PAKDD 2025
Sydney 10-13 June

Developing Data Driven Automated Negotiation Agents

Yasser Mohammad
Data Science Labs
NEC

May 25, 2025

Outline

Timeline

- 8:30 - 9:00: Automated Negotiation
- 9:00 - 9:30: Learning to Offer
- 9:30 - 9:45: Learning to Accept
- 9:45 - 10:30: Preference Learning (Opponent Modeling)

More Information

- Tutorial website: `http://www.yasserm.com/tutorials/pakdd.html`
- NegMAS Documentation: `https://negmas.readthedocs.io/en/latest/`

Table of Contents

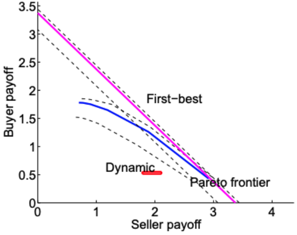
Automated Negotiation

Motivation

- Negotiation is ubiquitous in societal and business interactions.
- Increased utilization of AI agents in businesses → higher need and opportunity for automating negotiation.
- People are not very good at negotiation. Market studies routinely reveal somewhere between 9% to 20% value loss due to negotiation inefficiencies..

Automated negotiation can lead to

- **Better** agreements → Less money on the table.
- **Faster** agreements → More dynamic markets.
- **New** Opportunities → New applications.



9-20% loss based on 27,000 negotiations[1]

[1] Bradley J Larsen. "The efficiency of real-world bargaining: Evidence from wholesale used-auto auctions". In: *The Review of Economic Studies* 88.2 (2021), pp. 851–882

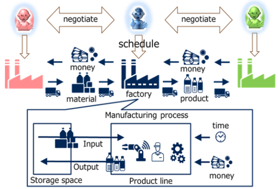
ML Motivation: For Automated Negotiation Researchers

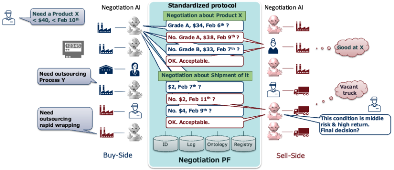
The need

- No general-purpose AN solution.
- No known equilibrium for bargaining with incomplete information [2].

The opportunity

- ML has been successfully applied to various domains, including games, robotics, and finance.
- Several AN subproblems can be cast as ML problems.





[2] Except in a vanishingly small set of scenarios.

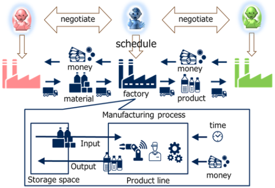
ML Motivation: For Automated Negotiation Researchers

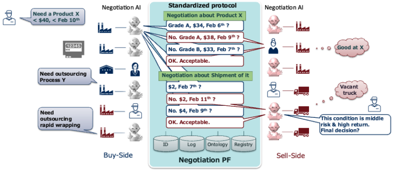
The need

- No general-purpose AN solution.
- No known equilibrium for bargaining with incomplete information [2].

The opportunity

- ML has been successfully applied to various domains, including games, robotics, and finance.
- Several AN subproblems can be cast as ML problems.





[2] Except in a vanishingly small set of scenarios.

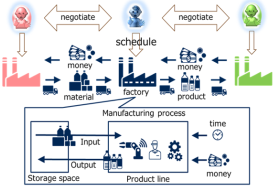
ML Motivation: For Automated Negotiation Researchers

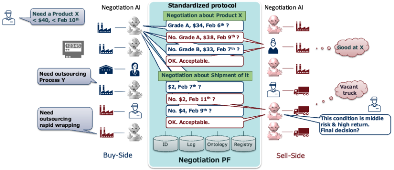
The need

- No general-purpose AN solution.
- No known equilibrium for bargaining with incomplete information [2].

The opportunity

- ML has been successfully applied to various domains, including games, robotics, and finance.
- Several AN subproblems can be cast as ML problems.





---

[2] Except in a vanishingly small set of scenarios.

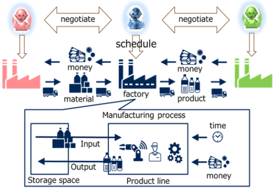
ML Motivation: For Automated Negotiation Researchers

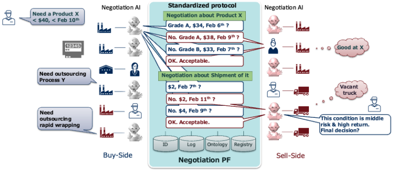
The need

- No general-purpose AN solution.
- No known equilibrium for bargaining with incomplete information [2].

The opportunity

- ML has been successfully applied to various domains, including games, robotics, and finance.
- Several AN subproblems can be cast as ML problems.



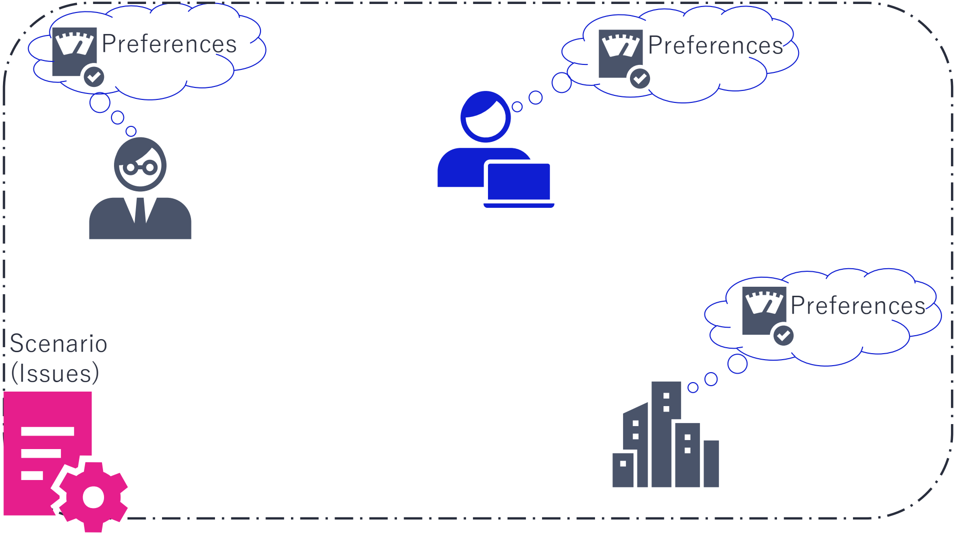


[2] Except in a vanishingly small set of scenarios.

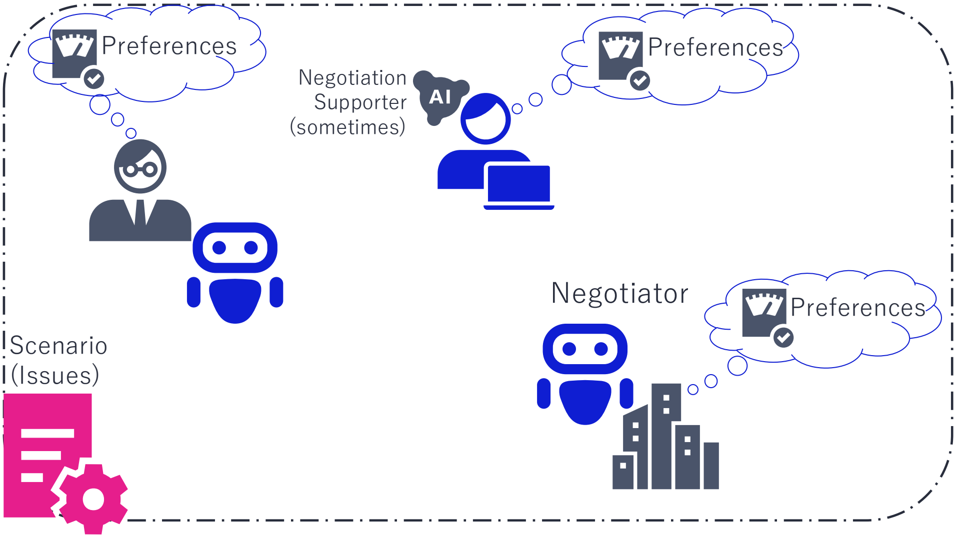
What is negotiation?

What is negotiation?

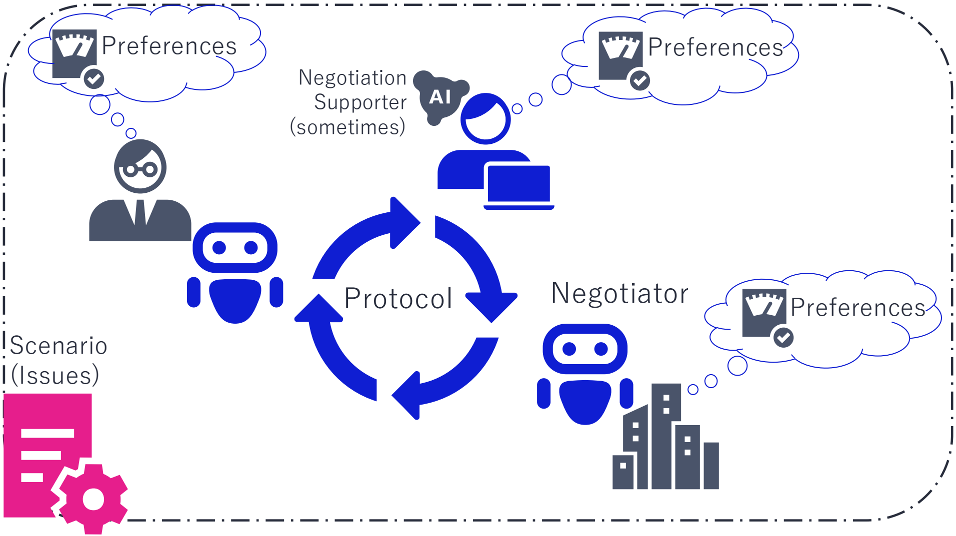
What is negotiation?



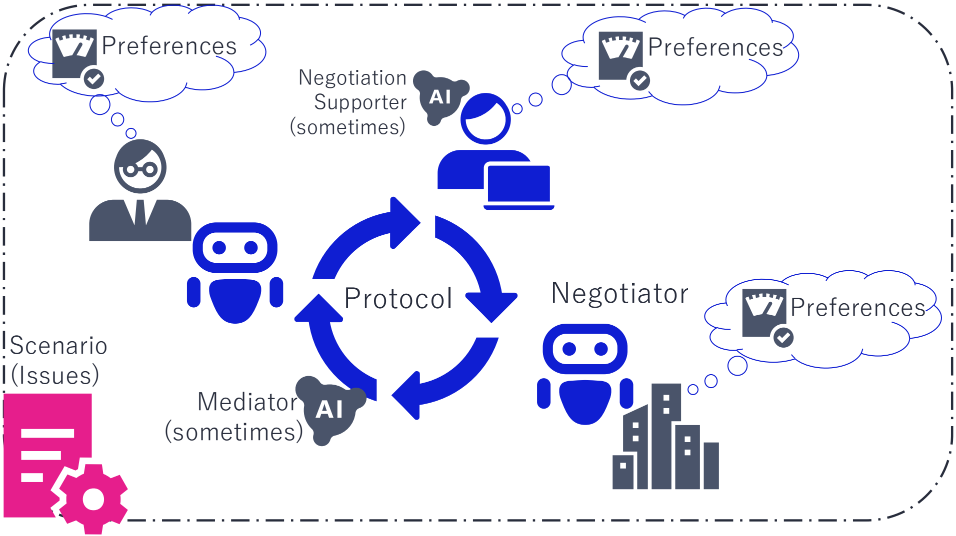
What is negotiation?



What is negotiation?



What is negotiation?



When do we need to negotiate?

- More than one actor (multiagent system).
- Actors have different **interests** represented by different **preferences**.
- There is a boundary: Actors cannot share information freely.
- Each actor thinks it may benefit from an agreement with the others.
- Actors can agree on a protocol and agenda for the negotiation.

When do we need to negotiate?

- More than one actor (multiagent system).
- Actors have different **interests** represented by different **preferences**.
- There is a boundary: Actors cannot share information freely.
- Each actor thinks it may benefit from an agreement with the others.
- Actors can agree on a protocol and agenda for the negotiation.

When do we need to negotiate?

- More than one actor (multiagent system).
- Actors have different **interests** represented by different **preferences**.
- There is a boundary: Actors cannot share information freely.
- Each actor thinks it may benefit from an agreement with the others.
- Actors can agree on a protocol and agenda for the negotiation.

When do we need to negotiate?

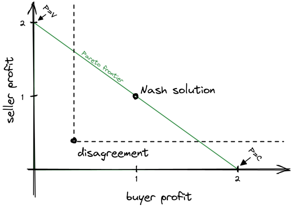
- More than one actor (multiagent system).
- Actors have different **interests** represented by different **preferences**.
- There is a boundary: Actors cannot share information freely.
- Each actor thinks it may benefit from an agreement with the others.
- Actors can agree on a protocol and agenda for the negotiation.

When do we need to negotiate?

- More than one actor (multiagent system).
- Actors have different **interests** represented by different **preferences**.
- There is a boundary: Actors cannot share information freely.
- Each actor thinks it may benefit from an agreement with the others.
- Actors can agree on a protocol and agenda for the negotiation.

