# Automated Negotiation: Challenges and Tools

**Yasser Mohammad**[1,2,3] and **Amy Greenwald**[4]

[1] NEC CORPORATION, Global Innovation Unit

[2] National Institute of Advanced Industrial Science and Technology (AIST)

[3] Assiut University, Egypt

[4] Brown University

February 23rd, 2022

# Outline

# Why Now?

1. Industries are moving online.
2. Automation: Factory floor
   The back office.

# Outline

1. Introduction and Classic Results (45min)
2. Protocols, Strategies and Platforms (45min)
    1. Hands On Experience
3. Recent Advances (45min):
4. Supply Chain Management Competition (30min)
    1. Hands On Experience
5. Challenges and Open Problems (30min)

# Materials

1. Tutorial Website: http://yasserm.com/aaai2022tutorial-automated_negotiation_challenges_and_tools/
2. Github Repository:

# Q&A

# Automated Negotiation: Challenges and Tools

## Introduction and Classic Results

**Yasser Mohammad**[1,2,3] and **Amy Greenwald**[4]

[1] NEC CORPORATION, Global Innovation Unit

[2] National Institute of Advanced Industrial Science and Technology (AIST)

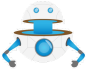
[3] Assiut University, Egypt

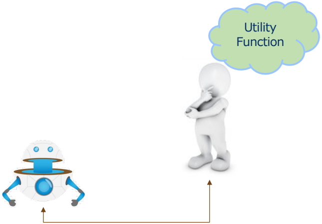
[4] Brown University

February 23rd, 2022

# Outline

# Outline




Utility
Function


Utility
Function
Utility
Function

# Why not Auctions

## or when to not use automated negotiation

# Definition

## Negotiation

$$\Upsilon \equiv \left(A, T, N, \Omega, M, \left\{\tilde{P}_a, P^n_{ab}\ \forall 1 \leq a, b \leq A,\ 0 \leq n \leq N\right\}\right)$$

$A \in \mathbf{I}^+ - \{1\}$: Number of agents/actors.

$T \in \mathbf{R} \bigcup \infty$: The allowed time of the negotiation.

$N \in \mathbf{I} \bigcup \infty$: The allowed number of rounds of the negotiation.

$\Omega \equiv \{\omega_j\} \bigcup \phi$: Possible outcomes including $\phi$ signifying disagreement.

$M$ The negotiation *mechanism* (protocol) defining rules of encounter for agents.

$\tilde{P}_a$: Preferences of **actor** $a$.

$P^n_{ab}$: Information available to **agent** $a$ about preferences of **actor** $b$ at the beginning of round $n$.

## Known Preferences Assumption

# Preference Representations

## Preference Types

Partial Ordering $\succcurlyeq$ Defines preference as a partial ordering over $\Omega$.

Ranking A total ordering over a subset of $\Omega$.

Utility Function $\tilde{u}$ Defines a numeric value for every outcome in $\Omega$.

$$\tilde{u} : \Omega \rightarrow \Re$$

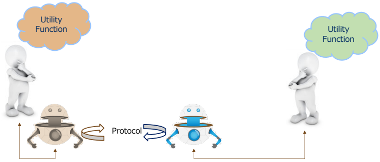
Probabilistic Utility Function $u$ Defines a distribution of values.

$$u : \Omega \times \Re \rightarrow [0, 1]$$

## Known Ufun Assumption

$$u_a^t(\omega, x) = u_a^0(\omega, x) = \begin{cases} 1 & \tilde{u}(\omega) = x \\ 0 & otherwise \end{cases}$$

# Components of the Negotiation Problem



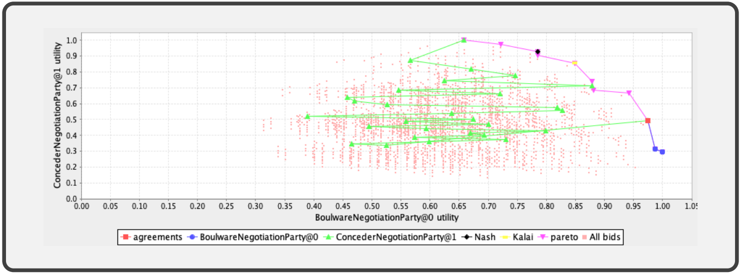

Negotiation Protocol Defines how negotiation is to be conducted [Mechanism Design Problem].

- Alternating Offers Protocol
- Single Text Protocol
- ...

Negotiation Strategy Defines how an agent behaves during the negotiation [Effective Negotiation Problem].

- Time-based strategies: Boulware, conceder, ...
- Tit-for-tat variations
- ...

# 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.

**Sub-game Perfect Equilibrium** A Nash Equilibrium in every sub-game.

# Types of Automated Negotiation Problems

NEC-AIST AI Cooperative Research Laboratory

## Negotiator type

1. Agent-Agent negotiation
2. Agent-Human negotiation

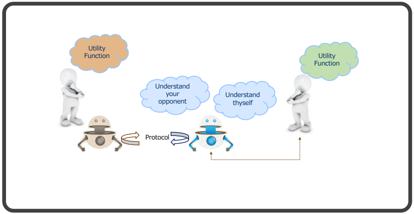
## Number of negotiators

1. Bilateral negotiation
2. Multilateral negotiation

## Outcome Space

1. Single Issue: $\Omega = \{\omega_0, \omega_1, ....\}$
2. Multiple Issues: $\Omega = \prod_{i=1}^{n_i} I_i$

## Protocol Type



# Outline

# Nash Bargaining Game: Solution

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

$$\underset{\omega_1,\omega_2}{\arg\max} \prod_{i=1}^{2} \left(\tilde{u}_i(\omega_i) - \tilde{u}_i(\phi)\right)$$

