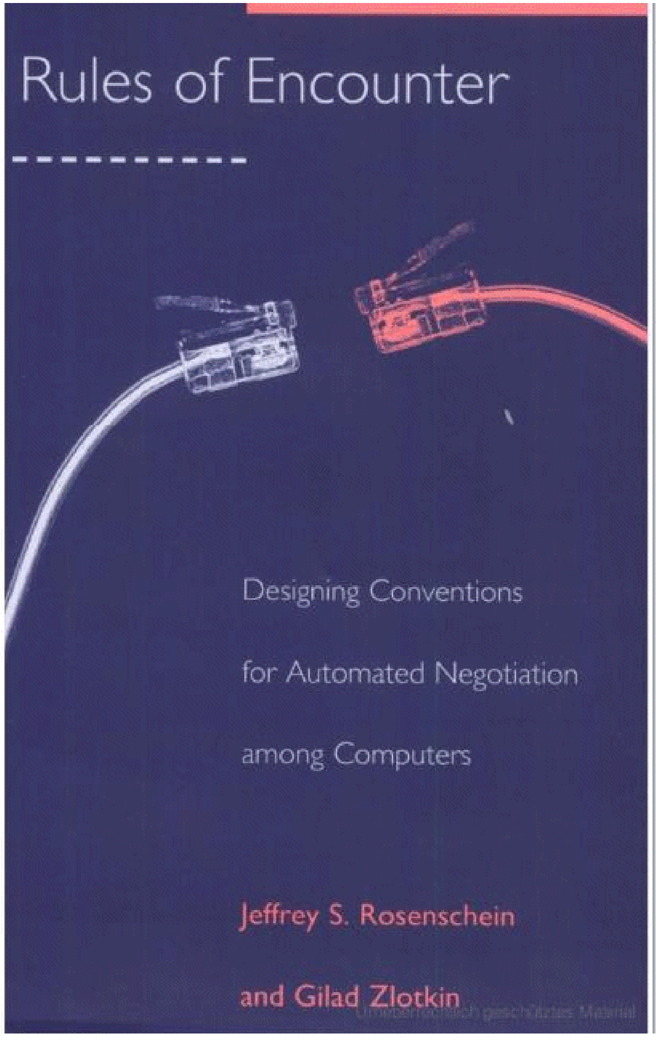


used in PrIAS

Agenda

- What is PrIAS
- Rules Of Encounter
  J.S. Rosenschein, G. Zlotckin

PrIAS

Robust **Pr**obabilistic Reasoning for **I**ntelligent **A**utonomous **S**ystems

- Goal of the project is to find the fundamentals for the development of formally modeled agent systems for building flexible and adaptable software systems.
- Will use concepts from “Probabilistic Logical Reasoning", "Mechanism Design" and "Artificial Knowledge Management“.
- First steps by designing and implementing a show case.
  - Jade, Jadex, Racer, …

The Show Case

Autonomous cargo transport system using LEGO Mindstorms

- Producers build cargo to be delivered
- Each cargo is associated with an agent that is responsible for the transport.
- A cargo agent searches for a transport agent that is able to fulfill his demands either via a broker agent or directly by contacting a known transport agent.
- A transport agent is responsible for the delivery either direct or by **cooperating** with other transport agents.

Rules Of Encounter

- Domain Categories
  - Task Oriented Domains "subset of"
  - State Oriented Domains "subset of"
  - Worth Oriented Domains
- High level Protocol for agent negotiation
- Strategies for agents in negotiation

Mechanism Design has to deal with protocol design and strategies

Negotiation

Negotiation is the process of reaching agreements on matters of common interest

- Any negotiation setting will have four components:
  - A **negotiation set**: possible proposals that agents can make
  - A **protocol**
  - **Strategies**, one for each agent, which might be private
  - A **rule** that determines when a deal has been struck and what the agreement deal is
- Negotiation usually proceeds in a series of rounds, with every agent making a proposal at every round

Heterogeneous, Self-motivated Agents

The systems:

- are not centrally designed
- do not have a notion of global utility
- are dynamic (e.g., new types of agents)
- will not act “benevolently” unless it is in their interest to do so

Broad Working Assumption

- Designers (from different companies, countries, etc.) come together to agree on *standards* for how their automated agents will interact (in a given domain)
- Discuss various possibilities and their tradeoffs, and agree on **protocols**, **strategies**, and **social laws** to be implemented in their machines

Attributes of Standards

ü *Efficient*:	Pareto Optimal
ü *Stable*:	No incentive to deviate
ü *Simple*:	Low computational and communication cost
ü *Distributed*:	No central decision-maker
ü *Symmetric*:	Agents play equivalent roles

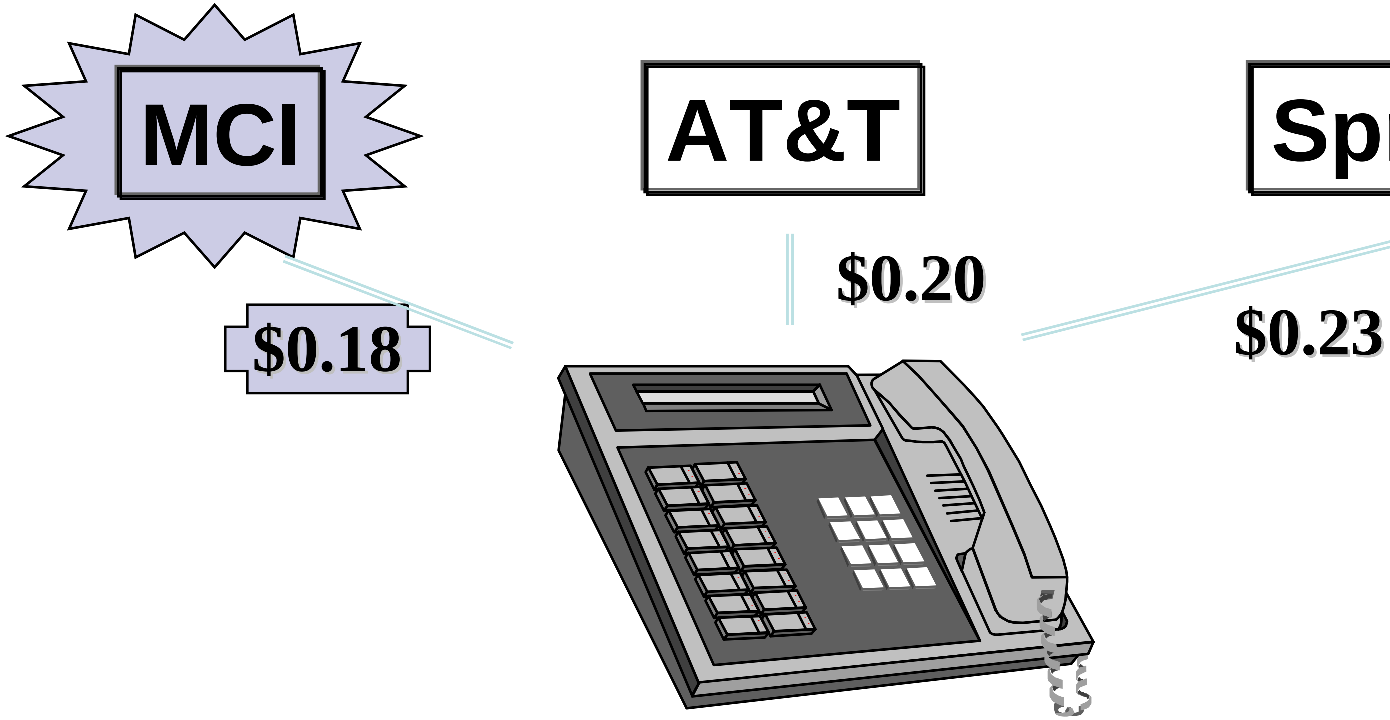
Designing protocols for specific classes of domains that satisfy some or all of these attributes

Phone Call Competition Example

- Customer wishes to place long-distance call
- Carriers simultaneously bid, sending proposed prices
- Phone automatically chooses the carrier (dynamically)

Best Bid Wins

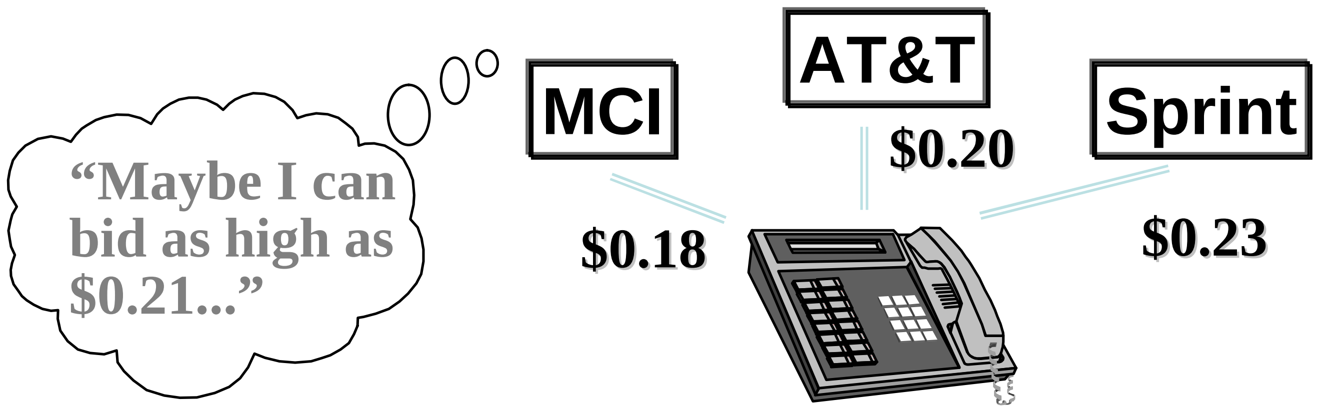
- Phone chooses carrier with lowest bid
- Carrier gets amount that it bid