A Simple Trading Problem

- A buyer values a good at V
- A seller can create the good at cost C
- If $V > C$, then there is surplus $V - C$ to be gained (value creation)
- **Bargaining problem:** how much should the buyer pay the seller for the good? (value division)
- We might also assume there is an outside option (e.g., eBay), if the negotiation breaks down (i.e., they do not reach an agreement):
  - The buyer (seller) can buy (sell) the good elsewhere for slightly less than V (more than C)



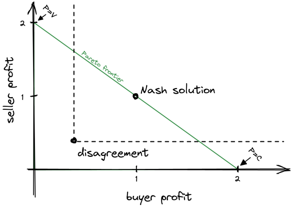
Sketch by Jackson de Campos

Slide by Amy Greenwald

A Simple Trading Problem

- A buyer values a good at V
- A seller can create the good at cost C
- If $V > C$, then there is surplus $V - C$ to be gained (value creation)
- **Bargaining problem:** how much should the buyer pay the seller for the good? (value division)

- We might also assume there is an outside option (e.g., eBay), if the negotiation breaks down (i.e., they do not reach an agreement):
  - The buyer (seller) can buy (sell) the good elsewhere for slightly less than V (more than C)

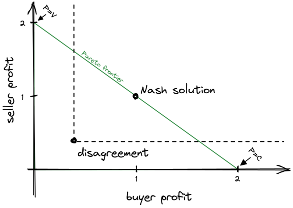


Sketch by Jackson de Campos

Slide by Amy Greenwald

A Simple Trading Problem

- A buyer values a good at V
- A seller can create the good at cost C
- If $V > C$, then there is surplus $V - C$ to be gained (value creation)
- **Bargaining problem:** how much should the buyer pay the seller for the good? (value division)
- We might also assume there is an outside option (e.g., eBay), if the negotiation breaks down (i.e., they do not reach an agreement):
  - The buyer (seller) can buy (sell) the good elsewhere for slightly less than V (more than C)



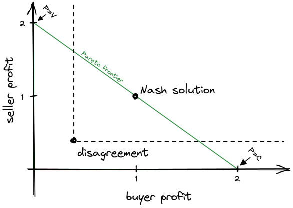
Sketch by Jackson de Campos

Slide by Amy Greenwald

A Simple Trading Problem

- A buyer values a good at V
- A seller can create the good at cost C
- If $V > C$, then there is surplus $V - C$ to be gained (value creation)
- **Bargaining problem:** how much should the buyer pay the seller for the good? (value division)

- We might also assume there is an outside option (e.g., eBay), if the negotiation breaks down (i.e., they do not reach an agreement):
  - The buyer (seller) can buy (sell) the good elsewhere for slightly less than V (more than C)



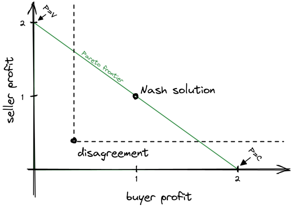
Sketch by Jackson de Campos

Slide by Amy Greenwald

A Simple Trading Problem

- A buyer values a good at V
- A seller can create the good at cost C
- If $V > C$, then there is surplus $V - C$ to be gained (value creation)
- **Bargaining problem:** how much should the buyer pay the seller for the good? (value division)

- We might also assume there is an outside option (e.g., eBay), if the negotiation breaks down (i.e., they do not reach an agreement):
  - The buyer (seller) can buy (sell) the good elsewhere for slightly less than V (more than C)

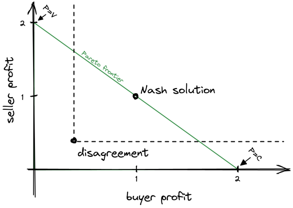


Sketch by Jackson de Campos

Slide by Amy Greenwald

A Simple Trading Problem

- A buyer values a good at V
- A seller can create the good at cost C
- If $V > C$, then there is surplus $V - C$ to be gained (value creation)
- **Bargaining problem:** how much should the buyer pay the seller for the good? (value division)



Sketch by Jackson de Campos

Slide by Amy Greenwald

- We might also assume there is an outside option (e.g., eBay), if the negotiation breaks down (i.e., they do not reach an agreement):
  - The buyer (seller) can buy (sell) the good elsewhere for slightly less than V (more than C)

Abstract Bargaining Problem

The two-agent bargaining problem can be defined abstractly by

- A set $F \subset \Omega$ of feasible outcomes
- Two agents with utility functions $u_1, u_2 : \Omega \to \mathbb{R}$
- A disagreement point $\phi \in \Omega$, also called the status quo. The value $u_i(\phi)$ is called agent i's reservation value.

Individual rationality assumption: No agent will ever agree to a utility below their reservation value.

An efficient outcome is one on the Pareto frontier, where neither agent can be made strictly better off without making the other worse off

Challenge: We seek a cooperative outcome (i.e., an efficient one) in a non-cooperative game

Abstract Bargaining Problem

The two-agent bargaining problem can be defined abstractly by

- A set $F \subset \Omega$ of feasible outcomes
- Two agents with utility functions $u_1, u_2 : \Omega \to \mathbb{R}$
- A disagreement point $\phi \in \Omega$, also called the status quo. The value $u_i(\phi)$ is called agent i's reservation value.

Individual rationality assumption: No agent will ever agree to a utility below their reservation value.

An efficient outcome is one on the Pareto frontier, where neither agent can be made strictly better off without making the other worse off

Challenge: We seek a cooperative outcome (i.e., an efficient one) in a non-cooperative game

Abstract Bargaining Problem

The two-agent bargaining problem can be defined abstractly by

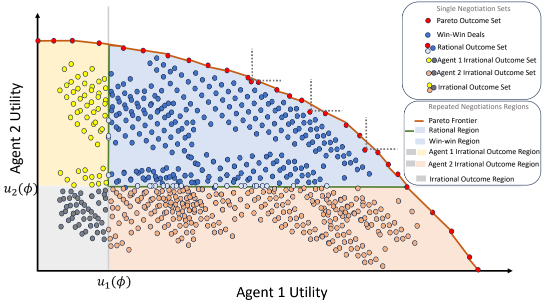
- A set $F \subset \Omega$ of feasible outcomes
- Two agents with utility functions $u_1, u_2 : \Omega \rightarrow \mathbb{R}$
- A disagreement point $\phi \in \Omega$, also called the status quo. The value $u_i(\phi)$ is called agent i's reservation value.

Individual rationality assumption: No agent will ever agree to a utility below their reservation value.

An efficient outcome is one on the Pareto frontier, where neither agent can be made strictly better off without making the other worse off

Challenge: We seek a cooperative outcome (i.e., an efficient one) in a non-cooperative game

Important Concepts



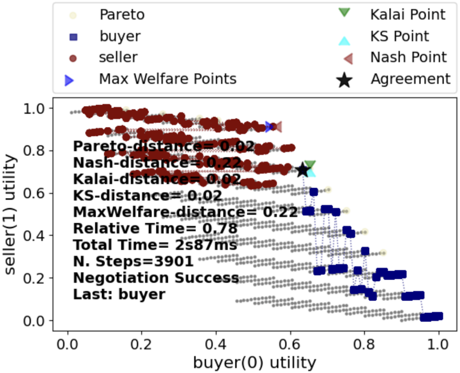
Pareto Frontier Outcomes that cannot be improved for one actor without making another worse off.

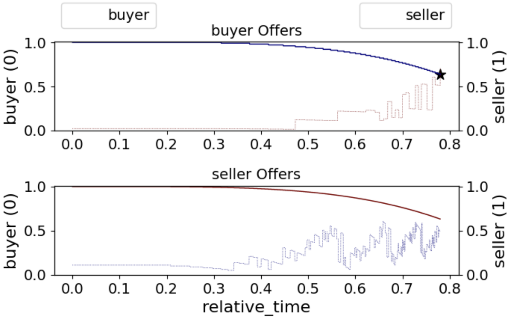
Welfare Total utility received by all actors.

Surplus utility Utility above disagreement utility.

Nash Equilibrium Strategies that are best responses to each other.

Visualizing a Negotiation

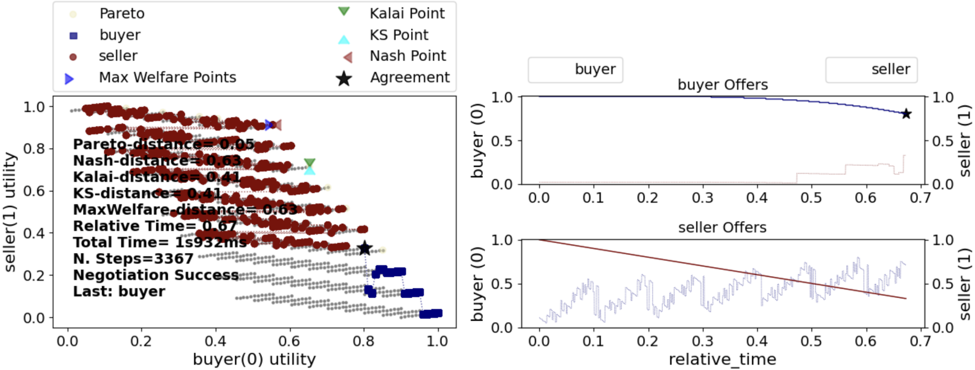




A buyer and a seller negotiating price, quantity and delivery date.

- Is this a zero-sum game?
- What are the reservation values?
- Is this a **good** result?

Visualizing a Negotiation



A buyer and a seller negotiating price, quantity and delivery date.

- Is this a zero-sum game?
- What are the reservation values?
- Is this a **good** result?

von Neumann-Morgenstern Utility Theorem

It is natural to express agent's preferences as comparisons: e.g., "I prefer apples to bananas."
We can also compare lotteries (i.e., randomized outcomes): e.g., when I am very hungry, "I prefer a banana with probability 90% to an apple with probability 50%."

Theorem[3] Given an agent with preferences over randomized outcomes that satisfy various axioms,
there exists a unique utility function $u : \Omega \to \mathbb{R}$ s.t. $\sigma \succ \tau$ iff $\mathbb{E}[u(\sigma)] > \mathbb{E}[u(\tau)]$), up to scaling.
The axioms are completeness, transitivity, continuity, and IIA.

Why is this relevant?

- Justifies focusing on bargaining assuming utility functions (hereafter, ufuns).
- Justifies modelling the preferences of negotiation partners (hereafter, opponents) via ufuns.

J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947

Nash Bargaining Game: Description

A single-step full-information bilateral negotiation with $\Omega = [0,1]^2$ and two utility functions $(\tilde{u}_1, \tilde{u}_2)$ such that:

- A feasible set of agreements F. A common example is to define F as all the outcomes for which the total utility received by negotiators is less than or equal to one:

$$F = \{(\omega_1, \omega_2) | \tilde{u}_2(\omega_2) + \tilde{u}_1(\omega_1) \leq 1\}.$$

- A disagreement point $d \equiv \tilde{u}_1(\phi) + \tilde{u}_2(\phi) \in \Re^2$ which is the utility value received by the two players in case of disagreement (reserved values).

Nash Bargaining Game: Description

A single-step full-information bilateral negotiation with $\Omega = [0, 1]^2$ and two utility functions $(\tilde{u}_1, \tilde{u}_2)$ such that:

- A feasible set of agreements F. A common example is to define F as all the outcomes for which the total utility received by negotiators is less than or equal to one:

$$F = \{(\omega_1, \omega_2) | \tilde{u}_2(\omega_2) + \tilde{u}_1(\omega_1) \leq 1\}.$$

- A disagreement point $d \equiv \tilde{u}_1(\phi) + \tilde{u}_2(\phi) \in \Re^2$ which is the utility value received by the two players in case of disagreement (reserved values).

Nash Bargaining Game: Description

A single-step full-information bilateral negotiation with $\Omega = [0,1]^2$ and two utility functions $(\tilde{u}_1, \tilde{u}_2)$ such that:

- A feasible set of agreements F. A common example is to define F as all the outcomes for which the total utility received by negotiators is less than or equal to one:

$$F = \{(\omega_1, \omega_2) | \tilde{u}_2(\omega_2) + \tilde{u}_1(\omega_1) \leq 1\}.$$

- A disagreement point $d \equiv \tilde{u}_1(\phi) + \tilde{u}_2(\phi) \in \Re^2$ which is the utility value received by the two players in case of disagreement (reserved values).

Nash Bargaining Game: Description

A single-step full-information bilateral negotiation with $\Omega = [0,1]^2$ and two utility functions $(\tilde{u}_1, \tilde{u}_2)$ such that:

- A feasible set of agreements F. A common example is to define F as all the outcomes for which the total utility received by negotiators is less than or equal to one:

$$F = \{(\omega_1, \omega_2) | \tilde{u}_2(\omega_2) + \tilde{u}_1(\omega_1) \leq 1\}.$$

- A disagreement point $d \equiv \tilde{u}_1(\phi) + \tilde{u}_2(\phi) \in \Re^2$ which is the utility value received by the two players in case of disagreement (reserved values).

Nash Bargaining Game: Description

A single-step full-information bilateral negotiation with $\Omega = [0,1]^2$ and two utility functions $(\tilde{u}_1, \tilde{u}_2)$ such that:

- A feasible set of agreements F. A common example is to define F as all the outcomes for which the total utility received by negotiators is less than or equal to one:

$$F = \{(\omega_1, \omega_2) | \tilde{u}_2(\omega_2) + \tilde{u}_1(\omega_1) \leq 1\}.$$

- A disagreement point $d \equiv \tilde{u}_1(\phi) + \tilde{u}_2(\phi) \in \Re^2$ which is the utility value received by the two players in case of disagreement (reserved values).