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

$$\underset{\omega_1,\omega_2 \in F}{\arg\max} \left(\omega_1 + \omega_2\right) \text{ s.t. } \left(\frac{\tilde{u}_1(\omega_1) - \tilde{u}_1(\phi)}{\tilde{u}_2(\omega_2) - \tilde{u}_2(\phi)} = \frac{\max_{v \in F}\left(\tilde{u}_1(v)\right) - \tilde{u}_1(\phi)}{\max_{v \in F}\left(\tilde{u}_2(v)\right) - \tilde{u}_2(\phi)}\right)$$

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

$$\underset{\omega_1,\omega_2 \in P}{\arg\max} \min_{i \in \{1,2\}} \left(\tilde{u}_i(\omega_i) - \tilde{u}_i(\phi)\right)$$

# Rubinstein's Bargaining Protocol: Description

- Two agents sharing a pie.
- Each agent is under a different time-pressure: $\tilde{u}_i^{t+\Delta}(\omega) < \tilde{u}_i^t(\omega)$. Examples of time-pressure:
  - Exponential $\tilde{u}_i^{t+\Delta}(\omega) = \delta_i^{\Delta} u_i^t(\omega)$.
  - Linear $\tilde{u}_i^{t+\Delta}(\omega) = u_i^t(\omega) - \Delta c_i$
- Actor's initial utility is the assigned part of the pie: $\tilde{u}_i^0 = \omega_i$.
- 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:

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

# Negotiation With Full information

## Hick's Paradox

Why do rational parties negotiate when they have full information?

## Becuase the world exists!! [Fernandez and Glazer, 1989]

- A union negotiating with management about a wage raise in rounds.
- The union *can* strike.
- Both parties are perfectly rational and fully informed.

**Main Findings:**

- Sub-game perfect equilibria exist in which there is some finite striking time followed by agreement.
- That happens in real time even when round length goes to zero.

# Negotiation With Incomplete Information

NEC-AIST AI Cooperative Research Laboratory

## Impossibility Result

Define a good mechanism as:

- Incentive compatible.
- No external subsidy.

Assuming rationality, there is *no* good mechanism that can guarantee agreement when it is dominant [Myerson and Satterthwaite, 1983].

## Example

- A buyer values a product at $v$.
- A seller can create the product at cost $c$.
- $v > c$.
- There is no way to design a good mechanism that results in agreement for all $v, c$ values.

# Outline

# References I

Fernandez, R. and Glazer, J. (1989). Striking for a bargain between two completely informed agents. Technical report, National Bureau of Economic Research.

Myerson, R. B. and Satterthwaite, M. A. (1983). Efficient mechanisms for bilateral trading. *Journal of economic theory*, 29(2):265–281.

# Automated Negotiation: Challenges and Tools

## Protocols and Strategies

**Yasser Mohammad**[1,2,3] and **Amy Greenwald**[4]

[1] NEC CORPORATION, Global Innovation Unit

[2] National Institute of Advanced Industrial Science and Technology (AIST)

[3] Assiut University, Egypt

[4] Brown University

February 23rd, 2022

# Outline

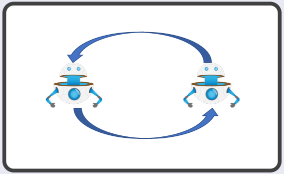
# Outline

# Unmediated Protocols

NEC-AIST AI Cooperative Research Laboratory

## 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.

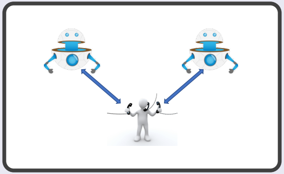
Rubinstein Bargaining Protocol: Infinite horizon, single issue, bilateral protocol with complete information [Rubinstein, 1982].

Stacked Alternating Offers Protocol: Finite horizon, multi-issue, multilateral protocol with partial information

# 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.

# Stacked Alternating Offers Protocol



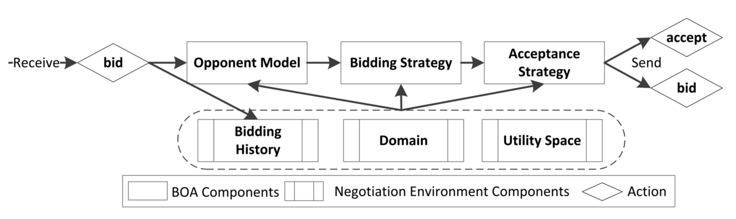
```
n_agreed, current = 0, randint(0, n_agents)
offer = agents[current].offer()
```

```
while True:
if timedout():
return 'TIME_OUT'
current = (current + 1) % n_agents
response = agents[current].respond(offer)
if response == 'accept':
n_agreed += 1
if n_agreed == n_agents:
return offer  # contract
elif response == 'end_negotiation':
return 'FAILURE'
elif response == 'reject':
offer = agents[current].offer()
```

# Outline

# Outline

# Negotiator Components [Baarslag et al., 2014][1]




## OBA Atchitecture

Opponent Model predicts opponent behavior.

Bidding Strategy Generates new bids.

Acceptance Strategy Decides when to accept.