Attributes of the Mechanism

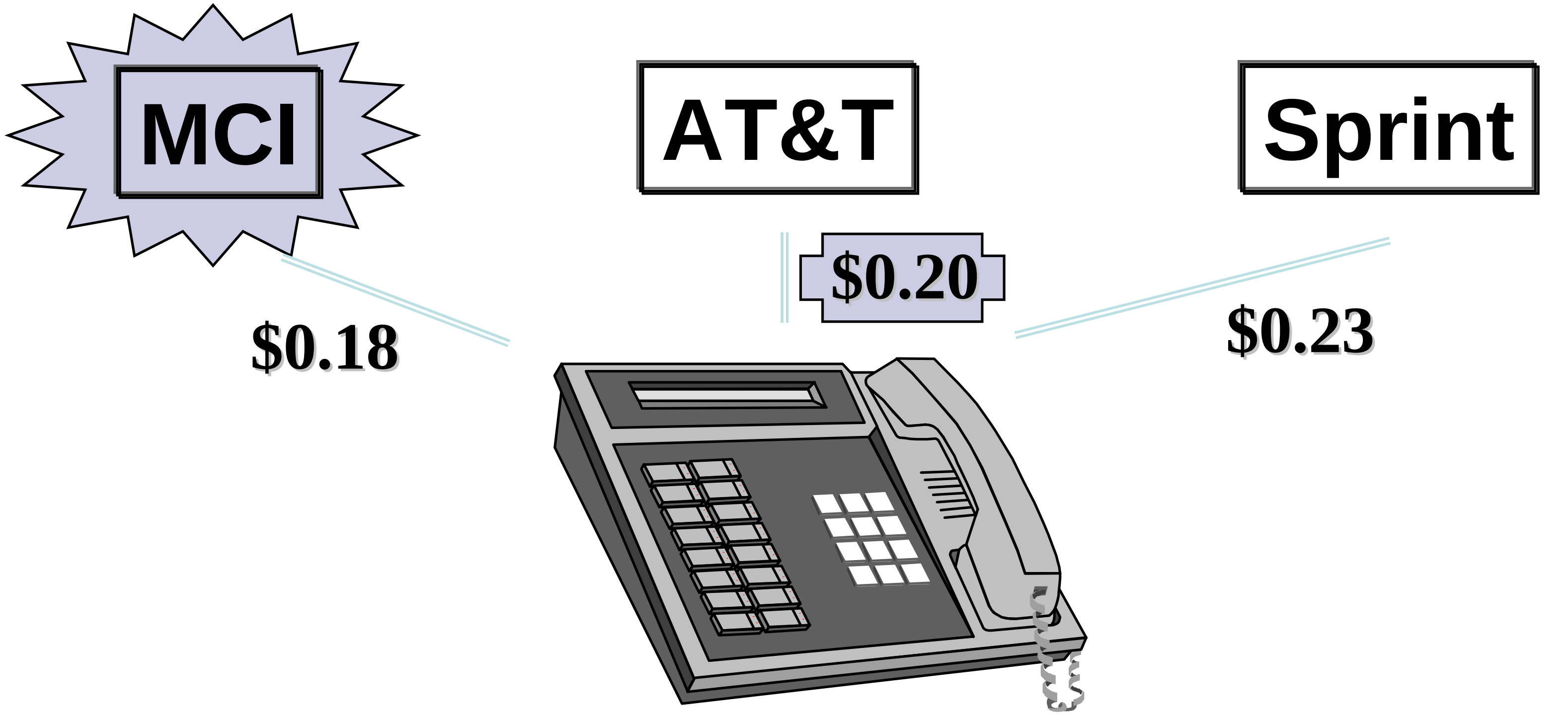
- ü *Distributed*
- ü *Symmetric*
- û *Stable*
- û *Simple*
- û *Efficient*

Carriers have an incentive to invest effort in strategic behavior



Best Bid Wins, Gets Second Price (Vickrey Auction)

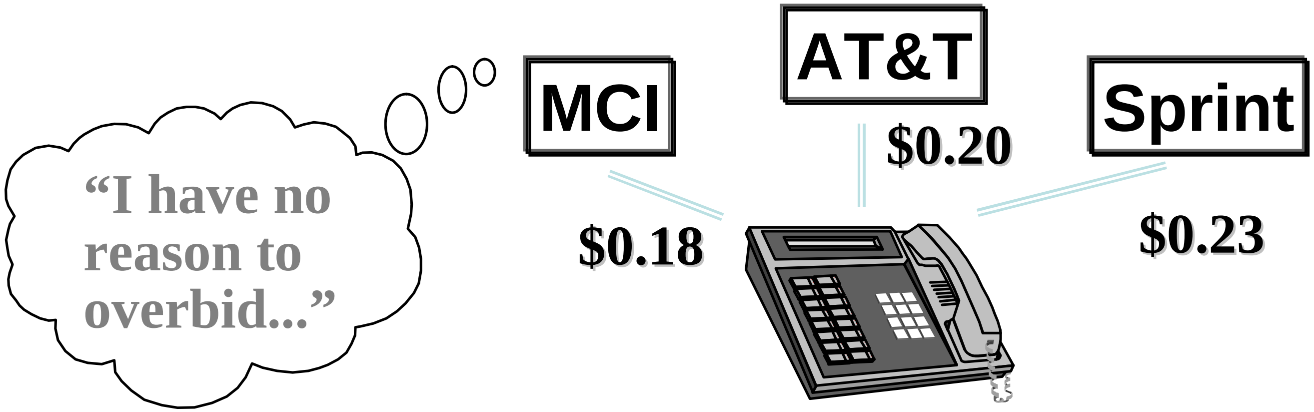
- Phone chooses carrier with lowest bid
- Carrier gets amount of second-best price



Attributes of the Vickrey Mechanism

- ü *Distributed*
- ü *Symmetric*
- ü *Stable*
- ü *Simple*
- ü *Efficient*

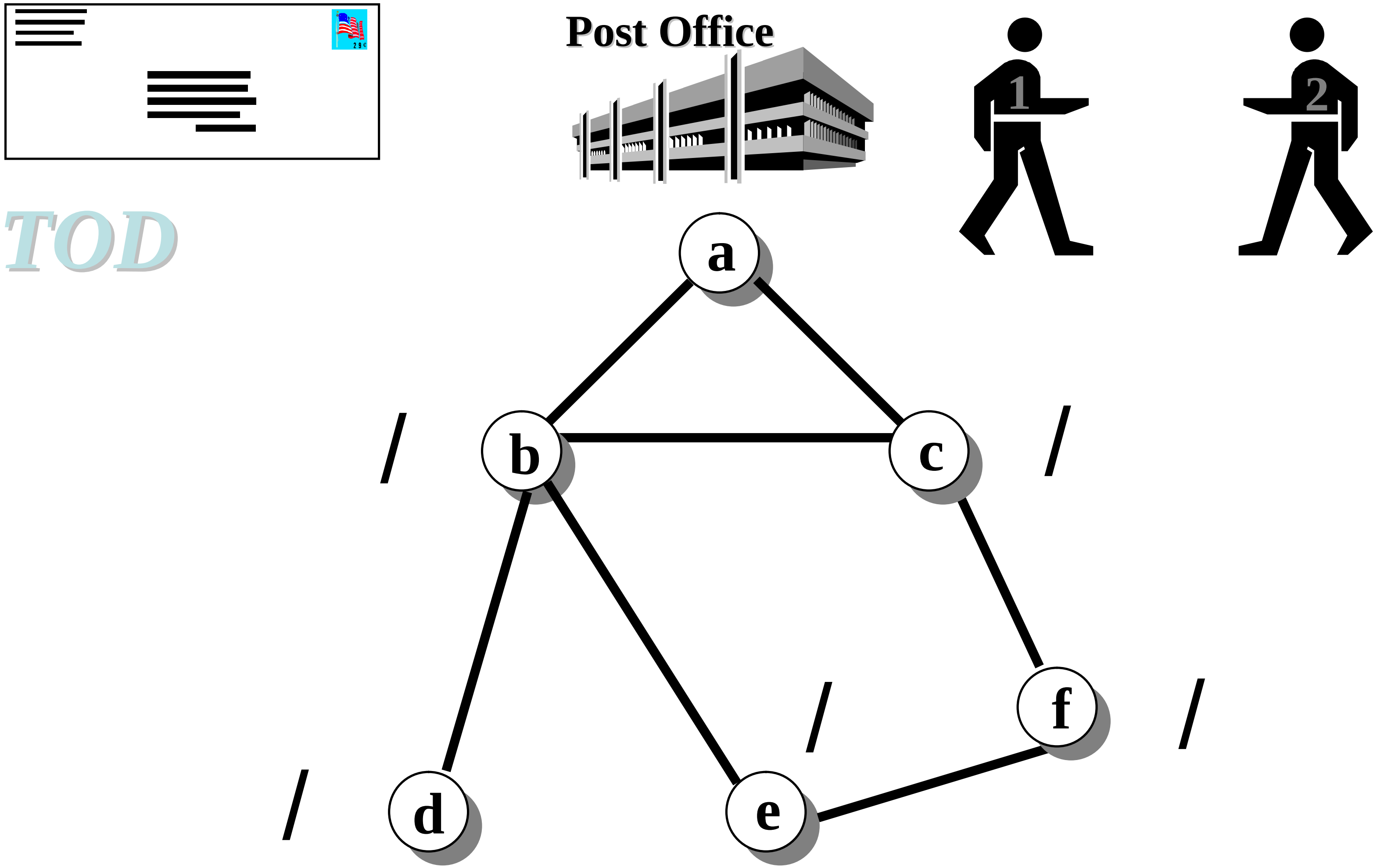
Carriers have *no* incentive to invest effort in strategic behavior



Domain Theory

- Task Oriented Domains
  - Agents have tasks to achieve
  - Task redistribution
- State Oriented Domains
  - Goals specify acceptable final states
  - Side effects
  - Joint plan and schedules
- Worth Oriented Domains
  - Function rating states' acceptability
  - Joint plan, schedules, and goal relaxation

Postmen Domain



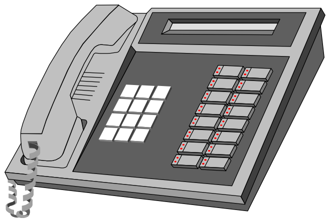
Postmen Domain

- **Description**:
  Agents have to deliver sets of letters to mailboxes, which are arranged on a weighted graph G=G(V,E). No limit on the number of letters that fit in a mailbox. Have to return to post office if finished. Agents can exchange letters at no cost while they are at the post office.
- **Task Set**:
  The set of all addresses in the graph. If address *x* is in an agent task set, it means that he has at least one letter to deliver to *x*.
- **Cost Function**:
  The cost of a subset of addresses $X \subseteq V$, i.e., $c(X)$, is the length of the minimal path that starts at the post office, visits all members of *X*, and ends at the post office.

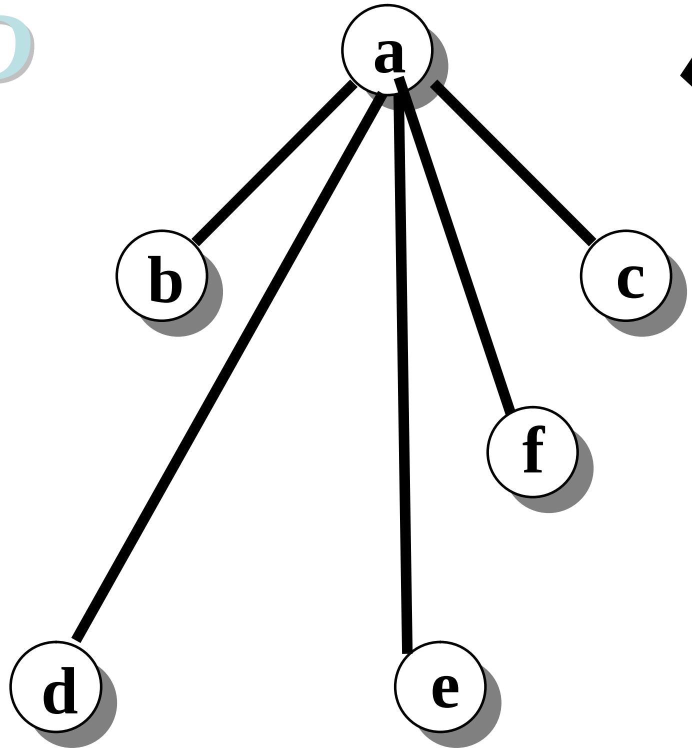
Delivery Domain

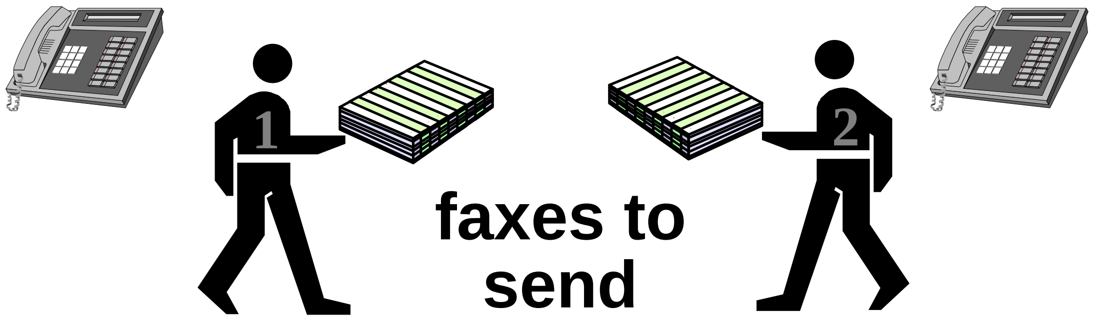
- **Description**:
  Agents have to deliver sets of containers to warehouses, which are arranged on a weighted graph G=G(V,E). No limit on the number of containers that fit in a warehouse. Agents can exchange containers at no cost while they are at the distribution point.
- **Task Set**:
  The set of all addresses in the graph. If address *x* is in an agent task set, it means that he has at least one container to deliver to *x*.
- **Cost Function**:
  The cost of a subset of addresses $X \subseteq V$, i.e., $c(X)$, is the length of the minimal path that starts at the post office, visits all members of X.

