Multiagent Systems

13. Bargaining

B. Nebel, C. Becker-Asano, S. Wölfl

Albert-Ludwigs-Universität Freiburg

July 16, 2014

Where are we?

- Different auction types and properties
- Combinatorial Auctions
- Bidding Languages
- The VCG mechanism

Today ...

- Bargaining

General setting

Bargaining

- **Aim**: Reaching agreement in the presence of conflicting goals and preferences
  (e.g., distribution of goods, prize of a good, political agreements, meeting place)
- . . . similar to a multi-step game with specific protocol
- General setting for bargaining/negotiation:
  - The **negotiation set** is the space of possible proposals
  - The **protocol** defines the proposals the agents can make, as a function of prior negotiation history
  - **Strategies** determine the proposals the agents will make (private)
  - A rule that determines when a deal has been struck (**agreement deal**)

Negotiation scenarions

- Number of issues:
  - **Single issue**, e.g. price of a good
  - **Multiple issues**, e.g. buying a car: price, extras, service
  - Concessions may be hard to identify in multiple-issue negotiations
  - Number of possible deals: m^n for n attributes with m possible values
- Number of agents:
  - **one-to-one**, simplified when preferences are symmetric
  - **many-to-one**, e.g. auctions
  - **many-to-many**, $n(n-1)/2$ negotiation threads for n agents

Conditions on negotiation protocols

Implementing negotiation in MAS needs interaction protocols.

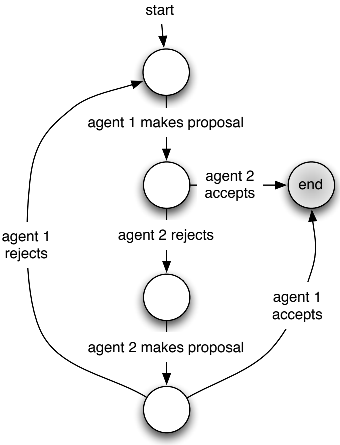
What are **good** protocols?

- **Efficiency**: Agreed solution does not waste utility (e.g., is Pareto optimal or maximizes social welfare)
- **Stability**: In the agreed-upon solution no agent has an incentive to deviate (Nash equilibrium)
- **Simplicity**: Required interaction according to the protocol has low computational overhead (e.g. for communication, determining optimal behavior)
- **Distribution**: Protocol does not require a central decision maker
- **Symmetry**: Negotiation process should not be biased against or towards one of the agents
- **Effectiveness**: When possible, agreement should be reachable, when all agents follow the protocol

Division of Resources

Alternating offers

A common one-to-one protocol: **alternating offers**



- Negotiation takes place in a sequence of rounds
- Agent 1 begins at round 0 by making a proposal x^0
- Agent 2 can either accept or reject the proposal
- If the proposal is accepted the deal x^0 is implemented
- Otherwise, negotiation moves to the next round where agent 2 makes a proposal

Example: Dividing the Pie

Scenario: Dividing the pie

- There is some resource whose value is 1
- The resource can be divided into two parts, such that the values of each part must be between 0 and 1
  the sum of the values of the parts sum to 1
- A **proposal** is a pair $(x, 1-x)$ (meaning: agent 1 gets x, agent 2 gets $1-x$)
- The **negotiation set** is: $\{(x, 1-x) \colon 0 \leq x \leq 1\}$

Example: Dividing the Pie

Scenario: Dividing the pie

- There is some resource whose value is 1
- The resource can be divided into two parts, such that the values of each part must be between 0 and 1
  the sum of the values of the parts sum to 1
- A **proposal** is a pair $(x, 1 - x)$ (meaning: agent 1 gets x, agent 2 gets $1 - x$)
- The **negotiation set** is: $\{(x, 1 - x) \colon 0 \leq x \leq 1\}$

Some assumptions:

- Disagreement is the worst outcome, we call this the **conflict deal** Θ
- Agents seek to maximize utility

Negotiation rounds

- Special case 1: one single negotiation round ($\rightsquigarrow$ **ultimatum game**)
  - Suppose that player 1 proposes to get all the pie, i.e. $(1, 0)$
  - Player 2 will have to agree to avoid getting the conflict deal Θ
  - Player 1 has all the power

Negotiation rounds

- Special case 1: one single negotiation round ($\rightsquigarrow$ **ultimatum game**)
  - Suppose that player 1 proposes to get all the pie, i.e. $(1, 0)$
  - Player 2 will have to agree to avoid getting the conflict deal Θ
  - Player 1 has all the power
- Special case 2: Two rounds of negotiation
  - Player 1 makes a proposal in the first round
  - Player 2 can reject and turn the game into an ultimatum