Other Bargaining Solutions

- Nash Point (1950): The point at which the product of surplus utility (above reservation value) of negotiators is maximized

$$\arg\max_{\omega_1,\omega_2} \prod_{i=1}^{2} \left(u_i(\omega_i) - u_i(\phi)\right)$$

- Kalai-Smorodinsky Point (1975): The Pareto outcome with equal ratios of achieved surplus utility and maximum feasible surplus utility

$$\arg\max_{\omega_1,\omega_2 \in F} (\omega_1 + \omega_2) \text{ s.t. } \left(\frac{u_1(\omega_1) - u_1(\phi)}{u_2(\omega_2) - u_2(\phi)} = \frac{\max_{v \in F}\left(u_1(v)\right) - u_1(\phi)}{\max_{v \in F}\left(u_2(v)\right) - u_2(\phi)}\right)$$

- Kalai Point (1977): The Pareto outcome maximizing the utility for the unfortunate player. Defining P as the Pareto front

$$\arg\max_{\omega_1,\omega_2 \in P} \min_{i \in \{1,2\}} \left(u_i(\omega_i) - u_i(\phi)\right)$$

Other Bargaining Solutions

- Nash Point (1950): The point at which the product of surplus utility (above reservation value) of negotiators is maximized

$$\arg\max_{\omega_1,\omega_2} \prod_{i=1}^{2} \left(u_i(\omega_i) - u_i(\phi)\right)$$

- Kalai-Smorodinsky Point (1975): The Pareto outcome with equal ratios of achieved surplus utility and maximum feasible surplus utility

$$\arg\max_{\omega_1,\omega_2 \in F} (\omega_1 + \omega_2) \text{ s.t. } \left(\frac{u_1(\omega_1) - u_1(\phi)}{u_2(\omega_2) - u_2(\phi)} = \frac{\max_{v \in F}(u_1(v)) - u_1(\phi)}{\max_{v \in F}(u_2(v)) - u_2(\phi)} \right)$$

- Kalai Point (1977): The Pareto outcome maximizing the utility for the unfortunate player. Defining P as the Pareto front

$$\arg\max_{\omega_1,\omega_2 \in P} \min_{i \in \{1,2\}} \left(u_i(\omega_i) - u_i(\phi)\right)$$

Other Bargaining Solutions

- Nash Point (1950): The point at which the product of surplus utility (above reservation value) of negotiators is maximized

$$\arg\max_{\omega_1,\omega_2} \prod_{i=1}^{2} \left(u_i(\omega_i) - u_i(\phi)\right)$$

- Kalai-Smorodinsky Point (1975): The Pareto outcome with equal ratios of achieved surplus utility and maximum feasible surplus utility

$$\arg\max_{\omega_1,\omega_2 \in F} (\omega_1 + \omega_2) \text{ s.t. } \left(\frac{u_1(\omega_1) - u_1(\phi)}{u_2(\omega_2) - u_2(\phi)} = \frac{\max_{v \in F}\left(u_1(v)\right) - u_1(\phi)}{\max_{v \in F}\left(u_2(v)\right) - u_2(\phi)}\right)$$

- Kalai Point (1977): The Pareto outcome maximizing the utility for the unfortunate player. Defining P as the Pareto front

$$\arg\max_{\omega_1,\omega_2 \in P} \min_{i \in \{1,2\}} \left(u_i(\omega_i) - u_i(\phi)\right)$$

Rubinstein's Bargaining Protocol: Description

The Game

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $u_i^{t+\Delta}(\omega) < u_i^t(\omega)$. Examples of time-pressure:

  Exponential $u_i^{t+\Delta}(\omega) = \delta_i^{\Delta} u_i^t(\omega)$.

  Linear $u_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$

- Agent's initial utility is the assigned part of the pie: $u_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^\tau(\phi) = 0 \ \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Description

The Game

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $u_i^{t+\Delta}(\omega) < u_i^t(\omega)$. Examples of time-pressure:

  Exponential $u_i^{t+\Delta}(\omega) = \delta_i^{\Delta} u_i^t(\omega)$.

  Linear $u_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$

- Agent's initial utility is the assigned part of the pie: $u_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^{\tau}(\phi) = 0 \ \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Description

The Game

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $u_i^{t+\Delta}(\omega) < u_i^t(\omega)$. Examples of time-pressure:
  - Exponential $u_i^{t+\Delta}(\omega) = \delta_i^{\Delta} u_i^t(\omega)$.
  - Linear $u_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$
- Agent's initial utility is the assigned part of the pie: $u_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^{\tau}(\phi) = 0 \ \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Description

The Game

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $u_i^{t+\Delta}(\omega) < u_i^t(\omega)$. Examples of time-pressure:

  Exponential $u_i^{t+\Delta}(\omega) = \delta_i^{\Delta} u_i^t(\omega)$.

  Linear $u_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$
- Agent's initial utility is the assigned part of the pie: $u_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^{\tau}(\phi) = 0 \; \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Description

The Game

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $u_i^{t+\Delta}(\omega) < u_i^t(\omega)$. Examples of time-pressure:
  - **Exponential** $u_i^{t+\Delta}(\omega) = \delta_i^{\Delta} u_i^t(\omega)$.
  - **Linear** $u_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$
- Agent's initial utility is the assigned part of the pie: $u_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^{\tau}(\phi) = 0 \ \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Description

The Game

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $u_i^{t+\Delta}(\omega) < u_i^t(\omega)$. Examples of time-pressure:

  Exponential $u_i^{t+\Delta}(\omega) = \delta_i^\Delta u_i^t(\omega)$.

  Linear $u_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$
- Agent's initial utility is the assigned part of the pie: $u_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^\tau(\phi) = 0 \ \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Description

The Game

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $u_i^{t+\Delta}(\omega) < u_i^t(\omega)$. Examples of time-pressure:
  - **Exponential** $u_i^{t+\Delta}(\omega) = \delta_i^{\Delta} u_i^t(\omega)$.
  - **Linear** $u_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$
- Agent's initial utility is the assigned part of the pie: $u_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^{\tau}(\phi) = 0 \ \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Description

The Game

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $u_i^{t+\Delta}(\omega) < u_i^t(\omega)$. Examples of time-pressure:
  - **Exponential** $u_i^{t+\Delta}(\omega) = \delta_i^{\Delta} u_i^t(\omega)$.
  - **Linear** $u_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$
- Agent's initial utility is the assigned part of the pie: $u_i^0 = \omega_i$.
- Time pressure and utility information are common knowledge.
- No externally imposed time-limit.
- Zero reservation value: $u_i^{\tau}(\phi) = 0 \ \forall \tau$.

Main Result

There is a unique *sub-game perfect equilibrium* that requires a single negotiation step in most cases.

Rubinstein's Bargaining Protocol: Equilibrium

Exponential Discounting

The negotiation ends in **one step** with the first agent proposing and the second agent accepting *for asymmetric cases*:

$$(\omega_1^*, \omega_2^*) = \left(\frac{1-\delta_2}{1-\delta_1\delta_2}, \frac{\delta_2(1-\delta_1)}{1-\delta_1\delta_2}\right)$$

Linear Discounting

The negotiation ends in **one step** with the first agent proposing and the second agent accepting:

$$(\omega_1^*, \omega_2^*) = \begin{cases} (c_2, 1-c_2) & c_1 > c_2 \\ (x, 1-x)\ \forall x \in [c_1, 1] & c_1 = c_2 \\ (1, 0) & c_1 < c_2 \end{cases}$$

Rubinstein's Bargaining Protocol: Equilibrium

Exponential Discounting

The negotiation ends in **one step** with the first agent proposing and the second agent accepting *for asymmetric cases*:

$$(\omega_1^*, \omega_2^*) = \left(\frac{1-\delta_2}{1-\delta_1\delta_2}, \frac{\delta_2\left(1-\delta_1\right)}{1-\delta_1\delta_2}\right)$$

Linear Discounting

The negotiation ends in **one step** with the first agent proposing and the second agent accepting:

$$(\omega_1^*, \omega_2^*) = \begin{cases} (c_2, 1-c_2) & c_1 > c_2 \\ (x, 1-x)\ \forall x \in [c_1, 1] & c_1 = c_2 \\ (1, 0) & c_1 < c_2 \end{cases}$$

Summary of Classical Results

Solved

- Singe-shot bilateral negotiation with complete information (Nash/Kalai/Kalai-Smorodinsky Solutions).
- Infinite horizon bilateral negotiation with time-pressure and complete information (Rubinstein's SPE).
- Sometimes we still need negotiations even with complete information (Hick's Paradox).
- Incomplete information: Myerson-Satterthwaite Impossibility Result.

Open

- What is the PBE of time-limited bargaining with unknown partner utility function?
- What is the PBE of time-limited bargaining with partially-known partner utility function?

Issues and Outcomes

Cartesian Outcome Space

The Cartesian product of a set of issues:
$\Omega = I_0 \times I_1 \times \cdots \times I_{N-1}$.

Issue Types

Catgorical	Set of values: $\{v_i \vert v_i \in I\}$
Ordinal	with defined order
Cardinal	with defined difference
Numeric	with defined numeric value (integer/real)

Issues and Outcomes

Cartesian Outcome Space

The Cartesian product of a set of issues:
$\Omega = I_0 \times I_1 \times \cdots \times I_{N-1}$.

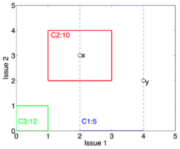
Issue Types

Catgorical	Set of values: $\{v_i \| v_i \in I\}$
Ordinal	with defined order
Cardinal	with defined difference
Numeric	with defined numeric value (integer/real)

Preferences and Utility Functions

- Partial Ordering $\omega_i \succeq \omega_j \forall \omega_i, \omega_j \in \Omega$
- Full Ordering $\omega_i \succ \omega_j \forall \omega_i, \omega_j \in \Omega$
- Cardinal $\delta_{ij} = \omega_i - \omega_j \in \Re \forall \omega_i, \omega_j \in \Omega$
- Utility Function $u(\omega) \in \Re \; \forall \omega \in \Omega$
- Normalized Utility Function $u(\omega) \in [0,1] \forall \omega \in \Omega$

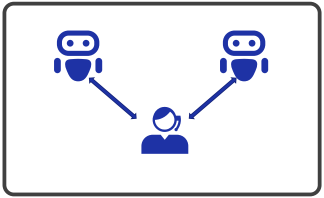
- Linear UFuns $u(\omega) = \sum_{i=0}^{|\omega|} \alpha_i \times \omega_i$
- Linear Additive UFuns $u(\omega) = \sum_{i=0}^{|\omega|} \omega_i \times f_i(\omega_i)$
- Generalized Additive UFuns $u(\omega) = \sum_{i=0}^{K} \omega_k \times f_k(\omega_j \forall j \in G_k)$
- Hyper Rectangle UFuns $u(\omega) = \sum_{k=0}^{K} c_k \times \delta[\omega \in C_k]$
- Genrealized Hyper Rectangle UFuns $u(\omega) = \sum_{k=0}^{K} f_k(\omega) \times \delta[\omega \in C_k]$



Mediated Protocols

Main Features

- Has A central *mediator*.
- Agents negotiate by exchanging *messages* with the *mediator*.
- Proposals can come from the mediator or the negotiators.



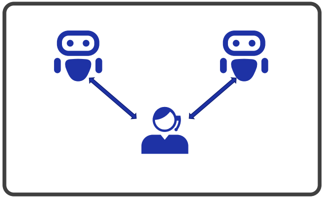
Examples

Single Text Protocol The mediator proposes a single hypothetical agreements, gets feedback about it and modifies it based on this feedback.

Mediated Protocols

Main Features

- Has A central *mediator*.
- Agents negotiate by exchanging *messages* with the *mediator*.
- Proposals can come from the mediator or the negotiators.



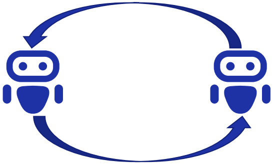
Examples

Single Text Protocol The mediator proposes a single hypothetical agreements, gets feedback about it and modifies it based on this feedback.

Unmediated Protocols

Main Features

- No central coordinator.
- Agents negotiate by exchanging *messages*.
- All proposals come from negotiators.



Examples

Nash Bargaining Game Single iteration, single issue, bilateral protocol with complete information.

Rubinstein Bargaining Protocol Infinite horizon, single issue, bilateral protocol with complete information[4].

Alternating Offers Protocol Finite horizon, multi-issue, multilateral protocol with partial information[5].

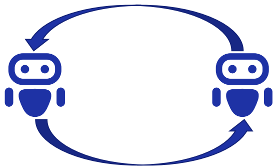
Ariel Rubinstein. "Perfect equilibrium in a bargaining model". In: *Econometrica: Journal of the Econometric Society* (1982), pp. 97–109

Reyhan Aydoğan et al. "Alternating offers protocols for multilateral negotiation". In: *Modern Approaches to Agent-based Complex Automated Negotiation*. Springer, 2017, pp. 153–167

Unmediated Protocols

Main Features

- No central coordinator.
- Agents negotiate by exchanging *messages*.
- All proposals come from negotiators.



Examples

Nash Bargaining Game: Single iteration, single issue, bilateral protocol with complete information.

Rubinstein Bargaining Protocol: Infinite horizon, single issue, bilateral protocol with complete information[4].

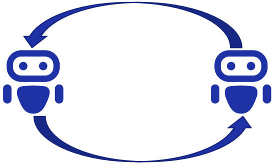
Alternating Offers Protocol: Finite horizon, multi-issue, multilateral protocol with partial information[5].

Ariel Rubinstein. "Perfect equilibrium in a bargaining model". In: *Econometrica: Journal of the Econometric Society* (1982), pp. 97–109

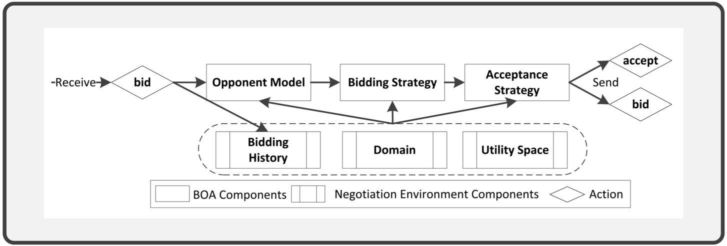
Reyhan Aydoğan et al. "Alternating offers protocols for multilateral negotiation". In: *Modern Approaches to Agent-based Complex Automated Negotiation*. Springer, 2017, pp. 153–167

Stacked Alternating Offers Protocol


```
n_agreed, current = 0, randint(0, n_agents)
offer = agents[current].offer()
while not timeout()
        current = (current + 1) % n_agents
        response = agents[current].respond(offer)
        if response == 'accept':
                n_agreed += 1
        if n_agreed == n_agents:
                return offer
        elif response == 'end_negotiation':
                return 'failed'
        elif response == 'reject':
                offer = agents[current].offer()
return "timedout"
```



Negotiator Components



BOA Architecture

Opponent Model Learns about the partner/opponent.