Fax Domain



TOD





Cost is only to establish connection

TODs Defined

- A TOD is a triple

  $$<T, Ag, c>$$

  where
  - T is the (finite) set of all possible tasks
  - $Ag = \{1,\ldots,n\}$ is the set of participating agents
  - $c = \wp(T) \rightarrow \Re^+$ defines the *cost* of executing each subset of tasks
- An *encounter* is a collection of tasks

  $$<T_1,\ldots,T_n>$$

  where $T_i \subseteq T$ for each $i \in Ag$

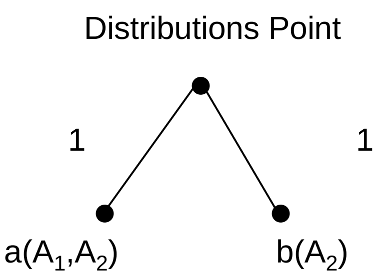
Building Blocks

- Domain
  - A precise definition of what a goal is
  - Agent operations
- Negotiation Protocol
  - A definition of a deal
  - A definition of utility
  - A definition of the conflict deal
- Negotiation Strategy
  - In Equilibrium
  - Incentive-compatible

Deals in TODs

- Given encounter $<T_1, T_2>$, a *deal* δ is an allocation of the tasks $T_1 \cup T_2$ to the agents 1 and 2. I.e. $\delta = <D_1, D_2>$ where $D_1 \cup D_2 = T_1 \cup T_2$
- The *cost* to i of deal $\delta = <D_1, D_2>$ is $c(D_i)$, and will be denoted $cost_i(\delta)$
- The *utility* of deal δ to agent i is:
$$utility_i(\delta) = c(T_i) - cost_i(\delta)$$
- The conflict deal, Θ, is the deal $<T_1, T_2>$ consisting of the tasks originally allocated.
Note that $utility_i(\Theta) = 0$ for all $i \in Ag$
- Deal δ is *individual rational* if it weakly dominates the conflict deal

Example of Deals



Standalone cost of agent 1 $cost_1(T_1) = 1$
Standalone cost of agent 2 $cost_2(T_2) = 3$

1. ({a}, {b}) gives agent 1 no utility and agent 2 utility of 2 (0,2).
2. ({a,b}, Ø) gives (-2, 3)
3. (Ø, {a,b}) gives (1, 0)
4. ({a}, {a,b}) gives (0,0)
5. ({a,b}, {a}) gives (-2,2)

Deal δ dominates δ‘ if δ is better for at least one agent and not worse for the other agent. Neither agent prefers δ‘ over δ and at least one agent prefers δ over δ‘.

Deals

Definition 6 A deal δ is called *individual rational* if $\delta \succeq \Theta$.

A deal is *pareto optimal* if there does not exist another deal that dominates it. A pareto optimal deal cannot be improved upon for one agent without lowering the other agent's utility from the deal.

Definition 7 A deal δ is called *pareto optimal* if there does not exist another deal δ' such that $\delta' \succ \delta$.

In our delivery example, we have the following relationship among deals: $(\emptyset, \{a, b\}) \succ (\{a\}, \{a, b\}) \succ (\{a, b\}, \{a, b\})$. In addition, $(\{a\}, \{b\}) \succ (\{a\}, \{a, b\}) \succ (\{a, b\}, \{a, b\})$. Finally, $(\{a\}, \{b\}) \equiv (\{b\}, \{a\})$
A deal is *individual rational* if it gives both agents non-negative utility.

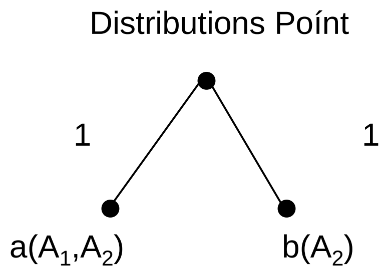
Remember:
1. ({a}, {b}) gives (0,2).
2. ({a,b}, Ø) gives (-2, 3)
3. (Ø, {a,b}) gives (1, 0)
4. ({a}, {a,b}) gives (0,0)
5. ({a,b}, {a}) gives (-2,2)

The Negotiation Set

- The set of deals over which agents negotiate are those that are:
  - individual rational **and**
  - pareto optimal

Individual Rational for Both
(eliminate any choices that are negative for either)

1. ({a}, {b}) = (0,2)
2. ({b}, {a})= (0,2)
3. ~~({a,b}, ϕ) =~~(-2,3)
4. (ϕ, {a,b}) =(1,0)
5. ({a}, {a,b})= (0,0)
6. ({b}, {a,b})= (0,0)
7. ~~({a,b}, {a})~~= (-2,2)
8. ~~({a,b}, {b})~~=(-2,2)
9. ~~({a,b}, {a,b}~~=(-2,0)



individual rational →

1. ({a}, {b})
2. ({b}, {a})
3. (ϕ, {a,b})
4. ({a}, {a,b})
5. ({b}, {a,b})

Pareto Optimal Deals

1. ({a}, {b}) = (0,2)
2. ({b}, {a})= (0,2)
3. ({a,b}, ϕ) =(-2,3)
4. (ϕ, {a,b}) =(1,0)
5. ~~({a}, {a,b}~~)= (0,0)
6. ~~({b}, {a,b}~~)= (0,0)
7. ~~({a,b}, {a}~~)= (-2,2)
8. ~~({a,b}, {b}~~)=(-2,2)
9. ~~({a,b}, {a,b}~~=(-2,0)

but nothing beats 3 for agent 2

Beaten by ({a}{b}) deal

Pareto Optimal

1. ({a}, {b})
2. ({b}, {a})
3. ({a,b}, ϕ)
4. (ϕ, {a,b})

Negotiation Set

Individual Rational Deals

({a}, {b})

({b}, {a})

(ϕ, {a,b})

({a}, {a,b})

({b}, {a,b})

Pareto Optimal Deals

({a}, {b})

({b}, {a})

({a,b}, ϕ)

(ϕ, {a,b})

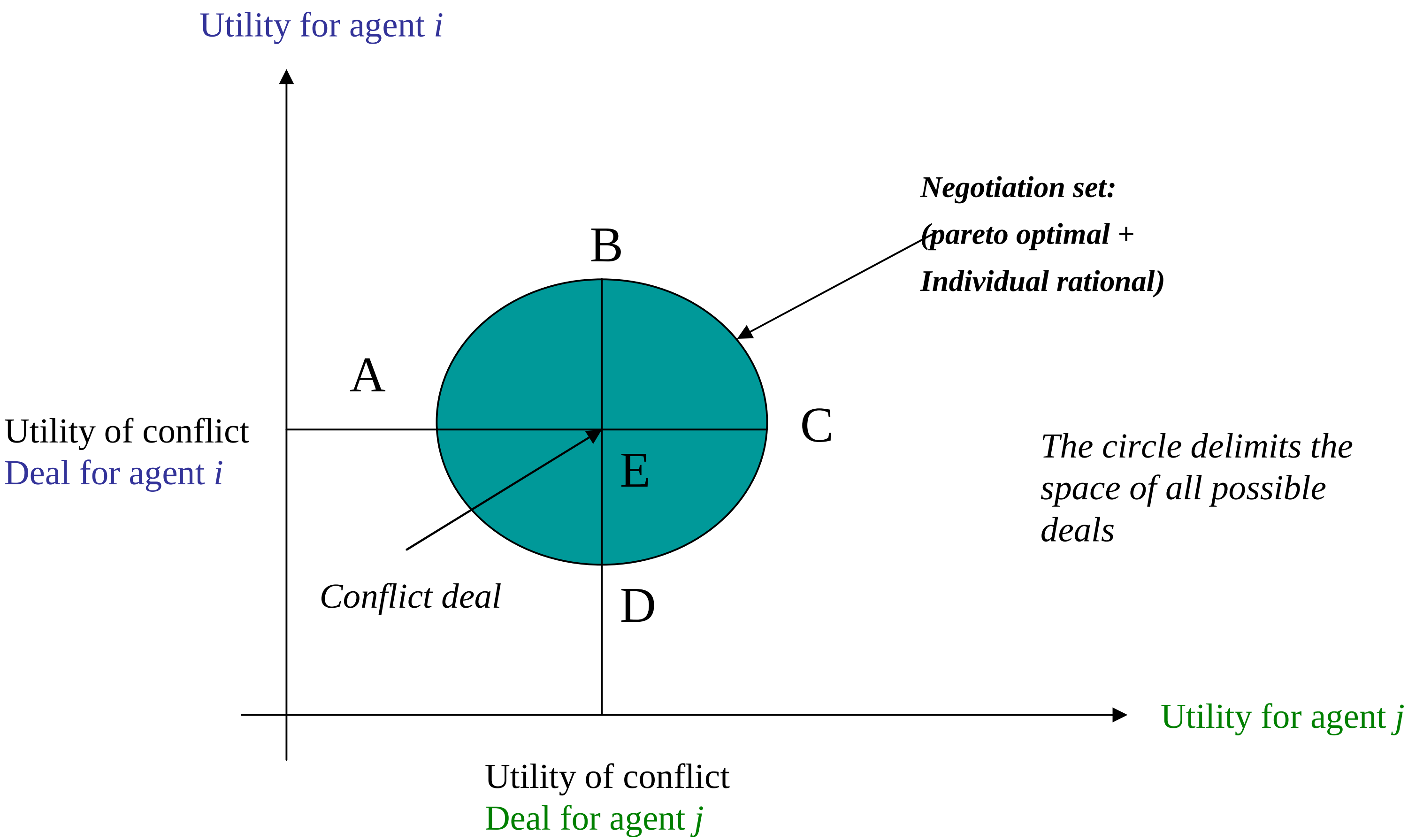
Negotiation Set

({a}, {b})

({b}, {a})

(ϕ, {a,b})

The Negotiation Set Illustrated



Negotiation Protocols

- Agents use a product-maximizing negotiation protocol (as in Nash bargaining theory)
- Examples:
  - monotonic concession protocol…
  - 1-step protocol,

- They need to know their utility functions (e.g., cost and tasks).

