

Computational applications of mechanism design

- ① Task scheduling
 - allocate tasks among agents to minimize **makespan**

Computational applications of mechanism design

- ① Task scheduling
 - allocate tasks among agents to minimize **makespan**
- ② Bandwidth allocation in computer networks
 - allocate the real-valued capacity of a single network link among users with different demand curves

Computational applications of mechanism design

- ① Task scheduling
 - allocate tasks among agents to minimize **makespan**
- ② Bandwidth allocation in computer networks
 - allocate the real-valued capacity of a single network link among users with different demand curves
- ③ Multicast cost sharing
 - share the cost of a multicast transmission among the users who receive it

Computational applications of mechanism design

- ① Task scheduling
 - allocate tasks among agents to minimize **makespan**
- ② Bandwidth allocation in computer networks
 - allocate the real-valued capacity of a single network link among users with different demand curves
- ③ Multicast cost sharing
 - share the cost of a multicast transmission among the users who receive it
- ④ Two-sided matching
 - pair up members of two groups according to their preferences, without imposing any payments
 - e.g., students and advisors; hospitals and interns; kidney donors and recipients

Constrained Mechanism Design

Imagine that the mechanism designer isn't allowed to change the agents' strategy spaces *or* utility functions.

- e.g., improve traffic flow without building new roads or closing existing ones
- e.g., broker peace between two warring countries without changing their military capabilities

Constrained Mechanism Design

Imagine that the mechanism designer isn't allowed to change the agents' strategy spaces *or* utility functions.

What can be done in such settings?

- ① Contracts
- ② Bribes
- ③ Mediators

Constrained Mechanism Design

Imagine that the mechanism designer isn't allowed to change the agents' strategy spaces *or* utility functions.

What can be done in such settings?

1 Contracts

- agents agree to a contract before playing the game
- if anyone deviates, he is punished
- folk-theorem-like result: can implement any social choice function that the agents prefer to the punishment

2 Bribes

3 Mediators

Constrained Mechanism Design

Imagine that the mechanism designer isn't allowed to change the agents' strategy spaces *or* utility functions.

What can be done in such settings?

① **Contracts**

② **Bribes**

- promise to pay agents if a given joint outcome is reached
- these promises can change the game's payoffs so that agents have dominant strategies
- any Nash equilibrium can be transformed into a dominant strategy equilibrium in this way, and the mechanism designer does not even have to pay the agents anything in equilibrium!

③ **Mediators**

Constrained Mechanism Design

Imagine that the mechanism designer isn't allowed to change the agents' strategy spaces *or* utility functions.

What can be done in such settings?

① **Contracts**

② **Bribes**

③ **Mediators**

- Offer to play on behalf of agents
- e.g., in Prisoner's Dilemma, play C if the other agent also uses the mediator; otherwise play D
- using the mediator's services is weakly dominant, and leads to both agents cooperating in equilibrium
- in a broad class of other games, mediators can be used to implement the optimal-surplus outcome