Mechanism Design

CS 886

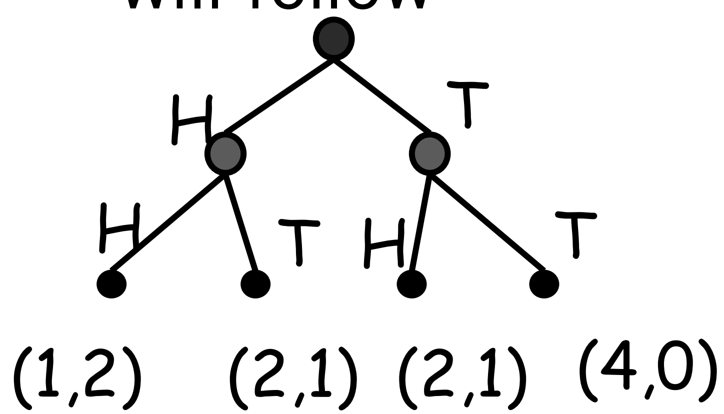
Multiagent Systems

University of Waterloo

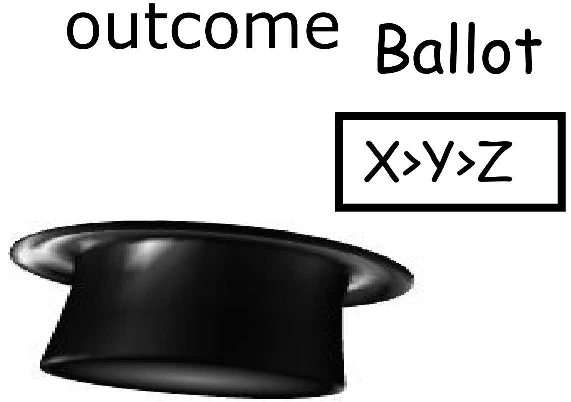
Introduction

So far we have looked at

- Game Theory
  - Given a game we are able to analyze the strategies agents will follow



- Social Choice Theory
  - Given a set of agents' preferences we can choose some outcome



Introduction

- Today, Mechanism Design
  - Game Theory + Social Choice
- Goal of Mechanism Design is to
  - Obtain some outcome
  - But agents are rational
- "Solution":
  - Define the rules of a game so that in equilibrium the agents do what we want
- CS Spin
  - Defining protocols for distributed systems

Example: London Bus System
(as of April 2004)

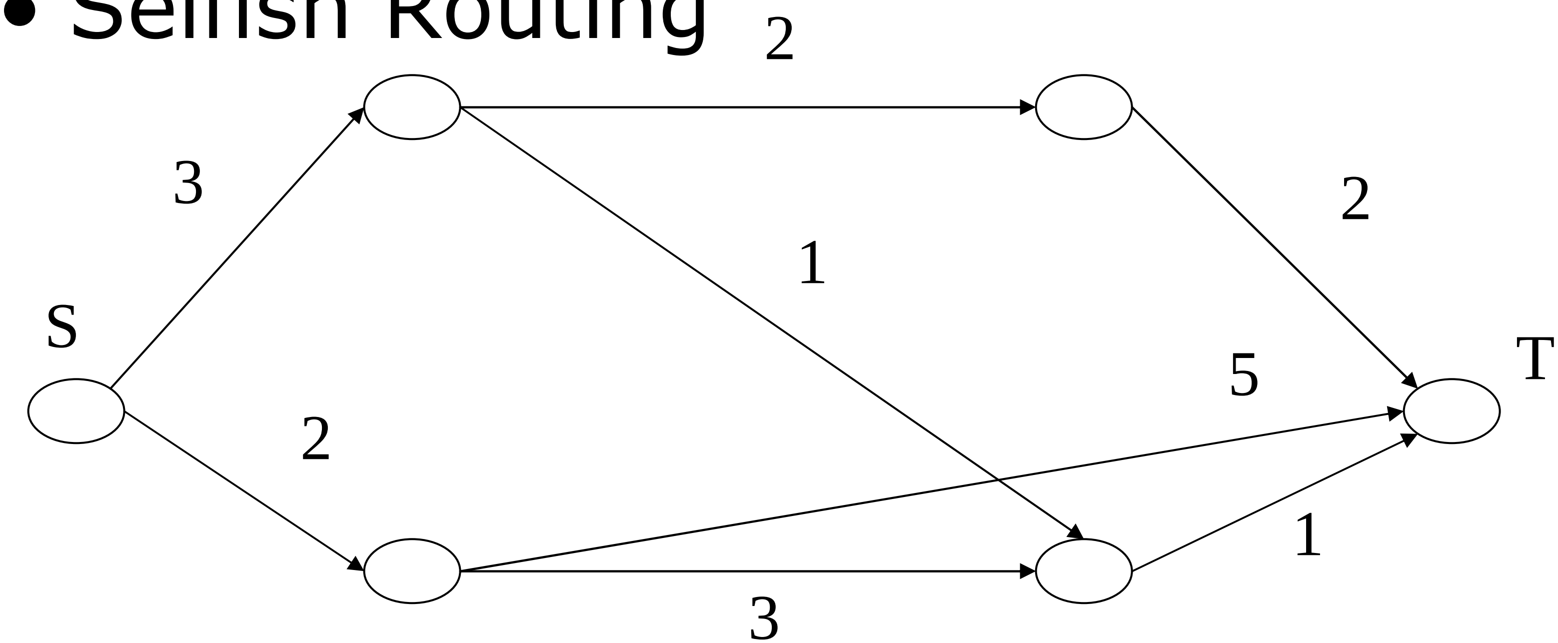
- 5 million passengers each day
- 7500 buses
- 700 routes

