CM30174: Intelligent Agents

Marina De Vos, Julian Padget

Negotiation / version 0.4

November 2, 2010

Authors/Credits for this lecture

- Chs. 14, 15 and 9 of "An Introduction to Multiagent Systems" [Wooldridge, 2009].

Content

Content

Reaching Agreements

- How do self-interested agents reach agreements?
- In an extreme case (zero sum encounter) no agreement is possible — but in most scenarios, a mutually beneficial agreement can be concluded
- The capabilities of negotiation and argumentation are central to the ability of an agent to reach such agreements.
  - Consider an offer as $\vec{v_i}$ s.t. $v_i \in \mathbb{R}^n$
  - Valuation is then $\Sigma_{i=0}^{n} w_i v_i$ such that given a threshold value, a decision can be made
  - Simple negotiation requires each agent to change $\vec{v_i}$ such that the valuation monotonically approaches the threshold
  - Argumentation is the process of one agent getting another to change its $\vec{w_i}$

Protocols and Strategies

- Negotiation is governed by a particular mechanism, or protocol.
- The mechanism defines the “rules of encounter” between agents.
  - Auctions are a large class of “useful” mechanisms
- Mechanism design is the process of designing mechanisms so that they have certain desirable properties.
- Given a particular protocol, how can a particular strategy be designed that individual agents can use?
  - What is the *dominant* strategy for a particular mechanism?

Content

Auctions

- An auction takes place between an agent known as the auctioneer and a collection of agents known as the bidders.
- The goal of the auction is for the auctioneer to allocate the good to one of the bidders.
- In most settings the auctioneer desires to maximise the price; bidders desire to minimise price.

Auction Parameters

- Goods can have:

  private value OR public/common value OR correlated value

- Winner may pay:

  first price OR second price OR n^{th} price

- Bids may be:

  open cry OR sealed bid

- Bidding may be:

  one shot OR ascending OR descending

English Auctions

- English auction characteristics:
  - first-price,
  - open cry,
  - ascending.
- Susceptible to:
  - Winner's curse
  - Shills
- Dominant strategy is for agent successively to bid a small amount more than the current highest bid until it reaches their valuation, then withdraw.

Dutch Auctions

- Dutch auctions characteristics:
  - open-cry
  - descending
  - auctioneer starts price at artificially high value;
  - auctioneer lowers offer price until some agent makes a bid equal to the current offer price;
  - the good is then allocated to the agent that made the offer.
- Best strategy is to bid only at own valuation

Sealed-Bid Auctions

- Sealed bid auction characteristics:
  - first-price
  - sealed-bid
  - one-shot
- Single round;
- Bidders submit a sealed bid for the good;
- Good is allocated to agent that made highest bid.
- Winner pays price of highest bid.
- Best strategy is to bid less than true valuation.

Vickrey Auctions

- Vickrey auctions characteristics:
  - second-price
  - sealed-bid
  - one-shot
- Good is awarded to the agent that made the highest bid; at the price of the second highest bid.
- Vickrey auctions susceptible to antisocial behavior: untruthful bids can distort market
- Bidding to your true valuation is dominant strategy in Vickrey auctions.
  - Overbid $\rightsquigarrow$ risk of paying above valuation
  - Underbid $\rightsquigarrow$ reduced chance of success

Continuous Double Auction

- Perhaps overlooked because it is simple, but it is also effective and the basis for many real-world mechanisms
- Given buyer *B* and seller *S*, proceed in rounds:
  - Step 1: The seller announces an offer price p_1
  - Step 2: The buyer announces a bid price p_2, assume $p_1 > p_2$
  - Step 3: If $p_2 \geq p_1$, sale is agreed at $\frac{p_1+p_2}{2}$
  - Step 4: Seller *reduces* offer giving $p_i, i \in 3, 5, ...$
  - Step 5: Buyer *increases* bid giving $p_j, j \in 4, 6, ...$
  - Return to Step 3
- Description in terms of two agents, but readily adapts for multiple buyers and multiple sellers
- Example of a mechanism that is more generally applicable in an electronic than a physical setting.

