Negotiations

• What are negotiations:
  – Negotiation is a process of communication whereby two or more parties, each with its own viewpoint and objectives, attempt to reach a mutually satisfactory result on a matter of common concern

• Why are mutually satisfactory results needed:
  – Otherwise one party at least does not take part to the negotiation

• What is the peculiarity of a negotiation:
  – Parties’ viewpoint and objectives are in conflict

• What is the object of a negotiation:
  – Essentially, the price of goods, services, etc.
**Negotiation Models**

- A negotiation is essentially a **strategic interaction situation** and is modelled as a **strategic game** [Kraus, 2000]
  - **Negotiation protocol** sets the rules of the dispute
    - Actions available to the agents (e.g., make an offer, accept, etc.)
    - Sequence of the interaction (e.g., agents act concurrently or in alternating fashion)
  - **Agents’ strategies**: define the behaviour of each agent
    - Actions to be employed by each agent at each single decision node
- Furthermore, as is in a game:
  - **Agents’ preferences**: each agent has preferences over all the possible negotiation outcomes
  - **Agents’ knowledge**: agents’ preferences can be known by the others or can be uncertain or can be unknown
  - **Agents’ rationality**: each agent act in order to maximize its expected payoff relying on its knowledge

**A Simple Protocol Example**

- **Agents**:
  - One seller
  - Many buyers
- **Allowed actions**:
  - Seller: “open”, “close”
  - Buyer: “offer”
- **Payoffs**: utility functions, e.g., $U = RP - price$
- **Interaction sequence**: any buyer can act at $t$
- **Information**: private other agents’ preferences ($RP$)

**Protocol Characteristics [Kraus, 2000]**

- **Distribution**: the decision making process should be distributed
- **Negotiation time**: negotiations that end without delay are preferable to negotiations that are time-consuming
- **Efficiency**: the efficiency of the agreement increases the number of agents that will be satisfied by the negotiation result
- **Simplicity**: negotiation processes that are simple and efficient are preferable to complex processes
- **Stability**: a set of negotiation strategies is stable if, given that all the other agents are following their strategies, it is beneficial to an agent to follow its strategy too; protocols with stable strategies are preferable
- **Money transfer**: side payments can be required from or provided to agents to resolve the conflicts; protocols without money transfer are preferable

**Protocol Classification**

- **Number of attributes**:
  - One (e.g., price)
  - Many (e.g., price and response time)
- **Number of agents**:
  - One-to-one (e.g., bilateral bargaining)
  - One-to-many (e.g., multilateral bargaining and auctions)
  - Many-to-many (e.g., auctions)
- **Number of units**:
  - One
  - Many
Automatic Negotiations

- **What are automatic negotiations:**
  - Electronic negotiations in which intelligent self-interested software agents negotiate with other agents on behalf of users for buying or selling services and goods [Sandholm, 2000]

- **Why do we need to develop automatic negotiations:**
  - Increasing efficiency by saving resources
    - **Human work:** the agents act on behalf of the man
    - **Time:** the agents are faster than man
    - **Money:** market competition is higher

- **What are the application domains:**
  - eCommerce (electronic markets)
  - Resource allocation

---

Involved Areas

**Economics**
- Microeconomics
- Game Theory
- Econometrics

**Computer Science**
- Software Engineering
- Data Base
- Security

**Law**
- Social Right
- Legal Aspects

**Artificial Intelligence**
- Multiagent Systems
- Decision Theory

**Procedures**

**Media and autonomy**

**Part 2**

Bilateral Bargaining
The Bargaining Problem [Nash, 1950]

- Bargaining is a socioeconomic problem involving **two parties**, who can **cooperate** towards the creation of a commonly desirable **surplus**, over whose **distribution** both parties are in **conflict**
- **Example**: two agents divide a pie
  - Each player prefers to reach an agreement, rather than abstain from doing so (disagreement)
  - Each agent prefers that agreement which most favors her interests (the largest piece of pie)

Bargaining in Economic Domains

- **Bilateral exchange situation**:
  - A buyer that wants to buy an item
  - A seller that wants to sell an item
  - They negotiate over the price \( p \)
- **Agents' utility function**:
  - Buyer agent: \( U_b(p) = RP_b - p \)
    \( U_b(\text{Disagreement}) = 0 \)
  - Seller agent: \( U_s(p) = p - RP_s \)
    \( U_s(\text{Disagreement}) = 0 \)
  - The surplus to be divided is: \( RP_b - RP_s \)
- **The bargaining problem**:
  - What is the optimal price?