The Monotonic Concession Protocol

Rules of this protocol are as follows…

- Negotiation proceeds in rounds
- On round 1, agents simultaneously propose a deal from the negotiation set
- Agreement is reached if one agent finds that the deal proposed by the other is at least as good or better than its proposal
- If no agreement is reached, then negotiation proceeds to another round of simultaneous proposals
- In round $u + 1$, no agent is allowed to make a proposal that is less preferred by the other agent than the deal it proposed at time u
- If neither agent makes a concession in some round $u > 0$, then negotiation terminates, with the conflict deal

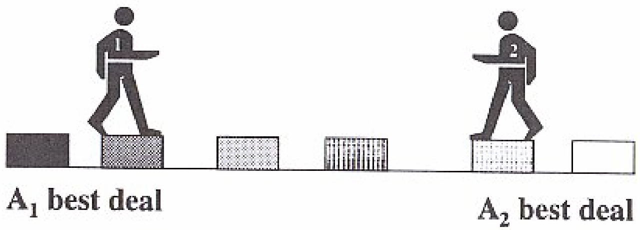


Figure 3.2
The Monotonic Concession Protocol

The Zeuthen Strategy

Three problems:

- What should an agent's first proposal be?

Its most preferred deal

- On any given round, *who should concede?*

The agent least willing to risk conflict

- If an agent concedes, then *how much* should it concede?

Just enough to change the balance of risk

Willingness to Risk Conflict

- Suppose you have conceded a *lot*. Then:
  - Your proposal is now near the conflict deal
  - In case conflict occurs, you are not much worse off
  - You are *more willing* to risk conflict
- An agent will be *more willing* to risk conflict if the difference in utility between its current proposal and the conflict deal is *low*

$$\text{Risk}_i^t = \begin{cases} 1 & \text{if Utility}_i(\delta_i^t) = 0 \\ \dfrac{\text{Utility}_i(\delta_i^t) - \text{Utility}_i(\delta_j^t)}{\text{Utility}_i(\delta_i^t)} & \text{otherwise} \end{cases}$$

In other words (as can be seen in Figure 3.3),

$$\text{Risk}_1^t = \frac{\text{utility agent 1 loses by conceding and accepting agent 2's offer}}{\text{utility agent 1 loses by not conceding and causing a conflict}}$$

Nash Equilibrium Again…

- The Zeuthen strategy is in Nash equilibrium: under the assumption that one agent is using the strategy the other can do no better than use it himself…
- This is of particular interest to the designer of automated agents. It does away with any need for secrecy on the part of the programmer. An agent's strategy can be publicly known, and no other agent designer can exploit the information by choosing a different strategy. In fact, it is desirable that the strategy be known, to avoid inadvertent conflicts.

A one-shot Negotiation Protocol

- Protocol: both agents suggest an agreement; the one giving a higher product of utilities wins (flip a coin in case of a tie)
- Obvious strategy: amongst the set of agreements with maximal product of utilities, propose the one that is best for you
- Properties: This mechanism is:
  – efficient: outcomes have maximal Nash product and are
- Pareto optimal (like MCP with Zeuthen Strategy)
  – stable: no agent has an incentive to deviate from the strategy (like MCP with extended Zeuthen Strategy)
- In addition, the one-shot protocol is also:
  – simple: only one round is required
- But why should anyone accept to use such a protocol? (Like dividing a candy bar – one divides and the other picks. There is no motivation to be less than fair.)

Recap: How did we get to this point?

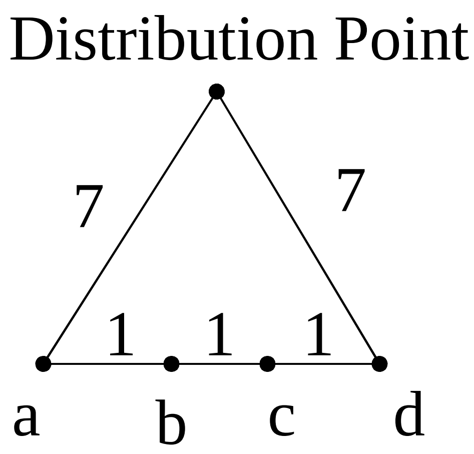
- Both agents making several small concessions until an agreement is reached is the most intuitive approach to one-to-one negotiation.

- The Monotonic Concession Protocol (MCP) is a straightforward formalization of the above intuition.

- The extended Zeuthen Strategy is also motivated by intuition ("willingness to risk conflict") and constitutes a stable and (almost) efficient strategy for the MCP.

- The one-shot protocol (together with the obvious strategy) produces similar outcomes as MCP/Zeuthen, but it is a much simpler mechanism.

Parcel Delivery Domain: Example 2 (don't return to dist point)



Conflict Deal:
({a,b,c,d}, {a,b,c,d})

All choices are IR, as can't do worse
({ac}{bd}) is dominated by ({a}{bcd})

Negotiation Set:
({a,b,c,d}, ϕ)
({a,b,c), {d})
({a,b}, {c,d})
({a}, {b,c,d})
(ϕ, {a,b,c,d})

Parcel Delivery Domain: Example 2 (Zeuthen works here both concede on equal risk)

Pure Deal	Agent 1's Utility	Agent 2's Utility
({a,b,c,d}, ϕ)	0	10
({a,b,c), {d})	1	3
({a,b}, {c,d})	2	2
({a}, {b,c,d})	3	1
(ϕ, {a,b,c,d})	10	0
Conflict deal	0	0

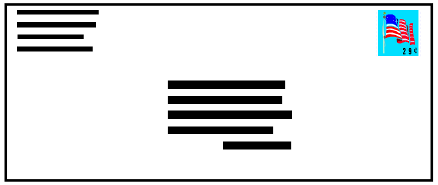
What bothers you about the previous agreement?

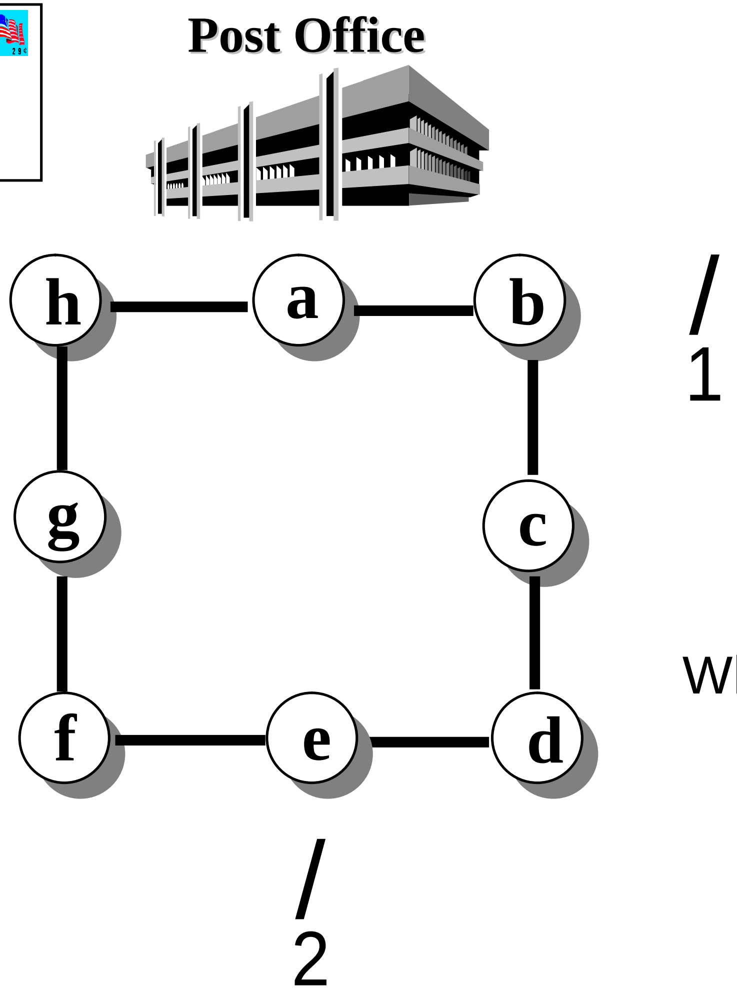
- Decide to both get (2,2) utility. This is not globally efficient.
- Is there a solution?
- Fair versus higher global utility.
- Restrictions of this method (no promises for future or sharing of utility)
- Efficiency can be increased with “Mixed Deals” (see Book for details)

Deception in TODs

- Deception can benefit agents in two ways:
  - *Phantom and Decoy tasks*
    Pretending that you have been allocated tasks you have not
  - *Hidden tasks*
    Pretending *not* to have been allocated tasks that you have been

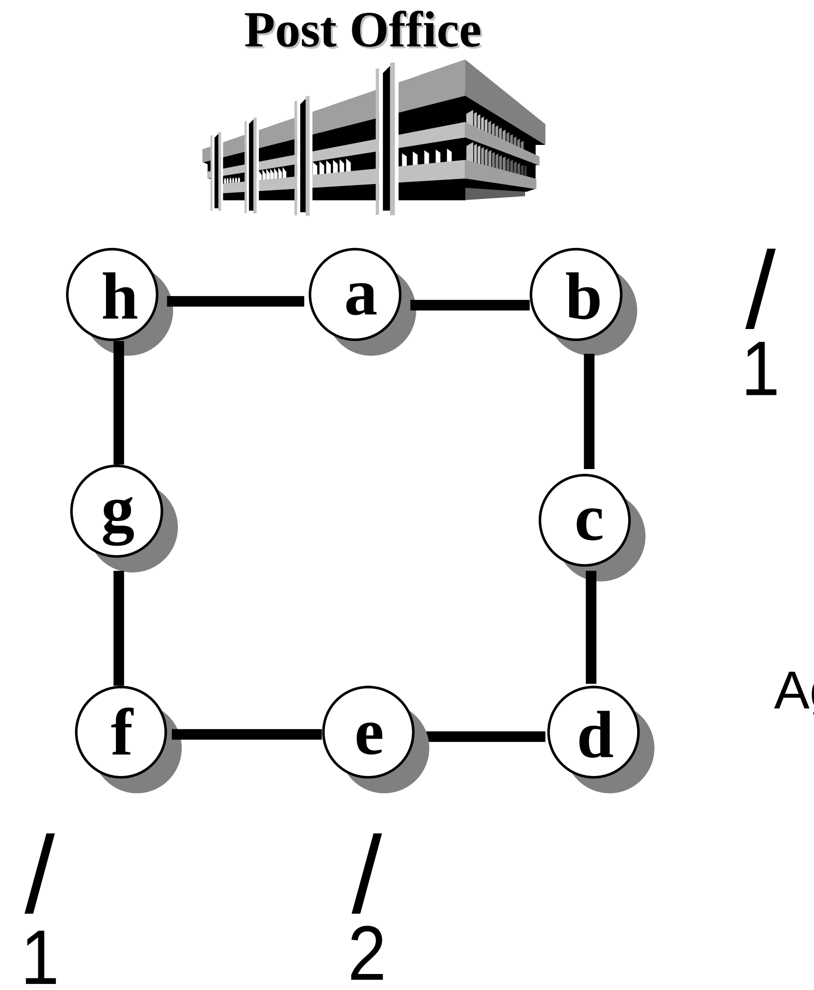
Negotiation with Incomplete Information

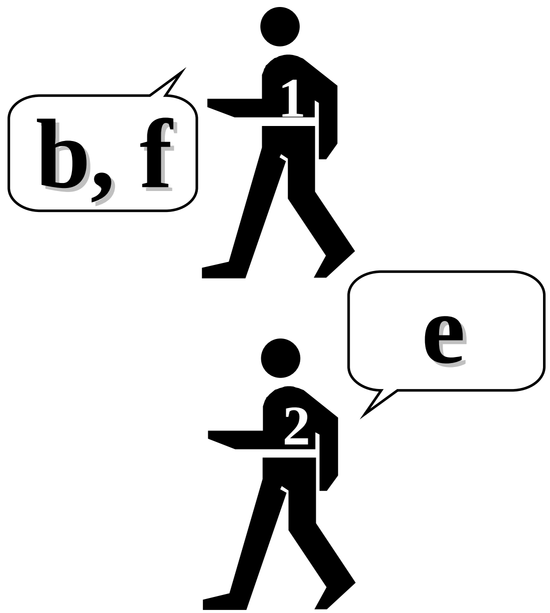




What if the agents don’t know each other’s letters?