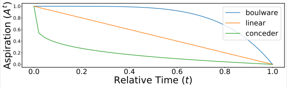
[1] Supported by Genius

# Bidding Strategy

NEC-AIST AI Cooperative Research Laboratory

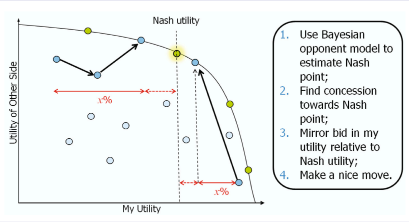
## Time-based strategies

Offer an outcome with a utility just above the current *aspiration level* which is monotonically decreasing.




## (Nice) Tit-for-Tat (bilateral) [Baarslag et al., 2013]

Concede as much as the opponent and do not retaliate.



# Opponent Modeling

NEC-AIST AI Cooperative Research Laboratory

## What is being modeled?

- Opponent preferences.
- Opponent strategy.
- Acceptance probability.
- Future offers.
- Opponent Type.

## When is it modeled?

- Before the negotiation.
- During the negotiation.

## Data

- This negotiation vs. past negotiations.
- This opponent vs. this opponent group vs. others.
- Exchanged offers vs. agreements

# Opponent Model: Example

## Bayesian Learning

**Hypothesis** A hypothesis about the opponent's behavior.

**Evidence** Behavior of the agent (e.g. its offers/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).

# Acceptance Strategy

## Examples

Accept if the utility of the offer $\succ$

- **Previous** my last offer.
- **Current** what I am about to offer.
- **Expected** the best offer I expect to receive (needs an opponent model).
- **Threshold** an offer with the current utility threshold ($\tau$).
  - **Constant** May be a fraction of maximum utility.
  - **Time-based** Monotonically decreasing with time.
  - **Predictive** Predicts the expected/max utility on rejection (e.g. Gaussian Process).

# Outline

# Platforms [Used in this tutorial]

## Genius [Lin et al., 2014]

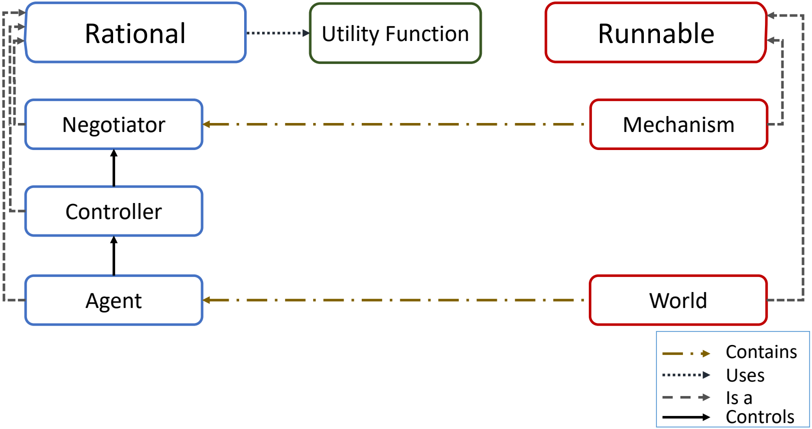
a Java-based negotiation platform to develop general negotiating agents and create negotiation scenarios. The platform can simulate negotiation sessions and tournaments and provides analytical tools to evaluate the agents' performance.

## NegMAS [Mohammad et al., 2019]

a Python-based negotiation platform for developing autonomous negotiation agents embedded in simulation environments.The main goal of NegMAS is to advance the state of the art in situated simultaneous negotiations.