Cooperative vs Non-Cooperative Bargaining Models

- **Cooperative approaches**:
  - Cooperative solutions attempt a prediction of what agreement two agents can be expected to reach in an **unspecified negotiation process**
  - They state **assumptions** on the **expected agreement** and find the agreement that satisfies the assumptions
  - **Examples**: Nash Bargaining solution [Nash, 1950], Kalai-Smorodinsky solution, Kalai solution, egalitarian solution, utilitarian solution
- **Non-cooperative approaches**:
  - Non-cooperative models consider bargaining as a **fully specified game**
  - **Example**: Rubinstein's alternating-offers protocol [Rubinstein, 1982]

Nash Bargaining Solution (1)

- **Nash's axioms**
  - **Individual rationality** (IR): the optimal agreement \( a \) must be such that \( U_i(a) \geq 0 \) and \( U_j(a) \geq 0 \)
  - **Pare efficiency** (PAR): the optimal agreement \( a \) must be Pareto efficient for the agents
  - **Invariance to equivalent utility representations** (INV): it satisfies affine transformations
  - **Independence of irrelevant alternatives** (IIA): removed all the non-optimal agreements, the optimal agreement holds to be
  - **Symmetry** (SYM): if the agents have the same preferences, then the agreement \( a \) must gives the same utilities to them
Nash Bargaining Solution (2)

\[ NBS = \arg \max_a \{ U_b(a) \cdot U_s(a) \} \]

(It is the tangency point between the Pareto frontier and a hyperbola of the form \( U_b \cdot U_s = \text{constant} \))

Alternating-Offers Protocol [Rubinstein, 1982]

- **The informal model**
  - Two agents want to divide a pie of size 1
  - Opposite preferences with temporal discounting factors
  - Extensive form game wherein agents alternately act
  - Infinite horizon

- **The formal model**
  - **Players**
    \[ \{1, 2\} \]
  - **Player function**
    \[ a(0) = t \]
    \[ a(t) \neq a(t - 1) \]
  - **Actions**
    \[ \text{offer}(x) \]
    \[ \text{accept} \]
  - **Preferences**
    \[ U_b(x, t) = (1 - x) \cdot \delta^t \]
    \[ U_s(x, t) = x \cdot \delta^t \]

Equilibrium in Alternating-Offers Protocol

- **Subgame Perfect Equilibrium** [Selten, 1972]
  - It defines the equilibrium strategies of each agent in each possible subgame
  - Typically, addressed by employing backward induction, but not in this case since the horizon is infinite

- **Rubinstein Solution** [Rubinstein, 1982]
  \[
  \sigma_b(t) = \begin{cases} 
  \text{accept} & \text{if } a(t) = \text{offer}(x) \text{ with } x \leq \frac{1 - \delta}{1 - \delta - \delta} \\
  \text{offer} & \text{if } a(t) = \text{offer}(x) \text{ with } x > \frac{1 - \delta}{1 - \delta - \delta} 
  \end{cases}
  \]

A Graphical View

- \( NBS \) is the tangency point between the Pareto frontier and a hyperbola of the form \( U_b \cdot U_s = \text{constant} \)

- \( U_s^* \) is the preference function for player 1.

- The dotted line represents the Pareto frontier.

- The vertical lines represent the time horizon.

- The points on the graph represent the offers made by each player in each time step.
Protocol Enrichments in Computer Science

- **Agents’ preferences:**
  - **Multiplicity of Issues**
    - The evaluation of each item takes into account several attributes $x'$
    - Each offer is defined on all the attributes of the item, being a tuple $x = < x_1, \ldots, x_m >$
  - **Reservation Values ($RV_j$)**
    - $RV_{bi}$: the maximum value of attribute $j$ at which the agent $b$ will buy the item
    - $RV_{sj}$: the minimum value of attribute $j$ at which the agent $s$ will sell the item
  - **Deadlines ($T_i$):** The time after which agent $i$ has not convenience to negotiate any more

- **Agents’ actions:**
  - **Exit Option:** Agent can make exit at any time point it plays

Revised Alternating-Offers Protocol

- **Players**
  - $b$ (buyer)
  - $s$ (seller)