- The system has been privatized since 1997 by using competitive tendering

- Idea: Run an auction (mechanism) to allocate routes to companies

Example

- Selfish Routing



Want to find the least-cost route from S to T.
You do not know costs.
You do know that each links wants to maximize revenue.

How do you use this information to extract information needed to find least-cost path?

Fundamentals

- Set of possible outcomes, O

- Agents $i \in N$, $|N|=n$, each agent i has type $\theta_i \in \Theta_i$
  - Type captures all private information that is relevant to agent's decision making

- Utility $u_i(o, \theta_i)$, over outcome $o \in O$

- Recall: goal is to implement some system-wide solution
  - Captured by a social choice function

$$\mathbf{f}:\Theta_1 \times \ldots \times \Theta_n \rightarrow \mathcal{O}$$

$\mathbf{f(\theta_1,\ldots\theta_n)=o}$ is a collective choice

Examples of social choice functions

- Voting: choose a candidate among a group

- Public project: decide whether to build a swimming pool whose cost must be funded by the agents themselves

- Allocation: allocate a single, indivisible item to one agent in a group

Mechanisms

- Recall: We want to implement a social choice function
  - Need to know agents' preferences
  - They may not reveal them to us truthfully
- Example:
  - 1 item to allocate, and want to give it to the agent who values it the most
  - If we just ask agents to tell us their preferences, they may lie

Mechanism Design Problem

- By having agents interact through an institution we might be able to solve the problem

- Mechanism:

$$M=(S_1,\ldots,S_n, g(\cdot))$$

Strategy spaces of agents

Outcome function

$$g: S_1 \times \ldots \times S_n \rightarrow O$$

Implementation

- A mechanism $M=(S_1,\ldots,S_n,g())$ implements social choice function $f(\theta)$ if there is an equilibrium strategy profile $s^*=(s_1^*,\ldots,s_n^*)$ of the game induced by M such that

  – $g(s_1^*(\theta_1),\ldots,s_n^*(\theta_n))=f(\theta_1,\ldots,\theta_n)$
  $\forall\ (\theta_1,\ldots,\theta_n)\in \Theta_1\times\ldots\times\Theta_n$

Implementation

- We did not specify the type of equilibrium in the definition
- Nash

$$u_i(s_i^*(\theta_i),s^*_{-i}(\theta_{-i}),\theta_i) \geq u_i(s_i'(\theta_i),s^*_{-i}(\theta_{-i}),\theta_i),\ \forall\, i,\ \forall\, \theta,\ \forall\, s_i' \neq s_i^*$$

- Bayes-Nash

$$E[u_i(s_i^*(\theta_i),s^*_{-i}(\theta_{-i}),\theta_i)] \geq E[u_i(s_i'(\theta_i),s^*_{-i}(\theta_{-i}),\theta_i)],\ \forall\, i,\ \forall\, \theta,\ \forall\, s_i' \neq s_i^*$$