Offer Policy Generates new bids, Also called Offer Policy

Acceptance Policy Decides when to accept, Also called Acceptance Policy.

Tim Baarslag et al. "Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies". In: *Novel Insights in Agent-based Complex Automated Negotiation*. Ed. by Ivan Marsa-Maestre et al. Tokyo: Springer Japan, 2014, pp. 61–83. ISBN: 978-4-431-54758-7. DOI: 10.1007/978-4-431-54758-7_4. URL: https://doi.org/10.1007/978-4-431-54758-7_4

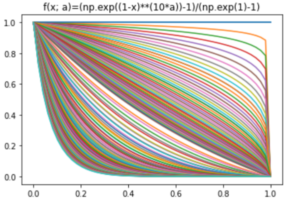
Time-based Offer Policy

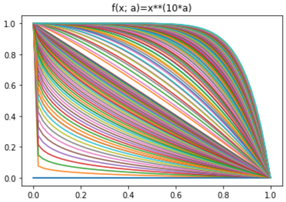
Time-based strategies

- The negotiator's offers and decisions (acceptance, ending) depend **only** on the relative negotiation time.
- Monotonically decreasing utility (usually).
- Usually requires an inverse utility function.

Common TB Strategies

- Boulware: Slow then fast concession (i.e. $a > 1$)
- Linear: Linear concession (i.e. $a = 1$)
- Conceder Fast then slow concession (i.e. $a < 1$)





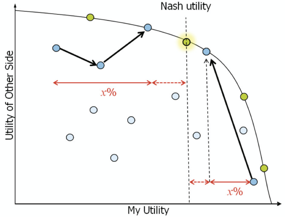
Behavioral Offer Policies

Behavior Based Strategies

- Responds to the opponent offers.
- Usualy Tit-for-Tat.
- Usually requires an opponent model.

(Nice) Tit-for-Tat (bilateral) [6]

Concede as much as the opponent toward the estimated nash-point and do not retaliate.



1. Use Bayesian opponent model to estimate Nash point;
2. Find concession towards Nash point;
3. Mirror bid in my utility relative to Nash utility;
4. Make a nice move.

Tim Baarslag, Koen Hindriks, and Catholijn Jonker. “A tit for tat negotiation strategy for real-time bilateral negotiations”. In: *Complex Automated Negotiations: Theories, Models, and Software Competitions*. Springer, 2013, pp. 229–233

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

Threshold a utility threshold (τ).

- **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
- **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).

Last my last offer (AC_{last}).

Next what I am about to offer (AC_{next}).

Best the best offer I received in a given window ($AC_{best}(K)$).

Average the average utility I received in a given window ($AC_{avg}(K)$).

Time the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)

Expected the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

- **Threshold** a utility threshold (τ).
  - **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
  - **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).
- **Last** my last offer (AC_{last}).
- **Next** what I am about to offer (AC_{next}).
- **Best** the best offer I received in a given window ($AC_{best}(K)$).
- **Average** the average utility I received in a given window ($AC_{avg}(K)$).
- **Time** the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)
- **Expected** the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

- **Threshold** a utility threshold (τ).
  - **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
  - **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).
- **Last** my last offer (AC_{last}).
- **Next** what I am about to offer (AC_{next}).
- **Best** the best offer I received in a given window ($AC_{best}(K)$).
- **Average** the average utility I received in a given window ($AC_{avg}(K)$).
- **Time** the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)
- **Expected** the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

- **Threshold** a utility threshold (τ).
  - **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
  - **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).
- **Last** my last offer (AC_{last}).
- **Next** what I am about to offer (AC_{next}).
- **Best** the best offer I received in a given window ($AC_{best}(K)$).
- **Average** the average utility I received in a given window ($AC_{avg}(K)$).
- **Time** the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)
- **Expected** the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

Threshold a utility threshold (τ).

- **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
- **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).

Last my last offer (AC_{last}).

Next what I am about to offer (AC_{next}).

Best the best offer I received in a given window ($AC_{best}(K)$).

Average the average utility I received in a given window ($AC_{avg}(K)$).

Time the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)

Expected the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

- **Threshold** a utility threshold (τ).
  - **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
  - **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).
- **Last** my last offer (AC_{last}).
- **Next** what I am about to offer (AC_{next}).
- **Best** the best offer I received in a given window ($AC_{best}(K)$).
- **Average** the average utility I received in a given window ($AC_{avg}(K)$).
- **Time** the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)
- **Expected** the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

- **Threshold** a utility threshold (τ).
  - **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
  - **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).
- **Last** my last offer (AC_{last}).
- **Next** what I am about to offer (AC_{next}).
- **Best** the best offer I received in a given window ($AC_{best}(K)$).
- **Average** the average utility I received in a given window ($AC_{avg}(K)$).
- **Time** the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)
- **Expected** the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

- **Threshold** a utility threshold (τ).
  - **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
  - **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).
- **Last** my last offer (AC_{last}).
- **Next** what I am about to offer (AC_{next}).
- **Best** the best offer I received in a given window ($AC_{best}(K)$).
- **Average** the average utility I received in a given window ($AC_{avg}(K)$).
- **Time** the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)
- **Expected** the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

- **Threshold** a utility threshold (τ).
  - **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
  - **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).
- **Last** my last offer (AC_{last}).
- **Next** what I am about to offer (AC_{next}).
- **Best** the best offer I received in a given window ($AC_{best}(K)$).
- **Average** the average utility I received in a given window ($AC_{avg}(K)$).
- **Time** the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)
- **Expected** the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Acceptance Policy

Accept if $\alpha u(\omega) + \beta$ is greater than:

- **Threshold** a utility threshold (τ).
  - **Constant** May be a fraction of maximum utility ($AC_{const}(\gamma)$).
  - **Time-based** Monotonically non-increasing with time ($AC_{monotonic}(t)$).
- **Last** my last offer (AC_{last}).
- **Next** what I am about to offer (AC_{next}).
- **Best** the best offer I received in a given window ($AC_{best}(K)$).
- **Average** the average utility I received in a given window ($AC_{avg}(K)$).
- **Time** the reserved value and $T \in [0, 1]$ fraction of the negotiation have passed ($AC_{time}(T)$)
- **Expected** the best offer I expect to receive (e.g. Gaussian Processes, needs opponent offer and acceptance policies).

Combining Acceptance Policies

Combined Acceptance Strategy[7]

- Combines multiple simple acceptance policies.
- $AC_{combi}(\tau, \gamma) = AC_{next} \vee (AC_{time}(\tau) \wedge AC_{const}(\gamma))$
- $AC^{best}_{combi}(\tau, W) = AC_{next} \vee (AC_{time}(\tau) \wedge AC_{best}(W))$
- $AC^{avg}_{combi}(\tau, W) = AC_{next} \vee (AC_{time}(\tau) \wedge AC_{avg}(W))$
- $AC^{best}_{combi}(\tau) = AC_{next} \vee (AC_{time}(\tau) \wedge AC_{best}(T))$

NegMAS

```
ACCombi = ACNext() or (ACTime(tau) and ACConst(gamma))
ACBest = ACNext() or (ACTime(tau) and ACBest(w))
ACAvg = ACNext() or (ACTime(tau) and ACLastKReceived(K, op=math.mean))
ACBestAll = ACNext() or (ACTime(tau) and ACBest())
```

[7] Tim Baarslag, Koen Hindriks, and Catholijn Jonker. "Effective acceptance conditions in real-time automated negotiation". In: *Decision Support Systems* 60 (2014), pp. 68–77

Opponent Modeling

Opponent Components

- Opponent preferences $u^o(\omega) \forall \omega \in \Omega$
- Offer Policy $\pi^o(s)$
- Acceptance Policy $a(\omega, s)$

What is being modeled?

- Any of the 3 components.
- Opponent Type.

When is it modeled?

- Before the negotiation: Static Model.
- During the negotiation: Dynamic Model.

Data

- This opponent vs. this opponent group vs. all opponents.
- Only agreements vs. All exchanged offers.

Opponent Modeling

Opponent Components

- Opponent preferences $u^o(\omega) \forall \omega \in \Omega$
- Offer Policy $\pi^o(s)$
- Acceptance Policy $a(\omega, s)$

When is it modeled?

- Before the negotiation: Static Model.
- During the negotiation: Dynamic Model.

Data

- This opponent vs. this opponent group vs. all opponents.
- Only agreements vs. All exchanged offers.

What is being modeled?

- Any of the 3 components.
- Opponent Type.

Opponent Modeling

Opponent Components

- Opponent preferences
  $u^o(\omega) \forall \omega \in \Omega$
- Offer Policy $\pi^o(s)$
- Acceptance Policy $a(\omega, s)$

What is being modeled?

- Any of the 3 components.
- Opponent Type.

When is it modeled?

- Before the negotiation: Static Model.
- During the negotiation: Dynamic Model.

Data

- This opponent vs. this opponent group vs. all opponents.
- Only agreements vs. All exchanged offers.

Some Automated Negotiation Platforms

Genius [8]

a Java-based negotiation platform to develop general negotiating agents and create negotiation scenarios.

GeniusWeb

A distributed platform for automated negotiation on the internet

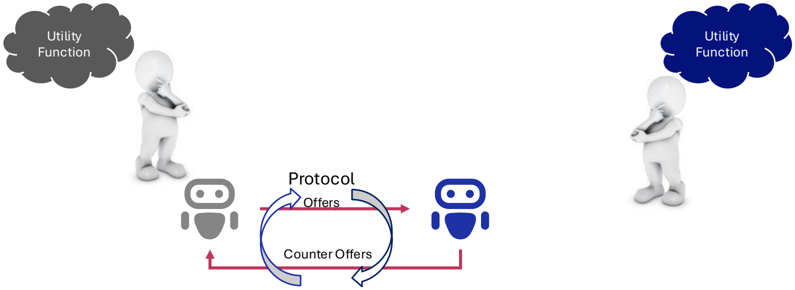
NegMAS [9]

a Python-based negotiation platform for developing autonomous negotiation agents embedded in simulation environments.

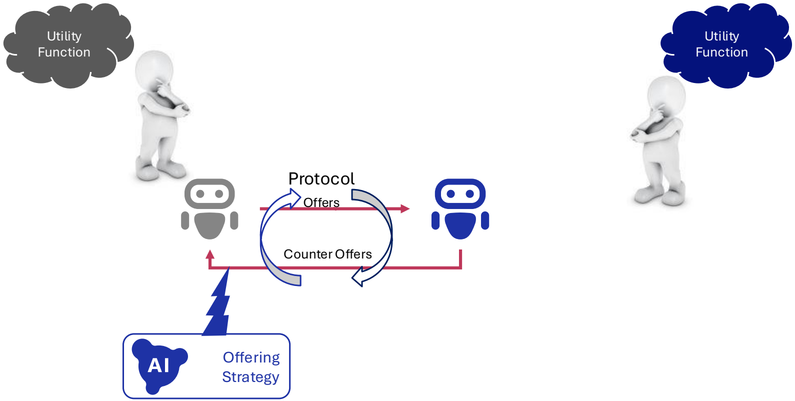
Raz Lin et al. "Genius: An Integrated Environment for Supporting the Design of Generic Automated Negotiators". In: *Computational Intelligence* 30.1 (2014), pp. 48–70. ISSN: 1467-8640. DOI: 10.1111/j.1467-8640.2012.00463.x. URL: http://dx.doi.org/10.1111/j.1467-8640.2012.00463.x

Yasser Mohammad, Amy Greenwald, and Shinji Nakadai. "NegMAS: A platform for situated negotiations". In: *Twelfth International Workshop on Agent-based Complex Automated Negotiations (ACAN2019) in conjunction with IJCAI*. Macau, China, 2019. URL: https://github.com/yasserfarouk/negmas