- **Player function**
  - $j(0) = i$
  - $j(t) = s(t-1)$

- **Actions**
  - offer:
  - accept
  - exit

- **Preferences**
  - $U_i(NoAgreement) = U_s(NoAgreement) = 0$
  - $U_i(x,s) = \sum_{j} (RV_{bi} - x_j) \cdot (y_j)^j$ for $t \leq T_i$
  - $-1$ for $t > T_i$

Solution with One Issue and Complete Information

- **By backward induction**
  - The game is not rigorously a finite horizon game
  - However, no rational agent will play after its deadline
  - Therefore, there is a time point from which we can build backward
  - We call it the **deadline of the bargaining**, i.e. $T_{in} = \min(T_b, T_s)$
  - The agents’ optimal offers are function of time $t$, we call $x^*(t)$
  - $x^*(t)$ is such that $x^*(T-1)=RV_{bi}$ and $U_{str}(x^*(t),t)=U_{str}(x^*(t+1),t+1)$

![A Graphical View](image)

- **Infinite Horizon Construction**
- **Finite Horizon Construction**
Significant Results in Literature (1)

- **Multi-issue bargaining:**
  - With complete information the problem of bargaining with multiple issues can be cast in the problem of bargaining one issue in time polynomial in the number of issues [Di Giunta et al., 2006], [Fatima et al., 2006]

- **Bargaining with uncertainty:**
  - In presence of uncertainty the bargaining game is an imperfect information extensive-form game and the appropriate solution concept is the sequential equilibrium of Kreps and Wilson
  - Examples of bargaining with uncertain information are [Gatti et al., 2008a], [Rubinstein, 1985], [Sandholm et al., 1999]

- **Bargaining in markets:**
  - Within markets, buyers are in competition over the purchase of an item and sellers over the sale of an item
  - Refinements of the bargaining protocol are considered to capture this competition [Serrano, 2008], [Gatti et al., 2008b]

- **Learning in bargaining:**
  - Learning is an interesting and promising technique to address negotiation, specially when agents are not perfectly rational
  - An example of the employment of learning techniques in bargaining is [Lazaric et al., 2007b]

Significant Results in Literature (2)

- **Bargaining with bounded rationality:**
  - Agents can follow predefined tactics, not searching for their optimal actions
  - Examples are [Binmore, 2007], [Faratin et al., 1998], [Fatima et al., 2002], [Fatima et al., 2004]

- **Evolutionary models of bargaining:**
  - Bargaining is studied as an evolutionary process by employing evolutionary game theory tools
  - Examples are [Binmore, 2007], [Napel, 2004]

Introduction to Auctions [Vidal, 2007]

- Auctions ask and answer the most fundamental questions in economics: **who should get the goods and at what prices?** [Cramton et al., 2006]
- Auctions provide the micro-foundation of markets
- Typically,
  - **An auctioneer:**
    - A seller who wants to sell goods
    - A buyer who wants to buy a good
  - **The bidders:**
    - Buyers who want to acquire goods
    - Sellers who want to sell their goods
- The agents are **self-interested and rational**: they play in the attempt to maximize their own payoffs
- The reservation prices are **private** information

Part 3
Auctions
Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bid</td>
<td>Bids are offered by bidders to buy or sell the auctioned item</td>
</tr>
<tr>
<td>Buy bid</td>
<td>The price that a bidder is willing to pay to own an item</td>
</tr>
<tr>
<td>Sell bid</td>
<td>The price that a bidder is willing to accept to sell an item</td>
</tr>
<tr>
<td>Reservation price</td>
<td>The maximum (minimum) price that a buyer (seller) is willing to pay (accept) for an item</td>
</tr>
<tr>
<td>Process bid</td>
<td>The auctioneer checks the validity of a bid according to the rules of the auction protocol</td>
</tr>
<tr>
<td>Price quote generation</td>
<td>The auction house via the auctioneer or by other means may provide information about the status of the bids</td>
</tr>
<tr>
<td>Bid quote</td>
<td>The amount a seller would have to offer to sell an item</td>
</tr>
<tr>
<td>Ask quote</td>
<td>The amount a buyer would have to offer to buy an item</td>
</tr>
<tr>
<td>Clearance</td>
<td>Through clearance buyers and sellers are matched and the transaction price is set</td>
</tr>
<tr>
<td>Clearing price</td>
<td>The final transaction price that the buyer pays and the seller receives</td>
</tr>
</tbody>
</table>

Classification of Auctions [Fasli, 2007] (1)

- **Three dimensions**: bidding rules, information revelation policy, and clearing policy

