LECTURE 10: REACHING AGREEMENT

An Introduction to Multiagent Systems

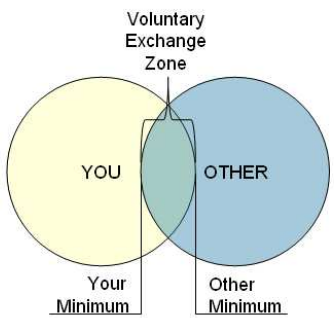
CIS 716.5, Spring 2010

Today

- How do agents *reach agreements* when they are self interested?
- In an extreme case (zero sum encounter) no agreement is possible — but in most scenarios, there is potential for *mutually beneficial agreement* on matters of common interest.
- The capabilities of:
  - *negotiation* and
  - *argumentation*

  are central to the ability of an agent to reach such agreements.
- This lecture will cover material from two chapters of the textbook (though obviously not *all* the material from the chapters)

Two pictures that summarise negotiation



Mechanisms, Protocols, and Strategies

- Negotiation is governed by a particular *mechanism*, or *protocol*.
- The mechanism defines the "rules of encounter" between agents.
- *Mechanism design* is designing mechanisms so that they have certain desirable properties.
  - Properties like Pareto efficiency
- Given a particular protocol, how can a particular *strategy* be designed that individual agents can use?

- Auctions are *only* concerned with the allocation of goods: 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.

- There are a number of aspects of negotiation that make it complex.
- Multiple issues
  - Number of possible deals is exponential in the number of issues.

    (Like the number of bundles in a combinatorial auction)
  - Hard to compare offers across multiple issues

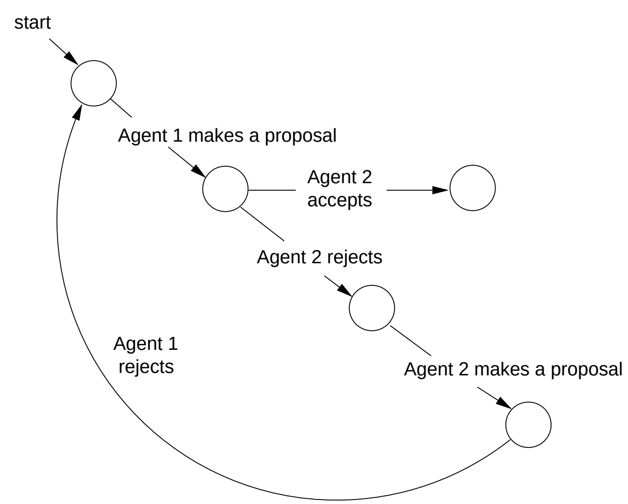
    The car salesman problem
- Multiple agents
  - One-to-one negotiation
  - Many-to-one negotiation
  - Many-to-many negotiation
- At the simple end there isn't much to distinguish negotiation from auctions.

Negotiation for Resource Division

- We will start by looking at Rubinstein's *alternating offers* model.
- This is a one-to-one protocol.
- Agents are 1 and 2, and they negotiate over a series of rounds:

$$0, 1, 2, \ldots$$

- In round 0, Agent 1 makes an offer x^0.
- Agent 2 either accepts A, or rejects R.
- If the offer is accepted, then the deal is implemented.
- If not, we have round 1, and Agent 2 makes an offer.



- The rules of the protocol don't mean that agreement will ever be reached.
  - Agents could just keep rejecting offers.
- If there is no agreement, we say the result is the *conflict deal* Θ.
- We make the following basic assumptions:
  - Disagreement is the worst ouctome
    Both agents prefer any agreement to none.
  - Agents seek to maximise utility
    Agents prefer to get larger utility values
- With this basic model, we get some odd results.

- Consider we are dividing a pie (m'mmmm, pie)

- Model this as some resource with value 1, that is divided into two parts.
  - Each part is between 0 and 1.
  - The two parts sum to 1

  so a proposal is $(x, 1 - x)$
- The set of possible deals is:

$$\{(x, 1 - x) : 0 \leq x \leq 1\}$$

- If you are Agent 1, what do you offer?