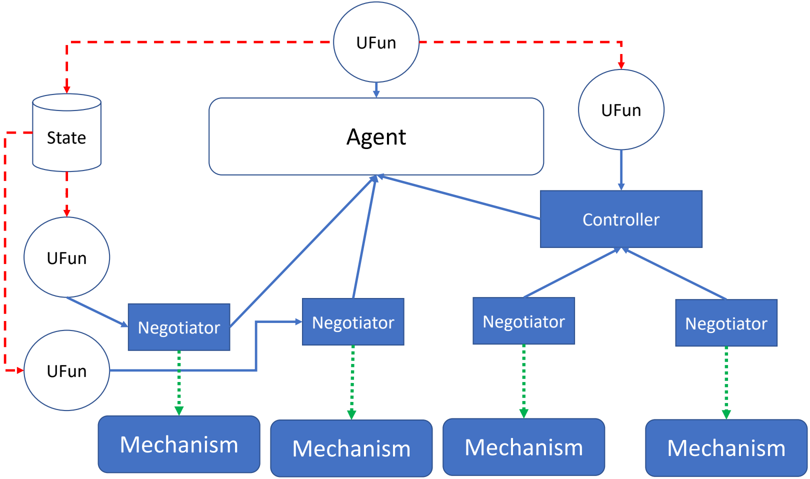
# Outline

# NegMAS[2] in two slides

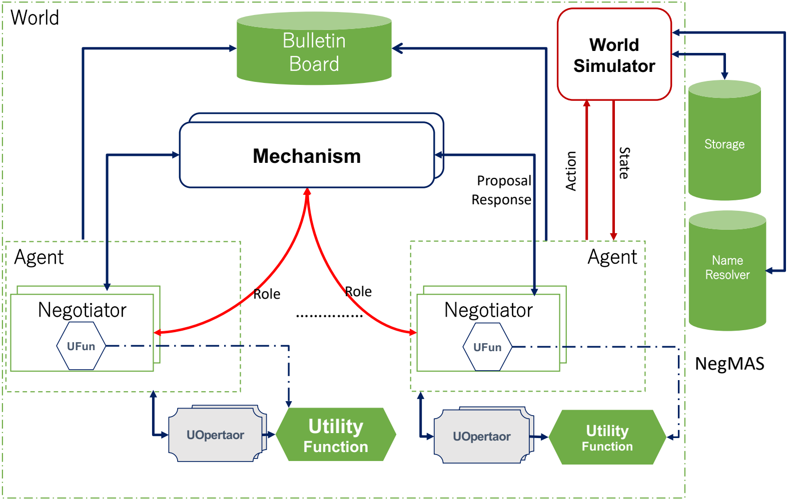



[2] https://www.github.com/yasserfarouk/negmas

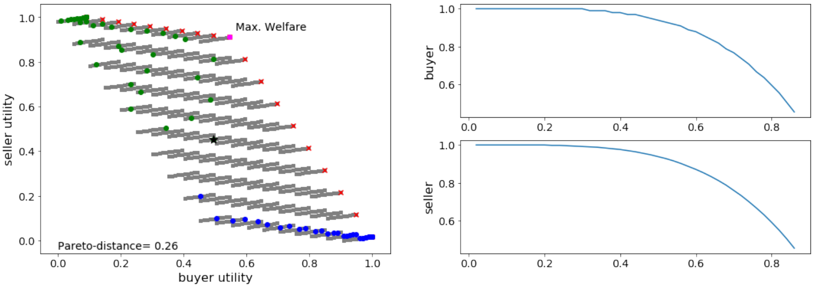
# NegMAS in two slides



# NegMAS in two slides (... OK 3)



# NegMAS in two slides (... really!!!)




- An Example negotiation.
- Can you spot a problem?

# Outline

# References I

Aydoğan, R., Festen, D., Hindriks, K. V., and Jonker, C. M. (2017). Alternating offers protocols for multilateral negotiation. In *Modern Approaches to Agent-based Complex Automated Negotiation*, pages 153–167. Springer.

Baarslag, T., Hindriks, K., Hendrikx, M., Dirkzwager, A., and Jonker, C. (2014). *Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies*, pages 61–83. Springer Japan, Tokyo.

Baarslag, T., Hindriks, K., and Jonker, C. (2013). A tit for tat negotiation strategy for real-time bilateral negotiations. In *Complex Automated Negotiations: Theories, Models, and Software Competitions*, pages 229–233. Springer.

Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., and Jonker, C. M. (2014). Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30(1):48–70.

# References II

Mohammad, Y., Greenwald, A., and Nakadai, S. (2019). Negmas: A platform for situated negotiations. In *Twelfth International Workshop on Agent-based Complex Automated Negotiations (ACAN2019) in conjunction with IJCAI*.

Rubinstein, A. (1982). Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, pages 97–109.

# Automated Negotiation: Challenges and Tools

## Recent Advances

**Yasser Mohammad**[1,2,3] and **Amy Greenwald**[4]

[1] NEC CORPORATION, Global Innovation Unit

[2] National Institute of Advanced Industrial Science and Technology (AIST)

[3] Assiut University, Egypt

[4] Brown University

February 23rd, 2022

# Outline

# Outline

# Negotiation With Incomplete Information

# Outline

# Recent Optimality Results

# Outline

# Supervised Learing in Automated Negotiation

# Outline

# Reinforcement Learing in Automated Negotiation

# Automated Negotiation: Challenges and Tools

## Automagted Negotiation is SCM

**Yasser Mohammad**[1,2,3] and **Amy Greenwald**[4]

[1] NEC CORPORATION, Global Innovation Unit

[2] National Institute of Advanced Industrial Science and Technology (AIST)

[3] Assiut University, Egypt

[4] Brown University

February 23rd, 2022

# Outline

# Outline

# Negotiation in SCM Business

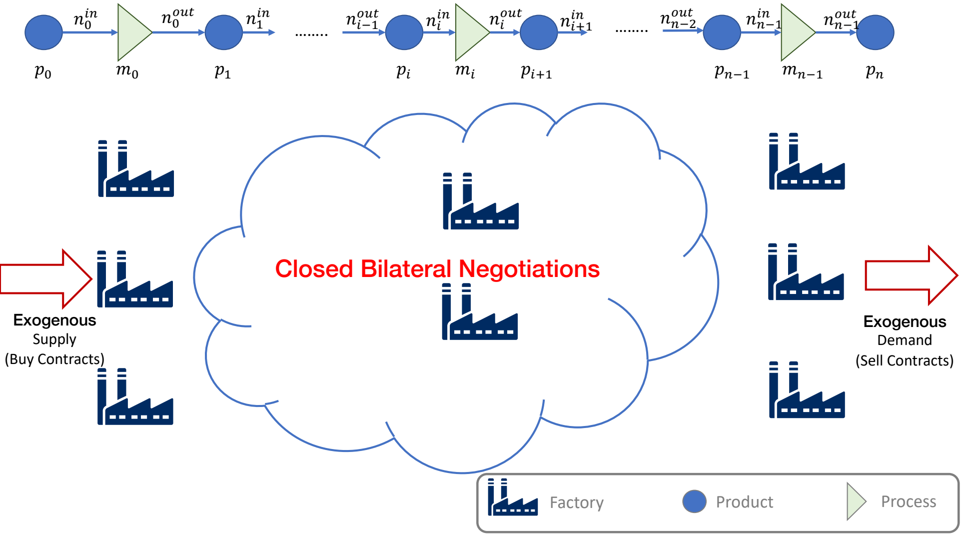
CONTRACTROOM pactum

- Human negotiations lead to an estimated 17-40% *value leakage* in some estimates [1]
- A recent study suggests that at least 15 companies are working in *contracting support systems* [2].
- A recent UNECE UN/CEFACT proposal to standardize negotiation protocols for SCM and other applications [3]
- More to come [Mohammad et al., 2019].