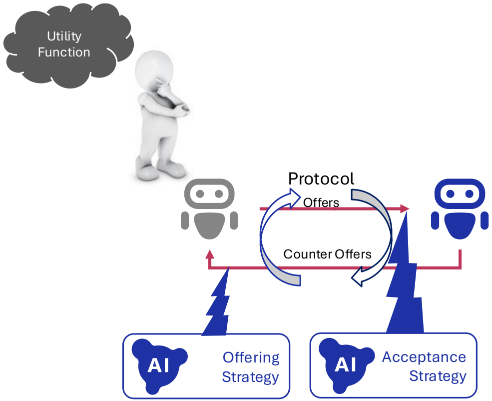
Role of ML in Automated Negotiation

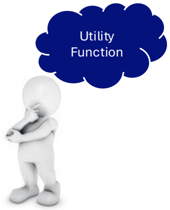


Role of ML in Automated Negotiation

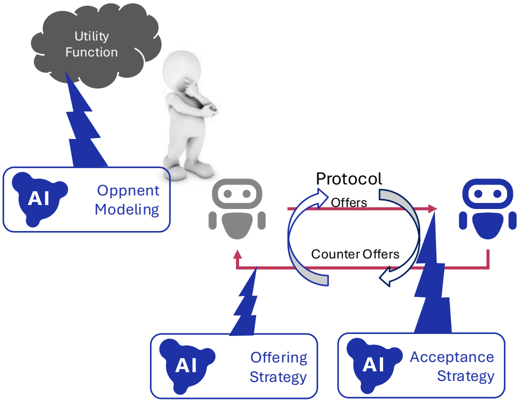


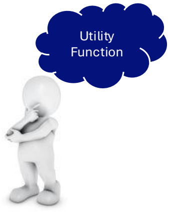
Role of ML in Automated Negotiation



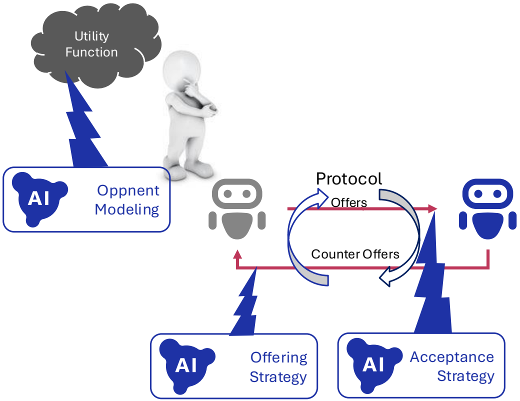


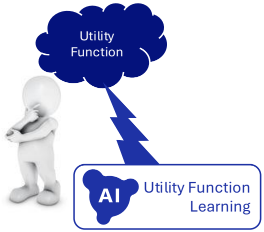
Role of ML in Automated Negotiation



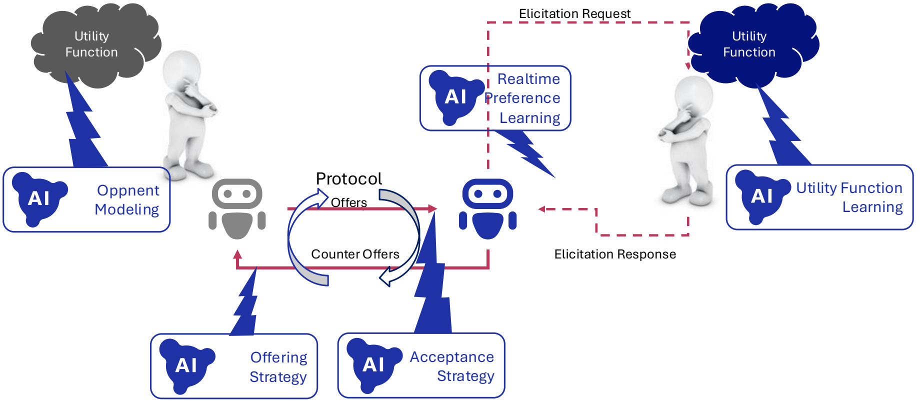


Role of ML in Automated Negotiation

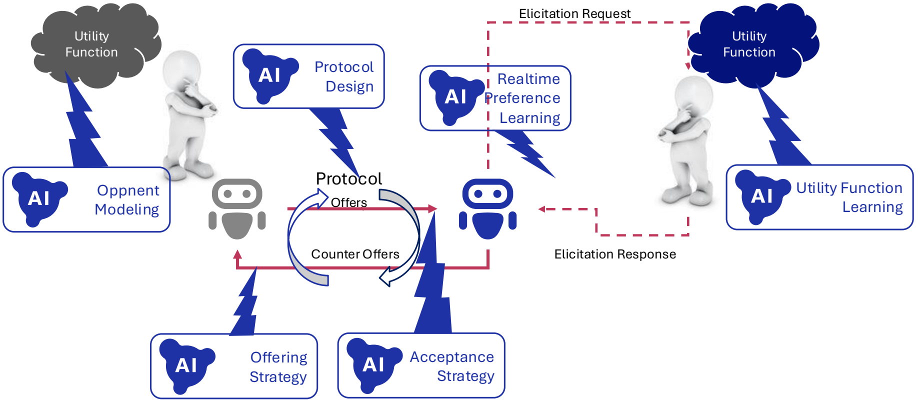




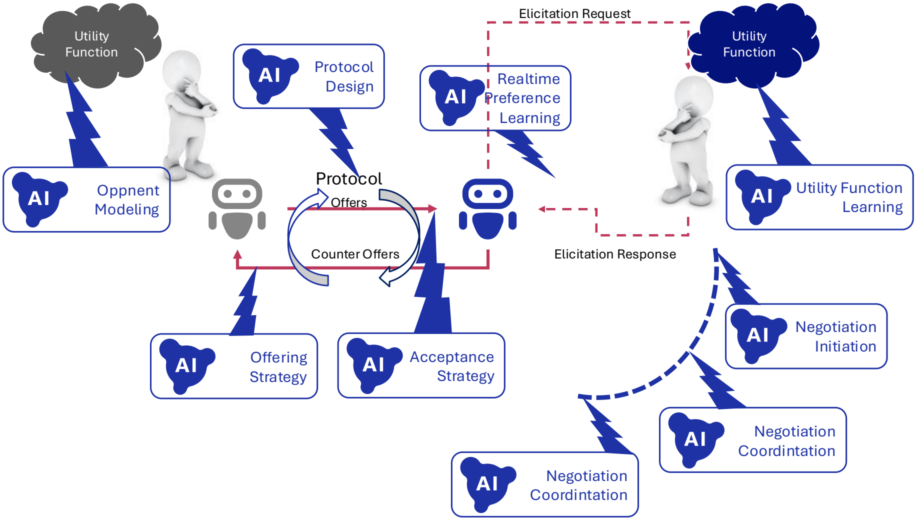
Role of ML in Automated Negotiation



Role of ML in Automated Negotiation



Role of ML in Automated Negotiation



Learning in Automated Negotiation

What?

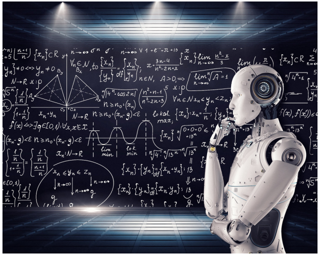
1. Acceptance Policy
2. Offering Policy
3. Opponent/Partner Model

When?

1. Within Negotiation
2. Between Negotiations

How?

1. Supervised Learning
2. Reinforcement Learning
3. Unsupervised Learning



"Artificial Intelligence & AI & Machine Learning" by mikemacmarketing

ANAC Competition

Automated Negotiation League

Explore the strategies and difficulties in creating efficient agents whose primary purpose is to negotiate with other agent's strategies.

Details | CFP | Live | Results 2024

Supply Chain Management League

Design and build an autonomous agent that negotiates on behalf of a factory manager situated in a supply chain management simulation.

Details | CFP | Live | Results 2024

- ANAC is the **Automated Negotiating Agents Competition** running since 2010.
- This year it will be with IJCAI. Submission is open until June 1st.
- ANAC's 2024 demo is scheduled for Thursday (22nd) 10AM Demo Session.

Partner Preferences

HardHeaded Opponent Modeling Strategy[10]

Context

- Winner of the ANAC 2011 competition.

$$u(\omega) = \sum_{i=1}^{n} \alpha_i F[i, \omega_i]$$

- Assumes a Discrete Outcome Space, Linear Additive Utility Function and a bilalateral negotiation.
- Learns while negotiaing.

Main Idea

- The opponent is likely to change values for issues that are less important.

Opponent Model: Bayesian

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its counteroffers/rejections).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Example

Hypothesis space: Utility function as a weighted sum of basis functions

$$u(\omega) = \sum_{i=1}^{n} \alpha_i f_i(\omega_i; \sigma_i)$$

Evidence: Rejection and offers (assuming a strategy).

Opponent Model: Bayesian

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its counteroffers/rejections).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Example

Hypothesis space: Utility function as a weighted sum of basis functions

$$u(\omega) = \sum_{i=1}^{n} \alpha_i f_i(\omega_i; \sigma_i)$$

Evidence: Rejection and offers (assuming a strategy).

Opponent Model: Bayesian

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its counteroffers/rejections).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Example

Hypothesis space: Utility function as a weighted sum of basis functions

$$u(\omega) = \sum_{i=1}^{n} \alpha_i f_i(\omega_i; \sigma_i)$$

Evidence: Rejection and offers (assuming a strategy).

Opponent Model: Bayesian

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its counteroffers/rejections).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Example

Hypothesis space: Utility function as a weighted sum of basis functions

$$u(\omega) = \sum_{i=1}^{n} \alpha_i f_i(\omega_i; \sigma_i)$$

Evidence: Rejection and offers (assuming a strategy).

Opponent Model: Bayesian

Bayesian Learning

Hypothesis A hypothesis about the opponent's behavior.

Evidence Behavior of the agent (e.g. its counteroffers/rejections).

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

Example

Hypothesis space: Utility function as a weighted sum of basis functions

$$u(\omega) = \sum_{i=1}^{n} \alpha_i f_i(\omega_i; \sigma_i)$$

Evidence: Rejection and offers (assuming a strategy).

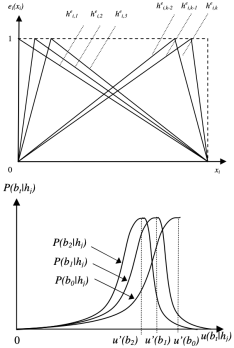
Bayesian Opponent Model Learner[11]

Assumptions

- Opponent has a Linear Additive UFun $(u(\omega) = \sum_{i=1}^{|\omega|} \alpha_i f_i(\omega_i, \sigma_i))$
- Value functions (f_i) are triangle like (or linear).

Settings

- Hypothesis Space: values of α_i and σ_i
- Evidence: $P(\omega|\alpha_i, \sigma_i) = \frac{1}{\sqrt{2\pi\sigma}} \exp \frac{(u(\omega|\alpha_i,\sigma_i) - \hat{u}(\omega))^2}{\sigma}$ with $\hat{u}(\omega) = 1 - \frac{t}{20}$.
- Estimated opponent utility value: $u^o(\omega) = \sum_{j=1}^{|H|} P(\alpha_j, \sigma_j|\Omega^o) u(\omega|\alpha_j, \sigma_j)$



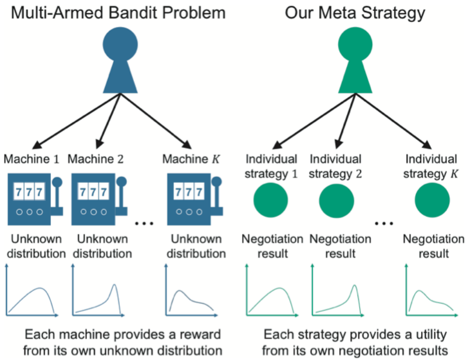
Offering Strategy

Multiaremed Bandits for Repeated Negotiations

Treat sub-negotiators as bandits in a standard multi-armed bandits problem.

- Base Strategies: Atlas3, CaduceusDC16, Kawaii, ParsCat, Rubick, YXAgent
- Method:
  - After every negotiation update the corresponding $\hat{\mu}_s$.
  - Use the slot machine (negotiator) that maximizes

$$UCB(s) = \hat{\mu}_s + c\sqrt{\frac{lnN}{N_s}}$$



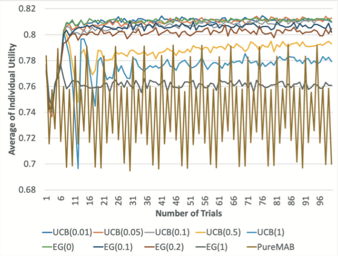
Ryohei Kawata and Katsuhide Fujita. "Meta-Strategy Based on Multi-Armed Bandit Approach for Multi-Time Negotiation". In: *IEICE TRANSACTIONS on Information and Systems* 103.12 (2020), pp. 2540–2548

Multiaremed Bandits for Repeated Negotiations

Treat sub-negotiators as bandits in a standard multi-armed bandits problem.

- Base Strategies: Atlas3, CaduceusDC16, Kawaii, ParsCat, Rubick, YXAgent
- Method:
  - After every negotiation update the corresponding $\hat{\mu}_s$.
  - Use the slot machine (negotiator) that maximizes

$$UCB(s) = \hat{\mu}_s + c\sqrt{\frac{lnN}{N_s}}$$



Agent	Individual utility	Social welfare
UCB(0.01)	**0.7734**	1.4575
Agent33	0.6901	**1.4579**
AgentNP2018	0.7082	1.4362
Appaloosa	0.7067	1.3706
Ellen	0.6083	1.2223
TimeTraveler	0.7142	1.4573

Ryohei Kawata and Katsuhide Fujita. "Meta-Strategy Based on Multi-Armed Bandit Approach for Multi-Time Negotiation". In: *IEICE TRANSACTIONS on Information and Systems* 103.12 (2020), pp. 2540–2548

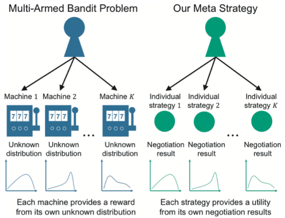
Multiarmed Bandits: Mapping

Components

ObservationManager N/A

RewardFunction utility

ActionManager strategy index



Supporting components

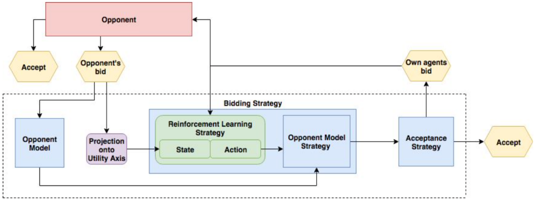
- Base strategies Atlas3, CaduceusDC16, Kawaii, ParsCat, Rubick, YXAgent

- Applies to **repeated** negotiations.

Training Method

- After every negotiation update the corresponding $\hat{\mu}_s$.
- Use the slot machine (negotiator) that maximizes $UCB(s) = \hat{\mu}_s + c\sqrt{\frac{lnN}{N_s}}$.

RLBOA: Learning Offering Strategy



Main Points

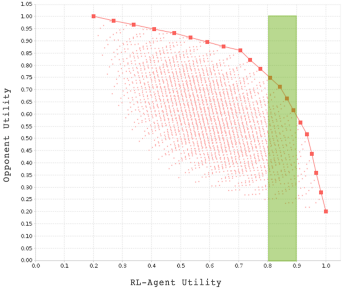
- Extends the BOA architecture.
- Learns only a bidding strategy:
  - The agent learns how to move *in its own utility axis*.