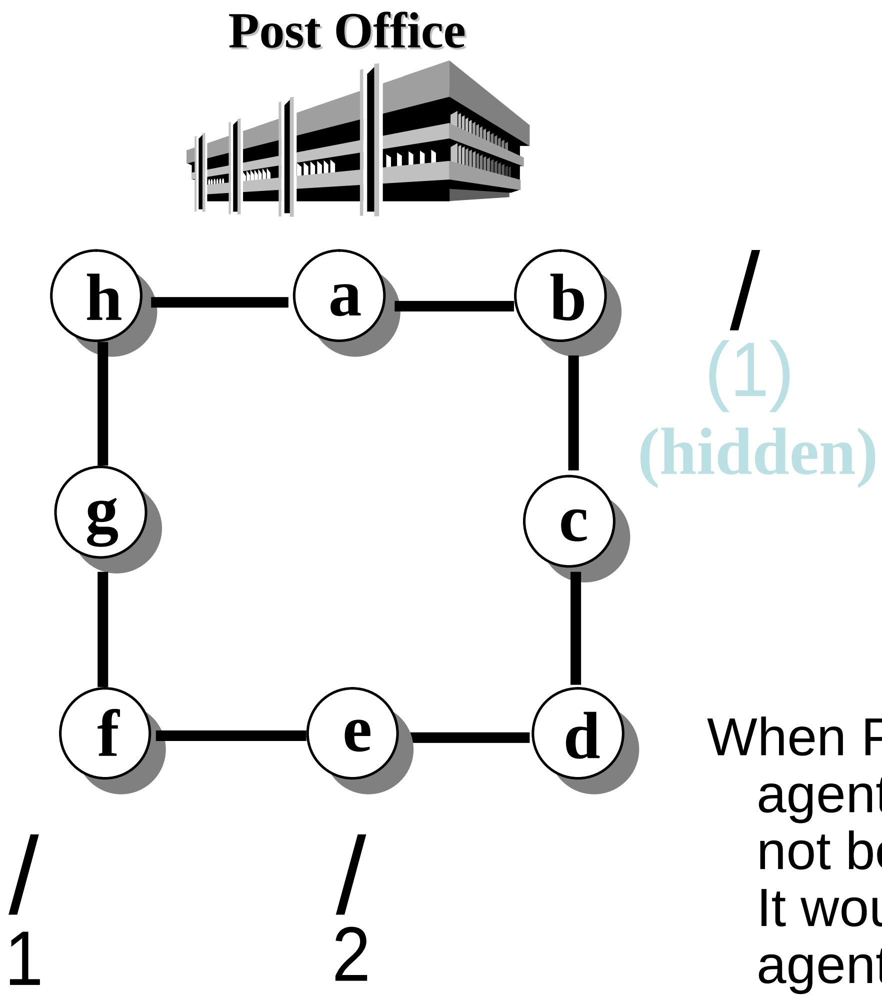
1 Phase Game: Broadcast Tasks

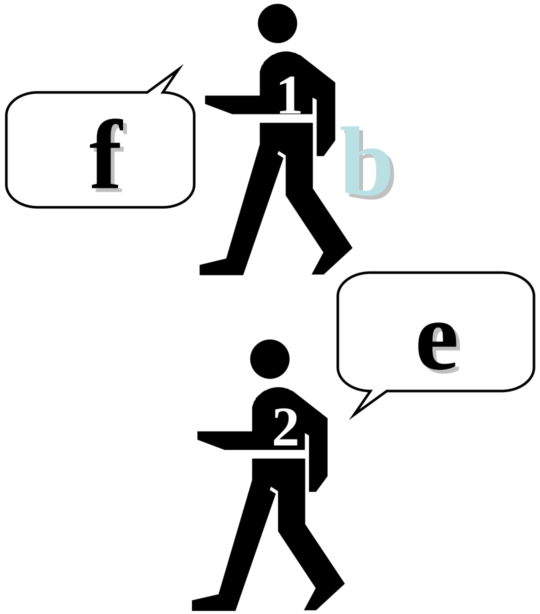




Agents will flip a coin to decide who delivers all the letters, because the standalone cost of each agent is 8.
The expected utility for both is 4.

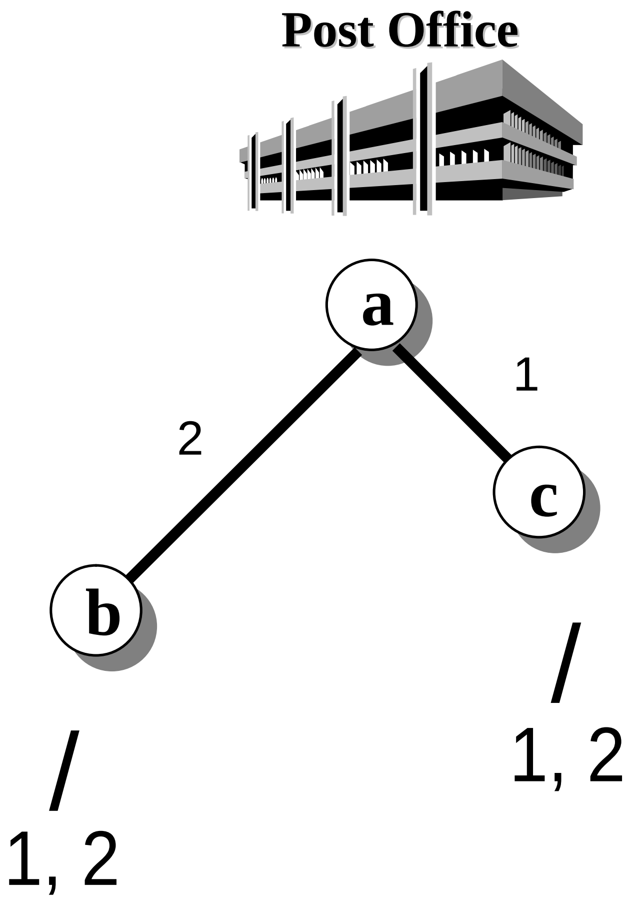
Hiding Letters

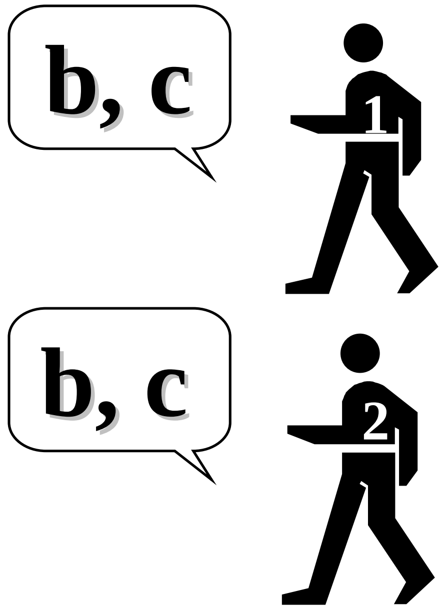




When PMM is used they agree that agent 2 delivers to f and e. It would not be rational for agent 1 to visit e. It would not be pareto optimal if agent 2 does not deliver the letter of agent 1.