- Dominant

$$u_i(s_i^*(\theta_i),s_{-i}(\theta_i),\theta_i) \geq u_i(s_i'(\theta_i),s_{-i}(\theta_{-i}),\theta_i),\ \forall\, i,\ \forall\, \theta,\ \forall\, s_i' \neq s_i^*,\ \forall\, s_{-i}$$

Direct Mechanisms

- Recall that a mechanism specifies the strategy sets of the agents
  - These sets can contain complex strategies
- **Direct mechanisms:**
  - Mechanism in which $S_i=\Theta_i$ for all i, and $g(\theta)=f(\theta)$ for all $\theta\in\Theta_1\times\ldots\times\Theta_n$
- **Incentive compatible:**
  - A direct mechanism is incentive compatible if it has an equilibrium s^* where $s^*_i(\theta_i)=\theta_i$ for all $\theta_i\in\Theta_i$ and all i
    - truth telling by all agents is an equilibrium
  - Strategy-proof if dominant-strategy equilibrium

Dominant Strategy Implementation

- Is a certain social choice function implementable in dominant strategies?
  - In principle we would need to consider all possible mechanisms

- **Revelation Principle**
  - Suppose there exists a mechanism $M=(S_1,\ldots,S_n,g(\cdot))$ that implements social choice function f() in dominant strategies.
  - Then there is a direct strategy-proof mechanism, M', which also implements f().

Revelation Principle

"*the computations that go on within the mind of any bidder in the nondirect mechanism are shifted to become part of the mechanism in the direct mechanism*" [McAfee&McMillian 87]

- Vickrey auction and English auction

Revelation Principle: Proof

- $M=(S_1,\ldots,S_n,g())$ implements SCF f() in dom str.
  - Construct direct mechanism $M'=(\Theta^n,f(\theta))$
  - By contradiction, assume

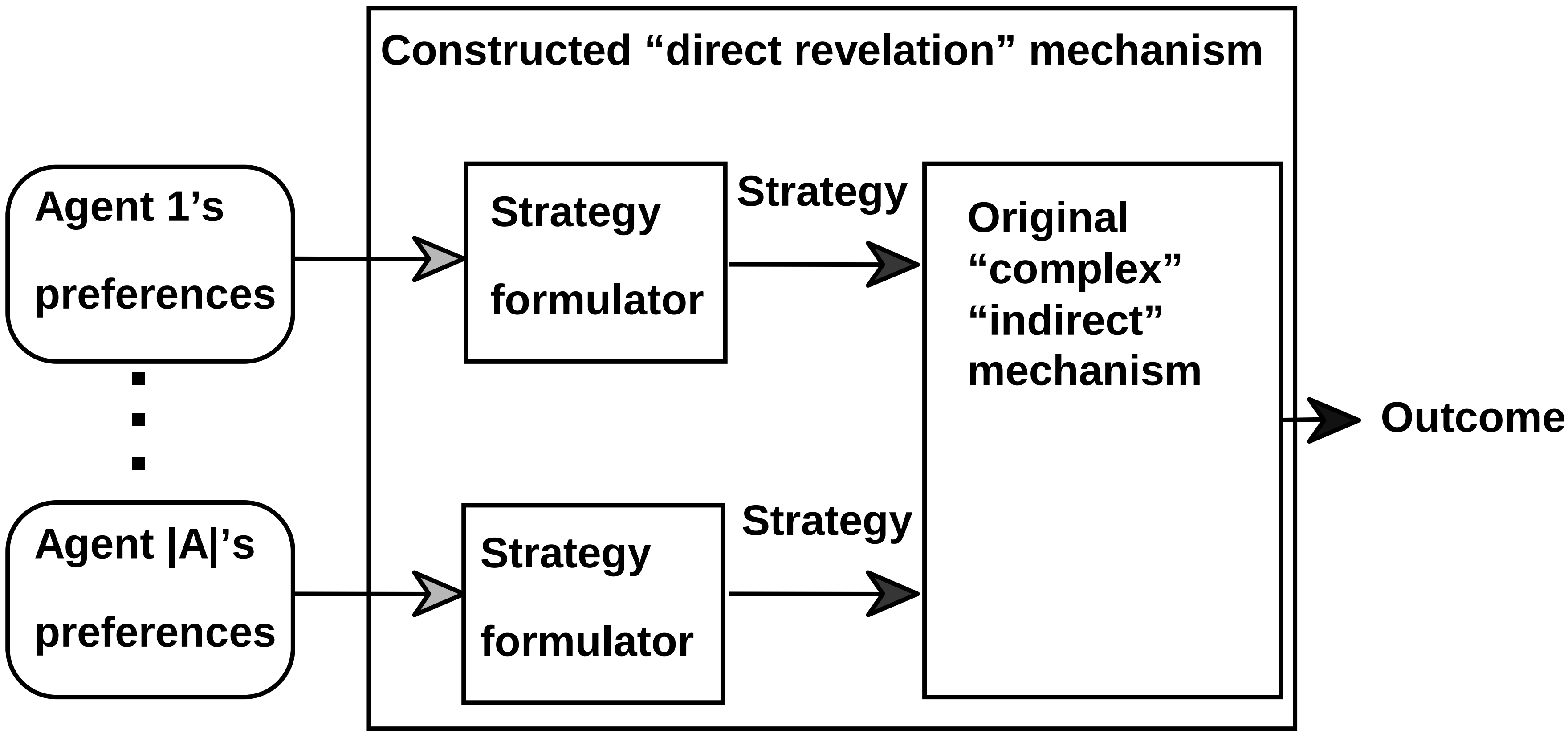
  $\exists\ \theta_i' \neq \theta_i$ s.t. $u_i(f(\theta_i',\theta_{-i}),\theta_i) > u_i(f(\theta_i,\theta_{-i}),\theta_i)$