Negotiation rounds

- Special case 1: one single negotiation round ($\rightsquigarrow$ **ultimatum game**)
  - Suppose that player 1 proposes to get all the pie, i.e. $(1, 0)$
  - Player 2 will have to agree to avoid getting the conflict deal Θ
  - Player 1 has all the power
- Special case 2: Two rounds of negotiation
  - Player 1 makes a proposal in the first round
  - Player 2 can reject and turn the game into an ultimatum
- More generally: If the number of rounds is **fixed**, whoever moves last gets all the pie …

Negotiation rounds

- If there are **no** bounds on the number of rounds:
  - Suppose agent 1's strategy is: propose $(1, 0)$, reject any other offer

Negotiation rounds

- If there are **no** bounds on the number of rounds:
  - Suppose agent 1's strategy is: propose $(1, 0)$, reject any other offer
  - If agent 2 rejects the proposal, the agents will never reach agreement (the conflict deal is enacted)
  - Agent 2 will have to accept to avoid Θ

Negotiation rounds

- If there are **no** bounds on the number of rounds:
  - Suppose agent 1's strategy is: propose $(1, 0)$, reject any other offer
  - If agent 2 rejects the proposal, the agents will never reach agreement (the conflict deal is enacted)
  - Agent 2 will have to accept to avoid Θ
  - Infinite set of Nash equilibrium outcomes (of course agent 2 must understand the situation, e.g. given access to agent 1's strategy)

Time

- Additional assumption: Time is valuable (agents prefer outcome x at time t_1 over outcome x at time t_2 if $t_2 > t_1$).
- Model agent i's patience using a **discount factor** δ_i $(0 \leq \delta_i \leq 1)$:
  the value of slice x at time 0 is $\delta_i^0 \cdot x = x$
  the value of slice x at time 1 is $\delta_i^1 \cdot x = \delta_i \cdot x$
  the value of slice x at time 2 is $\delta_i^2 \cdot x = \delta_i \cdot \delta_i \cdot x$

Time

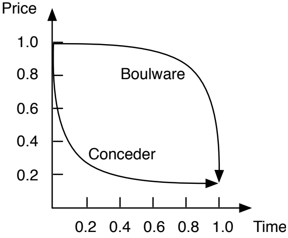
- Additional assumption: Time is valuable (agents prefer outcome x at time t_1 over outcome x at time t_2 if $t_2 > t_1$).
- Model agent i's patience using a **discount factor** δ_i $(0 \leq \delta_i \leq 1)$:
  the value of slice x at time 0 is $\delta_i^0 \cdot x = x$
  the value of slice x at time 1 is $\delta_i^1 \cdot x = \delta_i \cdot x$
  the value of slice x at time 2 is $\delta_i^2 \cdot x = \delta_i \cdot \delta_i \cdot x$

Interesting results:

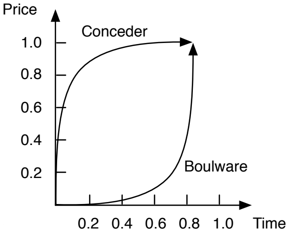
- More patient players (larger δ_i) have more power
- Games with two rounds of negotiation:
  - The best possible outcome for agent 2 in the second round is δ_2
  - If agent 1 initially proposes $(1 - \delta_2, \delta_2)$, agent 2 can do no better than accept
- Games with no bounds on the number of rounds
  - Agent 1 proposes what agent 2 can enforce in the second round

Negotiation Decision Functions

- Non-strategic approach, does not depend on how other's behave
- Agents use a time-dependent decision function to determine what proposal they should make
- **Boulware strategy**: exponentially decay offers to reserve price
- **Conceder strategy**: make concessions early, do not concede much as negotiation progresses



Seller strategies



Buyer strategies

Task Allocation

Task-oriented domains

To model the negotiation for re-allocating tasks we consider so-called **task-oriented domains** (Rosenschein & Zlotkin, 1994).

Simplifying assumptions:

- Each agent has a given set of tasks she has to achieve
- Tasks are indivisible units,
- ... can be carried out without interference from other agents, and
- ... all necessary resources are available
- Agents can redistribute their tasks by negotiation (thus improving their utility)
- TODs are inherently cooperative

Task-oriented domains (I)

Task-oriented domain

A **task-oriented domain** (TOD) is a triple $\langle T, Ag, c\rangle$ where:

- T a finite set of **tasks**,
- $Ag = \{1, \ldots, n\}$ is a set of **agents**, and
- $c\colon 2^T \to \mathbb{R}_0^+$ is function describing the **cost** of executing any set of tasks (symmetric for all agents) such that $c(\emptyset) = 0$, and that c is monotonic i.e.

$$T', T'' \subseteq T \text{ and } T' \subseteq T'' \implies c(T') \leq c(T'').$$

An **encounter** in a TOD is a collection $(T_1, \ldots, T_n)$ with $T_i \subseteq T$ for each agent $i \in Ag$ (T_i is the set of tasks to be performed by agent i).