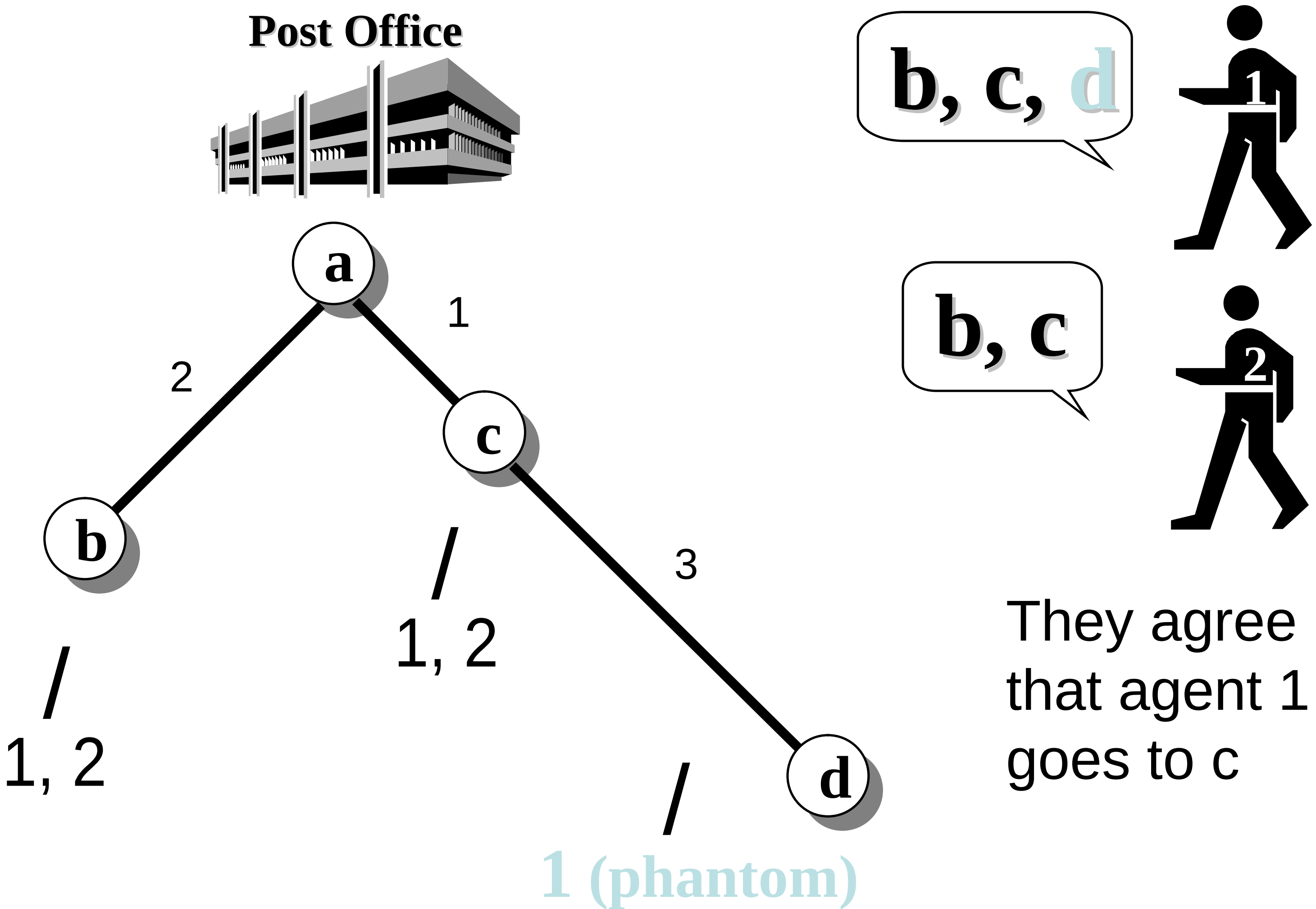
Another Possibility for Deception





They will agree to flip a coin to decide who goes to b and who goes to c

Phantom Letter

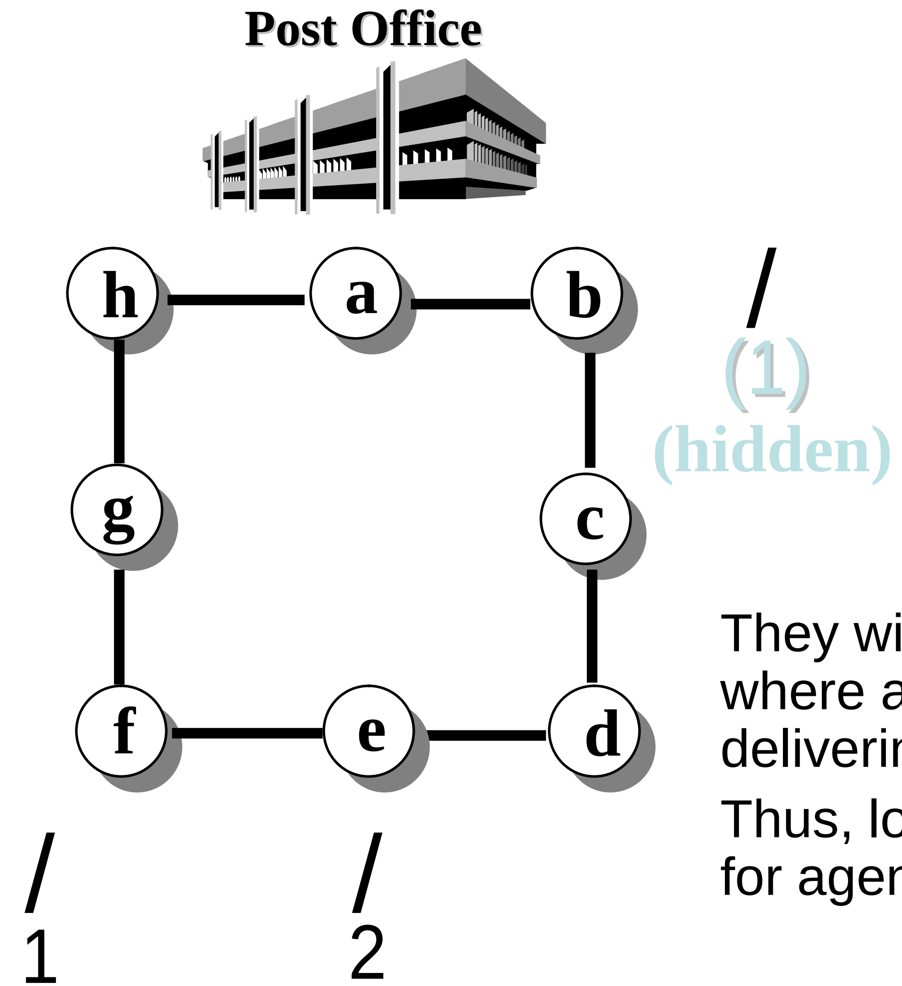


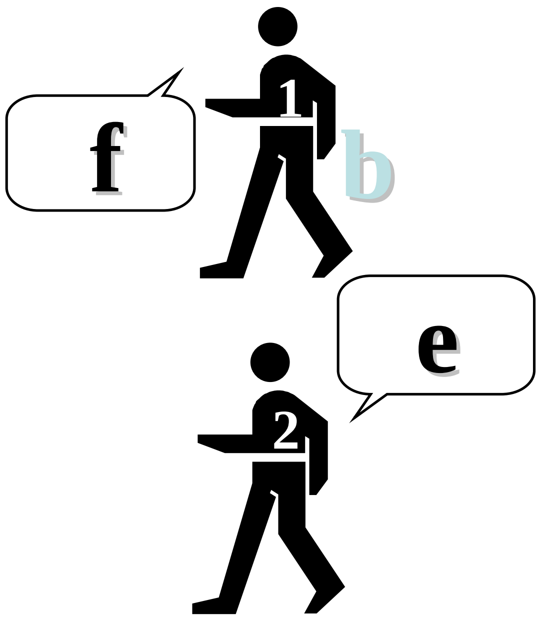
They agree that agent 1 goes to c

Negotiation over Mixed Deals

- Mixed deals transforms the discrete space into a continuous space of possible deals.
- Mixed deal $<D_1, D_2>$: p
  - The agents will perform $<D_1, D_2>$ with probability p, and the symmetric deal $<D_2, D_1>$ with probability 1 – p
- The definition of deal domination, individual rational, pareto optimal, and negotiation set are the same as for pure deals.
- *Theorem*: With mixed deals, agents can always agree on the "all-or-nothing" deal – where *D*1 is *T*1 $\cup$*T*2 and *D*2 is the empty set

Hiding Letters with Mixed All-or-Nothing Deals





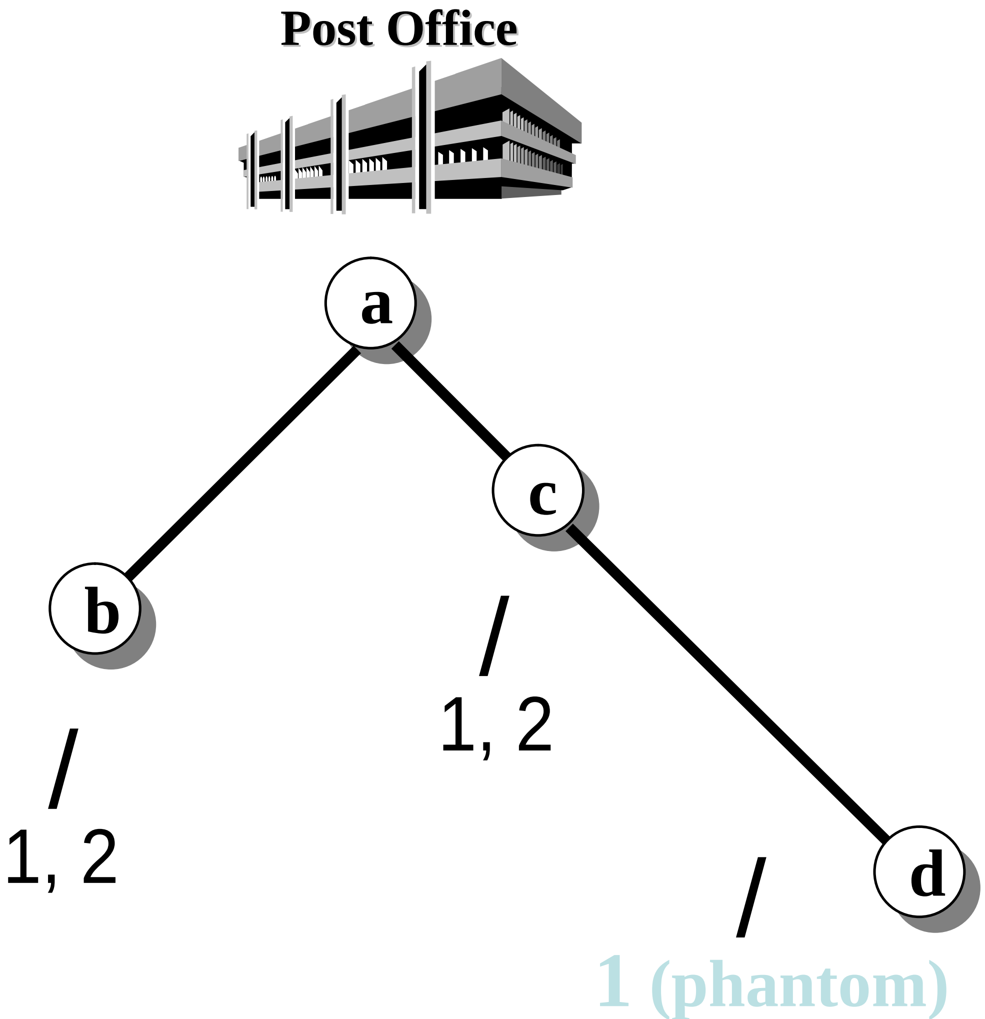
They will agree on the mixed deal where agent 1 has a 3/8 chance of delivering to f and e

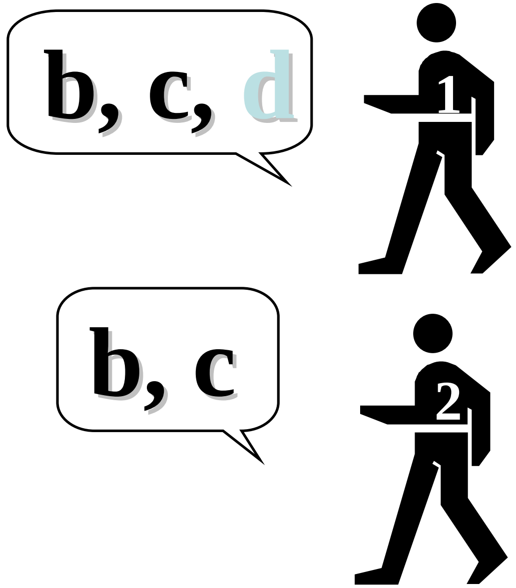
Thus, lowering the expected utility for agent 1 from 4 (no lying) to $3^3/_4$

Computation

- Not lying:
  - $8*p + (1-p)*0 = 0*p + (1-p)*8 \Rightarrow p=1/2$
  - The cost for 1 is $\frac{1}{2}*8+\frac{1}{2}*0=4$
    the expected utility is $8 – 4 = 4$
- Lying:
  - p can be found by
    $(-2p) + (1-p)\,6 = 8p + (1-p)\,0$
    $\Rightarrow p=3/8$
  - The cost for 1 is ${}^{3}/_{8}*8 + {}^{5}/_{8}*2=4^{1}/_{4}$
    the expected utility is then $8-4^{1}/_{4}= 3^{3}/_{4}$

Phantom Letters with Mixed Deals





- They will agree on the mixed deal where 1 has 3/4 chance of delivering all letters
- Because there is a chance that the phantom is detected introducing penalties will eliminate them.

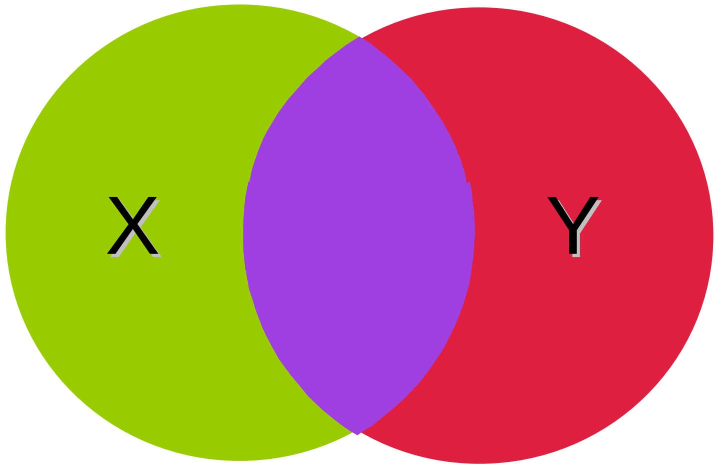
Sub-Additive TODs

All TODs seem to have a great deal in common.
But they differ significantly with respect to their cost functions.
We will categorize the TODs based on the properties:
subadditivity, *concavity* and *modularity*.