Zero Intelligence Traders

- Original idea set out in [Gode and Sunder, 1993]
- Trader submits random bids and offers
- Simulations using experimental CDA markets demonstrate the transaction price time-series is *human-like*:
  - Appearing to converge to the theoretical equilibrium price
  - Yielding allocative efficiency comparable to human markets
- Assertion: no intelligence required to trade in a CDA as long as not permitted to trade at a loss — it is the market structure that ensures allocative efficiency
- [Cliff and Bruten, 1997] identified pathological market conditions for ZI traders, supported by empirical results

Zero Intelligence Plus (ZIP)

- Augments ZI-traders with simple machine-learning mechanism
- ZIP succeeds where ZI fails
- ZIP traders adapt their profit margin on the basis of four factors:
  - whether an agent still needs to buy or sell
  - was the last quote an offer (seller) or a bid (buyer)
  - was the last quote accepted or rejected
  - was the last quote bigger or smaller than own quote
- At time t trader i calculates
  - the shout price $p_i(t)$ for a unit j with
  - limit price $\lambda_{i,j}$ using
  - profit-margin $\mu_i(t)$, such that $p_i(t) = \lambda_{i,j}(1 + \mu_i(t))$

ZIP profit margin algorithm

- For sellers:
  1. if last shout accepted at price q then
     (i) any seller s_i whose $p_i \leq q$ raises its margin
     (ii) if last shout was a bid then any seller s_i whose $p_i \geq q$ lowers its margin
  2. else
     (i) if last shout was an offer then any seller s_i whose $p_i \geq q$ lowers its margin
- For buyers:
  1. if last shout accepted at price q then
     (i) any buyer b_i whose $p_i \geq q$ raises its margin
     (ii) if last shout was an offer then any buyer b_i whose $p_i \leq q$ lowers its margin
  2. else
     (i) if last shout was a bid then any buyer b_i whose $p_i \leq q$ lowers its margin

ZIP Adaptation

- Adapation arises from the alteration of the profit margin using the Widrow-Hoff "delta rule", widely used in back-propagation: $A(t+1) = A(t) + \Delta(t)$
- $\Delta(t)$ is the change in output, determined by the product of a learning rate β and the difference between $A(t)$ and the desired output at time t, denoted $D(t)$:
  $\Delta(t) = \beta(D(t) - A(t))$
- if $D(t)$ is constant the update rule gives asymptotic convergence of $A(t)$ to $D(t)$ at a speed determined by β.
- When a trader has to change its profit margin, compute a target price $\tau_i(t)$ and use the Widrow-Hoff rule to compute the shout price at the next time step, $p_i(t+1)$
- Full details in [Cliff and Bruten, 1997]

Combinatorial Auctions 1/4

- Preceding mechanisms rely on “intelligence” on the part of the bidder as the source for (economically) efficient, effective allocation
- Need complete information for a Pareto optimal allocation
- CA aims to achieve optimal allocation by putting all the decision making in the auctioneer, hence we have
- The Winner Determination Problem: given a set of bids in a CA, find an allocation of items (not necessarily all) to bidders to maximize revenue
- Revenue is maximized by the allocation that maximizes the sum over all bidders of the bidders’ valuations for the subset of items they receive.
- Bids are specified as valuations for a subset of items—called a *bundle*

Combinatorial Auctions 2/4

- WDP can be written as an integer linear program that is equivalent to the weighted set packing problem and hence NP-hard.
- Problem is hard because need to check for each subset of the bids whether the subset is feasible (no bids share items) and how much revenue results... For k bids, there are 2^k subsets.
- Some special cases can be solved quickly, for example when the number of items is small ($O(3^m)$ for m items) or the number of bids is large.
- Can also approximate optimality to within some bound in polynomial time ($O(max(l^c, m^2 l^2))$ for m items and l bids.

Combinatorial Auctions 3/4

- Example: given n items for sale (1,2,3,4,5,6,7,8,9), can construct a set of tuples describing the bidder, their bundle and their valuations:

Bidder	Bid	Bundle
1	45	1,2
2	98	1,4,7,8
3	86	9,4,5,1,2
4	62	9,4

- Looking at individual bids, the revenue from bid 2 is greatest
- But bids 1 and 4 are for non-overlapping bundles, and so can *both* be satisfied and maximizes revenue by selling elements 1,2,4,9 for 107 (45+62).

Combinatorial Auctions 4/4

- (Naïve) Algorithmic solution builds a matrix of all possible combinations, then searches for sets that generate the greatest utility.
  - Take first combination
  - Then examine any other combinations that (a) match and (b) have a higher score, until done
- Practical solvers essentially take the same approach, but have good heuristics for pruning the search tree.

Which auction?