1. **Bidding rules**:
   - Single good or combinatorial
   - Single attribute or multi-attribute
   - Single or double
   - Open (outcry) or sealed-bid
   - Ascending or descending
   - Single unit or multi-unit

Classification of Auctions [Fasli, 2007] (2)

Classification of Auctions [Fasli, 2007] (3)

2. **Information revelation policy**:
   - When to reveal information: on each bid, at predetermined points in time, on inactivity, on market clears
   - What information:
     - **Bid**: the price a seller would have to offer in order to trade
     - **Ask**: the price a buyer would have to offer in order to trade
     - **Auction closure**: known, unknown, after a period of inactivity
   - To whom: participants only, everyone

3. **Clearing policy**:
   - When to clear: on each bid, on closure, periodically, after a period of inactivity
   - Who gets what: allocation and winner determination problem
   - At what prices: first, second price or other
Auctions and Mechanism Design

• Each auction is essentially a mechanism
  – A mechanism (from mechanism design) is an implementation of a social function
  – Given the preferences of all the participants and a social function, the mechanism chooses the winner
• Exactly as in mechanism design, the maximum efficiency is when agents are truth-revelling
  – Agents are truth-revelling when the mechanism is incentive-compatible
• The aim is the design of auction mechanism that be incentive-compatible

English Auction (1)

• Protocol (open-outcry ascending-price):
  – The auctioneer announces an opening or the reserve price
  – Bidders raise their bids and the auction proceeds to successively higher bids
  – The winner of the auction is the bidder of the highest bid
• Dominant strategy:
  – It is to bid a small amount above the previous high bid until one reaches its private value and then stop

English Auction (2)

Dutch Auction (1)

• Protocol (open-outcry descending-price):
  – The auctioneer announces a very high opening bid
  – The auctioneer keeps lowering the price until a bidder accepts it
  – The first bidder that accepts is the winner of the auction
• Dominant strategy:
  – No dominant strategy there is
  – Each agent acts on the basis of its prior
**English Auction (2)**

- Auctioneer
  - RP=5

  - Bidder 1
    - RP=10
  - Bidder 2
    - RP=8
  - Bidder 3
    - RP=9
  - Bidder 4
    - RP=7

  - accept

**Dutch Auction (3)**

- Properties:
  - The non-existence of the dominant strategy introduces inefficiencies in the solution
  - Real-time efficient: the auction closes really fast and the auctioneer can make it move even faster by lowering the price faster
  - It is used in The Netherlands for selling fresh flowers

**First-Price Sealed-Bid Auction (1)**

- Protocol (sealed-bid):
  - Each bidder submits its own bid without knowledge of the bids of the other bidders
  - The bids are opened and the winner is determined
  - The highest bidder wins and pays the amount it bids

- Dominant strategy:
  - No dominant strategy there is
  - Each agent acts on the basis of its prior

**First-Price Sealed-Bid Auction (2)**

- The winner is Bidder 3 and it pays 9

  - Bidder 1
    - RP=10
  - Bidder 2
    - RP=8
  - Bidder 3
    - RP=9
  - Bidder 4
    - RP=7
First-Price Sealed-Bid Auction (3)

- **Properties**:
  - The non-existence of the dominant strategy introduces inefficiencies in the solution

Second-Price Sealed-Bid Auction – Vickrey (1)

- **Protocol (sealed-bid)**:
  - Each bidder submits its own bid without knowledge of the bids of the other bidders
  - The bids are opened and the winner is determined
  - The highest bidder wins and pays the amount of the second-highest bid
- **Dominant strategy**:
  - The dominant strategy of an agent is to bid its reservation price

Second-Price Sealed-Bid Auction – Vickrey (2)

Auctioneer: RP=5

<table>
<thead>
<tr>
<th>Bidder</th>
<th>RP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bidder 1</td>
<td>10</td>
</tr>
<tr>
<td>Bidder 2</td>
<td>8</td>
</tr>
<tr>
<td>Bidder 3</td>
<td>9</td>
</tr>
<tr>
<td>Bidder 4</td>
<td>7</td>
</tr>
</tbody>
</table>

the winner is bidder 1 and it pays 9

Second-Price Sealed-Bid Auction – Vickrey (3)