  for some $\theta_i' \neq \theta_i$, some θ_{-i}.
  - But, because $f(\theta)=g(s^*(\theta))$, this implies

  $u_i(g(s_i^*(\theta_i'),s_{-i}^*(\theta_{-i})),\theta_i) > u_i(g(s^*(\theta_i),s^*(\theta_{-i})),\theta_i)$

  Which contradicts the strategy proofness of s^* in M

Revelation Principle: Intuition



Theoretical Implications

- Literal interpretation: Need only study direct mechanisms
    - This is a smaller space of mechanisms

  – Negative results
    - If no direct mechanism can implement SCF f() then no mechanism can do it

  – Analysis tool:
    - Best direct mechanism gives us an upper bound on what we can achieve with an indirect mechanism
    - Analyze all direct mechanisms and choose the best one

Practical Implications

- Incentive-compatibility is “free” from an implementation perspective
- **BUT!!!**
  - A lot of mechanisms used in practice are not direct and incentive-compatible
    - Maybe there are some issues that are being ignored here

Quick review

- We now know
  - What a mechanism is
  - What is means for a SCF to be dominant strategy implementable
  - Implementable in dominant strategies $\Rightarrow$ implementable by a direct incentive-compatible mechanism
- We do not know
  - What types of SCF are dominant-strategy implementable

Gibbard-Satterthwaite Thm

- Assume
  - $\mathcal{O}$ is finite and $|\mathcal{O}| \geq 3$
  - Each $o \in \mathcal{O}$ can be achieved by social choice function f() for some θ

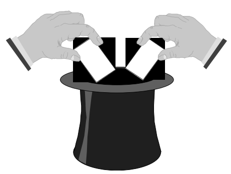
Then:

f() is truthfully implementable in

dominant strategies $\leftrightarrow$ f() is dictatorial

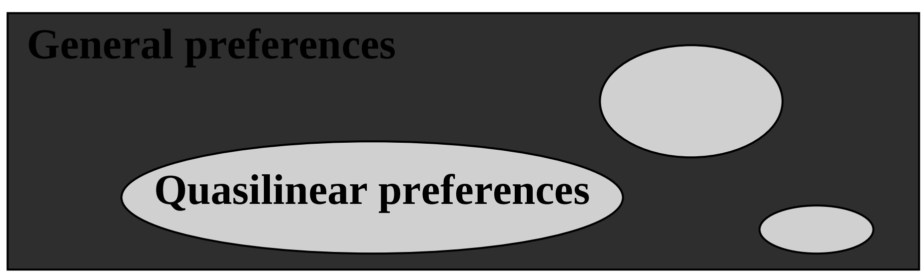
Circumventing G-S

- Use a weaker equilibrium concept
  - Nash, Bayes-Nash
- Design mechanisms where computing a beneficial manipulation is hard
  - Many voting mechanisms are NP-hard to manipulate (or can be made NP-hard with small "tweaks) [Bartholdi, Tovey, Trick 89] [Conitzer, Sandholm 03]
- Randomization