TOD $< T, Ag, c >$ is *sub-additive* if for all finite sets of tasks X, Y in T we have:

$$c(X \cup Y) \leq c(X) + c(Y)$$

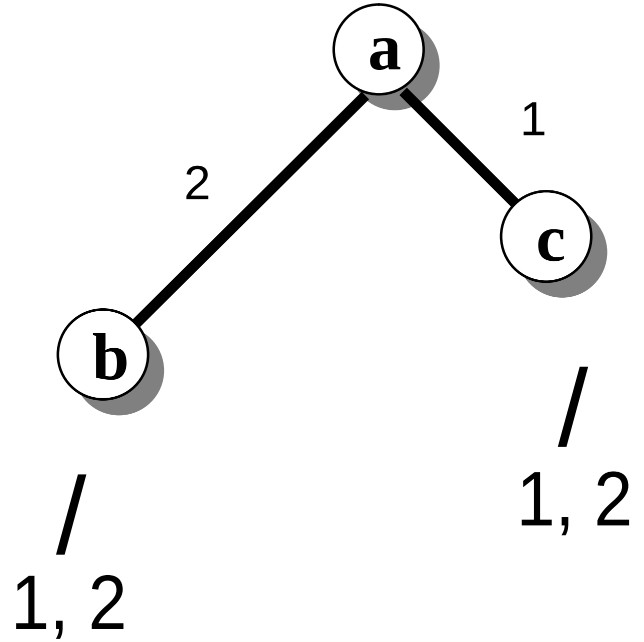
Sub-Additivity

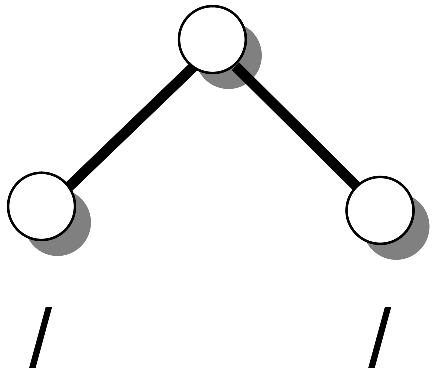


$$c(\boldsymbol{X} \cup \boldsymbol{Y}) \leq c(\boldsymbol{X}) + c(\boldsymbol{Y})$$

Sub-Additive TODs

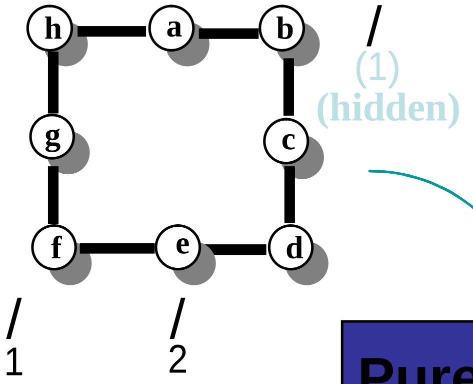
The Postmen Domain is sub-additive.

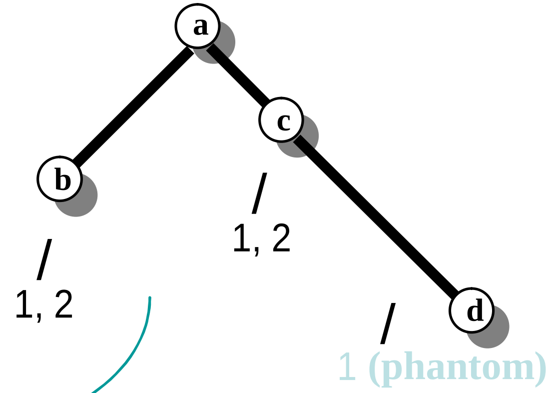




The “Delivery Domain” (where postmen don’t have to return to the Post Office) is not sub-additive

Incentive Compatible Mechanisms





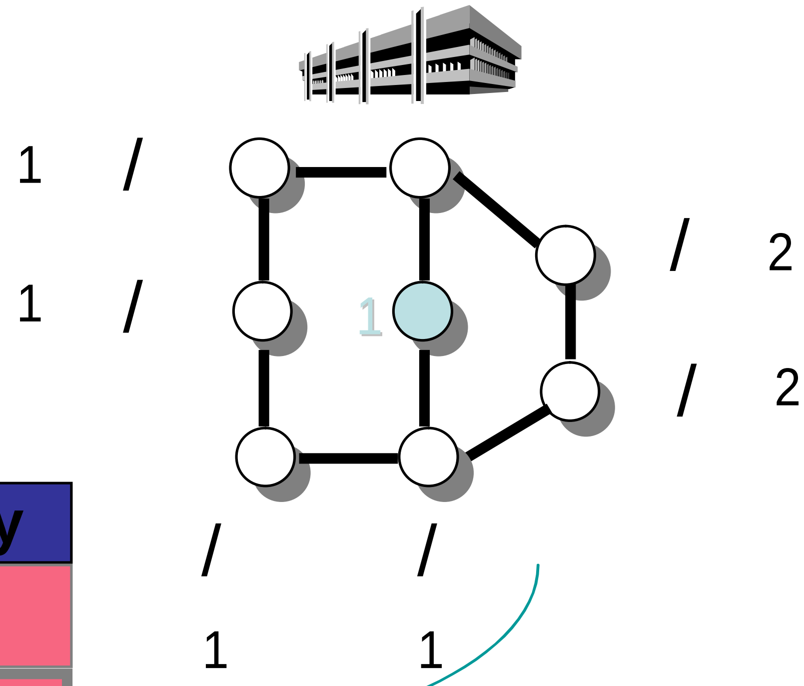
Sub-Additive

	Hidden	Phantom
Pure	L	L
A/N	T	T/P
Mix	L	T/P

Theorem: For all encounters in all sub-additive TODs, when using a PMM over all-or-nothing deals, no agent has an incentive to hide a task.

Decoy Tasks

Decoy tasks, however, can be beneficial even with all-or-nothing deals



Sub-Additive

	Hidden	Phantom	Decoy
Pure	L	L	L
A/N	T	T/P	L
Mix	L	T/P	L

Decoy lies are simply phantom lies where the agent is able to manufacture the task (if necessary) to avoid discovery of the lie by the other agent.

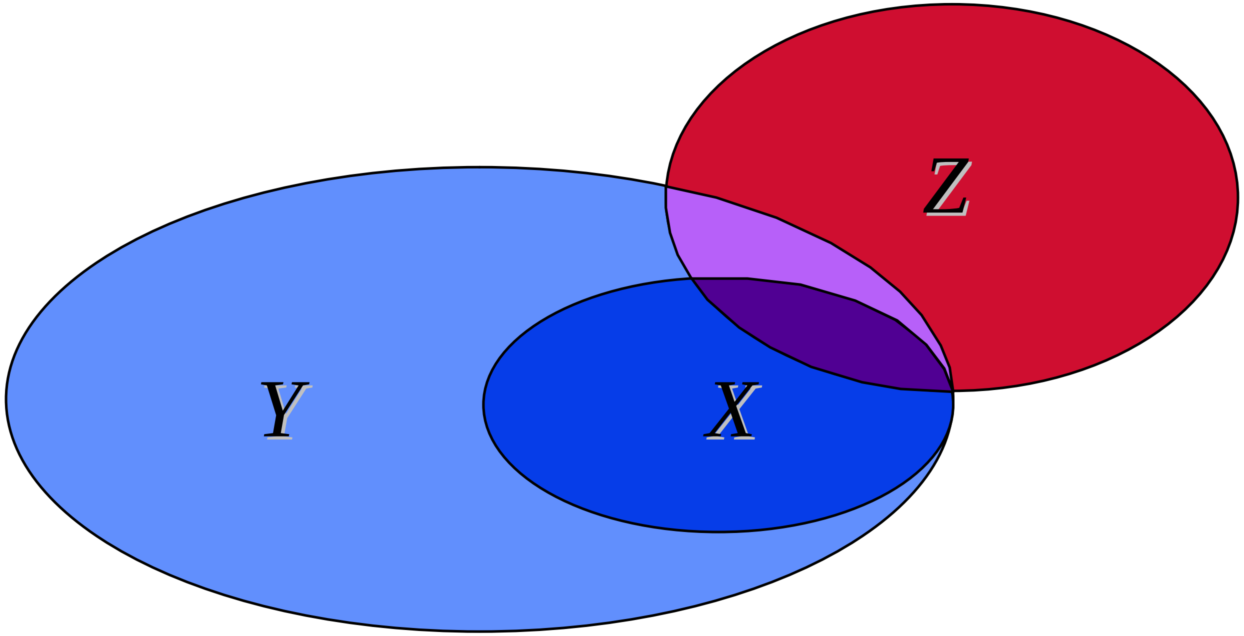
Concave TODs

TOD < T, Ag, c > is *concave* if for all finite sets of tasks Y and Z in T , and $X \subseteq Y$, we have:

$$c(Y \cup Z) - c(Y) \leq c(X \cup Z) - c(X)$$

Concavity implies sub-additivity

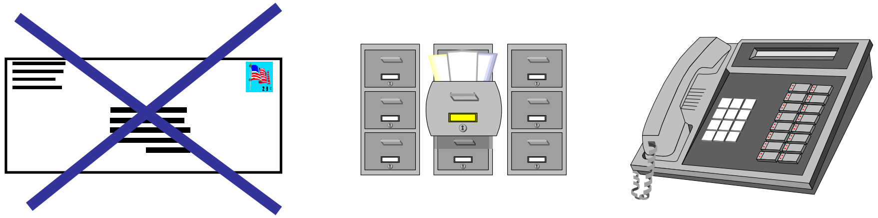
Concavity

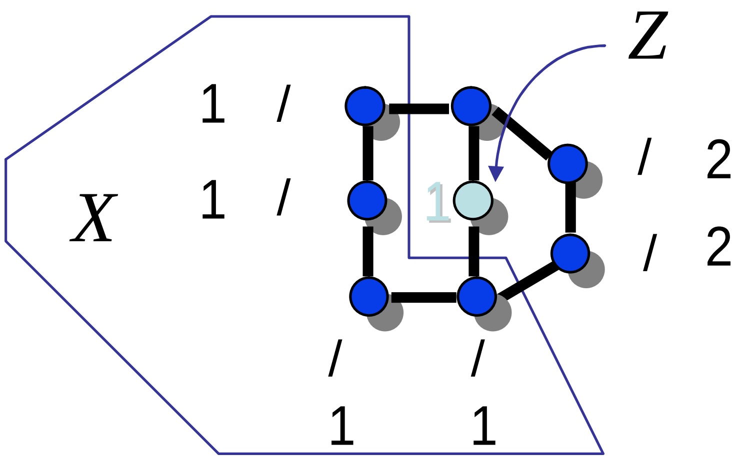


The cost Z adds to X is more than the cost it adds to Y.
(Z - X is a superset of Z - Y)

Concave TODs

Postmen Domain is not Concave, unless restricted to trees.





This example was not concave; Z adds 0 to X, but adds 2 to its superset Y (all blue nodes)

Three-Dimensional Incentive Compatible Mechanism Table

Theorem: For all encounters in all concave TODs, when using a PMM over all-or-nothing deals, no agent has any incentive to lie.

Concave

	Hidden	Phantom	Decoy
Pure	L	L	L
A/N	T	T	T
Mix	L	T	T

Sub-Additive

	Hidden	Phantom	Decoy
Pure	L	L	L
A/N	T	T/P	L
Mix	L	T/P	L

Theorem: For any encounter in all concave TODs, when using a PMM over mixed deals, every decoy lie is not beneficial.