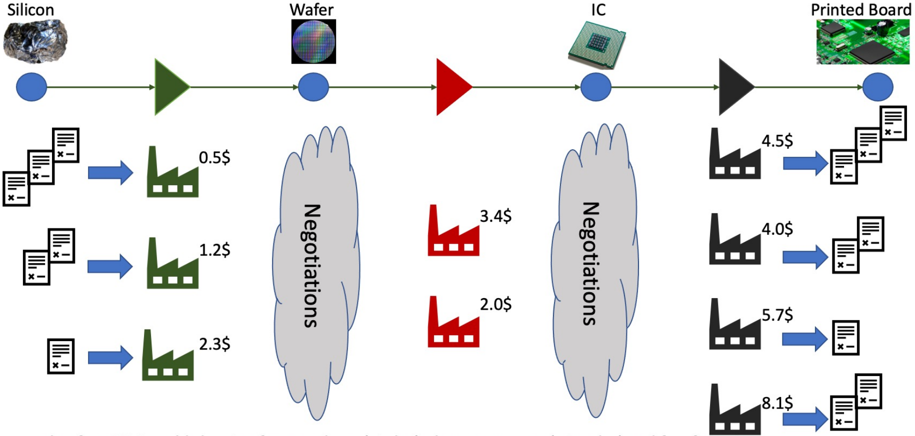
[1] KPMG report: https://bit.ly/3kDRy6I
[2] Forrester report: https://bit.ly/3nwXEaY
[3] UN/CEFACT Project website: https://bit.ly/38LOsLX

# Outline

# SCML Competition [Mohammad et al., 2019]



# Example Configuration



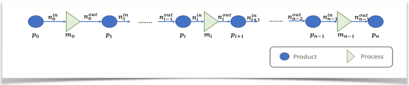

An example of an SCM world showing four products (circles), three processes (triangles) and few factories.
Each process consumes one item of its input and generates one output of its output in one day.
Each factory requires a different cost to run its process (shown in its top right).
Factories in the first level have exogenous contracts to buy raw material (silicon) and factories at the last level have exogenous contracts to sell the final product (printed boards). These contracts drive the market.

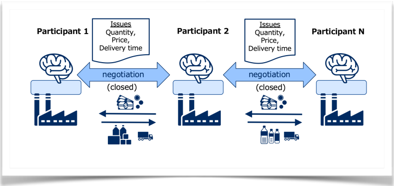
# SCML World

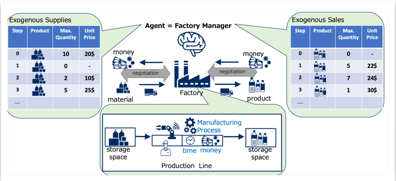


## Challenge

- Turn maximize profit into a ufun!!
- Dynamic interdependent ufuns.
- Sequential negotiations.
- Concurrent Negotiations.
- Negotiation under uncertainty.
- Adaptation and learning.
- Trust management.




## Information

- **Website** https://scml.cs.brown.edu/

# SCML Competition

## Competition Details

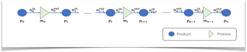
- Runs as part of ANAC IJCAI.
- You control one or more factories.
  - **Standard track** $\rightarrow$ one factory.
  - **Collusion track** $\rightarrow$ multiple factories (3).

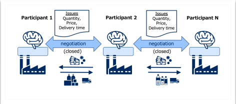
## Score Evaluation

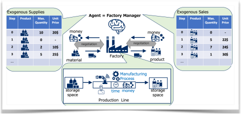
- **Per instantiation**: Total profit counting inventory at half the trading price.
- **Total**: median of per-instantiation scores.

## Flavors

- Online competition at https://scml.cs.brown.edu



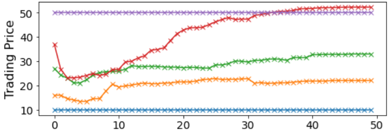

# Simulation Steps

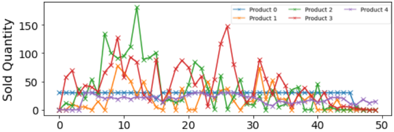


**Simulation Steps**

# Trading prices






## What is trading price and why is it calculated?

- A value calculated by the system for each product.
- Represents some estimate of the current price.
- Never revealed to agents.
- Usages:
  - Used at the end to value inventory.
  - Used when calculating spot-market price. during breach processing.

## How does the system calculate it?

$$\mathrm{tp}(p, s) = \frac{\beta^{s+1}\ Q_{-1}(p)\ \mathrm{cat}(p) + \sum_{i=0}^{s} \beta^{s-i}\ Q_i(p)\ \mu_i(p)}{\beta^{s+1} + \sum^{s} \cdots \beta^{s-i} \cdots},$$

# Trading prices: The details

NEC-AIST AI Cooperative Research Laboratory

## Quantities and prices

$$Q_i(p') = \sum_{\{c \in C^i | c.p = p'\}} c.\bar{q}$$

$$\mu_i(p') = \frac{\sum_{\{c \in C^i | c.p = p'\}} c.\bar{q} \times c.u}{Q_i(p')}$$

## How does the system calculate it?

$$\text{tp}(p, s) = \frac{\beta^{s+1}\ Q_{-1}(p)\ \text{cat}(p) + \sum_{i=0}^{s} \beta^{s-i}\ Q_i(p)\ \mu_i(p)}{\beta^{s+1} + \sum_{i=0|Q_i(p)>0}^{s} \beta^{s-i}},$$

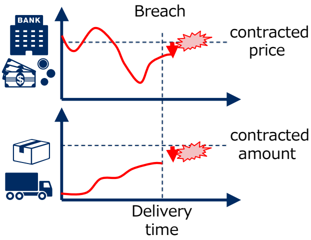
# When things go wrong

## What is a breach

- Insufficient funds or insufficient inventory.