- Let's assume that we will only have one round.
  Ultimatum game
- Agent 1 has all the power.
- If Agent 1 proposes $(1, 0)$, then this is still better for Agent 2 than the conflict deal.
- Agent 1 can do no better than this either.
- So we have a Nash equilibrium.

- If we have two rounds, the power passes to Agent 2.
- Whatever Agent 1 proposes, Agent 2 rejects it.
- Then Agent 2 proposes $(0, 1)$.
- Just as before this is still better for Agent 1 than the conflict deal and so it is accepted.
- A bit of thought shows that this will happen any time there is a fixed number of rounds.

- What if we have an indefinite number of rounds.
- Let's say that Agent 1 uses this strategy:

  Always propose $(1, 0)$ and always reject any offer from Agent 2
- How should Agent 2 respond?
- If she rejects, then there will never be agreement.
  - Conflict deal
- So accept. And there is no point in not accepting on the first round.
- In fact, whatever $(x, 1 - x)$ agent 1 proposes here, immediate acceptance is the Nash equilibrium so long as Agent 2 *knows* what Agent 1's strategy is.

Impatient players

- Since we have an infinite number of Nash equilibria, the solution concept of NE is too weak to help us.
- Can get unqiue results if we take time into account.

  For any outcome x and times $t_2 > t_1$, both agents prefer x at time t_1.
- A standard way to model this impatience is to discount the value of the outcome.
- Each agent has δ_i, $i \in \{1, 2\}$, where $0 \leq \delta < 1$.
- The closer δ_i is to 1, the more patient the agent is.

- If agent i is offered x, then the value of the slice is:
  - x at time 0
  - $\delta_i x$ at time 1
  - $\delta_i^2 x$ at time 2.

    ⋮
  - $\delta^k x$ at time k
- Now we can make some progress with the fixed number of rounds.
- A 1 round game is still an ultimatum game.
- A 2 round game means Agent 2 can play as before, but if so, will only get δ_2.

  Gets the whole pie, but it is worth less.

- Agent 1 can take this into account.
- If Agent 1 offers:

$$(1 - \delta_2, \delta_2)$$

  then Agent 2 might as well accept — can do no better.
- So this is now a Nash equilibrium.

- In the general case, agent 1 makes the proposal that gives Agent 2 what Agent 2 would be able to enforce in the second round.
- Agent 1 gets:

$$\frac{1 - \delta_2}{1 - \delta_1\delta_2}$$

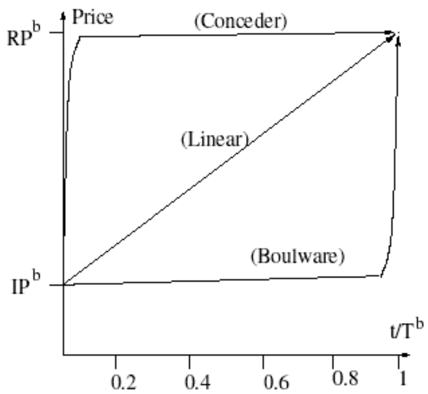
- Agent 2 gets:

$$\frac{\delta_2(1 - \delta_1)}{1 - \delta_1\delta_2}$$

- Note that the more patient either agent is, the more pie they get.

Heuristic approach

- The approach we just talked about relies on strageic thinking about the other player.
- A simpler approach is to use some heuristic approximation of how the value of the pie varies for the players.
- Some common approximations:
  - Linear
  - Boulware
  - Conceder
- We can see what these look like for buyers.



- Linear
  - Linear increase from initial price at the start time to reserve price at the deadline.
- Boulware
  - Very slow increase until close to deadline and then an exponential increase.
- Conceder
  - Inital exponential increase to close to the reserve price and then not much change.