- **Proof of truth-revealing** (it is similar to prove that a strategy is a Nash equilibrium):
  - Suppose that bidder \( b_i \) bids \( x < v \) where \( v \) is its true valuation
    - Suppose that the other highest bid is \( w > v \)
      - If \( x > w \), then \( b_i \) wins and pays \( w \), therefore \( b_i \) does not gain more by bidding \( x \) rather than \( v \)
      - If \( w > x \), then \( b_i \) looses and gains 0, therefore \( b_i \) gains lesser by bidding \( x \) rather than \( v \)
    - When the other highest bid is \( w > v \), \( b_i \) cannot gain more by bidding \( x \)
  - Suppose \( x > v \)
    - Suppose that the other highest bid is \( w < v \)
      - If \( x > w \), then \( b_i \) wins and pays \( w \), therefore \( b_i \) does not gain more by bidding \( x \) rather than \( v \)
      - If \( w > x \), then \( b_i \) looses and gains 0, therefore \( b_i \) gains lesser by bidding \( x \) rather than \( v \)
    - When the other highest bid is \( w > v \), \( b_i \) cannot gain more by bidding \( x \)
Auction Properties

- An auction is **incentive compatible** if truth-revelation is a dominant strategy for the agents.
- An auction is **individually rational** if its allocation does not make any agent worse off than had the agent not participated.
- An allocation of goods is **efficient** if there can be no more gains from trade.
  - No mechanism is individually rational, efficient and incentive compatible for both sellers and buyers.

Strategic Equivalence of Dutch and FPSB

- The “strategy space” is the same in the Dutch and FPSB auctions, hence they are said “strategically equivalent”.
- Since these auction mechanisms do not admit any dominant strategy, we resort to Bayes-Nash.
- We assume that agents be risk neutral and that their valuations are drawn uniformly from [0,1]
- We assume that the information is common.
- The **equilibrium strategy** of each bidder $b_i$ is to bid exactly $(N-1/N) \cdot v_{b_i}$ where $N$ is the number of bidders.

Revenue Equivalence Theorem

- **Theorem**: Assume that each of $n$ risk-neutral agents has a cumulative distribution $F(v)$ that is strictly increasing and atomless on [0,1]. Then any auction mechanism in which:
  - the good will be allocated to the agent with valuation 1, and
  - any agent with valuation 0 has an expected utility of 0,
yields the same expected revenue, and hence results in any bidder with valuation $v$ making the same expected payment.
- The theorem shows that in presence of a Bayesian prior all the auctions mechanism are equivalent for the auctioneer.

Auction Advantages and Drawbacks

- **Advantages**: Flexibility, as protocols can be tailor-made.
  - Less time-consuming and expensive than negotiating a price, e.g. in bargaining.
  - Simplicity in determining the market prices.
- **Drawbacks**: Collusion, Lying auctioneer.
Collusion (1)

- Bidders can collude and form an auction ring
- In order for rings to be successful, agreement has to be self-enforcing
- In the Dutch auction and the first-price sealed-bid auction the collusion agreement is not self-enforcing:
  - Bidders decide what is the designated “winner”
  - This bidder make a bid equal to the seller’s reservation price
  - All the other ring members are asked to refrain from bidding
  - However, each of the ring members can gain by placing a slightly higher bid in violation of the ring agreement
  - Therefore agreement is not self-enforcing

Collusion (2)

- In the English auction and in the Vickrey auction the collusion agreement is self-enforcing:
  - Bidders decide what is the designated “winner”
  - This bidder make a bid equal to its reservation price
  - All the other ring members are asked to refrain from bidding
  - None can gain from breaching the agreement, because none will ever exceed the designated bidder’s limit
  - Therefore agreement is self-enforcing

Collusion (3)

- Consider a setting wherein there are two bidders $b_1$ and $b_2$ with $v_1=100$ and $v_2=50$, and with agreement 40
- In the English auction:
  - $b_1$ can observe $b_2$’s bids, if $b_2$ decides to bid more than the agreed 40, $b_1$ can observe this and adjust its bid
  - Therefore, $b_2$’s optimal strategy is to bid no more than 40
- In the Vickrey auction:
  - $b_1$ submits its reservation price (100) while $b_2$ submits 40
  - $b_2$’s utility cannot increase if its bid exceeds the agreed price 40

Lying Auctioneer

- Overstate reservation price
- Phantom bidders
- In the English auction: use of shills that constantly raise the bids
- In the Vickrey auction: the auctioneer may overstate the second highest bid to the winner in order to increase revenue
Double Auctions (1)

- They capture the settings wherein there are more **buyers** and more **sellers**
- Each buyer and each seller make one bid
- The sellers' and buyers' bids are ranked highest to lowest
- **Two issues:**
  - What is the **clearing price**?
  - What are the **matchings** between buyers and sellers?