## Breach Processing

- A breach report is **always** published.
  - who and fraction.
- Insufficient products → **forced to** buy from the spot market.
- Insufficient funds → bankruptcy.
  - bankruptcy → liquidation.



# Spot Market

NEC-AIST AI Cooperative Research Laboratory

## Spot market

- **No-choice:** Can be used **only** for insufficient inventory breaches.
- **Penalizing:** Entails higher price than the trading price
- **Personalized:** More breaches → higher spot price **for you**

## Calculation

- Agent's spot price penalty: $\mathrm{ip}_a(p, s) = \lambda \sum_{i=0}^{s} \alpha^{s-i} q_a(p, i)$ ,
- Breaching agents buy **expensive**:

$$\mathrm{tp}(p, s) \times (1 + \mathrm{gp}) \times (1 + \mathrm{ip}_a(p, s))$$

- Bankrupt agents are liquidated **cheap**:

$$\mathrm{tp}(p, s)/((1 + \mathrm{gp}) \times (1 + \mathrm{ip}_a(p, s))$$

# Bankruptcy Processing

## Bankruptcy conditions

- Insufficient money for a buy contract.
- Insufficient money to buy from the spot market for as sell contract.

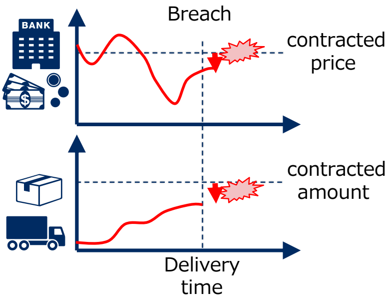
## What exactly happens?

1. The agent is stopped from every buying or selling.
2. Its inventory is sold on the spot market.
3. All agents are informed.
4. Agents with future contracts with it are informed about the expected level of breach.
5. The agent's score is set to $-1$.

# When things go wrong: Summary

## Summary

- An unfulfilled contract is reported to the **breach-list** (who and fraction).
- Insufficient funds → bankrupt.
- Insufficient product → buy at high cost.
  - Cannot buy → bankrupt.
- Bankrupt → **really really bad**
  - No more trade.
  - Very low score.
  - All inventory is liquidated.
  - May hurt other agents.



# Outline

# Agent Knowledge

## About itself

- **Its capabilities:** lines and production cost.
- **Its location:** input and output products.
- **Its partners:** suppliers and consumers (and competitors).
- **Its state:** inventory, wallet, contracts, and negotiations.

## About the market

- The production graph and factories at each level.
- Time and simulation length.

## About others

- **Financial Reports:** balance, assets, breach fraction/probability.
- **Past Interactions:** negotiations and contracts between itself and that agent.

# Information about previous simulations

- Agents **cannot** pass information between simulations.
- We release datasets from the online competition including:
  - World parameters.
  - Agent types.
  - Financial reports.
  - Contracts.
  - Breaches.
- This data is anonymous:
  - You **cannot** associate an agent type with a specific participant.
  - You **can** associate agent types across simulations.

# Development Approaches

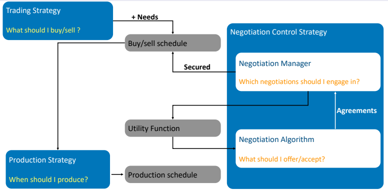
NEC-AIST AI Cooperative Research Laboratory

## Monolithic Agent

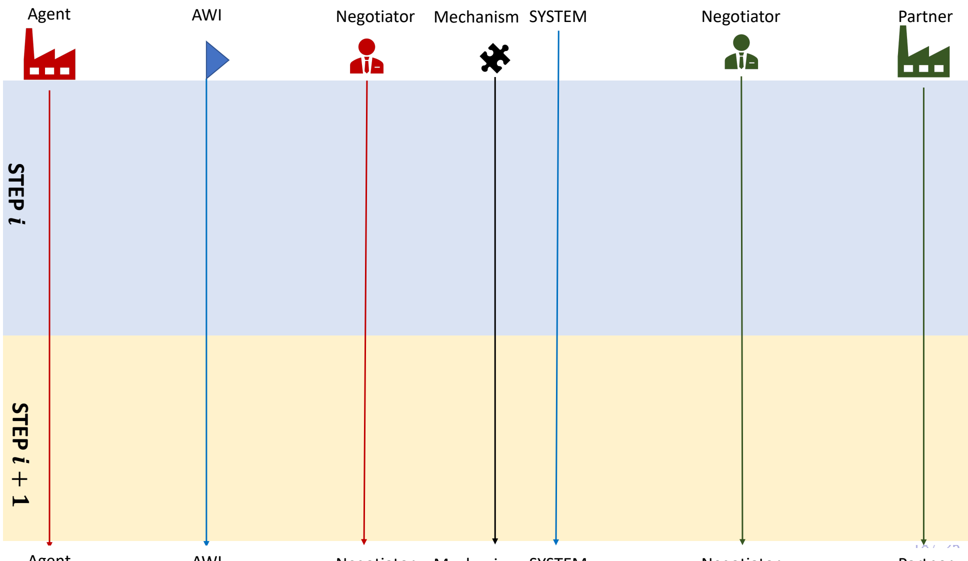
- Respond to callbacks in the Agent class.
- Functionality is distributed **among callbacks**.
- Everything is in one place (the agent class).
- Harder to reuse.