Negotiation in Task-Oriented Domains

Imagine that you have three children, each of whom needs to be delivered to a different school each morning. Your neighbour has four children, and also needs to take them to school. Delivery of each child can be modelled as an indivisible task. You and your neighbour can discuss the situation, and come to an agreement that it is better for both of you (for example, by carrying the other's child to a shared destination, saving him the trip). There is no concern about being able to achieve your task by yourself. The worst that can happen is that you and your neighbour won't come to an agreement about setting up a car pool, in which case you are no worse off than if you were alone. You can only benefit (or do no worse) from your neighbour's tasks. Assume, though, that one of my children and one of my neigbours's children both go to the same school (that is, the cost of carrying out these two deliveries, or two tasks, is the same as the cost of carrying out one of them). It obviously makes sense for both children to be taken together, and only my neighbour or I will need to make the trip to carry out both tasks.

TODs Defined

- A task-oriented domain (TOD) is a triple

$$\langle T, Ag, c \rangle$$

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

$$\langle T_1, \ldots, T_n \rangle$$

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

Deals in TODs

- Given encounter $\langle T_1, T_2 \rangle$, a *deal* will be an allocation of the tasks $T_1 \cup T_2$ to the agents 1 and 2.
- The *cost* to i of deal $\delta = \langle D_1, D_2 \rangle$ 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 $\langle T_1, T_2 \rangle$ consisting of the tasks originally allocated.
  Note that

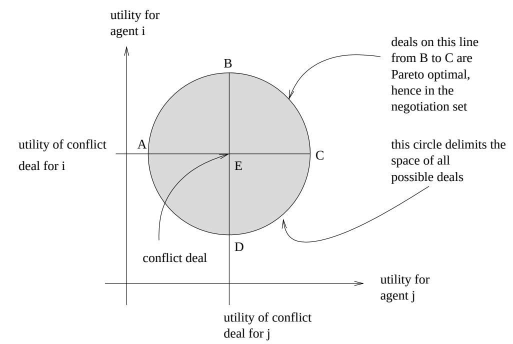
$$utility_i(\Theta) = 0 \quad \text{for all } i \in Ag$$

- Deal δ is *individual rational* if it gives positive utility.

The Negotiation Set

- The set of deals over which agents negotiate are those that are:
  - individual rational
  - pareto efficient.
- Individually rational: agents won't be interested in deals that give negative utility since they will prefer the conflict deal.
- Pareto efficient: agents can always transform a non-Pareto efficient deal into a Pareto efficient deal by making one agent happier and none of the others worse off.

The Negotiation Set Illustrated



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.

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 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 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.

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.

Argumentation

- Argumentation is the process of attempting to agree about what to believe.
- Only a question when information or beliefs are contradictory.
  - If everything is consistent, just merge information from multiple agents.
- Argumentation provides principled techniques for resolving inconsistency.
- Or at least, sensible rules for deciding what to believe in the face of inconsistency.
- The difficulty is that when we are presented with p and $\neg p$ it is not at all clear what we should believe.

- Gilbert (1994) identified 4 modes of argument:
  1. *Logical mode* — akin to a proof.
     "If you accept that A and that A implies B, then you must accept that B".
  2. *Emotional mode* — appeals to feelings and attitudes.
     "How would you feel if it happened to you?"
  3. *Visceral mode* — physical and social aspect.
     "Cretin!"
  4. *Kisceral mode* – appeals to the mystical or religious
     "This is against Christian teaching!"
- Depending on circumstances, some of these might not be accepted.

Abstract Argumentation

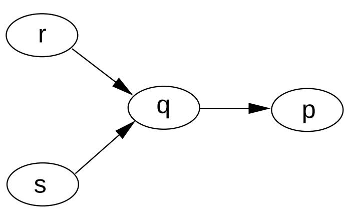
- Concerned with the overall structure of the set of arguments
  – (rather than internals of individual arguments).
- Write $x \rightarrow y$
  – "argument x attacks argument y";
  – "x is a counterexample of y; or
  – "x is an attacker of y".

  where we are not actually concerned as to what x, y are.
- An *abstract argument system* is a collection or arguments together with a relation "$\rightarrow$" saying what attacks what.

