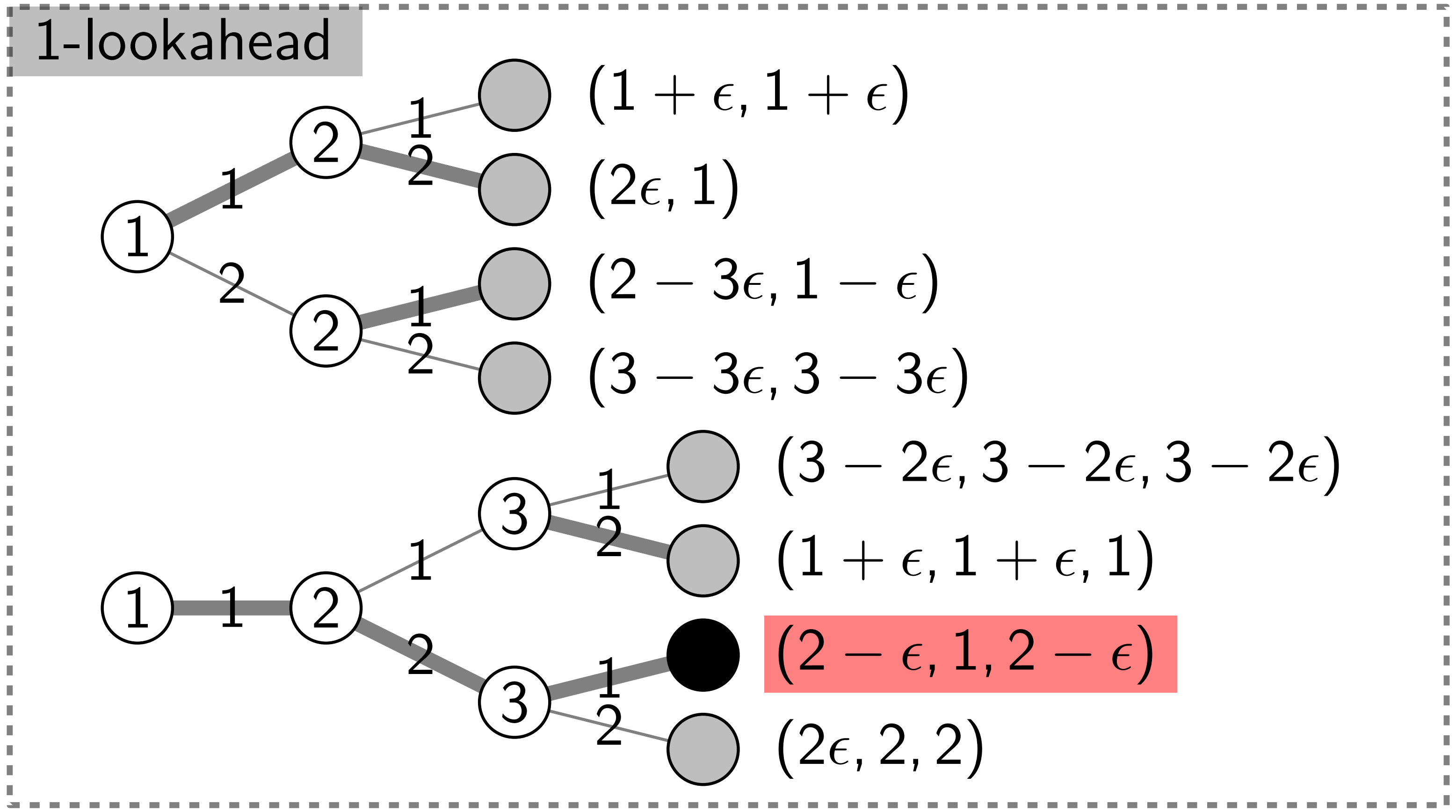
  Frank Hahn remarks that "there is only one way to be perfectly rational, while there are an infinity of ways to be partially rational..."

(1) Players with *k*-lookahead

- each player only considers the next k successors' information for computing his decision

  e.g., SPoA ≈ 2

	job 1	job 2	job 3
machine 1	2ϵ	$1-\epsilon$	$2-3\epsilon$
machine 2	$2-3\epsilon$	1	1



Modeling the bounded rationality

- [Simon, 1955]:

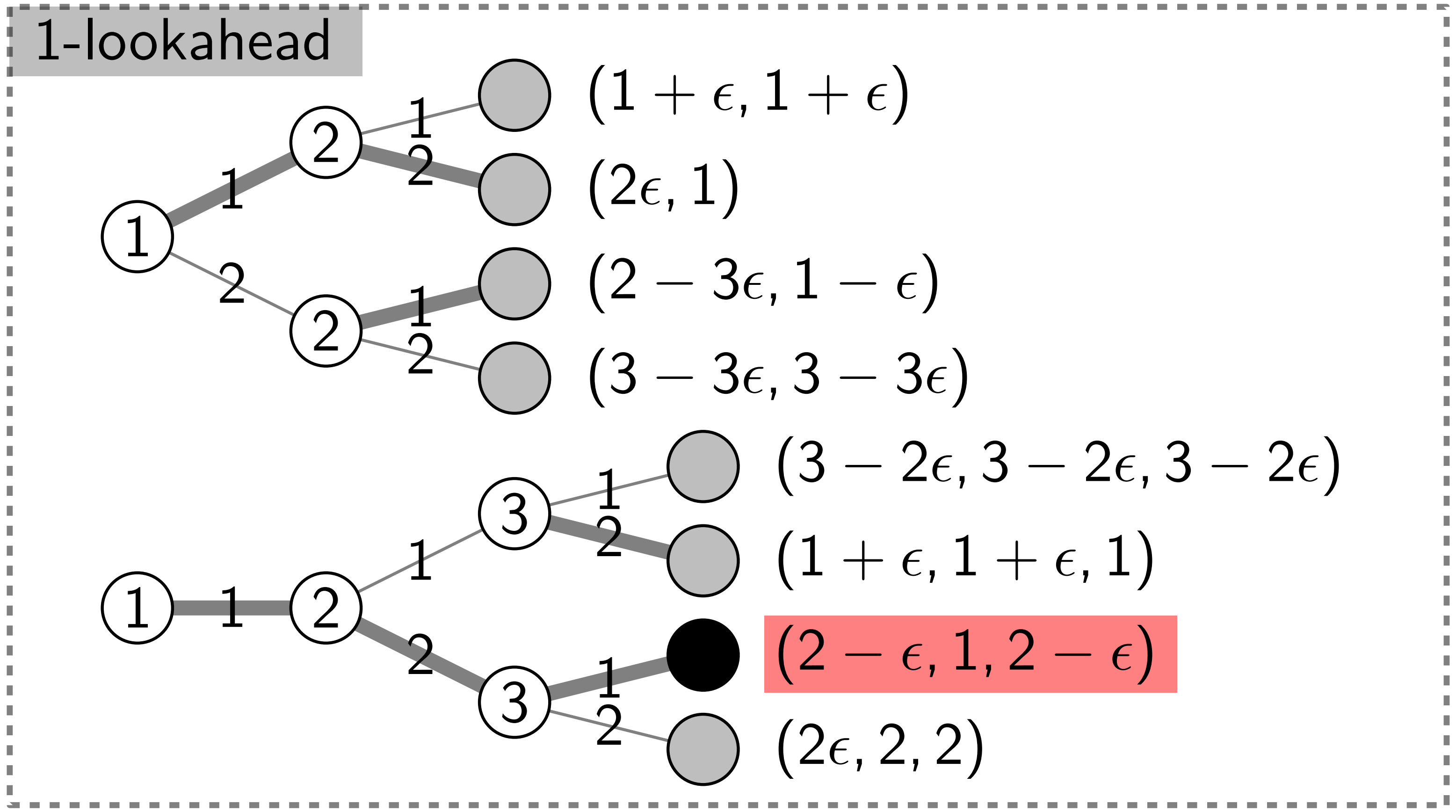
Bounded rationality — “rational choice that takes account the cognitive limitations of the decision-maker – limitations of both knowledge and computational capacity”

Frank Hahn remarks that “there is only one way to be perfectly rational, while there are an infinity of ways to be partially rational...”

(1) Players with k-lookahead

- each player only considers the next k successors’ information for computing his decision
  e.g., SPoA ≈ 2
  Similar settings can also be found in:
- [Mirrokni, Thain, Vetta, 2012]
- [Bilò, Fanelli, Moscardelli, 2017]
- [Groenland, Schäfer, 2018]
- [Kroer, Sandholm, 2020]

	job 1	job 2	job 3
machine 1	2ϵ	$1-\epsilon$	$2-3\epsilon$
machine 2	$2-3\epsilon$	1	1



Modeling the bounded rationality

(2) Simple-minded players

- A simple-minded player simply assumes the successors will choose machines with minimum processing times, so he/she can easily find a best choice.