## Component-Based Agent

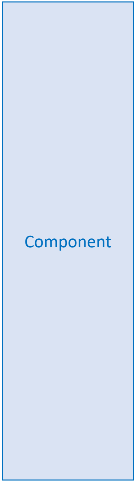
- Divides the agent into semi-independent components.
- Functionality is distributed between **components**.
- Easier to reuse.



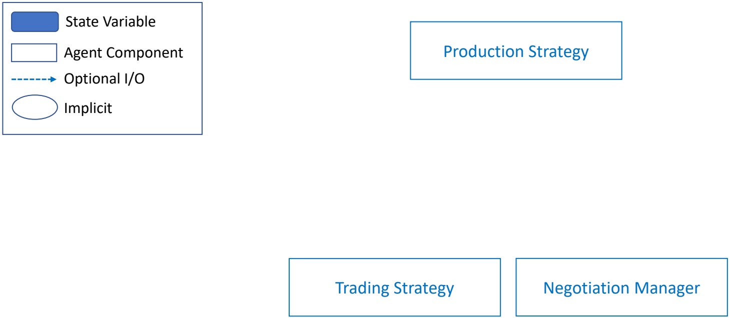
# Callbacks and Timing



# What is a component?



# SCML Agent Components



# Trading Strategy

## Role

- Decides an overall **business plan**.
- Keeps track of buy/sell **needs**.

## Built-in Options

No Strategy yes, just do nothing.

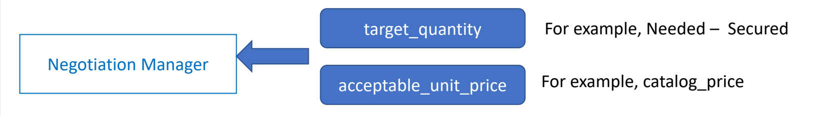
Reactive Zero needs.

Prediction Based trade and execution prediction.

- Needs come from trade predictions.
- Secured inputs/outputs come from execution prediction.

# Negotiation Manager

## Role




Controls negotiation.

- Sets negotiation agendas → **Proactive** .
- Accepts/rejects negotiation requests → **Reactive** .
- Defines utility functions.
- Goal: Achieve the target put by the trading strategy.

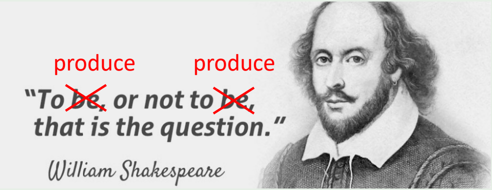
## Built-in Options

Independent Negotiations buy cheap ASAP, sell expensive ALAP.

Moving Range Creates one controller for selling and another for buying.

# Production Strategy

## Role




## Built-in Options

**Supply Driven** Produce based on **buy** contracts.

- Inventory is always valued at the end.

**Demand Driven** Produce based on **sell** contracts.

- … but inventory is valued at half the *trading* price!

# Outline

# A brief history of ANAC



- The Automated Negotiating Agents Competition (ANAC)
- Started on 2010 and is currently in its 11th incarnation.
- Was conducted in conjunction with AAMAS but currently IJCAI.


| Year | Challenge | Year | Challenge |
|---|---|---|---|
| 2010 | Domain Independence | 2011 | Linear Ufuns |
| 2012 | Reservation Value | 2014 | Learning and Adaptation |
| 2015 | Three-party negotiation | 2016 | Energy Grid Theme |
| 2017 | Repeated Negotiations, Diplomacy, HAN | 2018 | Repeated Negotiations, Diplomacy, HAN |

# Outline

# References I

Mohammad, Y., Viqueira, E. A., Ayerza, N. A., Greenwald, A., Nakadai, S., and Morinaga, S. (2019). Supply chain management world. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 153–169. Springer.

# Automated Negotiation: Challenges and Tools

## Future Challenges and Open Problems

**Yasser Mohammad**[1,2,3] and **Amy Greenwald**[4]

[1] NEC CORPORATION, Global Innovation Unit

[2] National Institute of Advanced Industrial Science and Technology (AIST)

[3] Assiut University, Egypt

[4] Brown University

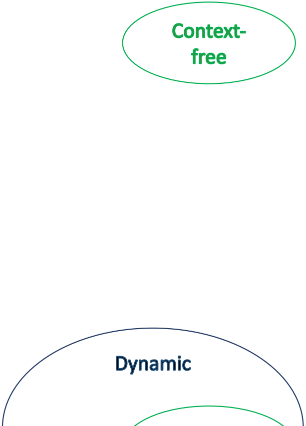
February 23rd, 2022

# Outline

# Outline

# Situated Negotiation

NEC-AIST AI Cooperative Research Laboratory



# Outline

# Negotiation Under Uncertainty

## The challenge

How to negotiate when you have *partial* information about your actor's utility function?

## The Game

- ANAC 2019 agent game @ IJCAI introduced the first competition in this domain.
- Input is a ranking of a subset of the outcomes.

## Example Solutions

- Regress the utility of all outcomes using a polynomial model.
- Use a GP to create a probabilistic ufun.