Task allocation: An example

The Postmen Domain

Several postmen have to deliver letters to mailboxes located in the same neighborhood, and then return to the post office.

Representation: The addresses on the letters are represented by the node set of a weighted graph $G = \langle V, E \rangle$, where the weights on edges represent distances between neighbored mailboxes.

Task set: Each task is given by a address (i.e., deliver at least one letter to the address); hence the set of all tasks is V.

Costs: The cost of $X \subseteq V$ is the length of the shortest path starting in the post office, visiting all nodes in V, and ending in the post office.

Task-oriented domains (II)

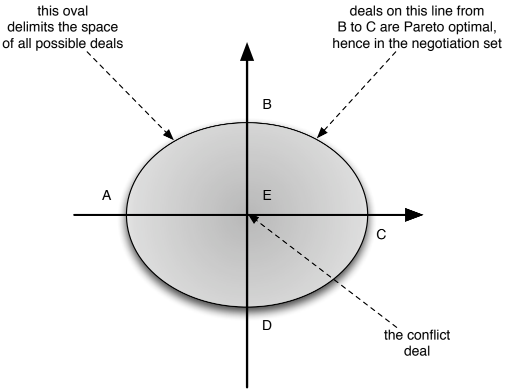
Following, we only consider encounters in two-agent TODs. A **deal** is a pair $\delta = (D_1, D_2)$ such that $D_1 \cup D_2 = T_1 \cup T_2$ (agent i is committed to perform tasks D_i in such a deal). Def. $\mathit{cost}_i(\delta) := c(D_i)$, and $\mathit{util}_i(\delta) := c(T_i) - \mathit{cost}_i(\delta)$.

- Utility represents how much agent gains from the deal
- If no agreement is reached, **conflict deal** is $\Theta = (T_1, T_2)$
- A deal δ_1 **dominates** another deal δ_2 (symb. $\delta_1 > \delta_2$) if δ_1 is at least as good as δ_2 for every agent (i.e. $\mathit{util}_i(\delta_1) \geq \mathit{util}_i(\delta_2)$, for $i = 1, 2$) and better for at least some agent (i.e. $\mathit{util}_i(\delta_1) > \mathit{util}_i(\delta_2)$, for $i = 1$ or $i = 2$)
- If δ is not dominated by any other δ', then δ is called **Pareto optimal**.
- A deal is **individual rational** if it weakly dominates (i.e. is at least as good as) the conflict deal Θ.

Negotiation sets

Negotiation set: set of deals that are individual rational and Pareto-optimal.

- Each agent can guarantee to get utility 0 (by always rejecting). Rational agent will not accept deals with negative utility.
- Agreeing on not Pareto-optimal deals is inefficient.



The monotonic concession protocol

- Start with simultaneous deals proposed by both agents (i.e., a pair of deals (δ_1, δ_2)) and proceed in rounds
- Agreement reached if

$$\text{either } \mathit{util}_1(\delta_2) \geq \mathit{util}_1(\delta_1) \text{ or } \mathit{util}_2(\delta_1) \geq \mathit{util}_2(\delta_2)$$

- If both proposals match or exceed other's offer, outcome is chosen at random between δ_1 and δ_2.
- If no agreement, in round $t + 1$ agents are not allowed to make deals less preferred by other agent than proposal made in round t.
- If no proposals are made or both do not concede, negotiation terminates with outcome Θ.

Protocol is verifiable and guaranteed to terminate, but not necessarily efficient (exponential in the number of tasks that are to allocated).

The Zeuthen strategy (I)

- The above protocol doesn't describe **when** and **how much** to concede
- Intuitively, agents will be more willing to risk conflict if difference between current proposal and conflict deal is low
- Model how much agent i's is willing to risk a conflict at round t by sticking to her last proposal:

$$\mathit{risk}_i^t = \frac{\text{utility lost by conceding and accepting } j\text{'s offer}}{\text{utility lost by not conceding and causing conflict}}$$

- Formally, we can calculate risk as a value between 0 and 1:

$$\mathit{risk}_i^t = \begin{cases} 1 & \text{if } \mathit{util}_i(\delta_i^t) = 0 \\ \dfrac{\mathit{util}_i(\delta_i^t) - \mathit{util}_i(\delta_j^t)}{\mathit{util}_i(\delta_i^t)} & \text{otherwise} \end{cases}$$

The Zeuthen strategy (II)

Zeuthen strategy

1. Start negotiation by proposing a deal that is best for you among all deals in the negotiation set.
2. In every following round t calculate $risk_i^t$ for you and opponent. If your risk is smaller or equal to the other's risk value, propose a deal with minimal concession such that the balance of risk is changed.

