Agent-mediated negotiation and auction protocols: an overview

Tutorial notes for Santa Fe Institute
Complex Systems Summer School
22 June 2006

Valentin Robu

Dutch Center for Mathematics and Computer Science
Amsterdam, The Netherlands

Overview of the problem

- Resource and task allocation problems occur in: electronic commerce, distributed logistics, scheduling, supply chain management, bandwidth usage etc.
- Many types of mechanisms exist, which can be classified by:
  - Number of parties (2 or more)
  - Number of issues (1 or more)
  - Complexity of preferences over those issues
  - Degree of self interest (cooperative / competitive)
  - Information shared (direct vs. indirect mech.)
  - Centralized vs. decentralized
  - One-shot vs. multiple offers etc., etc.

Types of resource allocation mechanisms

- Negotiation (bargaining) mechanisms: typically decentralized, incomplete information
  - Bilateral negotiation (one/more issues)
  - One-many, many-many negotiations
  - Contract-net protocols
  - Coalition formation
- Auction mechanisms: usually centralised, direct revelation
  - English, Dutch and sealed-bid (Vickrey) auctions
  - Combinatorial auctions (VCG etc.)
  - Concurrent / sequential auctions
  - Continuous Double Auctions (CDA-s)
  - Preference elicitation mech.

Bilateral multi-issue negotiation: example case study

- Two agents negotiating over several issues (attributes) simultaneously, leading to a large space of possible contracts
- Attributes can de discrete (e.g. quality level) or continuous (price)
- Bargaining follows an alternative offers protocol
- Applications: e-commerce, agents within the same organisation, work contract negotiations, scheduling
- Our example: buying of a car

Multi-issue (multi-attribute) negotiations

- Fully Open Truthful Exchange (H. Raiffa)
  - Both parties reveal their preferences to a central "mediator" agent, who computes optimal outcomes
  - In an electronic environment, who controls the mediator agent? Can one prove its impartiality?
- Reasons for not revealing full preferences:
- Fear the other may use it to get a better deal
- Privacy concerns
- Preference elicitation problem (n items = 2^n bundles)
- Heuristic search: Can we guess opponent's preferences based on his past bids (offers) ?

Example set-up: sale of a car

- Four attributes (CD player, Extra speakers, Tow hedge, Air conditioning) have value labels, and each party assigns to them an evaluation.
- For Buyer: good =100, fairly good = 85, standard =70, meager = 20, none = 0
- For Seller: good =30, fairly good = 65, standard = 80, meager = 65, none =100
- The evaluation of price is described by a linear function (ascending or descending)

Example set-up (2)

- Each attribute is given a preference weight coefficient.

$$U_{contract} = \sum w_i * U_i \quad \text{for all items i}$$

- Symmetrical vs. asymmetrical preferences

	Buyer	Seller
Airco	90 (18%)	15 (3%)
Dr. hook	90 (18%)	15 (3%)
CD player	15 (3%)	90 (18%)
Speakers	15 (3%)	90 (18%)
Price	300 (59%)	-

EXAMPLE TRACE

BUYER'S INTERFACE

```
round price  drawing     airco     extra    cd_player  utility   utility
             hook                  speakers            own bid   others

1     18000  good        good      good     good       1         0.740741
2     17450  fairly good standard  meager   meager     0.92037   0.829185
3     18222  fairly good standard  none     standard   0.909481  0.839926
..9   18583  fairly good standard  none     standard   0.882741  0.867407
```

SELLLER'S INTERFACE

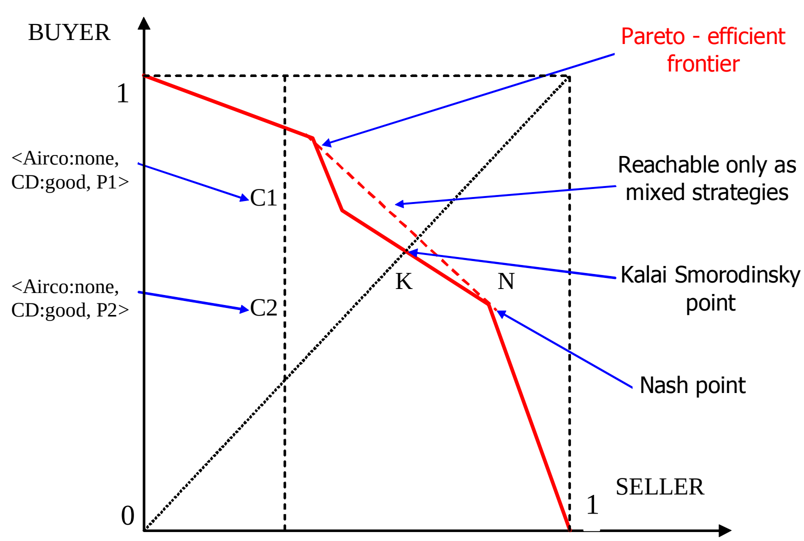
```
round price  drawing     airco     extra     cd_player  utility   utility
             hook                  speakers             own bid   others

1     16900  none        none      none      none       1         0.316667
2     19306  fairly good standard  none      standard   0.938269  0.595321
3     19161  fairly good standard  none      standard   0.919679  0.799295
..9   18790  fairly good standard  none      standard   0.872115  0.845577
```