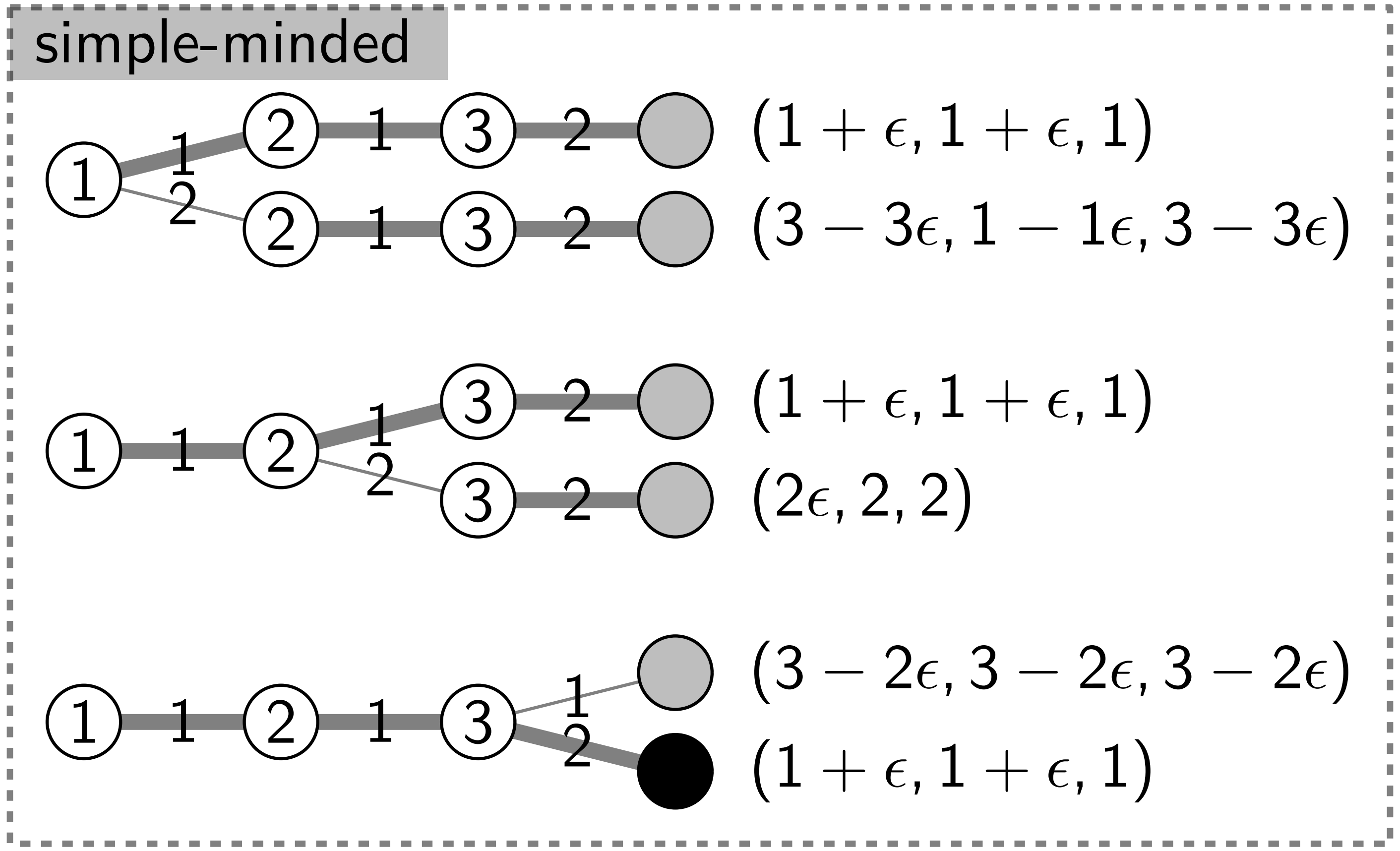
	job 1	job 2	job 3
machine 1	2ϵ	$1-\epsilon$	$2-3\epsilon$
machine 2	$2-3\epsilon$	1	1

Modeling the bounded rationality

(2) Simple-minded players

- A simple-minded player simply assumes the successors will choose machines with minimum processing times, so he/she can easily find a best choice.

	job 1	job 2	job 3
machine 1	2ϵ	$1-\epsilon$	$2-3\epsilon$
machine 2	$2-3\epsilon$	1	1



