Autonomous Agents and Multi-Agent Systems* 2015/2016

Lecture Reaching Agreements

Manuel LOPES

* These slides are based on the book by Prof. M. Wooldridge “An Introduction to Multiagent Systems” and the online slides compiled by Professor Jeffrey S. Rosenschein. Modifications introduced by Prof. Ana Paiva, Prof. César Pimentel and Prof. Manuel Lopes
The Haggle - Monty Python's Life Of Brian
The purpose of negotiation is to reach an agreement in the presence of conflicting goals and preferences.

How can negotiation, or bargaining, be formalized and implemented?
The Aim of the Research

- Social engineering for communities of machines
  - The creation of interaction environments that foster certain kinds of social behavior

The exploitation of game theory tools for high-level protocol design
Two Players: The Divider-Chooser Method

(a) Chocolate, Strawberry
(b) 50%
(c) 100%

Cleo
Damian

(a) Cleo
(b) Damian
(c) Cleo
Two Players: The Divider-Chooser Method

(a)  

(b)
Share Rent
Moving into a new apartment with roommates? Create harmony by fairly assigning rooms and sharing the rent.

START >

Split Fare
Fairly split taxi fare, or the cost of an Uber or Lyft ride, when sharing a ride with friends.

START >

Assign Credit
Determine the contribution of each individual to a school project, academic paper, or business endeavor.

START >

Divide Goods

Distribute Tasks

Suggest an App

http://www.spliddit.org/
Organ donor
Reaching Agreements

• How do agents *reaching 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
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

• Given a particular protocol, how can a particular *strategy* be designed that individual agents can use?
Negotiation Parameters

• Negotiation set
  – Space of *possible proposals* for an agent

• Protocol
  – Defines the *legal proposals* as a function of prior negotiation history

• Strategies
  – What *proposals the agents will make* (usually private)

• Rule
  – Determines when an *agreement deal* is reached and what it is
Negotiation Complexity

- **Single-issue negotiation**
  - E.g. two agents *arguing for a price*: Since preferences are symmetric it is obvious what a *concession* is!

- **Multiple-issue negotiation**
  - Values of *multiple attributes* (may be interrelated). E.g. *buying a car* (price, guarantee, extras, etc.)

- **One-to-one negotiation**
  - Simplest case: symmetric preferences

- **Many-to-one negotiation**
  - E.g. auctions
  - Can be treated as concurrent one-to-one negotiations

- **Many-to-many negotiation**
  - Very hard to handle
Individual Rationality

- A protocol satisfies individual rationality if its outcomes are responses to the interests of the negotiation participants.
- A deal is individual rational if its utility is positive or zero for every agent (Intuitively: It’s at least as good as the conflict deal, for every agent!)
Attributes of Standards

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

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

- We focus on the most commonly analyzed:
  - Single-issue
  - Symmetric
  - One-to-one
Convergence

A protocol is said to converge if it guarantees success in the negotiation.
A protocol maximizes the social welfare if the outcome maximizes the sum of the utilities of the negotiation participants.
Pareto Optimal Deal

- A deal is *Pareto optimal* if there is *no alternative* deal that would *increase the utility for one* agent *without decreasing* the utility for some *other* agent.
- Intuitively: *There isn’t an alternative that an agent prefers and the others don’t mind!*
Utilitarian solution

Nash bargaining deal

\[ \delta = \arg \max_{\delta'} \prod u_i(\delta'). \]
How to divide the cake?
How to divide the cake?
An egalitarian solution
How to divide the cake?