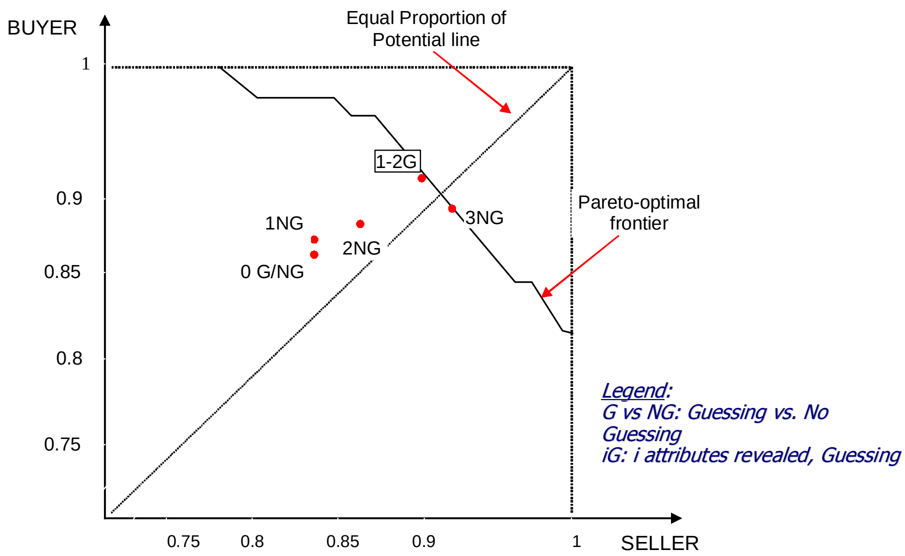
Measures of efficiency and fairness

- Pareto-optimal contract: A contract is said to be Pareto-optimal if no further improvement is possible in the utility of one agent, without reducing the utility of the other agents
- Pareto frontier: Set of all Pareto-optimal contracts
- Among the set of Pareto-optimal points, there are a few so-called solution "concepts"
  - Utilitarian: contract combination which maximizes the sum of utilities of the agents
  - Egalitarian (Kalai-Smorodinsky): maximizes the MINIMUM of the two utilities
  - Nash point: maximizes the PRODUCT of the utility functions of both agents

Multi-issue contract space



Experimental results



Multi-issue negotiation heuristics

- Many techniques exist to learn based on other counter proposals
  - Probabilistic
  - Fuzzy logic
  - Distance-based
- BUT remember: this example is for linear utility functions only!
- Non-linear utility function case is much harder: especially for high-dimensional, incomplete preference information

Bargaining with more than 2 agents

- Contract net protocol [Smith, 1977]
  - General mechanism for contracting and sub-contracting tasks
  - Two types of agents: initiators (managers) and participants (contractors)

1. Each initiator sends out a call for proposals (units, price, response deadline)
2. Participants review CFP-s and bid on feasible ones, accordingly.
3. After the deadline, the initiator chooses the best bid and awards the contract to the respective participant.
4. The initiator rejects the other bids.

After step 3, contractors may decompose/subcontract (parts of) tasks.

- Original CNP protocol assumes cooperative behavior, many extensions
- Original protocol assumes cooperation, many extensions exist!

Coalition formation & Shapley values

- Coalitions: groups of agents get higher payoff than individual agents
  - Many concepts for forming coalitions and dividing joint gains (core, kernel, Shapley values)
- Shapley value example (after [Vidal '05])

Coalition	Value
None	0
A1 only	1
A2 only	3
A1 and A2	6

Q: How to divide the joint gains of 6?