- Systems like this are called *Dung-style* after their inventor.
- A set of Dung-style arguments:

$$\langle\{p, q, r, s, \}, \{(r, q), (s, q), (q, p)\}\rangle$$

  meaning that r attacks q, s attacks q and q attacks p.



- The question is, given this, what should we believe?

Preferred extensions

- There is no universal agreement about what to believe in a given situation, rather we have a set of criteria.
- A *position* is a set of arguments.
  – Think of it as a viewpoint
- A position S is *conflict free* if no member of S attacks another member of S.
  – Internally consistent
- The conflict-free sets in the previous system are:

$$\emptyset, \{p\}, \{q\}, \{r\}, \{s\}, \{r, s\}, \{p, r\}, \{p, s\}, \{r, s, p\}$$

- If an argument a is attacked by another a', then it is *defended* by a'' if a'' attacks a'.
- Thus p is defended by r and s.

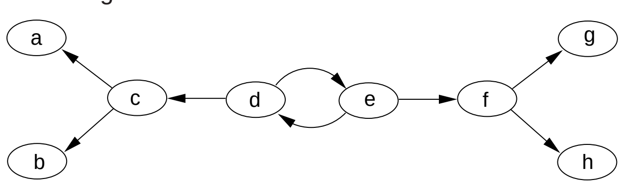
- A position S is *mutually defensive* if every element of S that is attacked is defended by some element of S.
  - Self-defence is allowed
- These positions are mutually defensive:

$$\emptyset, \{r\}, \{s\}, \{r,s\}, \{p,r\}, \{p,s\}, \{r,s,p\}$$

- A position that is conflict free and mutually defensive is *admissible*.
- All the above positions are admissible.
- Admissibility is a minimal notion of a reasonable position — it is internally consistent and defends itself against all attackers.

- A *preferred extension* is a maximal admissible set.
  - adding another argument will make it inadmissible.
- In other words S is a preferred extension if S is admissible and no supreset of S is admissible.
- Thus $\emptyset$ is not a preferred extension, because $\{p\}$ is admissible.
- Similarly, $\{p,r,s\}$ is admissible because adding q would make it inadmissible.
- A set of arguments always has a preferred extension, but it may be the empty set.

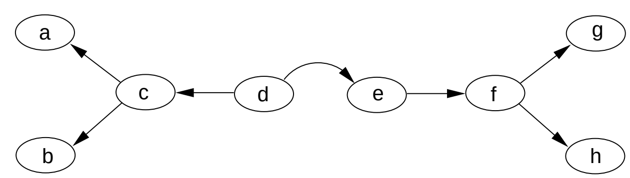
- With a larger set of arguments it is exponentially harder to find the preferred extension.
- n arguments have 2^n possible positions.
- This set of arguments:



has two preferred extensions:

$$\{a,b,d,f\} \qquad \{c,e,g,h\}$$

- In contrast:

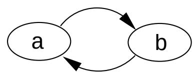


has only one:

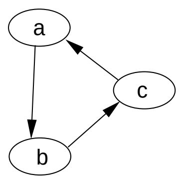
$$\{a,b,d,f\}$$

since c and e are now attacked but undefended, and so can't be in an admissible set.

- Two rather pathological cases are:



with preferred extension $\{a\}$ and $\{b\}$, and:



which has only $\emptyset$ as a preferred extension.

Credulous and sceptical acceptance

- To improve on preferred extensions we can define

  An argument is sceptically accepted if it is a member of *every* preferred extension.

  and

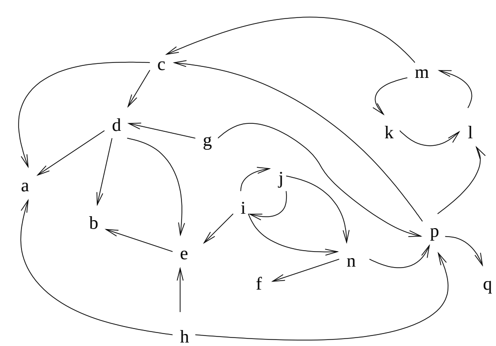
  An argument is credulously accepted if it is a member of *at least one* preferred extension.