Jasper Bakker et al. "RLBOA: A modular reinforcement learning framework for autonomous negotiating agents". In: *Proceedings of the 18th international conference on autonomous agents and multiagent systems*. 2019, pp. 260–268

RLBOA: The details

- State Space: $\left\{\hat{u}(\omega^s_t), \hat{u}(\omega^s_{t-1}), \hat{u}(\omega^p_t), \hat{u}(\omega^p_{t-1}), t\right\}$.
  - $\hat{u}(\omega) = \lfloor N \times u(\omega) \rfloor$[12]
- Action Space: $\leftarrow, -, \rightarrow$.
  - First step $\rightarrow i \in [0, N-1]$
  - Out-of-boundary correction: $-$.
- Training Method: Q-learning
- Acceptance Strategy [Recommended]: $AC_{next}(\alpha = 1, \beta = 0)$[13]

$$a(\omega) = \begin{cases} \text{Accept}, & \text{if } \alpha u(\omega) + \beta \geq u(o(s)) \\ \text{Reject}, & \text{otherwise} \end{cases}$$



RLBOA: Mapping

Components

ObservationManager Utility range (discrete).

RewardFunction utility

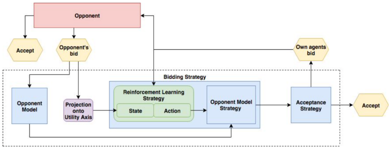
ActionManager Up to one utility band (3 valued).

Supporting components

- **Utility Inverter** Opponent model.
- **Utility Inverter** Samples an outcome in a range of utilities that maximizes the partner's utility.

Training Method

- Q-learning.



- Applies within a single negotiation.

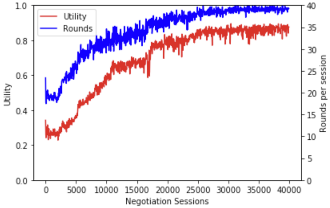
RLBOA: Observation/Action Manager Code

```python
@define
class RLBoaEncoder(DictEncoder):
    """The observation encoder of RLBOA according to the paper."""

    children: tuple[ObservationEncoder, ...] = field(init=False, factory=tuple)
    names: tuple[str, ...] = field(init=False, factory=tuple)

    def __attrs_post_init__(self):
        self.names = ("time", "utility")
        self.children = (
            DTimeEncoder(n_levels=5),
            DWindowedUtilityEncoder(
                n_offers=4,
                n_levels=10,
            ),
        )

@define
class RLBoaDecoder(DRelativeUtilityDecoder1D):
    """The action decoder for RLBOA"""
```

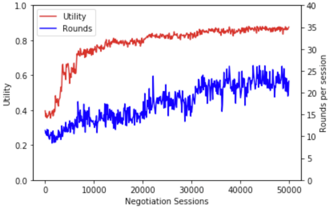
RLBOA: Negotiator Code

```python
class RLBoa(SAORLNegotiator):
    """RLBOA implementation"""

    @classmethod
    def default_trainer_type(cls) -> type["BaseAlgorithm"]:
        from stable_baselines3.ppo import PPO

        return PPO

    @classmethod
    def default_obs_encoder_type(cls) -> type[RLBoaEncoder]:
        return RLBoaEncoder

    @classmethod
    def default_action_decoder_type(cls) -> type[RLBoaDecoder]:
        return RLBoaDecoder
```

RLBOA: Evaluation and Results



(a) Scenario generality experiment against the Boulware agent.



(b) Opponent generality experiment in the medium sized domain with low opposition.

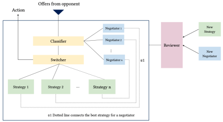
- Partners: TFT, Boulware TB
- Projection into one's utility space is surprisingly effective.
- Faster and better agreements!

Domain	**Outcome space**	**Low opp.**	**High opp.**
Small	256	0.2615	0.5178
Medium	3.125	0.3111	0.5444
Large	46.656	0.2595	0.5250

A Framework for Learning Offer Strategies

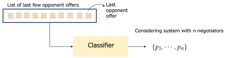
Main Idea[14]

- Uses RL for learning approximate best responses to some agents.
- Uses Supervised Learning to learn a realtime switching strategy between learned best responses.
- Uses a form of Unsupervised Learning for adapting the system to new partner types.

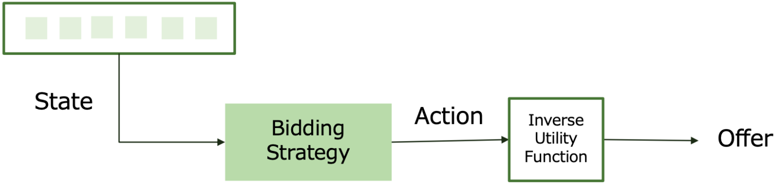


Phases

- Before Negotiation Learn approximate best responses to **a few** agents.
- During Negotiation Switch to the most appropriate **learned app. best response**
- After Negotiation Add a new **best response**?



Before: Learning Approximate Best Response



The RL Component

State Self utility of last N offers plus relative time.

Action Utility of next offer $\in [0, 1]$.

Reward Agreement/disagreement utility.

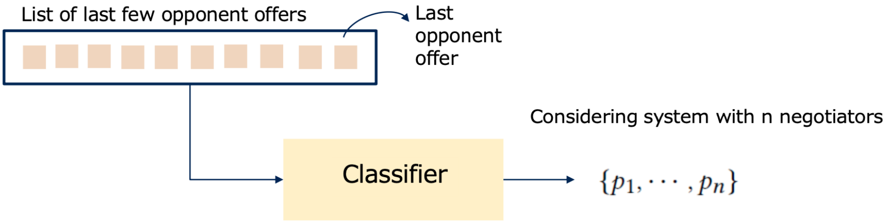
Trainer Soft Actor Critic (SAC)

$$\begin{aligned} s_t = & \{t_r, U_s(\omega_s^{t-2}), U_s(\omega_o^{t-2}), U_s(\omega_s^{t-1}), \\ & U_s(\omega_o^{t-1}), U_s(\omega_s^{t}), U_s(\omega_o^{t})\} \\ a_t = & \quad u_s^{t+1} \quad \text{such that} \quad u_r < u_s \leq 1 \end{aligned}$$

$$U_s^{-1}(u_s) = \underset{\omega}{\operatorname{argmin}} f(\omega), \quad \text{where}$$

$$f(\omega) = (U_s(\omega) - u_s)^2 \quad \forall \quad \omega \in \Omega.$$

During: Learning realtime Partner Classification



The SL Components

Features Opponent last K offers.

Target Opponent Type (discrete set)

After: Reviewing New Pairs

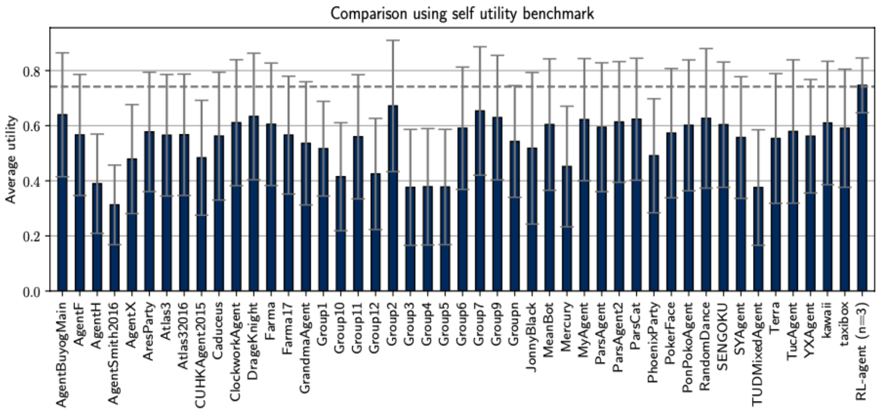
New Partner Type (N_{new}) Ecncountered

- Train a best response (using SAC) $\rightarrow S_{new}$.
- Evaluate S_{new} against $N_{new} \rightarrow U(S_{new})$
- Evaluate *Current* against $N_{new} \rightarrow U(Current)$
- Add (S_{new}, N_{new}) iff $\beta U(Current) < U(S_{new})$
- Update best responses $\downarrow$.

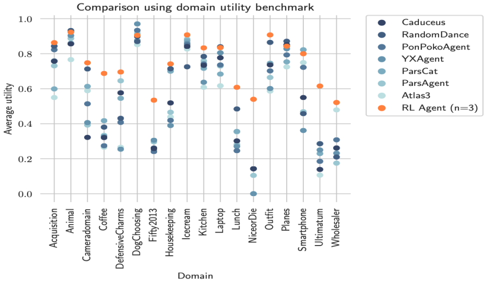
Update Best Responses

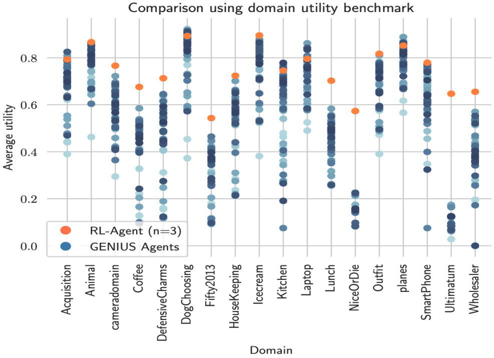
- For every learned ABR, negotiator pair (S, N):
  - Evaluate S_{new} against $N \rightarrow U(S_{new})$
  - Evaluate S against $N \rightarrow U(S)$
  - Replace S with S_{new} iff $\alpha U(S) < U(S_{new})$

Results: Against Different Opponents

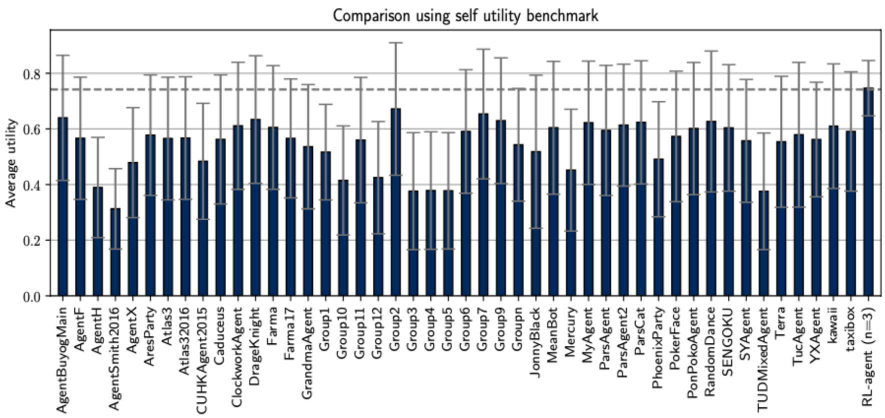


Results: In Different Domains

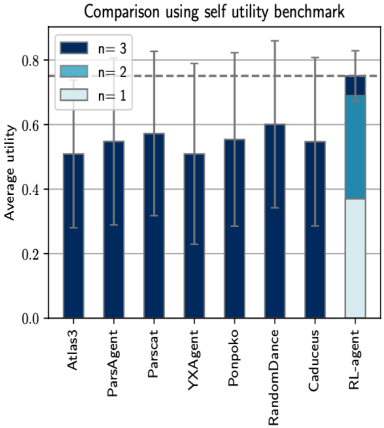




Results: Compared with SOTA Agents



Results: Improvement with new best responses



Acceptance Strategy

DQN for learning Acceptance Strategy

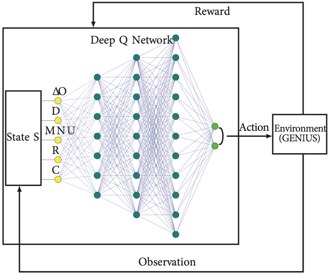
Main Idea

- Learning the acceptance strategy for a fixed offering strategy.

Settings

- State Space $u(\omega) - u(\phi), 1 - t, u(o(s)), u_t, u(\omega)$
  - u_t is a relatively large target utility (e.g. 0.8).
- Action Space Accept/Reject
- Reward

$$r = \begin{cases} -2^{|u_t - u_f|}, & \text{if } u_t > u(\omega_a) \\ +2^{|u_t - u_f|}, & \text{if } u_t < u(\omega_a) \\ 0 & \text{if non-terminal} \end{cases}$$

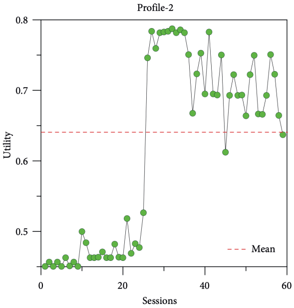


Yousef Razeghi, Celal Ozan Berk Yavuz, and Reyhan Aydoğan. "Deep reinforcement learning for acceptance strategy in bilateral negotiations". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 28.4 (2020), pp. 1824–1840

Evaluation

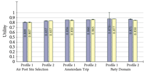
Training

- Domain England-Zimbabwe (576 outcomes)
- Partner Gahboninho
- Offering Strategy AgentK
- Opponent Model AgentLG, Not TFT.

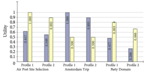


Testing

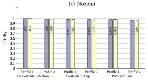
- Domains Party (3072), Amsterdam (3024), Airport (420)
- Partners Agent Smith, Yushu, FSEGA, IAMHaggler, ParsAgent, Nozomi
- Baseline ACnext



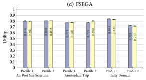
(c) Nozomi



(d) FSEGA



(e) Agent Smith



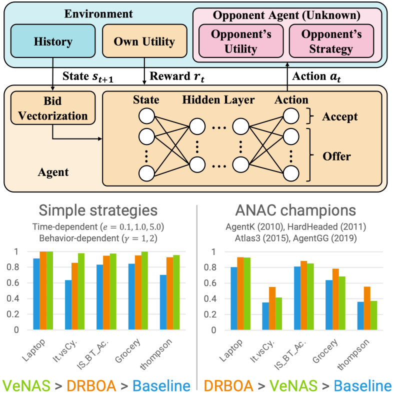
(f) Pars Agent

Learning Offer and Acceptance Policies

- Fixed domain (i/o using outcomes).
- Discrete Issues: One hot encoding per issue.