A: Consider all possible orders of joining the coalition

Sh({1,2}, 1)=½*[v(1)-v()+v(1,2)-v(2)]=2

Sh({1,2}, 2)=½*[v(1,2)-v(1)+v(2)-v()]=4

More centralized approaches: auctions

- Very popular an widely researched
  - One shot, single unit: English, Dutch, sealed bid, Vickrey etc.
  - Many units, many items (combinatorial)
- Mechanism Design
- Designing the mechanism (auction protocol) in such a way that:
  - It is truthful
  - Efficiently computable
  - Simplify the computation problem for the agents
- Designing the bidding strategies of the agents
- Many situations are inherently sequential (e.g. real-time planning and scheduling)
- No equilibrium bidding strategies exists for many auctions – e.g. sequential auctions with complementarities, CDA-s etc.
- A variety of machine learning strategies can be used (TAC literature)

Single-item auctions: Open cry

- Ascending English auction:
  - In each round, all parties can submit a price that is higher than the one announced in the previous round
  - The auctioneer selects the highest price and announces it
  - Game repeats until no agent offers more
  - The good is given to the highest price agent at the price offered
- Descending Dutch auction:
  - Auctioneer starts from the highest price, and reduces it, in subsequent rounds
  - Auction stops when one agent offers to buy the item at the current price

Single-item auctions: sealed bids

- Imagine all the bidders submitting their offer for good G in a sealed envelope, without knowing what the other bidders offered
- The bidder with the highest bid gets the item and pays:

- First price sealed bid:
  - Gets the item and pays the price he offered
  - NOT incentive compatible, the likely winner "shades" her bid (i.e. bids less than what it's truly worth to him/her)
- Second price sealed bid (Vickrey, 1961):
  - The winner gets the item, BUT pays the price of the second highest bidder + ε
  - Bidders will always bid their true worth for an item!

Sequential auctions & the bidding problem

- Example adapted after (Wellman et al., '98)
- Suppose we have a scheduling problem on a machine and 2 time slots: S1, S2
- The two time slots will be auctioned off SEQUENTIALLY
- We have 2 agents that need to use the machine A1 and A2:
  - A1 need both slots and is willing to pay $300, nothing for one slot
  - A2 needs exactly one slot (either S1 OR S2) is willing to pay $200
- No efficient equilibrium bidding strategy (check!)

Sequential auctions: bidding strategies

- Many (if not most) real-world settings involve many parties and dynamic environments (e.g. transportation logistics, electricity markets, travel reservations (TAC), dynamic supply chain chains, bandwidth demand in Starbucks, etc.
- Many machine learning techniques have been used: RL, evolutionary, Bayesian, fuzzy, other heuristics etc.
- To test the efficacy of these techniques against each other => Trading Agent Competition (TAC)
  - TAC Classic: inter-dependent reservations of hotel, flight and entertainment tickets
  - TAC Supply Chain Management: buy of computer parts and sell of ready assembled computers

Combinatorial auctions

- Set of items k items has to be distributed among n agents
- Agents bid a value for all bundles that have a value for them
- Bidding languages, such as XOR of ANDs and k-additive

- Mechanism Design
- Designing the mechanism (auction protocol) in such a way that:
  - It is truthful
  - Efficiently computable
  - Simplify the computation problem for the agents
- Economic literature: generally considers the cases where equilibriums are exactly computable (restricted set)
- Computer Science/OR literature:
  - Considers combinatorial cases, where it's computationally intractable for agents to "beat" the system.
  - Graph theory/OR approximation and other methods are used for this.

Vickrey-Clarke-Groves mechanism

- Set of allocations: $A = (a_1 ... a_M)$
- Agents declare values: $\theta = (\theta_1 ... \theta_N)$
- Center selects allocation a* which maximizes the sum of values of agents i: $\sum_i v_i(a, \theta_i)$
- Each agent pays: $\sum_{j \neq i} v_j(a^{-1}, \theta_j) - \sum_{j \neq i} v_j(a^*, \theta_j)$
- Where a^{-1} solves for: $\max_{a \in A} \sum_{j \neq i} v_j(a \ , \theta_j)$
- Intuition: Each agent pays the difference from the allocation which does not include her

VCG example

- Example and notation in previous slide: [Parkes,'04]
- Agents 1, 2, 3 and items A, B

Buyer 3 wins and pays 10-0 =10.

	A	B	AB
1	5	0	5
2	0	5	5
3			12

Buyer 1 and 2 win and pay 7-5=2 each.

	A	B	AB
1	5	0	5
2	0	5	5
3	0	0	7

Other issues & conclusions

- Preference elicitation
  - The number of combinations is exponential in the number of items
  - Even if we can solve the winner determination problem in polynomial time, this has limited applicability if the agents themselves may find it hard to specify their full preferences => new research area of pref. elicitation

- Overall conclusion:
- There are many mechanisms for resource allocation: choose one appropriate for your problem
- In any distributed application (e.g. planning, scheduling, networking etc.), allocating resources when agents are self-interested is much harder than the cooperative case