Double Auction (2)

<table>
<thead>
<tr>
<th>buy bids</th>
<th>sell bids</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Double Auction (3)

- **Matching:**
  - The **transaction set**: it is the set composed of the matched buyers and sellers, e.g. \( T = \{(4,4),(8,6),\ldots\} \)
  - The determination of \( T \) is tackled as follows:
    - \( T \) is initialized as empty
    - While the highest remaining buy bid is greater than or equal to the lowest sell bid, remove these bids and add this pair of bids to \( T \)

Double Auction (4)

\[ T = \{(13,4),(10,6),(9,7)\} \]
Double Auction (5)

- **Matching:**
  - The transaction set: it is the set composed of the matched buyers and sellers, e.g. \( T = \{(4,4),(8,6),\ldots\} \).
  - The determination of \( T \) is tackled as follows:
    - \( T \) is initialized as empty
    - While the highest remaining buy bid is greater than or equal to the lowest sell bid, remove these bids and add this pair of bids to \( T \).

- **Clearing price:**
  - Set the clearing price equal to the \( M \)th highest bid (\( M \)th price rule), where \( M \) is the number of the sellers
  - Set the clearing price equal to the \( M+1 \)st highest bid (\( M+1 \)st price rule), where \( M \) is the number of the sellers

Double Auction (6)

With \( T = \{(13,4), (10,6), (9,7)\} \):

- **\( M \)th price rule:**
  - Clearing price = 9
  - (13,4): the buyer pays 9 and the seller receives 9
  - (10,6): the buyer pays 9 and the seller receives 9
  - (9,7): ...

- **\( M+1 \)st price rule:**
  - Clearing price = 8
  - (13,4): the buyer pays 8 and the seller receives 8
  - (10,6): the buyer pays 8 and the seller receives 8
  - (9,7): ...

Combinatorial Auctions (1)

- The most useful auction for multiagent systems is the combinatorial auction
  - \( M \) items to sell/buy there are
  - Agents’ preferences are complex, depending on the set of items they buy (sell)
  - Agents can place bids for sets of items

- Example (4 items and 2 bidders):
  - Items = \{A, B, C, D\}
  - Bidder 1’s bids:
    - 1 for \( \{A\} \)
    - 2 for \( \{B\} \)
    - 1 for \( \{C\} \)
    - 4 for \( \{A,B\} \)
    - …

Combinatorial Auctions (2)

- Bidder 2’s bids:
  - 2 for \( \{A\} \)
  - 2 for \( \{B\} \)
  - 1 for \( \{C\} \)
  - 5 for \( \{A,B\} \)
  - …

- The largest number of bids for each bidder is \( 2^M \)
- A bidder may not bid over some possible sets of items

- Example:

<table>
<thead>
<tr>
<th>Items</th>
<th>Bidder 1</th>
<th>Bidder 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B, C, D</td>
<td>1 for ( {A} )</td>
<td>2 for ( {B} )</td>
</tr>
<tr>
<td></td>
<td>2 for ( {B} )</td>
<td>1 for ( {C,D} )</td>
</tr>
<tr>
<td></td>
<td>3 for ( {A,B} )</td>
<td>3 for ( {A,C,D} )</td>
</tr>
<tr>
<td></td>
<td>4 for ( {A,B,C} )</td>
<td>4 for ( {B,C,D} )</td>
</tr>
<tr>
<td></td>
<td>5 for ( {A,B,C,D} )</td>
<td>6 for ( {A,B,C,D} )</td>
</tr>
</tbody>
</table>
Combinatorial Auctions (3)

- The principal problem in a combinatorial auction is the determination of the winning bids in order to maximize the auctioneer's revenue.
- The winner determination is NP-hard [Rothkopf et al., 1998].
- If prices can be attached to single items in the auction, the winner problem can be reduced to a linear programming problem and, therefore, solved in polynomial time [Nisam, 2000].
- An approach is to conduct one of the standard AI-search over all possible allocations, given the bids submitted.
- Two approaches:
  - Branch-on-items search tree
  - Branch-on-bids search tree

Branch-on-Items (1)