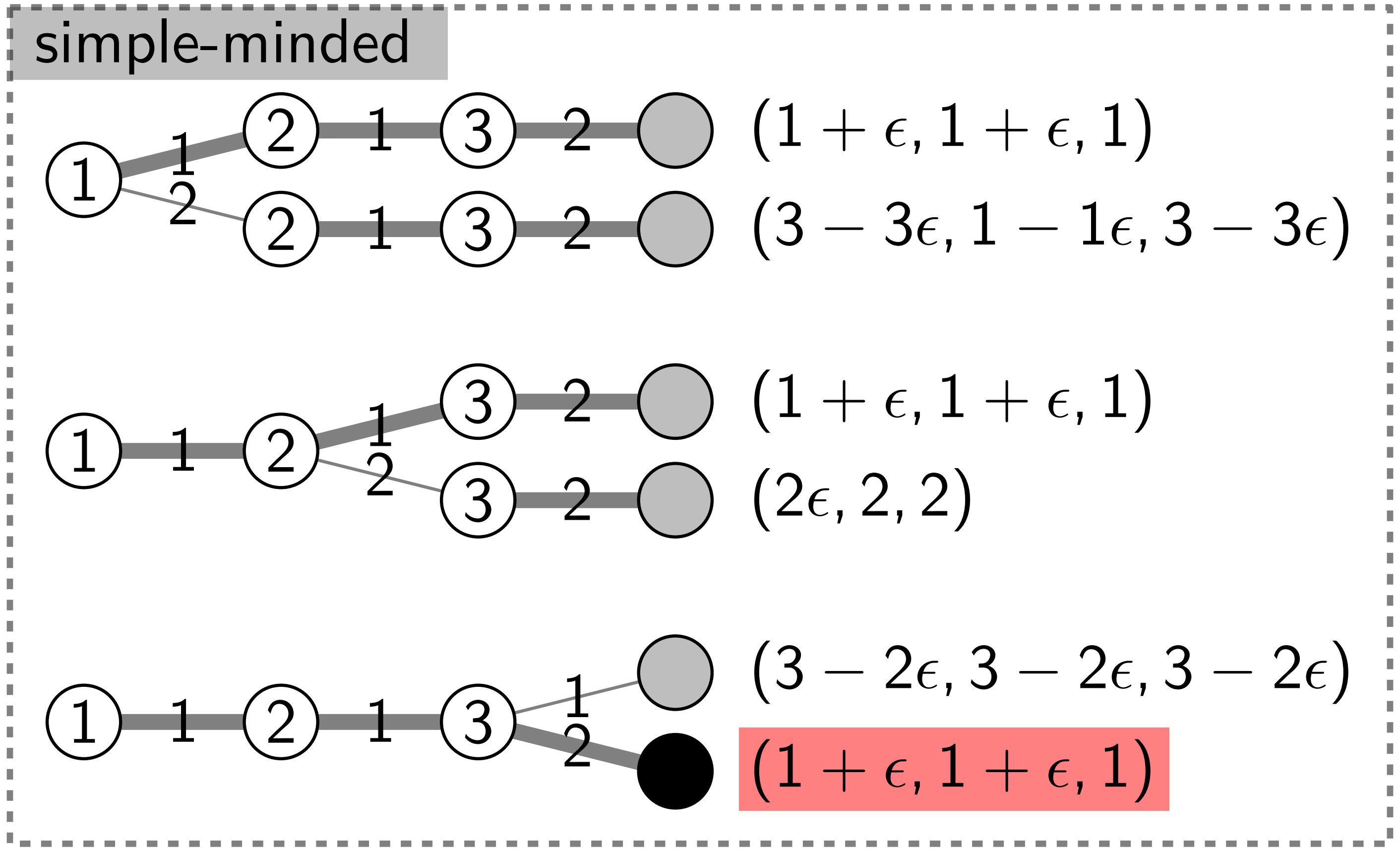
Modeling the bounded rationality

(2) Simple-minded players

- A simple-minded player simply assumes the successors will choose machines with minimum processing times, so he/she can easily find a best choice.

  e.g., SPoA $= 1$

	job 1	job 2	job 3
machine 1	2ϵ	$1-\epsilon$	$2-3\epsilon$
machine 2	$2-3\epsilon$	1	1



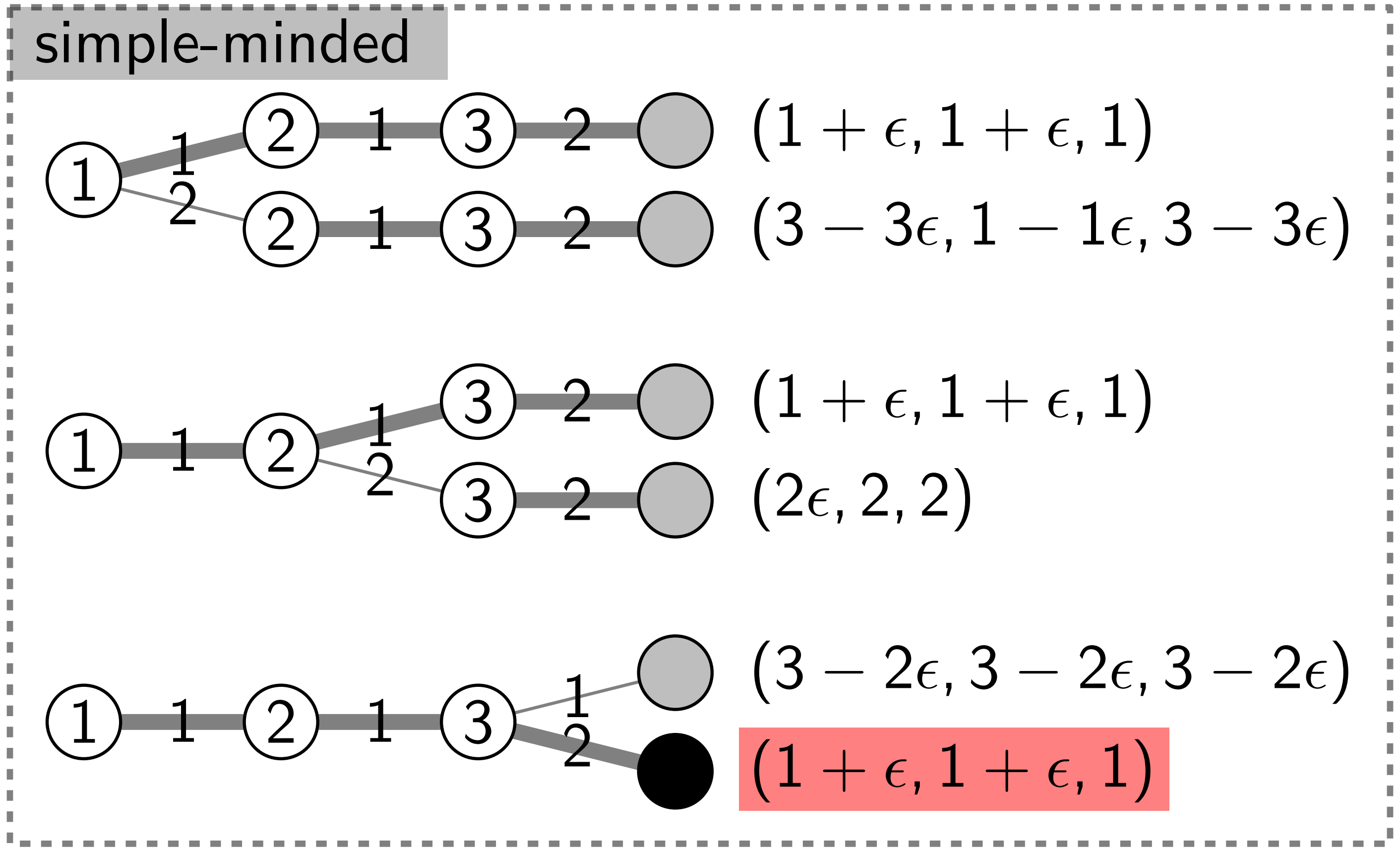
Modeling the bounded rationality

(2) Simple-minded players

- A simple-minded player simply assumes the successors will choose machines with minimum processing times, so he/she can easily find a best choice.