- Problem if agents have equal risk: we have to flip a coin, otherwise one of them could defect (and conflict would occur)
- Looking at our protocol criteria:
  Protocol terminates, doesn't always succeed, simplicity? (too many deals), Zeuthen strategies are Nash, no central authority needed, individual rationality (in case of agreement), Pareto optimality

Resource Allocation

Bargaining for resource allocation (I)

Resource allocation setting

A **resource allocation setting** is a tuple $\langle Ag, \mathcal{Z}, v_1, \ldots, v_n \rangle$, with:

- agents $Ag = \{1, \ldots, n\}$,
- resources $\mathcal{Z} = \{z_1, \ldots, z_m\}$,
- valuation functions $v_i \colon 2^{\mathcal{Z}} \to \mathbb{R}$ (one for each agent)

An **allocation** is a partition $(Z_1, \ldots, Z_n)$ of the resources over the agents.

Idea: Starting from some initial allocation $P^0 = (Z_1^0, \ldots, Z_n^0)$ agents can bargain to improve the value of package of resources assigned to them.

Negotiating a change from Z_i to Z_i' ($Z_i, Z_i' \subseteq \mathcal{Z}$ and $P_i \neq Q_i$) will lead to:

- $v_i(Z_i) < v_i(Z_i')$, $v_i(Z_i) = v_i(Z_i')$, or $v_i(Z_i) > v_i(Z_i')$

Bargaining for resource allocation (II)

Agents can make **side payments** as compensation for loss in utility: $p_i < 0$ means that agent i receives $-p_i$; $p_i > 0$ means that i contributes p_i to the amount that is distributed among the agents with negative pay-off.

- A **pay-off vector** is a tuple $p = (p_1, p_2, \ldots, p_n)$ of side payments such that $\sum_i p_i = 0$.
- A **deal** is a triple $\langle Z, Z', p\rangle$, where $Z, Z' \in \mathit{alloc}(\mathcal{Z}, \mathit{Ag})$ are distinct allocations and p is a pay-off vector.
- A deal $\langle Z, Z', p\rangle$ is **individually rational** if

$$v_i(Z'_i) - p_i > v_i(Z)$$

  for each $i \in \mathit{Ag}$ (p_i is allowed to be 0 if $Z_i = Z'_i$).
- **Pareto-optimal allocation**: every other allocation that makes some agents strictly better off makes some other agent strictly worse off

Protocol for resource allocation

Resource allocation

1. Start with initial allocation Z^0.
2. **Current allocation** is Z^0 with 0 side payments.
3. Any agent is permitted to put forward a deal $\langle Z, Z', p \rangle$ where Z is the current allocation.
4. If all agents agree and the **termination condition** is satisfied (i.e. Pareto optimality), then the negotiation terminates and deal Z' is implemented with payments p.
5. If all agents agree but the termination condition is not satisfied, then set current allocation to Z' with payments p and continue in step 3.
6. If some agent is not satisfied with the deal, go to step 3.

Restricted deals

Finding optimal deals is NP-hard, focus on restricted deals

- **One-contracts**: move only one resource and one side payment
  - Restricts search space, agent needs to consider $|Z_i| \cdot (n-1)$ deals
  - Can always lead to socially optimal outcome, but requires agents to accept deals that are not individually rational
- **Cluster-contracts**: transfer of any number of resources greater than 1 from one agent to another one (do not receive any resources in return)
- **Swap-contracts**: swap one resource and make side payment
- **Multiple-contracts**: three agents, each transferring a single resource

- C-contracts, S-contracts and M-contracts do not always lead to an optimal allocation

Summary

Summary

- Bargaining
- Alternating offers
- Negotiation decision functions
- Task-oriented domains
- Bargaining for resource allocation

- Next time: **Argumentation in Multiagent Systems**

Acknowledgments

These lecture slides are based on the following resources:

- Dr. Michael Rovatsos, The University of Edinburgh `http://www.inf.ed.ac.uk/teaching/courses/abs/abs-timetable.html`
- Michael Wooldridge: **An Introduction to MultiAgent Systems**, John Wiley & Sons, 2nd edition 2009.
- Jeffrey Rosenschein and Gilad Zlotkin: **Rules of Encounter**, PIT Press, 1994, 1998.