Almost need this much

- Agents' preferences have special structure

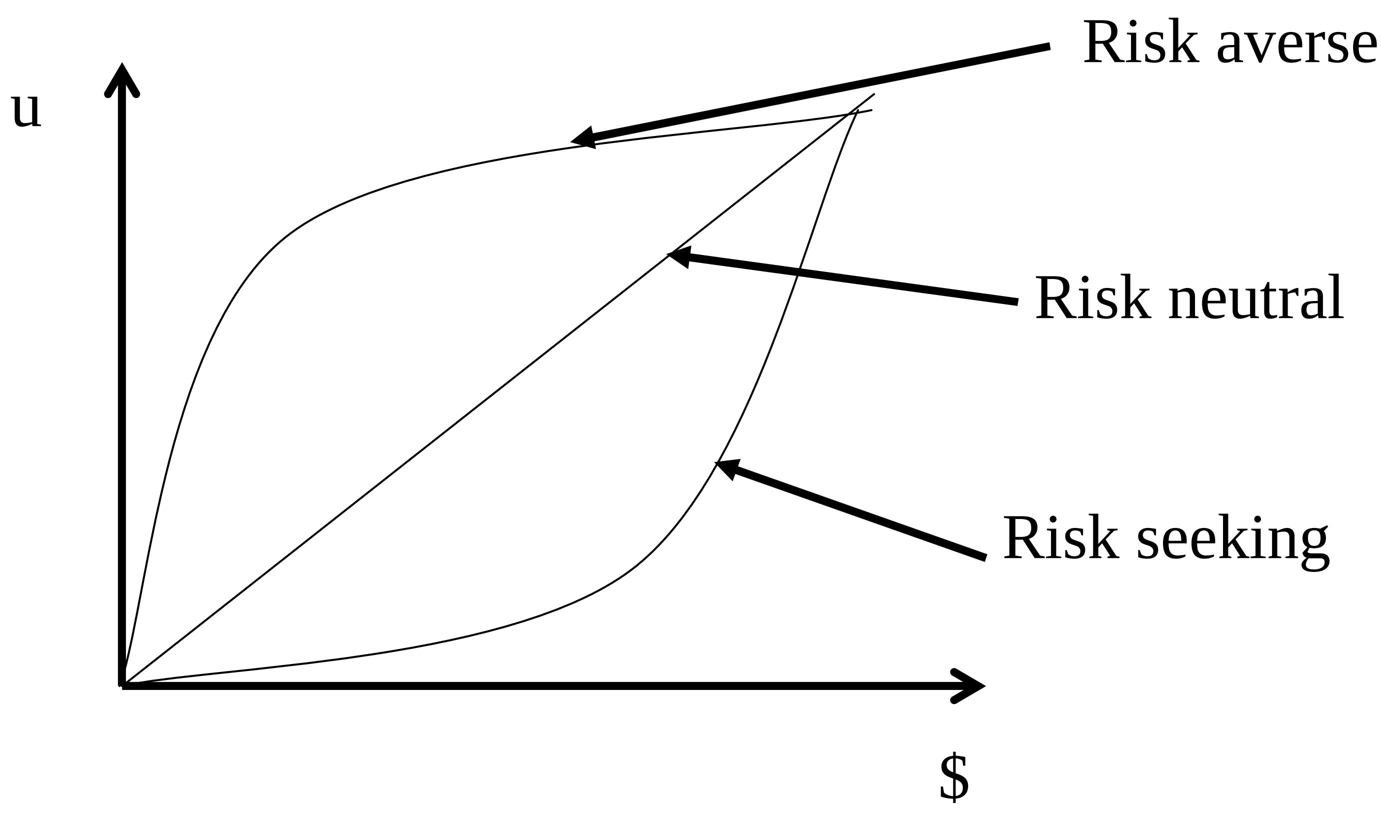


Quasi-Linear Preferences

- Outcome: $o=(x,t_1,\ldots,t_n)$
  - x is a "project choice"
  - t_i is a "monetary" transfer
- Utility of agent i:
  - $U_i(o,\theta_i)=u_i(x,\theta_i)-f(t_i)$
  - Preference of x is independent from the payment
  - Can choose to reward or punish by a monetary amount