  e.g., SPoA $= 1$

	job 1	job 2	job 3
machine 1	2ϵ	$1-\epsilon$	$2-3\epsilon$
machine 2	$2-3\epsilon$	1	1



- full rationality: SPoA ≈ 3
- 1-lookahead: SPoA ≈ 2
- simple-minded: SPoA $= 1$

Warm-up: 1-lookahead on 2 unrelated machines

- **Lemma**:

$L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

Warm-up: 1-lookahead on 2 unrelated machines

- **Lemma**:

$L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

Since $OPT \geq \sum_{j \in N} p_j / 2$, we obtain the following theorem:

- **Theorem**:

For the sequential scheduling game on two unrelated machines where players have 1-lookahead, SPoA $= 2$

Warm-up: 1-lookahead on 2 unrelated machines

- **Lemma**:

$L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

Since $OPT \geq \sum_{j \in N} p_j / 2$, we obtain the following theorem:

- **Theorem**:

For the sequential scheduling game on two unrelated machines where players have 1-lookahead, SPoA $= 2$

Proof:

(1) SPoA $= \frac{L_{\max}}{OPT} \leq \frac{\sum_{j \in N} p_j}{\sum_{j \in N} p_j / 2} = 2$

(2)

	job 1	job 2
machine 1	$1 + \epsilon$	2
machine 2	1	$1 + \epsilon$

$\Rightarrow$ SPoA ≥ 2

Warm-up: 1-lookahead on 2 unrelated machines

- **Lemma**:

  $L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

Warm-up: 1-lookahead on 2 unrelated machines

- **Lemma**:

  $L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

$$J = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ \dots\ , n-5,\ n-4,\ n-3,\ n-2,\ n-1,\ n\}$$

Warm-up: 1-lookahead on 2 unrelated machines

- **Lemma**:

  $L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

$$J = \{\underbrace{1,\ 2,\ 3,}\big|\underbrace{4,\ 5,\ 6,\ 7,}\big|\underbrace{\ldots\ ,n-5,\ n-4,}\big|\underbrace{n-3,\ n-2,\ n-1,\ n}\}$$

Warm-up: 1-lookahead on 2 unrelated machines

- **Lemma**:

  $L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

$$J = \{\underbrace{1,\ 2,\ 3,}\,\Big|\,\underbrace{4,\ 5,\ 6,\ 7,}\,\Big|\,\underbrace{\dots\ ,n-5,\ n-4,}\,\Big|\,\underbrace{n-3,\ n-2,\ n-1,\ n}\}$$

		$u+1$	$u+2$	$\cdots$	$v-1$	v
machine 1	$D_1(u)$	$p_{1,u+1}$	$p_{1,u+2}$	$\cdots$	$p_{1,v-1}$	$p_{1,v}$
machine 2	$D_2(u)$	$p_{2,u+1}$	$p_{2,u+2}$	$\cdots$	$p_{2,v-1}$	$p_{2,v}$

- $D_1(u) + p_{1,u+1} \leq D_2(u) + p_{2,u+1}$
- jobs $u+1$ to $v-1$ choose machine 2, and job v chooses machien 1

Warm-up: 1-lookahead on 2 unrelated machines

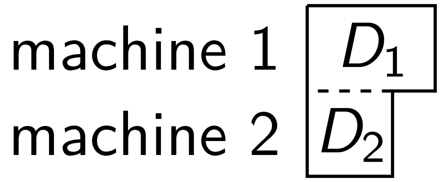
- **Lemma**:

  $L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

$$J = \{\underbrace{1,\ 2,\ 3,}_{}\Big|\underbrace{4,\ 5,\ 6,\ 7,}_{}\Big|\underbrace{\ldots\ ,n-5,\ n-4,}_{}\Big|\underbrace{n-3,\ n-2,\ n-1,\ n}_{}\}$$

		$u+1$	$u+2$	$\cdots$	$v-1$	v
machine 1	$D_1(u)$	$p_{1,u+1}$	$p_{1,u+2}$	$\cdots$	$p_{1,v-1}$	$p_{1,v}$
machine 2	$D_2(u)$	$p_{2,u+1}$	$p_{2,u+2}$	$\cdots$	$p_{2,v-1}$	$p_{2,v}$

- $D_1(u) + p_{1,u+1} \leq D_2(u) + p_{2,u+1}$
- jobs $u+1$ to $v-1$ choose machine 2, and job v chooses machien 1



Warm-up: 1-lookahead on 2 unrelated machines

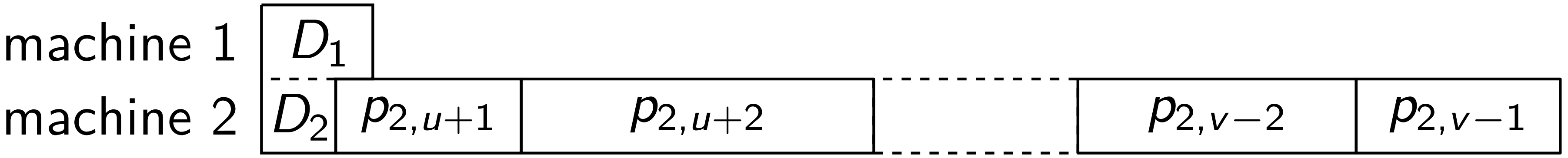
- **Lemma**:

  $L_{\max} \le \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

$$J = \{\underbrace{1,\ 2,\ 3,}\Big|\underbrace{4,\ 5,\ 6,\ 7,}\Big|\underbrace{\ldots\ ,n-5,\ n-4,}\Big|\underbrace{n-3,\ n-2,\ n-1,\ n}\}$$

		$u+1$	$u+2$	$\cdots$	$v-1$	v
machine 1	$D_1(u)$	$p_{1,u+1}$	$p_{1,u+2}$	$\cdots$	$p_{1,v-1}$	$p_{1,v}$
machine 2	$D_2(u)$	$p_{2,u+1}$	$p_{2,u+2}$	$\cdots$	$p_{2,v-1}$	$p_{2,v}$

- $D_1(u) + p_{1,u+1} \le D_2(u) + p_{2,u+1}$
- jobs $u+1$ to $v-1$ choose machine 2, and job v chooses machien 1



Warm-up: 1-lookahead on 2 unrelated machines

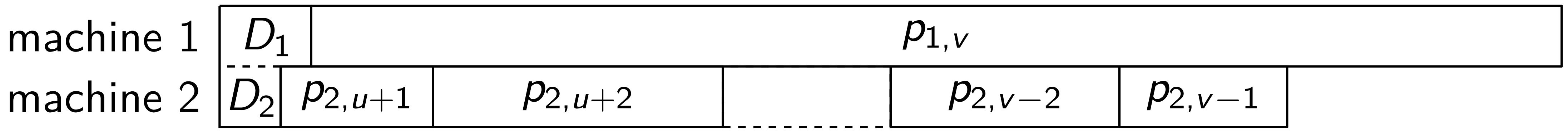
- **Lemma**:

  $L_{\max} \le \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

$$J = \{\underbrace{1,\ 2,\ 3,}\Big|\underbrace{4,\ 5,\ 6,\ 7,}\Big|\underbrace{\ldots\ ,n-5,\ n-4,}\Big|\underbrace{n-3,\ n-2,\ n-1,\ n}\}$$

		$u+1$	$u+2$	$\cdots$	$v-1$	v
machine 1	$D_1(u)$	$p_{1,u+1}$	$p_{1,u+2}$	$\cdots$	$p_{1,v-1}$	$p_{1,v}$
machine 2	$D_2(u)$	$p_{2,u+1}$	$p_{2,u+2}$	$\cdots$	$p_{2,v-1}$	$p_{2,v}$

- $D_1(u) + p_{1,u+1} \le D_2(u) + p_{2,u+1}$
- jobs $u+1$ to $v-1$ choose machine 2, and job v chooses machien 1



Warm-up: 1-lookahead on 2 unrelated machines

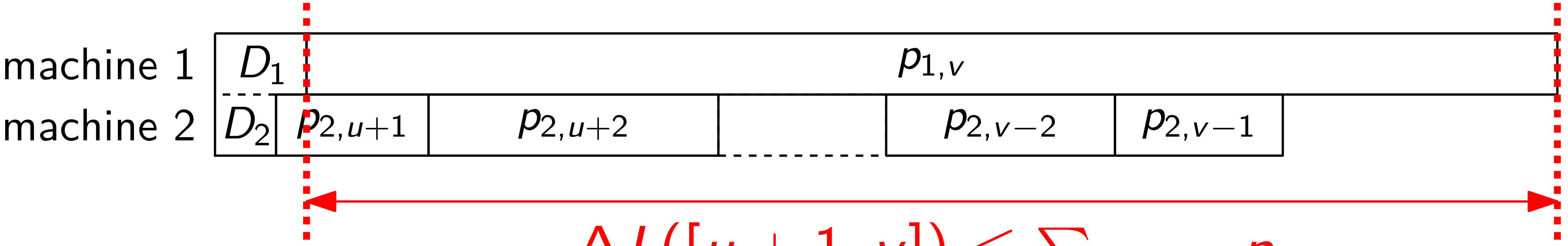
- **Lemma**:

  $L_{\max} \leq \sum_{j \in N} p_j$, where $p_j = \min_{i \in M} p_{i,j}$, i.e., the minimum processing time of job j

$$J = \{\underbrace{1,\ 2,\ 3,}\,\Big|\,\underbrace{4,\ 5,\ 6,\ 7,}\,\Big|\,\underbrace{\dots\ , n-5,\ n-4,}\,\Big|\,\underbrace{n-3,\ n-2,\ n-1,\ n}\}$$

		$u+1$	$u+2$	$\cdots$	$v-1$	v
machine 1	$D_1(u)$	$p_{1,u+1}$	$p_{1,u+2}$	$\cdots$	$p_{1,v-1}$	$p_{1,v}$
machine 2	$D_2(u)$	$p_{2,u+1}$	$p_{2,u+2}$	$\cdots$	$p_{2,v-1}$	$p_{2,v}$

- $D_1(u) + p_{1,u+1} \leq D_2(u) + p_{2,u+1}$
- jobs $u+1$ to $v-1$ choose machine 2, and job v chooses machien 1