- State Space $\omega^s, \omega^o, t, \eta_t$
- Action Space $\Omega \wedge$ Accept
- Reward $= \begin{cases} u(\omega_a), & \text{At the end} \\ 0 & \text{non terminal state} \end{cases}$

VeNAS: Versatile Negotiating Agent Strategy



Toki Takahashi et al. "VeNAS: Versatile Negotiating Agent Strategy via Deep Reinforcement Learning". In: *AAAI 2022*. 2022

Own Preferences

Preference Elicitation

The challenge

How to reduce Uncertainty in user preferences:

- before negotiation (offline preference elicitation).
- while negotiating (online preference elicitation).

Types of questions

Utility Value what is $\tilde{u}(\omega)$?

Utility Constraint Is $\tilde{u}(\omega) \geq x$? Usually implemented as a standard gamble.

Utility Comparison Is $\omega_1 \succ \omega_2$?

Preference Elicitation

The challenge

How to reduce Uncertainty in user preferences:

- before negotiation (offline preference elicitation).
- while negotiating (online preference elicitation).

Types of questions

Utility Value what is $\tilde{u}(\omega)$?

Utility Constraint Is $\tilde{u}(\omega) \geq x$? Usually implemented as a standard gamble.

Utility Comparison Is $\omega_1 \succ \omega_2$?

Elicitation Procedures

1. Long history in the decision support and economics research community.
2. Take away message: .
3. Practical elicitation uses a **series** of comparisons between outcomes to assess utilities.

Elicitation Procedures

1. Long history in the decision support and economics research community.
2. Take away message: Do not ask about the utility directly..
3. Practical elicitation uses a **series** of comparisons between outcomes to assess utilities.

Elicitation Procedures

1. Long history in the decision support and economics research community.
2. Take away message: Do not ask about the utility directly..
3. Practical elicitation uses a **series** of comparisons between outcomes to assess utilities.

Elicitation Procedures

1. Long history in the decision support and economics research community.
2. Take away message: Do not ask about the utility directly..
3. Practical elicitation uses a **series** of comparisons between outcomes to assess utilities.

A Gamble

(ω^*, ω_*, p) : Getting ω^* with probability p otherwise ω_*

Elicitation Procedures

1. Long history in the decision support and economics research community.
2. Take away message: Do not ask about the utility directly..
3. Practical elicitation uses a **series** of comparisons between outcomes to assess utilities.

A Gamble

(ω^*, ω_*, p) : Getting ω^* with probability p otherwise ω_*

Example query

Do you prefer to get ω for certain over (ω^*, ω_*, p)?

Elicitation Procedures/Strategies

Probability Equivalence

find p so that $\omega = (\omega^*, \omega_*, p)$

Certainty Equivalence

find ω so that $\omega = (\omega^*, \omega_*, p)$

- Both require *normalized* utilities.
- Both require knowledge of $\omega^* \succ \omega \succ \omega_*$.
- Lead to different biases.

Comparison-only Procedures

1. Titration-down: $p_k = 1 - s \times k$
2. Titration-up: $p_k = s \times k$
3. Ping-pong: $p_k = \begin{cases} s \times \lfloor k/2 \rfloor & \text{k is odd} \\ 1 - s \times k/2 & \text{k is even} \end{cases}$

Importance of Elicitation

Negotiation with Elicitation

$m, \Omega, R, \tilde{U}_i \; \forall 1 \leq i \leq m \;, \hat{U}_i^0 \; \forall 1 \leq i \leq m$

m Number of agents/actors

$\Omega = \{\omega_j\}$ Possible outcomes (assumed countable)

n Number of outcomes $|\Omega|$

$R(i) \equiv r_i$ Reserved value for agent i

$\tilde{U}_i : \Omega \to [0,1]$ Utility of outcomes to **actor** i

$\hat{U}_i^0 : \Omega \to P$ Probability distribution of utility values for **agent** i

$\hat{U}_{ij}^0 \equiv \hat{U}_i^0(\omega_j)$

$P : \{[0,1] \to [0,1]\}$ A probability distribution on the closed interval $[0,1]$

What is Elicitation Doing?

Reduces uncertainty in $\hat{U}$

State of the Art

- Lots of work on preferences/utility elicitation in decision making domain.
- Some work on incremental utility elicitation.
- Few works on incremental utility elicitation during negotiations

Why Is Negotiation Different

1. The acceptance model changes over time $\rightarrow$ environment dynamics are not static.
2. Exploration is extremely costly.
3. Usually negotiations are not repeated much.
4. Cannot train on a simulator (in most cases).

State of the Art

- Lots of work on preferences/utility elicitation in decision making domain.
- Some work on incremental utility elicitation.
- Few works on incremental utility elicitation during negotiations

Why Is Negotiation Different

1. The acceptance model changes over time $\rightarrow$ environment dynamics are not static.
2. Exploration is extremely costly.
3. Usually negotiations are not repeated much.
4. Cannot train on a simulator (in most cases).

Pandora's Problem [Economics]

1. A set of n boxes ($\{\omega_j\}$).
2. Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j.
3. Future rewards are discounted with a known factor β.
4. Pandora's Problem:
   4.1 What is the optimal order to open the boxes?
   4.2 When should she stop?
5. Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.

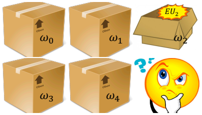
Solution: Pandora's Rule [15]

For each box j, find z_j which is the solution to:

$$c_j = \beta_j \int_{z_j}^{\infty} (u - z_j)\, p_j(u)\, du - (1 - \beta_j)\, z_j$$

Pandora's Problem [Economics]

1. A set of n boxes ($\{\omega_j\}$).
2. Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j.
3. Future rewards are discounted with a known factor β.
4. Pandora's Problem:
    4.1 What is the optimal order to open the boxes?
    4.2 When should she stop?
5. Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.



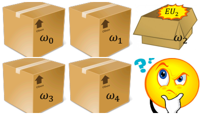
Solution: Pandora's Rule [15]

For each box j, find z_j which is the solution to:

$$c_j = \beta_j \int_{z_j}^{\infty} (u - z_j)\, p_j(u)\, du - (1 - \beta_j)\, z_j$$

Pandora's Problem [Economics]

1. A set of n boxes ($\{\omega_j\}$).
2. Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j.
3. Future rewards are discounted with a known factor β.
4. Pandora's Problem:
   - 4.1 What is the optimal order to open the boxes?
   - 4.2 When should she stop?
5. Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.



Solution: Pandora's Rule [15]

For each box j, find z_j which is the solution to:

$$c_j = \beta_j \int_{z_j}^{\infty} (u - z_j)\, p_j(u)\, du - (1 - \beta_j)\, z_j$$

Pandora's Problem [Economics]

1. A set of n boxes ($\{\omega_j\}$).
2. Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j.
3. Future rewards are discounted with a known factor β.
4. Pandora's Problem:
    - 4.1 What is the optimal order to open the boxes?
    - 4.2 When should she stop?
5. Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.

Solution: Pandora's Rule [15]

For each box j, find z_j which is the solution to:

$$c_j = \beta_j \int_{z_j}^{\infty} (u - z_j)\, p_j(u)\, du - (1 - \beta_j)\, z_j$$

Pandora's Problem [Economics]

1. A set of n boxes ($\{\omega_j\}$).
2. Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j.
3. Future rewards are discounted with a known factor β.
4. Pandora's Problem:
   4.1 What is the optimal order to open the boxes?
   4.2 When should she stop?
5. Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.

Solution: Pandora's Rule [15]

For each box j, find z_j which is the solution to:

$$c_j = \beta_j \int_{z_j}^{\infty} (u - z_j)\, p_j(u)\, du - (1 - \beta_j)\, z_j$$

Pandora's Problem [Economics]

1. A set of n boxes ($\{\omega_j\}$).
2. Opening a box j gives a reward between 0 and ∞ according to distribution p_j after t_j time-steps, and costs c_j.
3. Future rewards are discounted with a known factor β.
4. Pandora's Problem:
   4.1 What is the optimal order to open the boxes?
   4.2 When should she stop?
5. Similar to elicitation (boxes = outcomes, open = query) but assumes that uncertainty is completely removed.

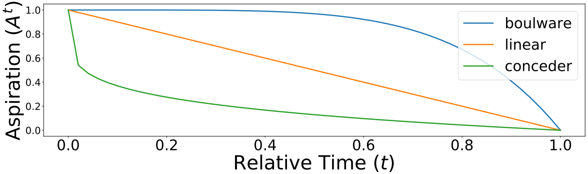
Solution: Pandora's Rule [15]

For each box j, find z_j which is the solution to:

$$c_j = \beta_j \int_{z_j}^{\infty} (u - z_j)\, p_j(u)\, du - (1 - \beta_j)\, z_j$$

Optimal Elicitation[16]

Adapts Pandora's Rule to the negotiation context:



1. $\beta = 1.0$
2. Define aspiration level as: $A^t \equiv r_i + \left(A^0 - r_i\right) \times \left(1 - \frac{t}{N}^{1/e}\right)$
   $e > 1 \rightarrow$ Boulware, $e = 1 \rightarrow$ Linear, $e < 1 \rightarrow$ Conceder
3. $p_j = \Lambda_i^t(\omega_j) \times \mathbb{E}\left(\hat{U}_{ij}^t\right) + (1 - \Lambda_i^t(\omega_j)) \times A^t(\omega_j)$
4. Assume that there is an open box giving r_i with outcome index 0.
5. End the negotiation once the best box is 0.

Why is OE sub-optimal?

Main Issue

Assuming that all uncertainty is removed by elicitation.

1. Assuming that $\hat{U}_{ij} \rightarrow \delta\left[u = \tilde{U}_i\left(\omega_j\right)\right]$
2. Consider any practical strategy (e.g. titration-down):
   - After the first question: $\hat{U}_{ij}^{t} \rightarrow \hat{U}_{ij}^{t+1}$
   - z_j was calculated using $\hat{U}_{ij}^{t}$ and must be recalculated.

Take-away message

Avoid deep-elicitation.

Why is OE sub-optimal?

Main Issue

Assuming that all uncertainty is removed by elicitation.

1. Assuming that $\hat{U}_{ij} \to \delta\left[u = \tilde{U}_i\left(\omega_j\right)\right]$
2. Consider any practical strategy (e.g. titration-down):
   - After the first question: $\hat{U}^t_{ij} \to \hat{U}^{t+1}_{ij}$
   - z_j was calculated using $\hat{U}^t_{ij}$ and must be recalculated.

Take-away message

Avoid deep-elicitation.

Extensions to Pandora's algorithm

Closed-form Calculation of z-index [17]

$$z_j = \begin{cases} \frac{a+b}{2}\beta - c_j & z_j \leq a \\ \frac{-\lambda \pm \sqrt{\lambda^2 - 4\zeta}}{2} & a < z_j \leq b \end{cases}$$

λ $-2\left(b + \frac{a-\beta}{\beta}(b-a)\right)$

ζ $b^2 - \frac{2c_j}{\beta}(b-a)$

The balanced expectation operator

$$\mathcal{E}(\hat{U}_{ij}^t) = \frac{t}{N} \times Min\left(\hat{U}_{ij}^t\right) + \left(1 - \frac{t}{N}\right) \times Max\left(\hat{U}_{ij}^t\right)$$

Min/Max a *biased* estimator that exaggerate the lower/upper part of its input. For $U(a, b)$, $Min, Max = a, b$.

Value of Information Algorithm

- Based on[18] in decision-support context.
- Adapted to the negotiation context.

Main Idea

- Assume an accurate opponent model (acceptance probability)
- Given a set of queries $Q \rightarrow$ find the one with the maximum difference between the expected expected utility before and after asking it[19,20].