Quasi-linear preferences

- $U_i(o,\theta_i)=u_i(x,\theta_i)-f_i(t_i)$
- $f_i()$ gives i's risk attitude



SCF and quasi-linear settings

- $f:\Theta \rightarrow (x(\Theta),t(\Theta))$

- SCF f is efficient if for all types $\theta=(\theta_1,\ldots,\theta_{\mathbf{n}})$
  - $\sum_{i=1}^{n} u_i(x(\theta),\theta_i) \geq \sum_{i=1}^{n} u_i(x'(\theta),\theta_i) \quad \forall\, x'(\theta)$
  - Aka social welfare maximizing

- SCF f is budget-balanced if $\sum^{n}_{i=1} t_i(\theta)=0$

- SCF f is weakly budget-balanced if $\sum^{n}_{i=1} t_i(\theta) \geq 0$

Mechanisms and quasi-linear utilities

- $M=(S_1,\ldots,S_n,(x(S),t(S))$
- Valuation for choice x $v_i(x)=u_i(x,\theta_i)$
- Agents reveal their valuation functions in a direct mechanism
  - v'_i denotes the valuation that agent i declares to the mechanism (may be different from true valuation v_i)
  - $v=(v'_1,\ldots,v'_n)$

Properties of mechanisms

- **Truthful**: $\forall$ i $\forall$ v_i, the equilibrium strategy for agent i is to adopt $v_i' = v_i$
- **Efficient**: Mechanism selects choice x such that $\forall$ I $\forall$ v_i $\forall$ x′ $\sum_i v_i(x) \geq \sum_i v_i(x')$
- **Budget balanced**: $\forall$ v′ $\sum_i t_i(v') = 0$
- **Individually rational**: $v_i(s^*(v)) - t_i(s(v)) \geq 0$ where S^* is the equilibrium

Groves Mechanisms

[Groves 1973]

- A **Groves mechanism**, $M=(S_1,\ldots,S_n, (x,t_1,\ldots,t_n))$ is defined by

  - Choice rule $x^*(\theta')=\arg\max_x \sum_i v_i(x,\theta_i')$
  - Transfer rules
    - $t_i(\theta')=h_i(\theta_{-i}')-\sum_{j\neq i} v_j(x^*(\theta'),\theta'_j)$

  where $h_i(\cdot)$ is an (arbitrary) function that **does not depend** on the reported type θ_i' of agent i

Groves Mechanisms

- **Thm:** Groves mechanisms are strategy-proof and efficient (We have gotten around Gibbard-Satterthwaite!)
- Proof: Agent i's utility for strategy θ_i', given θ_{-i}' from agents $j \neq i$ is

  $U_i(\theta_i') = v_i(x^*(\theta'), \theta_i) - t_i(\theta')$

  $= v_i(x^*(\theta'), \theta_i) + \sum_{j \neq i} v_j(x^*(\theta'), \theta'_j) - h_i(\theta'_{-i})$

  Ignore $h_i(\theta_{-i})$. Notice that

  $x^*(\theta') = \text{argmax} \sum_i v_i(x, \theta'_i)$

  i.e. it maximizes the sum of reported values.

  Therefore, agent i should announce $\theta_i' = \theta_i$ to maximize its own payoff

Thm: *Groves mechanisms are unique (up to $h_i(\theta_{-i})$)*

VCG Mechanism

(aka Clarke mechanism aka Pivotal mechanism)

- Def: Implement efficient outcome,

$$x^*=\max_x \sum_i v_i(x,\theta_i')$$

Compute transfers

$$t_i(\theta')=\sum_{j\neq i} v_j(x^{-i},\theta'_j) -\sum_{j\neq i} v_j(x^*, \theta_i')$$