$\Delta L([u+1, v]) \leq \sum_{u<j\leq v} p_j$

k-lookahead on m machines

- **Lemma**:

$\Delta L([\ell : n]) - \Delta L([\ell + 1 : n]) \leq p_\ell + \Delta L(K_\ell)$ for $\ell = 1, 2, \dots, n$

k-lookahead on m machines

- **Lemma**:

$\Delta L([\ell : n]) - \Delta L([\ell + 1 : n]) \leq p_\ell + \Delta L(K_\ell)$ for $\ell = 1, 2, \ldots, n$

$$J = \{1,\ 2,\ 3,\ \ldots,\ \boxed{\ell,}\ \ell+1,\ \ell+2,\ \ldots,\ \ell+k,\ \ldots,\ n-3,\ n-2,\ n-1,\ n\}$$

k-lookahead on m machines

- **Lemma**:

$\Delta L([\ell : n]) - \Delta L([\ell + 1 : n]) \leq p_\ell + \Delta L(K_\ell)$ for $\ell = 1, 2, \dots, n$

$$J = \{1,\ 2,\ 3,\ \dots,\ \underbrace{\boxed{\ell,}\ \ell+1,\ \ell+2,\ \dots,\ \ell+k,\ \dots,\ n-3,\ n-2,\ n-1,\ n}_{\Delta L([\ell\,:\,n])}\}$$

k-lookahead on m machines

- **Lemma**:

  $\Delta L([\ell : n]) - \Delta L([\ell+1 : n]) \leq p_\ell + \Delta L(K_\ell)$ for $\ell = 1, 2, \ldots, n$

$$J = \{1,\ 2,\ 3,\ \ldots,\ \underbrace{\boxed{\ell,}\ \underbrace{\ell+1,\ \ell+2,\ \ldots,\ \ell+k,\ \ldots,\ n-3,\ n-2,\ n-1,\ n}_{\Delta L([\ell+1\,:\,n])}}_{\Delta L([\ell\,:\,n])}\}$$

k-lookahead on m machines

- **Lemma**:

$\Delta L([\ell : n]) - \Delta L([\ell + 1 : n]) \leq p_\ell + \Delta L(K_\ell)$ for $\ell = 1, 2, \dots, n$

$$J = \{1,\ 2,\ 3,\ \dots,\ \underbrace{\overbrace{\boxed{\ell},\ \ell+1,\ \ell+2,\ \dots,\ \ell+k}^{\Delta L(K_\ell)},\ \dots,\ n-3,\ n-2,\ n-1,\ n}_{\Delta L([\ell : n])}\}$$

$$\underbrace{\ell+1,\ \dots,\ n}_{\Delta L([\ell + 1 : n])}$$

k-lookahead on m machines

- **Lemma**:

  $\Delta L([\ell : n]) - \Delta L([\ell + 1 : n]) \leq p_\ell + \Delta L(K_\ell)$ for $\ell = 1, 2, \dots, n$

$$J = \{1,\ 2,\ 3,\ \dots,\ \underbrace{\overbrace{\boxed{\ell,}\ \ell+1,\ \ell+2,\ \dots,\ \ell+k,}^{\Delta L(K_\ell)}\ \dots,\ n-3,\ n-2,\ n-1,\ n}_{\Delta L([\ell : n])}\}$$

$$\underbrace{\ell+1,\ \dots,\ n}_{\Delta L([\ell+1 : n])}$$

- $\Delta L([1 : n]) \leq \Delta L([2 : n]) + p_1 + \Delta L(K_1) \leq \dots \leq \sum_{j=1}^{n} p_j + \sum_{j=1}^{n} \Delta L(K_j)$

k-lookahead on m machines

- **Lemma**:

  $\Delta L([\ell : n]) - \Delta L([\ell + 1 : n]) \leq p_\ell + \Delta L(K_\ell)$ for $\ell = 1, 2, \dots, n$

$$J = \{1,\ 2,\ 3,\ \dots,\ \underbrace{\overbrace{\boxed{\ell},\ \ell + 1,\ \ell + 2,\ \dots,\ \ell + k}^{\Delta L(K_\ell)},\ \dots,\ n - 3,\ n - 2,\ n - 1,\ n}_{\Delta L([\ell : n])}\}$$

$$\underbrace{\ell + 1,\ \ell + 2,\ \dots,\ \ell + k,\ \dots,\ n - 3,\ n - 2,\ n - 1,\ n}_{\Delta L([\ell + 1 : n])}$$

- $\Delta L([1 : n]) \leq \Delta L([2 : n]) + p_1 + \Delta L(K_1) \leq \dots \leq \sum_{j=1}^{n} p_j + \sum_{j=1}^{n} \boxed{\Delta L(K_j)}$
- $\Delta L(K_j)$: k players with full rationality

k-lookahead on m machines

- [Chen, Giessler, Mamageishvili, Mihalák, Penna, 2020]:

  For 2 unrelated machines: $\Delta L(K_j) \leq (k-1) \sum_{j \in K_j} p_j$

- [Leme, Syrgkanis,Tardos, 2012]:

  For m unrelated machines: $\Delta L(K_j) \leq 2^k \sum_{j \in K_j} p_j$

- [Bilò, Flammini, Monaco, Moscardelli, 2015]:

  For m unrelated machines: $\Delta L(K_j) \leq 2^k \max_{j \in K_j} p_j$

k-lookahead on m machines

- [Chen, Giessler, Mamageishvili, Mihalák, Penna, 2020]:

For 2 unrelated machines: $\Delta L(K_j) \leq (k-1)\sum_{j \in K_j} p_j$

- [Leme, Syrgkanis,Tardos, 2012]:

For m unrelated machines: $\Delta L(K_j) \leq 2^k \sum_{j \in K_j} p_j$

- [Bilò, Flammini, Monaco, Moscardelli, 2015]:

For m unrelated machines: $\Delta L(K_j) \leq 2^k \max_{j \in K_j} p_j$

$$\Delta L([1:n]) \leq \Delta L([2:n]) + p_1 + \Delta L(K_1) \leq \ldots \leq \sum_{j=1}^{n} p_j + \sum_{j=1}^{n} \Delta L(K_j)$$

- **Theorem**:

For the sequential scheduling game where players have k-lookahead, the SPoA is at most $O(k^2)$ for the two unrelated machines case, and at most $O(2^k \cdot \min\{mk, n\})$ for the m unrelated machines case.

Simple-minded Players

- **Theorem**:

  For the sequential scheduling game on m unrelated machines where players are *simple-minded*, SPoA $= m$.

Simple-minded Players

- **Theorem**:

  For the sequential scheduling game on m unrelated machines where players are *simple-minded*, SPoA $= m$.

Proof (of SPoA $\leq$ m):

- Define: $A_i(j) = D_i(j) + P_i([j+1:n])$

  $D_i(j)$: the load of machine i due to the first j jobs

  $P_i([j+1:n])$: the total processing time of the jobs who are assumed by job j to choose machine i (i.e. the jobs have minimum processing time on machine i)

- Claim: $A_{\max}(\ell) \leq A_{\max}(\ell - 1)$

- SPoA $= \frac{L_{\max}}{OPT} = \frac{A_{\max}(n)}{OPT} \leq \frac{A_{\max}(0)}{OPT} \leq \frac{\sum_{j=1}^{n} A_j(0)}{OPT} \leq \frac{\sum_{j=1}^{n} p_j}{\sum_{j=1}^{n} p_j/m} = m$

Simple-minded Players

- **Theorem**:

For the sequential scheduling game on m unrelated machines where players are *simple-minded*, SPoA $= m$.

Proof (of SPoA $\geq$ m):

	job 1	job 2	job 3	job 4
machine 1	$1-\epsilon$	$1-\epsilon$	$1-\epsilon$	$1-\epsilon$
machine 2	$4-5\epsilon$	1	∞	∞
machine 3	∞	$3-4\epsilon$	1	∞
machine 4	∞	∞	$2-3\epsilon$	1

Simple-minded Players

- **Theorem**:

For the sequential scheduling game on m unrelated machines where players are *simple-minded*, SPoA $= m$.

Proof (of SPoA $\geq$ m):

	job 1	job 2	job 3	job 4
machine 1	$1-\epsilon$	$1-\epsilon$	$1-\epsilon$	$1-\epsilon$
machine 2	$4-5\epsilon$	1	∞	∞
machine 3	∞	$3-4\epsilon$	1	∞
machine 4	∞	∞	$2-3\epsilon$	1

$L_{\max} \approx 4$

Simple-minded Players

- **Theorem**:

  For the sequential scheduling game on m unrelated machines where players are *simple-minded*, SPoA $= m$.

Proof (of SPoA $\geq$ m):

	job 1	job 2	job 3	job 4
machine 1	$1-\epsilon$	$1-\epsilon$	$1-\epsilon$	$1-\epsilon$
machine 2	$4-5\epsilon$	1	∞	∞
machine 3	∞	$3-4\epsilon$	1	∞
machine 4	∞	∞	$2-3\epsilon$	1

$L_{\max} \approx 4$

$OPT = 1$

- SPoA $\geq$ m

Summary

A summary of the results

	2 machines	m machines
0-lookaahead (online greedy)	2	m
1-lookaahead		
k-lookaahead		
n-lookaahead (full rationality)	$\Theta(n)$	$O(2^n)$
Simple-minded		

Summary

A summary of the results

	2 machines	m machines
0-lookaahead (online greedy)	2	m
1-lookaahead	2	$O(m)$
k-lookaahead	$O(k^2)$	$O(2^k \min\{mk, n\})$
n-lookaahead (full rationality)	$\Theta(n)$	$O(2^n)$
Simple-minded	2	m

Summary

A summary of the results

	2 machines	m machines
0-lookaahead (online greedy)	2	m
1-lookaahead	2	$O(m)$
k-lookaahead	$O(k^2)$	$O(2^k \min\{mk, n\})$
n-lookaahead (full rationality)	$\Theta(n)$	$O(2^n)$
Simple-minded	2	m

Concluding remark

- reconsidering the "perfect rationality" assumption for the players

Summary

A summary of the results

	2 machines	m machines
0-lookaahead (online greedy)	2	m
1-lookaahead	2	$O(m)$
k-lookaahead	$O(k^2)$	$O(2^k \min\{mk, n\})$
n-lookaahead (full rationality)	$\Theta(n)$	$O(2^n)$
Simple-minded	2	m

Concluding remark

- reconsidering the "perfect rationality" assumption for the players
- Future work:

  - to improve the bounds for the SPoA of k-lookahead model

  - to further understand the role that bounded rationality plays in other games

THANK YOU!