- For bundles: CA is optimal, but not necessarily practical
- For simple encounters: CDA is effective
- Economic theory says there is no difference between the rest in general, although individuals may differ

Content

Negotiation

- Auctions are only concerned with the allocation of resource: richer techniques for reaching agreements are required.
- 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 are 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.

Task Oriented Domains

- A TOD is a triple: $\langle T, Ag, c\rangle$ where:
  - T is the (finite) set of all possible tasks
  - $Ag = \{A_1, \ldots, A_n\}$ is set of participating agents
  - $c : \mathcal{P}(T) \rightarrow \mathbb{R}^+$ defines cost of executing each subset of tasks
  - The cost function is *monotonic*: adding tasks never decreases the cost, i.e.

$$T_1, T_2 \subseteq T \text{and} T_1 \subseteq T_2 \Rightarrow c(T_1) \leq c(T_2)$$

- An encounter occurs when the agents A_i are assigned tasks from $T = \langle T_1, \ldots, T_n\rangle$, where $A_i \in Ag$, $T_i \subseteq T$
- But can agent A_i do better by negotiating a reallocation of its tasks?

Deals in TODs

- Given agents $\{A_1, A_2\}$ and encounter $\langle T_1, T_2\rangle$, a deal is the allocation of the tasks $T_1 \cup T_2$ to the agents A_1 and A_2
- A deal is $\langle D_1, D_2\rangle$, where agent A_i is allocated tasks D_i with no tasks left over: $D_1 \cup D_2 = T_1 \cup T_2$
- The cost of such a deal to agent A_i is $\text{cost}_i(D_1, D_2) = c(D_i)$
- The utility of a deal δ to agent i is the difference between the cost of fulfulling its original allocation and the cost of the tasks in the deal: $\text{utility}_i(\delta) = c(T_i) - cost_i(\delta)$
- In the absence of agreement, the agents must accept the conflict deal, Θ that is the deal $\langle T_1, T_2\rangle$ consisting of the tasks originally allocated. Note that

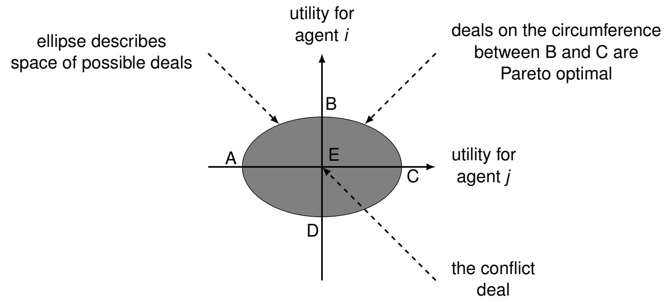
$$utility_i(\Theta) = 0, \ \forall A_i \in Ag$$

- Deal δ is individual rational if it gives positive utility

Deal Dominance

- The deal $\delta_1 \succ \delta_2$ (dominates) iff:
  - C1: Deal δ_1 is at least as good as δ_2 for every agent:
  $$\forall i \in \{1, 2\}, \mathit{utility}_i(\delta_1) \geq \mathit{utility}_i(\delta_2)$$
  - C2: Deal δ_1 is better than δ_2 for some agent:
  $$\exists i \in \{1, 2\}, \mathit{utility}_i(\delta_1) > \mathit{utility}_i(\delta_2)$$
- A deal δ_1 weakly dominates δ_2 if at least C1 holds
- A deal that is not dominated by any other deal is Pareto optimal
- A deal is individual rational if it weakly dominates the conflict deal
- The negotiation set is the set of deals that are: individual rational *and* Pareto efficient

The negotiation set



- ABCD = possible deals
- utility(E)>all deals left of B-D for agent*j*
- utility(E)>all deals below A-C for agent*i*

The Monotonic Concession Protocol

- 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.

The Zeuthen Strategy

Three questions:

- 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 to conflict deal.
  - In case conflict occurs, you are not much worse off.
  - You are more willing to risk confict.
- An agent will be more willing to risk conflict if the difference in utility between its current proposal and the conflict deal is low.

Nash Equilibrium

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 itself...

> *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.*

Working Together

- Why and how to get agents work together?
  - task sharing: components of a task are distributed to component agents
  - result sharing: information (partial results etc) is distributed.