Egalitarian
Egalitarian social welfare
Utilitarian
Nash barging deal
A protocol is should lead to *stable outcomes*, i.e., outcomes that *finalize the negotiation* and parties can *no longer change* to other outcomes.
Machines Controlling and Sharing Resources

- Electrical grids (load balancing)
- Telecommunications networks (routing)
- PDA’s (schedulers)
- Shared databases (intelligent access)
- Traffic control (coordination)
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
• 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
Distributed Artificial Intelligence (DAI)

- Distributed Problem Solving (DPS)
  - Centrally designed systems, built-in cooperation, have *global* problem to solve

✓ Multi-Agent Systems (MAS)

✓ Group of utility-maximizing heterogeneous agents co-existing in same environment, possibly competitive
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
Type of Negotiation

- For resource division
- For task allocation
- For resource allocation
Postmen Domain
“All female employees with more than three children.”

“All female employees making over $50,000 a year.”
Fax Domain

Cost is only to establish connection

TOD
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
Task-Oriented Domains

• Agents:
  – Have tasks to carry out
  – May be able to benefit from reorganizing the distribution of tasks

• How to agree on who will do which tasks?
Task-Oriented Domains

• Example:
  – A parent needs to deliver his three children in different schools
  – His neighbor needs to deliver his four children in school as well
  – Some of the schools are the same
  – Both parents can save trips if children that go to the same school are delivered by one of the parents
Task-Oriented Domains

Task-oriented domain:

\[ \langle T, Ag, c \rangle \]

- \( T \) - finite set of all possible tasks
- \( Ag = \{1, ..., n\} \) - finite set of negotiation participant agents
- \( c : 2^T \rightarrow \mathbb{R}_+ \) - function defining the cost of executing each subset of tasks
Cost Function

c has two constraints:

- **Monotonic**
  - Adding tasks never decreases cost

- \( c(\emptyset) = 0 \)
  - The cost of doing nothing is zero
An encounter in $\langle T, Ag, c \rangle$ is:

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

- where each $T_i \subseteq T$ is the set of tasks assigned to agent $i \in Ag$
- We focus on one-to-one scenarios: $Ag = \{1,2\}$
A deal for a one-to-one encounter \( \langle T_1, T_2 \rangle \) is:

\[ \langle D_1, D_2 \rangle \]

- where \( D_1 \cup D_2 = T_1 \cup T_2 \)

- Agent 1 is \textit{committed} to performing tasks \( D_1 \) and agent 2 is \textit{committed} to performing tasks \( D_2 \)
Utility of a Deal

• Consider deal $\delta = \langle D_1, D_2 \rangle$

• The utility of deal $\delta$ for agent $i$ measures the decrease of cost in its assigned tasks:
  $$\text{utility}_i(\delta) = c(T_i) - c(D_i)$$
Conflict Deal

• If the agents \textit{fail} to reach an agreement, they must perform their \textit{originally assigned tasks}!

• This is known as the \textit{conflict deal}:
  \[
  \Theta = \langle T_1, T_2 \rangle
  \]

Notice: \textit{utility}_i(\Theta) = 0 for all \(i \in Ag\)
Negotiation Set

- The space of *possible proposals* that an agent *can make*, i.e. the *negotiation set*, is the set of *deals* that are:
  
  i. **Individual rational**
     
     *(no point in proposing a deal that is worse than the conflict deal, for some agent)*
  
  ii. **Pareto optimal**

     *(no point in making a proposal, if there is a better one, for some agent, at nobody’s expense)*
Negotiation Set

- Deals on this line from B to C are Pareto optimal, hence in the negotiation set.
- This circle delimits the space of all possible deals.
Monotonic Concession Protocol

• Negotiation proceeds in rounds of simultaneous proposals, $\delta_1$ and $\delta_2$, from the negotiation set.