Where $x^{-i}=\max_x \sum_{j\neq i} v_j(x,\theta_j')$

VCG are efficient and strategy-proof

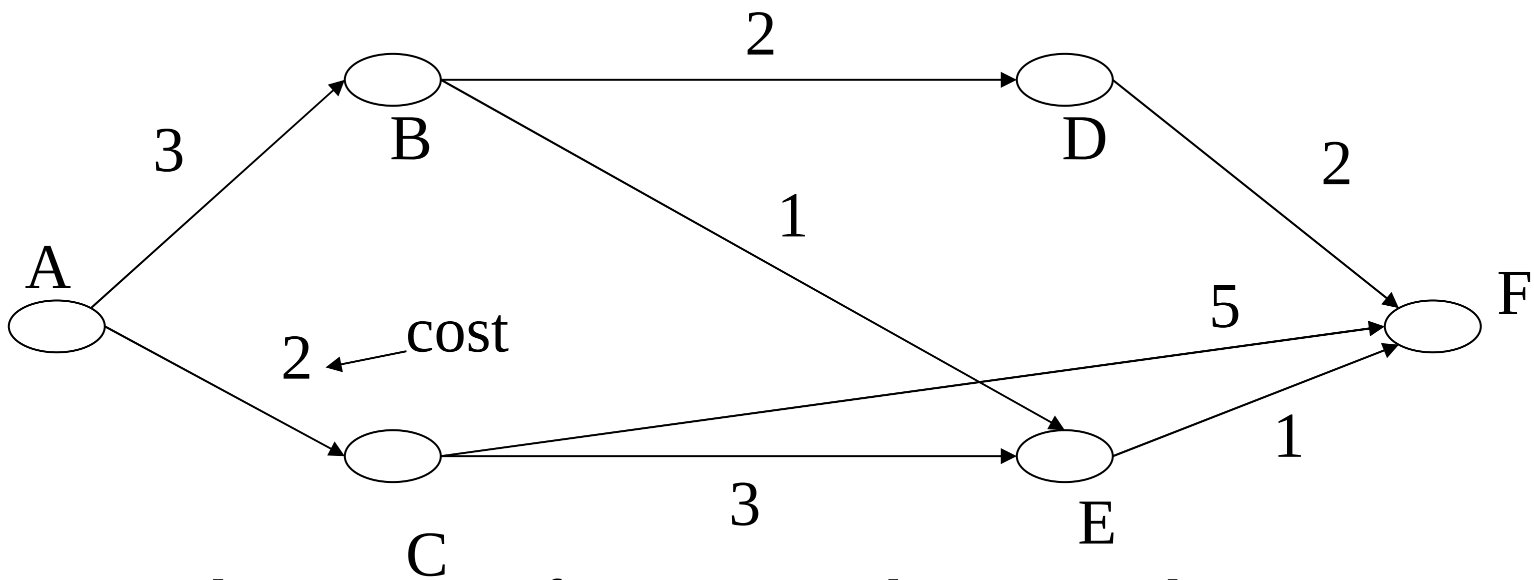
Agent's equilibrium utility is:

$$u_i(x^*,t_i,\theta_i)=v_i(x^*,\theta_i)-[\sum_{j\neq i} v_j(x^{-i},\theta_j) -\sum_{j\neq i} v_j(x^*,\theta_j)]$$

$$= \sum_j v_j(x^*,\theta_j) - \sum_{j\neq i} v_j(x^*,\theta_j)$$

= marginal contribution to the welfare of the system

Example: Selfish Routing



Agent's utility is –cost if its route is chosen, 0 otherwise

$$x(v)=\text{argmax} \sum_i v_i(x) = ABEF$$

Payments:

T_{AC}=5-5=0

T_{CE}=5-5=0

T_{BD}=5-5=0

T_{DF}=5-5=0

T_{DF}=5-5=0

Payments (Pivotal Agents):

T_{AB}=2-6=-4 (paid 4 for its contribution)

T_{BE}=4-6=-2 (paid 2 for its contribution)

T_{EF}=4-7=-3 (paid 3 for its contribution)

"Market Power"

Example: Building a pool

- The cost of building the pool is \$300
- If together all agents value the pool more than \$300 then it will be built
- Clarke Mechanism:
  - Each agent announces their value, v_i
  - If $\sum v_i \geq 300$ then it is built and each pays 100
  - Payments $t_i(\theta_i') = \sum_{j \neq i} v_j(x^{-i}, \theta_j') - \sum_{j \neq i} v_j(x^*, \theta_i')$ if built, 0 otherwise

v1=50, v2=50, v3=250

After each pay 100

Pool should be built

t_1=(250+50)-(250+50)=0

t_2=(250+50)-(250+50)=0

t_3=(0)-(100)=-100

Not budget balanced

Example: Vickrey Auction

- Highest bidder gets item, and pays second highest amount

- Also a VCG mechanism
  - Allocation rule: get item if $b_i = \max_i[b_j]$
  - Every agent pays

$$t_i(\theta_\iota') = \sum_{j\neq i} v_j(x^{-i},\theta'_j) - \sum_{j\neq i} v_j(x^*, \theta_i')$$

$\max_{j\neq i}[b_j]$

$\max_{j^1 i}[b_j]$ if i is not the highest bidder, 0 if it is