# Outline

1. Situated Negotiation
2. Optimality Results
    - Against Known Opponents
    - In Specific Settings
3. Preference Elicitation During Negotiation
4. Concurrent Negotiation
5. References

# Outline

# Outline

# Outline

1 Situated Negotiation

2 Optimality Results

3 Preference Elicitation During Negotiation
- Procedure and Strategies
- Optimal Elicitation Algorithm
- Value of Information Algorithm

4 Concurrent Negotiation

5 References

# Outline

1. Situated Negotiation
2. Optimality Results
3. Preference Elicitation During Negotiation
   - Procedure and Strategies
   - Optimal Elicitation Algorithm
   - Value of Information Algorithm
4. Concurrent Negotiation
5. References

# 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

NEC-AIST AI Cooperative Research Laboratory

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

$(\omega^*, \omega_*, p)$ : Getting $\omega^*$ with probability $p$ otherwise $\omega_*$

### Example query

Do you prefer to get $\omega$ for certain over $(\omega^*, \omega_*, p)$?

# Elicitation Procedures/Strategies

NEC-AIST AI Cooperative Research Laboratory

**Probability Equivalence**

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

**Certainty Equivalence**

find $\omega$ 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 $\lvert\Omega\rvert$ |
| $R(i) \equiv r_i$ | Reserved value for agent $i$ |
| $\tilde{U}_i : \Omega \rightarrow [0,1]$ | Utility of outcomes to **actor** $i$ |
| $\hat{U}_i^0 : \Omega \rightarrow P$ | Probability distribution of utility values for **agent** $i$ $\hat{U}_{ij}^0 \equiv \hat{U}_i^0(\omega_j)$ |
| $P : \{[0,1] \rightarrow [0,1]\}$ | 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).

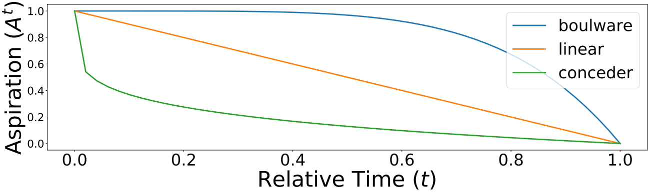
# Outline

## Pandora's Problem [Economics]

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

# Optimal Elicitation [Baarslag and Gerding, 2015]

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

## Take-away message

Avoid deep-elicitation.

# Extensions to Pandora's algorithm

NEC-AIST AI Cooperative Research Laboratory

## Closed-form Calculation of z-index[Mohammad and Nakadai, 2018b]

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

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

$\zeta$ $\quad b^2 - \frac{2c_j}{\beta}(b-a)$

## 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 $U(a,b)$, $Min, Max = a, b$.

# Outline

1. Situated Negotiation
2. Optimality Results
3. Preference Elicitation During Negotiation
   - Procedure and Strategies
   - Optimal Elicitation Algorithm
   - Value of Information Algorithm
4. Concurrent Negotiation
5. References

# Value of Information Algorithm

- Based on [Chajewska et al., 2000] 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[Baarslag and Kaisers, 2017, Mohammad and Nakadai, 2018a].

# VOI Based Elicitation

NEC-AIST AI Cooperative Research Laboratory

## 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 [Boutilier, 2003]

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

## Optimal Policy

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

# VOI Based Elicitation II

## Questions and Answers

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

## 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

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

# VOI main Issues

NEC-AIST AI Cooperative Research Laboratory

## Accurate Agreement Model Assumption

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

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

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

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

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

$$EVOI\left(q^I, \left\{\hat{U}^t_\omega\right\}\right) =$$

$$\mathbb{E}_s\left(\max_\pi EEU\left(\pi, Ans^I_s\right)\right) - \max_\pi EEU\left(\pi, \left\{\hat{U}^t_\omega\right\}\right)$$

# Outline

1. Situated Negotiation
2. Optimality Results
3. Preference Elicitation During Negotiation
4. Concurrent Negotiation
5. References

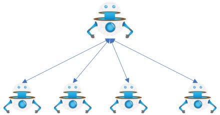
# Concurrent Negotiation

## Generality

- Specific scenario (buyer-seller).
- General domain

## Decommitment

- Symmetric de-commitment.
- Asymmetric de-commitment.
- No de-commitment.



## Timing

- Synchronous.
- Any-time.

# Outline

# References I


Baarslag, T. and Gerding, E. H. (2015). Optimal incremental preference elicitation during negotiation. In *IJCAI*, pages 3–9.

Baarslag, T. and Kaisers, M. (2017). 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*, pages 391–400. International Foundation for Autonomous Agents and Multiagent Systems.

Boutilier, C. (2003). On the foundations of expected expected utility. In *IJCAI*, volume 3, pages 285–290. Citeseer.

Chajewska, U., Koller, D., and Parr, R. (2000). Making rational decisions using adaptive utility elicitation. In *AAAI/IAAI*, pages 363–369.

Mohammad, Y. and Nakadai, S. (2018a). Fastvoi: Efficient utility elicitation during negotiations. In *International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, pages 560–567. Springer.

# References II

Mohammad, Y. and Nakadai, S. (2018b). Utility elicitation during negotiation with practical elicitation strategies. In *IEEE SMC*.

Weitzman, M. L. (1979). Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society*, pages 641–654.

# Conclusion

- Automated negotiation can enhance societal welfare.
- Genius and NegMAS as open-ended platforms for research in automated negotiation.
- Classical automated negotiation research in economics focused on simplified situations and provided performance guarantees.
- Many open questions:
  - General environment and unknown opponents.
  - Incomplete information about self.
  - Concurrent negotiations.
  - Negotiations with non-stationary utilities.
  - When to use negotiation?