• In every round $u+1$, no agent is allowed to make a proposal that is less preferred, by the other agent, than what it proposed in round $u$. 
Monotonic Concession Protocol

- Negotiation *stops* if either:
  1. There is a round where *no agent makes a concession* (result: *conflict deal*)
  2. *Agreement* is reached when one of the agents *finds the other agent’s proposal at least as good as its own*, i.e.:
     
     \[
     \text{util}_1(\delta_2) \geq \text{util}_1(\delta_1) \\
     \lor \quad \text{util}_2(\delta_1) \geq \text{util}_2(\delta_2)
     \]

*Agreement deal:*
- *The proposal that matches or exceeds* the other’s proposal
- *If both do:* choose one of them *randomly*
Monotonic Concession Protocol - Properties

• Verifiable
  – Both parties see that the rules are being adhered to

• Guaranteed to end
  – Negotiation ends in a finite number of rounds (since number of possible deals is finite)
  – Ends with or without agreement

• Might not be quick!
Strategies

• What to offer?
• When to concede?
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*
Zeuthen Strategy

Q: What should an agent’s first proposal be?
A: Its most preferred deal

Q: On any given round, who should concede?
A: The agent less willing to risk conflict

Q: If an agent concedes, then how much should it concede?
A: Just enough so that it will no longer be the less willing agent to risk conflict
Zeuthen Strategy

• Intuitively, an agent is more willing to risk conflict if:
  – The difference, in utility, between its current proposal and the conflict deal is low (not much to lose in conflict)
  – The difference, in utility, between its current proposal and the other agent’s proposal is high (a lot to lose with conceding)
Zeuthen Strategy

- Agent $i$’s **willingness to risk conflict** at round $t$:

$$risk_i^t = \frac{utility\ i\ loses\ by\ accepting\ j’s\ offer}{utility\ i\ loses\ by\ causing\ conflict}$$

$$1 - \frac{utility_i(\delta_j^t)}{utility_i(\delta_i^t)}$$
Zeuthen Strategy

• Agent $i$’s *willingness to risk conflict* at round $t$:

\[
\text{risk}_i^t = \begin{cases} 
1, & \text{if } \text{utility}_i(\delta_i^t) = 0 \\
\frac{\text{utility}_i(\delta_i^t) - \text{utility}_i(\delta_j^t)}{\text{utility}_i(\delta_i^t)}, & \text{otherwise}
\end{cases}
\]
Zeuthen Strategy

• First proposal:
  – The agent’s *most preferred* deal

• In each round:
  – The agent *less willing to risk* conflict *should concede* (if equal, flip a coin)
  – It should concede *just enough* so that it will *no longer be the less willing agent to risk* conflict
    • If less: has to concede again (inefficient)
    • If more: wastes utility
Monotonic Concession Protocol + Zeuthen Strategy

• Does *not guarantee success* but guarantees *termination*

• Does *not guarantee to maximize social welfare* but guarantees that *if success is reached, it is Pareto optimal*

• It is *individual rational*
Monotonic Concession Protocol + Zeuthen Strategy

- Does *not require arbiter*
- *High computational cost*
- Zeuthen strategy is in *Nash equilibrium* (assuming one agent uses the strategy, the other can do no better than to use it as well)
Divide the following cakes using the Zeuthen Strategy

<table>
<thead>
<tr>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Deception

- **Phantom and decoy tasks**
  Agents can *pretend to have been allocated* a task or *produce artificial tasks* to manipulate the outcome of the negotiation.

- **Hidden tasks**
  Agents can *hide allocated tasks* that would otherwise take the negotiation to undesirable outcomes.
Negotiation with Incomplete Information

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

Post Office

Envelope

Mailboxes: h, a, b, g, f, e, d

1

2
-1 Phase Game: Broadcast

Tasks

Agents will flip a coin to decide who delivers all the letters.
Hiding Letters

They then agree that agent 2 delivers to f and e.
Another Possibility for Deception

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

Post Office

They agree that agent 1 goes to c
Conclusions

• By appropriately adjusting the *rules of encounter* by which agents must interact, we can influence the private strategies that designers build into their machines.

• The interaction mechanism should ensure the *efficiency* of multi-agent systems.
Conclusions

• To maintain efficiency over time of dynamic multi-agent systems, the rules must also be stable

• The use of formal tools enables the design of efficient and stable mechanisms, and the precise characterization of their properties