- If there is not any singleton bid on item, this is added with price zero.
- All the children of the root are bids that have a 1 in them.
- The children of every node will be all the bids that contain the smallest number that is not on the path from the root to the node.
- If the node is a leaf and the set of bids from root to leaf constitutes one possible working bid set.
- Depth-first search (non mandatory)

Branch-on-Items (2)

<table>
<thead>
<tr>
<th>Bid Set</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>2</td>
</tr>
<tr>
<td>(3)</td>
<td>0</td>
</tr>
<tr>
<td>(3,3)</td>
<td>3</td>
</tr>
<tr>
<td>(1,2,3)</td>
<td>4</td>
</tr>
</tbody>
</table>

Significant Results in Literature

- In branch-on-items search:
  - [Fujishima et al., 1999] has developed a branch and bound algorithm that reduces the space of search on the basis of heuristics.
- A different search strategy:
  - Branch-on-bids: it produces a binary tree wherein each node is a bid and each edge represents whether or not that particular bid is in the solution [Sandholm, 2002].
  - [Sandholm et al., 2003] shows that the brach-on-bids search in much more efficient than branch-on-items search.
Auction Design Problem

- **Auction design** problem is a **mechanism design** problem
- The problem is to design protocols that are:
  - Incentive compatible
  - Individually rational
- Moreover, the mechanism should be robust with respect to **collusions** (group deviation)

Part 4
Auction Platforms

AuctionBot Architecture [Wurman et al., 1998]

![AuctionBot Architecture Diagram]

AuctionBot Description

- **Web interface**: interface for humans via web forms
- **TCP/IP interface**: interface for software agents
- **Database**: store the bids
- **Scheduler**: a daemon process that continuously monitors the database for auctions that have events to process or bids to verify
- **Auctioneer**: it loads the auction parameters and the set of current bids from the database
- **Bidding restrictions**:
  - **Participation**: \{1 : many\}, \{many : 1\}, \{many : many\}
  - **Bid rules**:
    - An agent’s new bid must dominate its previous bid
    - The bids must be discrete
### e-Game Architecture [Fasli et al., 2007]

- **Web interface**
- **e-Game package**
- **game API**
- **game managers**
- **scheduler**
- **Agent interface**
- **Game database**
- **Auctioneer processes**
- **Game data**
- **Game parameters**
- **Auction data**
- **Update events**
- **Retrieve**
- **Update**

### e-Game Description [Fasli et al., 2007]

- **Main features:**
  - Both human and artificial agents can access to
  - It supports a range of auction protocols that can be parameterised
  - More auction and other negotiation protocols can be developed
  - It supports the development of market scenarios by third parties
  - It is developed in Java

### Trading Agent Competition

- A **non-profit organization** that aims to promote research in market mechanisms and trading agents
- The effort was started in 2000
- Three **benchmark** problems have been created as testbeds to test one’s approaches and strategies:
  - The travel agent game (**CLASSIC**) – no more in use
  - The supply chain management game (**SCM**)  
    - It simulates a dynamic supply chain environment where agents compete to secure customer orders and components required for production of these orders
  - The market design game (**CAT**)  
    - CAT software agents represent brokers whose goals are to attract potential buyers and sellers as customers, and then to match buyers with sellers

### TAC SCM

- **TAC/SCM Scenario**
  - **Suppliers**
    - Variable supply and prices
  - **Manufacturers**
    - Limited capacity, competition for demand, suppliers, and interests
  - **Customers**
    - Different levels of demand, **“hurt”** due dates
  - **Agent**
    - Optimizing
  - Production schedule
  - Delivery schedule
  - RFQs & orders
  - Offers
TAC SCM Description (1)

- Six agents play in the game and start with no order from customers, no inventory, 0 back balance
- Agents do not know who the identity of the player they are playing against
- The objective is to maximize the profit through assembling PCs from different types of components and selling them at a profit to customers
- Highest bank balance wins
- 16 different types of PCs can be manufactured from 10 components which can be purchased from suppliers
- Factory capacity is limited

TAC SCM Description (2)

- An agent needs to perform the following tasks every day D
  - Negotiate supply contracts with suppliers
    - Send RFQs to suppliers
    - Receive offers on the RFQs sent on D-1
    - Decide which offers to accept from the suppliers
  - Bid for customer orders
    - Receive RFQs from customers
    - Decide which of these to bid on and send offers
    - Receive confirmations to orders for those offers sent on D-1
  - Manage assembly line and delivery schedule

References

References on Bargaining


References on Auctions