VOI Based Elicitation

Policy

$$\pi^t = \left(\omega^t, \omega^{t+1}, \omega^N\right) \text{ where } \omega^x \in \Omega \; K\left(\omega|\pi\right) \equiv \text{index of } \omega \text{ in } \pi, \; \pi\left(k\right) = \omega \text{ where } K\left(\omega|\pi\right) = k$$

Probability of Agreement

$$Pa^t\left(\omega|\pi\right) = \begin{cases} \Lambda^t\left(\omega\right)\prod_{k=1}^{K_\pi(\omega)-1}\left(1-\Lambda^t\left(\pi\left(k\right)\right)\right) & \omega \in \pi \\ 0 & otherwise \end{cases}$$

Expected Expected Utility[21]

$$EEU^t\left(\pi, \left\{\hat{U}^t_\omega\right\}\right) = \sum_{\omega\in\Omega} Pa\left(\omega|\pi\right)\mathbb{E}\left(\hat{U}^t_\omega\right)$$

Optimal Policy

$$\pi^{t*} = \arg\max_\pi EEU^t\left(\pi, \left\{\hat{U}^t_\omega\right\}\right)$$

VOI Based Elicitation

Policy

$$\pi^t = \left(\omega^t, \omega^{t+1}, \omega^N\right) \text{ where } \omega^x \in \Omega \; K\left(\omega|\pi\right) \equiv \text{index of } \omega \text{ in } \pi, \; \pi\left(k\right) = \omega \text{ where } K\left(\omega|\pi\right) = k$$

Probability of Agreement

$$Pa^t\left(\omega|\pi\right) = \begin{cases} \Lambda^t\left(\omega\right)\prod_{k=1}^{K_\pi(\omega)-1}\left(1-\Lambda^t\left(\pi\left(k\right)\right)\right) & \omega \in \pi \\ 0 & otherwise \end{cases}$$

Expected Expected Utility[21]

$$EEU^t\left(\pi, \left\{\hat{U}^t_\omega\right\}\right) = \sum_{\omega\in\Omega} Pa\left(\omega|\pi\right)\mathbb{E}\left(\hat{U}^t_\omega\right)$$

Optimal Policy

$$\pi^{t*} = \arg\max_\pi EEU^t\left(\pi, \left\{\hat{U}^t_\omega\right\}\right)$$

VOI Based Elicitation

Policy

$$\pi^t = \left(\omega^t, \omega^{t+1}, \omega^N\right) \text{ where } \omega^x \in \Omega \; K\left(\omega|\pi\right) \equiv \text{index of } \omega \text{ in } \pi, \; \pi\left(k\right) = \omega \text{ where } K\left(\omega|\pi\right) = k$$

Probability of Agreement

$$Pa^t\left(\omega|\pi\right) = \begin{cases} \Lambda^t\left(\omega\right) \prod_{k=1}^{K_\pi(\omega)-1} \left(1 - \Lambda^t\left(\pi\left(k\right)\right)\right) & \omega \in \pi \\ 0 & otherwise \end{cases}$$

Expected Expected Utility[21]

$$EEU^t\left(\pi, \left\{\hat{U}^t_\omega\right\}\right) = \sum_{\omega\in\Omega} Pa\left(\omega|\pi\right) \mathbb{E}\left(\hat{U}^t_\omega\right)$$

Optimal Policy

$$\pi^{t*} = \arg\max_\pi EEU^t\left(\pi, \left\{\hat{U}^t_\omega\right\}\right)$$

VOI Based Elicitation II

Questions

$$Q \equiv \{q_l\}$$
$$q_l \equiv \left\{\left(Ans_s^l, p_s\right)\right\}$$

Answers

$$Ans_s^l \equiv \left\{\hat{U}_\omega^{t+1}\right\}$$
$$\sum_s p_s = 1$$

Expected value of information

$$EVOI\left(q^l, \left\{\hat{U}_\omega^t\right\}\right) = \mathbb{E}_s\left(\max_\pi EEU\left(\pi, Ans_s^l\right)\right) - \max_\pi EEU\left(\pi, \left\{\hat{U}_\omega^t\right\}\right)$$

Elicitation

Ask q^* where

$$q^* = \arg\max_q \left(EVOI\left(q^l, \left\{\hat{U}_\omega^t\right\}\right) - c_q\right)$$

c_q Cost of asking question q

VOI Based Elicitation II

Questions

$$Q \equiv \{q_l\}$$
$$q_l \equiv \left\{\left(Ans_s^l, p_s\right)\right\}$$

Answers

$$Ans_s^l \equiv \left\{\hat{U}_\omega^{t+1}\right\}$$
$$\sum_s p_s = 1$$

Expected value of information

$$EVOI\left(q^l, \left\{\hat{U}_\omega^t\right\}\right) = \mathbb{E}_s\left(\max_\pi EEU\left(\pi, Ans_s^l\right)\right) - \max_\pi EEU\left(\pi, \left\{\hat{U}_\omega^t\right\}\right)$$

Elicitation

Ask q^* where

$$q^* = \arg\max_q \left(EVOI\left(q^l, \left\{\hat{U}_\omega^t\right\}\right) - c_q\right)$$

c_q Cost of asking question q

VOI Based Elicitation II

Questions

$$Q \equiv \{q_l\}$$
$$q_l \equiv \left\{\left(Ans_s^l, p_s\right)\right\}$$

Answers

$$Ans_s^l \equiv \left\{\hat{U}_\omega^{t+1}\right\}$$
$$\sum_s p_s = 1$$

Expected value of information

$$EVOI\left(q^l, \left\{\hat{U}_\omega^t\right\}\right) = \mathbb{E}_s\left(\max_\pi EEU\left(\pi, Ans_s^l\right)\right) - \max_\pi EEU\left(\pi, \left\{\hat{U}_\omega^t\right\}\right)$$

Elicitation

Ask q^* where

$$q^* = \arg\max_q \left(EVOI\left(q^l, \left\{\hat{U}_\omega^t\right\}\right) - c_q\right)$$

c_q Cost of asking question q

VOI main Issues

Accurate Agreement Model Assumption

- Everything depends on the probability of agreement (Pa)
- Pa depends on the **product** of probabilities in the acceptance model (Λ^t)

$$Pa^t\left(\omega|\pi\right) = \begin{cases} \Lambda^t\left(\omega\right)\prod_{k=1}^{K_\pi(\omega)-1}\left(1-\Lambda^t\left(\pi\left(k\right)\right)\right) & \omega \in \pi \\ 0 & otherwise \end{cases}$$

Speed: Complexity $= O\left(nN|Q||Ans|\right)$

- Too many *argmax* and $\mathbb{E}$ operations.
- Every policy extends to the end of the negotiation.

$$q^* = \arg\max_q\left(EVOI\left(q^l, \left\{\hat{U}^t_\omega\right\}\right) - c_q\right)$$

$$EVOI\left(q^l, \left\{\hat{U}^t_\omega\right\}\right) = \mathbb{E}_s\left(\max_\pi EEU\left(\pi, Ans^l_s\right)\right) - \max_\pi EEU\left(\pi, \left\{\hat{U}^t_\omega\right\}\right)$$

$$\pi^{t*} = \arg\max_\pi EEU^t\left(\pi, \left\{\hat{U}^t_\omega\right\}\right)$$

VOI main Issues

Accurate Agreement Model Assumption

- Everything depends on the probability of agreement (Pa)
- Pa depends on the **product** of probabilities in the acceptance model (Λ^t)

$$Pa^t(\omega|\pi) = \begin{cases} \Lambda^t(\omega)\prod_{k=1}^{K_\pi(\omega)-1}\left(1-\Lambda^t(\pi(k))\right) & \omega \in \pi \\ 0 & otherwise \end{cases}$$

Speed: Complexity $= O(nN|Q||Ans|)$

- Too many *argmax* and $\mathbb{E}$ operations.
- Every policy extends to the end of the negotiation.

$$q^* = \arg\max_q \left(EVOI\left(q^l, \left\{\hat{U}_\omega^t\right\}\right) - c_q\right)$$

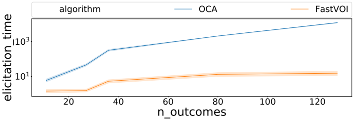
$$EVOI\left(q^l, \left\{\hat{U}_\omega^t\right\}\right) = \mathbb{E}_s\left(\max_\pi EEU\left(\pi, Ans_s^l\right)\right) - \max_\pi EEU\left(\pi, \left\{\hat{U}_\omega^t\right\}\right)$$

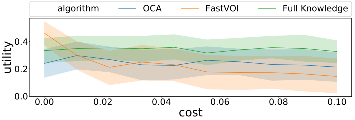
$$\pi^{t*} = \arg\max_\pi EEU^t\left(\pi, \left\{\hat{U}_\omega^t\right\}\right)$$

Extending VOI

FastVOI[22]

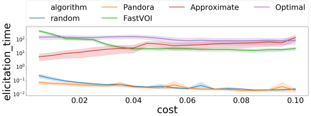
- A faster approximate version of VOI

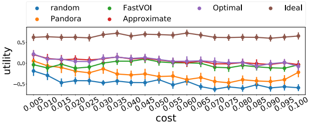




OptimalVOI[23]

- Extends Applicability to Infinite N. Questions.

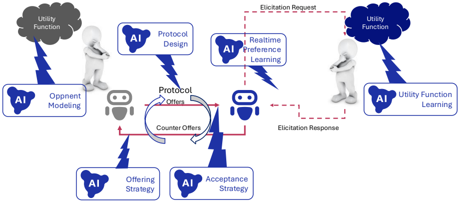




Wrap up

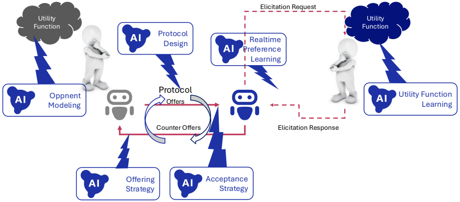
Wrap Up

- Automated negotiation (AN) is a well-defined long-standing challenging problem in multi-agent coordination.
- AN is becoming more relevant to real world business applications due to the faster pace of automation.
- Data-driven approaches have been applied to almost all facets of automated negotiation.
- negmas-rl simplifies the process of developing such data-driven approaches.



Wrap Up

- Automated negotiation (AN) is a well-defined long-standing challenging problem in multi-agent coordination.
- AN is becoming more relevant to real world business applications due to the faster pace of automation.
- Data-driven approaches have been applied to almost all facets of automated negotiation.
- negmas-rl simplifies the process of developing such data-driven approaches.



Thank you

y.mohammad@nec.com

References I

Aydoğan, Reyhan et al. "Alternating offers protocols for multilateral negotiation". In: *Modern Approaches to Agent-based Complex Automated Negotiation*. Springer, 2017, pp. 153–167.

Baarslag, Tim and Enrico H Gerding. "Optimal Incremental Preference Elicitation during Negotiation.". In: *IJCAI*. 2015, pp. 3–9.

Baarslag, Tim, Koen Hindriks, and Catholijn Jonker. "A tit for tat negotiation strategy for real-time bilateral negotiations". In: *Complex Automated Negotiations: Theories, Models, and Software Competitions*. Springer, 2013, pp. 229–233.

— . "Effective acceptance conditions in real-time automated negotiation". In: *Decision Support Systems* 60 (2014), pp. 68–77.

Baarslag, Tim and Michael Kaisers. "The value of information in automated negotiation: A decision model for eliciting user preferences". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2017, pp. 391–400.

Baarslag, Tim et al. "Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies". In: *Novel Insights in Agent-based Complex Automated Negotiation*. Ed. by Ivan Marsa-Maestre et al. Tokyo: Springer Japan, 2014, pp. 61–83. ISBN: 978-4-431-54758-7. DOI: 10.1007/978-4-431-54758-7_4. URL: https://doi.org/10.1007/978-4-431-54758-7_4.

Bakker, Jasper et al. "RLBOA: A modular reinforcement learning framework for autonomous negotiating agents". In: *Proceedings of the 18th international conference on autonomous agents and multiagent systems*. 2019, pp. 260–268.

Boutilier, Craig. "On the foundations of expected expected utility". In: *IJCAI*. Vol. 3. Citeseer. 2003, pp. 285–290.

Chajewska, Urszula, Daphne Koller, and Ronald Parr. "Making rational decisions using adaptive utility elicitation". In: *AAAI/IAAI*. 2000, pp. 363–369.

Hindriks, Koen and Dmytro Tykhonov. "Opponent modelling in automated multi-issue negotiation using bayesian learning". In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. 2008, pp. 331–338.

References II

- Kawata, Ryohei and Katsuhide Fujita. "Meta-Strategy Based on Multi-Armed Bandit Approach for Multi-Time Negotiation". In: *IEICE TRANSACTIONS on Information and Systems* 103.12 (2020), pp. 2540–2548.
- Krimpen, Thijs van, Daphne Looije, and Siamak Hajizadeh. "HardHeaded". In: *Complex Automated Negotiations: Theories, Models and Software Competitions*. Ed. by Ito Takayuki et al. Springer, 2013, pp. 223–227.
- Larsen, Bradley J. "The efficiency of real-world bargaining: Evidence from wholesale used-auto auctions". In: *The Review of Economic Studies* 88.2 (2021), pp. 851–882.
- Lin, Raz et al. "Genius: An Integrated Environment for Supporting the Design of Generic Automated Negotiators". In: *Computational Intelligence* 30.1 (2014), pp. 48–70. ISSN: 1467-8640. DOI: 10.1111/j.1467-8640.2012.00463.x. URL: http://dx.doi.org/10.1111/j.1467-8640.2012.00463.x.
- Mohammad, Yasser, Amy Greenwald, and Shinji Nakadai. "NegMAS: A platform for situated negotiations". In: *Twelfth International Workshop on Agent-based Complex Automated Negotiations (ACAN2019) in conjunction with IJCAI*. Macau, China, 2019. URL: https://github.com/yasserfarouk/negmas.
- Mohammad, Yasser and Shinji Nakadai. "Fastvoi: Efficient utility elicitation during negotiations". In: *International Conference on Principles and Practice of Multi-Agent Systems*. Springer. 2018, pp. 560–567.
- — ."Optimal value of information based elicitation during negotiation". In: *Proceedings of the 18th international conference on autonomous agents and multiagent systems*. AAMAS '19. Montreal QC, Canada: International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 242–250.
- — ."Utility Elicitation During Negotiation with Practical Elicitation Strategies". In: *IEEE SMC*. 2018.
- Neumann, J. von and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947.

References III

- Razeghi, Yousef, Celal Ozan Berk Yavuz, and Reyhan Aydoğan. "Deep reinforcement learning for acceptance strategy in bilateral negotiations". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 28.4 (2020), pp. 1824–1840.
- Rubinstein, Ariel. "Perfect equilibrium in a bargaining model". In: *Econometrica: Journal of the Econometric Society* (1982), pp. 97–109.
- Sengupta, Ayan, Yasser Mohammad, and Shinji Nakadai. "An Autonomous Negotiating Agent Framework with Reinforcement Learning Based Strategies and Adaptive Strategy Switching Mechanism". In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '21. Virtual Event, United Kingdom: International Foundation for Autonomous Agents and Multiagent Systems, 2021, pp. 1163–1172. ISBN: 9781450383073.
- Takahashi, Toki et al. "VeNAS: Versatile Negotiating Agent Strategy via Deep Reinforcement Learning". In: *AAAI 2022*. 2022.
- Weitzman, Martin L. "Optimal search for the best alternative". In: *Econometrica: Journal of the Econometric Society* (1979), pp. 641–654.

\Orchestrating a brighter world