Benevolent Agents

- If we “own” the whole system, we can design agents to help each other whenever asked.
- In this case, we can assume agents are benevolent: our best interest is their best interest.
- Problem-solving in benevolent systems is cooperative distributed problem solving (CDPS).
- Benevolence simplifies the system design task enormously!

Self-Interested Agents

- If agents represent individuals or organisations, (the more general case), then we cannot make the benevolence assumption:
- Agents will be assumed to act to further their own interests, possibly at expense of others.
- Potential for conflict.
- May complicate the design task enormously.

The Contract Net

- Contract Net is a protocol for task-sharing in five phases:
  1. Recognition
  2. Announcement
  3. Bidding
  4. Awarding
  5. Expediting.

Phase 1: Recognition

- In this stage, an agent recognises it has a problem it wants help with. Agent has a goal, and either...
  - Realises it cannot achieve the goal in isolation — does not have capability
  - Realises it would prefer not to achieve the goal in isolation (typically because of solution quality, deadline, etc)

Phase 2: Announcement

- In this stage, the agent with the task sends out an announcement of the task which includes a specification of the task to be achieved.
- Specification must encode:
  - Description of task itself (maybe executable)
  - Any constraints (e.g., deadlines, quality constraints).
  - Meta-task information (e.g., “bids must be submitted by... ”)
- The announcement is then broadcast.

Phase 3: Bidding

- Agents that receive the announcement decide for themselves whether they wish to bid for the task.
- Factors:
  - Agent must decide whether it is capable of expediting task
  - Agent must determine quality constraints and price information (if relevant).
- If they do choose to bid, then they submit a tender.

Phases 4+5: Awarding & Expediting

- Agent that sent task announcement must choose between bids and decide to which to award the contract
- The result of this process is communicated to agents that submitted a bid.
- The successful contractor then expedites the task.
- May involve generating further manager-contractor relationships: sub-contracting.
  - Recursive: a contractor issues a CFP etc.
  - Iterated: manager does not refuse bids, but negotiates with potential contractor

Content

Summary

- Conventional auctions
- Zero Intelligence Traders
- Combinatorial auctions
- Task Oriented Domains
- Contract net protocol

Recommended Reading

- "An Introduction to Multiagent Systems" [Wooldridge, 2009]:
  - Chapter 6, 7 and 9
  - Chapter 14, pp299-310, combinatorial auctions
- Two papers [Bigham and Du, 2003] and [Bussmann and Schild, 2000] illustrate applications of negotiation
- Cramton [Cramton et al., 2005] examines combinatorial auctions in exhaustive detail, wherein Lehmann and Sandholm [Lehmann et al., 2005] discuss the Winner Determination Problem.
- [Zlotkin and Rosenschein, 1993] is the original TOD paper.
- [Cliff and Bruten, 1997] describes ZIP traders.

References I

Bigham, J. and Du, L. (2003).
Cooperative negotiation in a multi-agent system for real-time load balancing of a mobile cellular network.
In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 568–575. ACM Press.

Bussmann, S. and Schild, K. (2000).
Self-organizing manufacturing control: An industrial application of agent technology.
In *ICMAS*, pages 87–94. IEEE Computer Society.

Cliff, D. and Bruten, J. (1997).
Minimal-intelligence agents for bargaining behaviours in market-based environments.
Technical Report HPL-97-91, Hewlett-Packard Laboratories.
Available via `http://www.hpl.hp.com/techreports/97/HPL-97-91.html`, last accessed November 2007.

References II

- Cramton, P., Shoham, Y., and Steinberg, R., editors (2005).
  Combinatorial Auctions.
  MIT Press.
  ISBN: 0-262-03342-9.
- Gode, D. K. and Sunder, S. (1993).
  Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality.
  The Journal of Political Economy, 101(1):119–137.
- Lehmann, D., Müller, R., and Sandholm, T. (2005).
  The Winner Determination Problem, chapter 12.
  MIT Press.
  ISBN: 0-262-03342-9. This chapter available via `http://www.cs.cmu.edu/~sandholm/winner-determination-final.pdf`.
- Wooldridge, M. (2002).
  An introduction to multiagent systems.
  Wiley.
  ISBN: 0 47149691X.

References III

- Wooldridge, M. (2009).
  An introduction to multiagent systems (second edition).
  Wiley.
  ISBN: 978-0-470-51946-2.
- Zlotkin, G. and Rosenschein, J. S. (1993).
  A domain theory for task oriented negotiation.
  In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 416–422, Chambery, France.