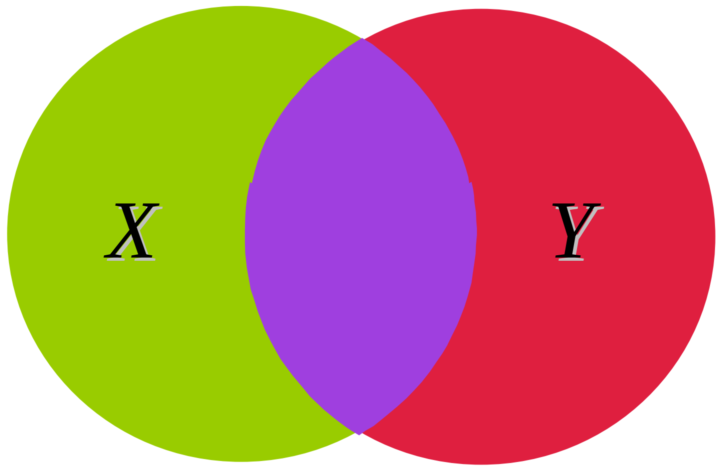
Modular TODs

TOD $< T, Ag, c >$ is *modular* if for all finite sets of tasks X, Y in T we have:

$$c(X \cup Y) = c(X) + c(Y) - c(X \cap Y)$$

Modularity implies concavity

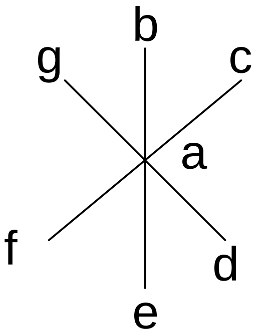
Modularity

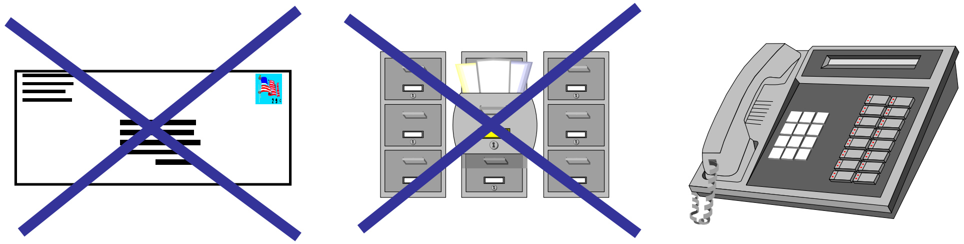


$$c(X \cup Y) = c(X) + c(Y) - c(X \cap Y)$$

Modular TODs

The Fax Domain is modular (not the Database Domain nor the Postmen Domain, unless restricted to a star topology or fully connected homogenous graphs).





$C(\{b,c\} \cup \{c,f\}) = 6$

$C(\{b,c\}) + C(\{c,f\}) – C(\{c\}) = 6$

Even in modular TODs, hiding tasks can be beneficial in general mixed deals

Three-Dimensional Incentive Compatible Mechanism Table

Sub-Additive

	H	P	D
Pure	L	L	L
A/N	T	T/P	L
Mix	*L	T/P	*L

Concave

	H	P	D
Pure	L	L*	L
A/N	T	T	* T
Mix	L	T	T

Modular

	H	P	D
Pure	L*	T	T*
A/N	T	T	T
Mix	L	T	T

* Müssen gezeigt werden, der Rest ergibt sich.

State oriented domain is a bit more powerful than TOD

- Joint plan is used to mean “what they both do” not “what they do together” – just the joining of plans. There is no joint goal!
- The actions taken by agent k in the joint plan are called k’s role and is written as J_k
- $C(J)_k$ is the cost of k’s role in joint plan J.
- In TOD, you cannot do another’s task as a side effect of doing yours or get in their way.
- In TOD, coordinated plans are never worse, as you can just do your original task. With SOD, you may get in each other’s way
- Don’t accept partially completed plans.

Examples

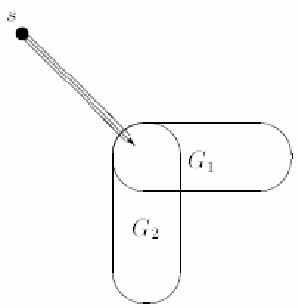
- Delivery Domain where with bounded storage at warehouse.
- Restricted Usage Shared Resource Domain
 Use cost is 0, Wait cost is 1, NOOP cost is 0

		Joint Plan			World States		
		A_1	A_2	A_3	A_1	A_2	A_3
T	0	Use	Use	Wait	(Use,0)	(Use,0)	(Wait,0)
i	1	Use	Use	Wait	(Use,1)	(Use,1)	(Wait,0)
m	2	NOP	Use	Use	(NOP,2)	(Use,2)	(Use,0)
e	3	NOP	NOP	Use	(NOP,2)	(NOP,3)	(Use,1)
	4	NOP	NOP	NOP	(NOP,2)	(NOP,3)	(NOP,2)

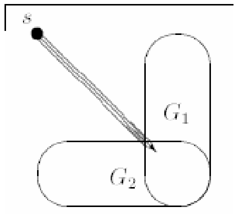
but not individual rational.

Interaction Types

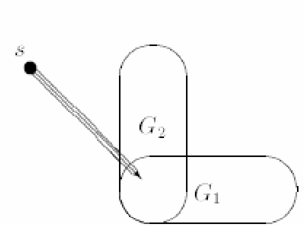
- *Symmetric cooperative*
  There is a joint plan in the negotiation set that is preferred over achieving the goals alone



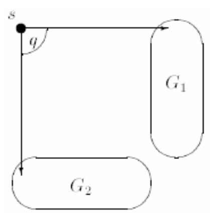
- *Symmetric compromise*
  There are individual rational deals, but both would prefer to be alone.

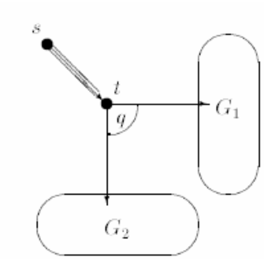


- *Non-Symmetric cooperative/compromise*
  One views the interaction as cooperative, while the other views the interaction as a compromise.



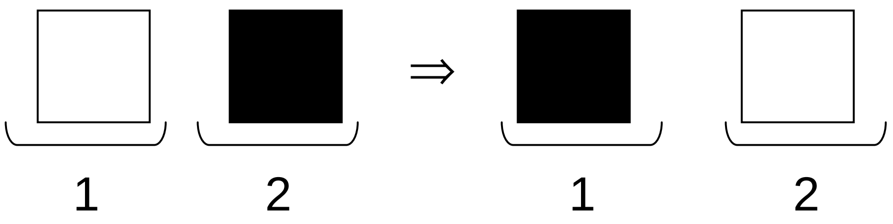
- *Conflict*
  Negotiation set is empty. Semi-Cooperative cooperation is possible.





Examples: Cooperative
Each is helped by joint plan

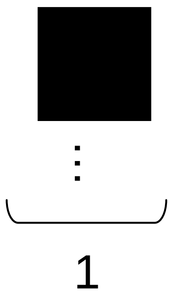
- Slotted blocks world:
  initially white block is at 1 and black block at 2. Agent 1 wants black in 1. Agent 2 wants white in 2. (Both goals are compatible.)
- Assume pick up is cost 1 and set down is one.
- Mutually beneficial – each can pick up at the same time, costing each 2 – Win – as didn't have to move other block out of the way!
- If done by one, cost would be 4 – so utility to each is 2.



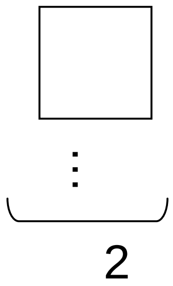
Examples: Compromise
Both can succeed, but worse for both than if other agent weren't there.

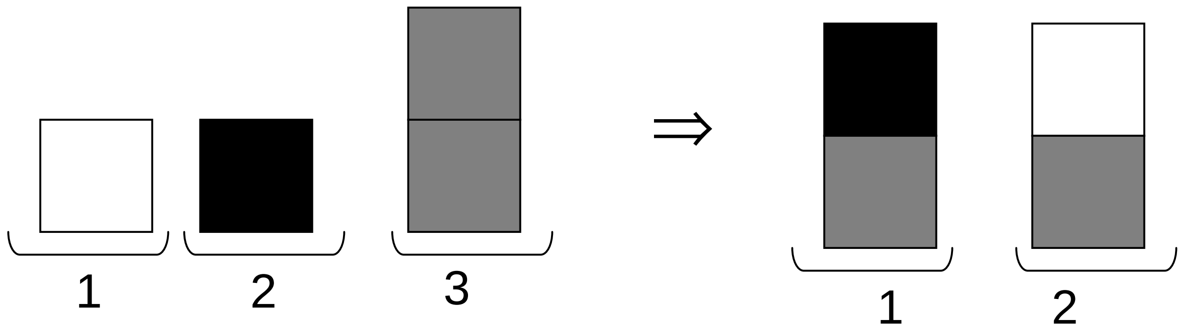
- Slotted blocks world: initially white block is at 1 and black block at 2, two gray blocks at 3. Agent 1 wants black in 1, but not on table. Agent 2 wants white in 2, but not directly on table.
- Alone, agent 1 could just pick up black and place on white. Similarly, for agent 2. But would undo others goal.
- But together, all blocks must be picked up and put down. Best plan: one agent picks up black, while other agent rearranges (cost 6 for one, 2 for other)
- Can both be happy, but unequal roles.

Agent 1 goal:



Agent 2 goal:





Compromise, continued

- Who should get to do the easier role?
- **If you value it more, shouldn't you do more of the work to achieve a common goal?**
- Look at worth. If A_1 assigns worth (utility) of 3 and A_2 assigns worth (utility) of 6 to final goal, we could use probability to make it "fair".
- Assign ({2}{6}) p of the time.
- Utility for agent 1= p(1) + (1-p)(-3)   // loses utility if takes 6 for benefit 3
- Utility for agent 2 = p(0) + (1-p)4
- Solving for p by setting utilities equal
- 4p-3 = 4-4p
- p = 7/8
- Thus, I can take an unfair division and make it fair!

Example: conflict

- I want black on white (in slot 1)
- You want white on black (in slot 1)
- Can't both win. Could flip a coin to decide who wins. Better than both losing. Weightings on coin needn't be 50-50.
- May make sense to have person with highest worth get his way – as utility is greater. (Would accomplish his goal alone) Efficient but not fair?
- What if we could transfer half of the gained utility to the other agent? This is not normally allowed, but could work out well.

Example:semi-cooperative

- Both agents want contents of slots 1 and 2 swapped (and it is more efficient to cooperate).
- Both have (possibly) conflicting goals for other slots
- To accomplish one Agent's goal by oneself is 26:
  8 for each swap and 10 for rest (pulling numbers out of the air)
- Cooperative swap is 4 (pulling numbers out of air).
- Idea, work together to swap, and then flip coin to see who gets his way for rest.

Example: semi-cooperative, cont

- Winning agent: utility: 26-4-10 = 12
- Losing agent: utility: -4 (as helped with swap)
- So with ½ probability: 12*1/2 -4*1/2 = 4
- Note, kept just the joint part of the plan that was more efficient, and gambled on the rest (to remove the need to satisfy the other)

Further …

- Need for analyzing "lying" situations – worth and goals
- Extending the protocol to handle "semi-cooperative" deals.
- Extending the possible deal set by "multi-plan" deals. Pref-flip vs. Post-flip (semi-cooperative)

  "*Although we are able to construct an incentive compatible mechanism to be used when worths are unknown, we are unable to construct such a mechanism in State Oriented Domains to be used when the other's goals are unknown*".
- What about worth oriented domains (with goal relaxation) (Delivery Domain with bounded storage).