- Clearly anything that is sceptically accepted is also credulously accepted.
- On our original example, p, q and r are all sceptically accepted, and q is neither sceptically or credulously accepted.

Grounded extensions

- Another approach, perhaps better than preferred extension.
- Arguments are guaranteed to be acceptable if they aren't attacked.
  – No reason to doubt them
- They are IN
- Once we know which these are, any arguments that they attack must be unacceptable.
- They are OUT — delete them from the graph.
- Now look again for IN arguments. . .
- And continue until the graph doesn't change.
- The set of IN arguments — the ones left in the graph — make up the *grounded extension*.

- Consider computing the grounded extension of:



- We can say that:
  - h is not attacked, so IN.
  - h is IN and attacks a, so a is OUT.
  - h is IN and attacks p, so p is OUT.
  - p is OUT and is the only attacker of q so q is IN.
- There is always a grounded extension, and it is always unique (though it may be empty)

Deductive Argumentation

Basic form of deductive arguments is as follows:

$$Database \vdash (Sentence, Grounds)$$

where:

- $Database$ is a (possibly inconsistent) set of logical formulae;
- $Sentence$ is a logical formula known as the *conclusion*; and
- $Grounds$ is a set of logical formulae such that:
  1. $Grounds \subseteq Database$; and
  2. $Sentence$ can be proved from $Grounds$.

Attack and Defeat

- Argumentation takes into account the relationship between arguments.
- Let (ϕ_1, Γ_1) and (ϕ_2, Γ_2) be arguments from some database Δ . . . Then (ϕ_2, Γ_2) can be defeated (attacked) in one of two ways:
  1. (ϕ_1, Γ_1) *rebuts* (ϕ_2, Γ_2) if $\phi_1 \equiv \neg\phi_2$.
  2. (ϕ_1, Γ_1) *undercuts* (ϕ_2, Γ_2) if $\phi_1 \equiv \neg\psi$ for some $\psi \in \Gamma_2$.
- A rebuttal or undercut is known an *attack*.
- Once we have identified attacks, we can look at preferred extensions or grounded extensions to determine what arguments to accept.

Argumentation and Communication

- We have two agents, P and C, each with some knowledge base, Σ_P and Σ_C.
- Each time one makes an assertion, it is considered to be an addition to its *commitment store*, $CS(P)$ or $CS(C)$.
- Thus P can build arguments from $\Sigma_P \cup CS(C)$, and C can use $\Sigma_C \cup CS(P)$.
- We assume that dialogues start with P making the first move.
- The outcomes, then, are:
  - P generates an argument both classify as IN, or
  - C makes P's argument OUT.
- Can use this for negotiation if the language allows you to express offers.

Argumentation Protocol

- A typical persuasion dialogue would proceed as follows:
  1. P has an acceptable argument (S, p), built from Σ_P, and wants C to accept p.
  2. P asserts p.
  3. C has an argument $(S', \neg p)$.
  4. C asserts $\neg p$.
  5. P cannot accept $\neg p$ and challenges it.
  6. C responds by asserting S'.
  7. P has an argument $(S'', \neg q)$ where $q \in S'$, and challenges q.
  8. ...

Argumentation Protocol II

- This process eventually terminates when

$$\Sigma_P \cup CS(P) \cup CS(C)$$

and

$$\Sigma_C \cup CS(C) \cup CS(P)$$

eventually provide the same set of IN arguments and the agents agree.

- Clearly here we are looking at grounded extensions.

Different dialogues

- Information seeking
  – Tell me if p is true.
- Inquiry
  – Can we prove p?
- Persuasion
  – You're wrong to think p is true.
- Negotiation
  – How do we divide the pie?
- Deliberation
  – Where shall we go for dinner?

Summary

- This lecture has looked at different mechanisms for reaching agreement between agents.
- We started by looking at negotiation, where agents make concessions and explore tradeoffs.
- Finally, we looked at argumentation, which allows for more complex interactions and can be used for a range of tasks that include negotiation.