Automated Negotiation

Catholijn M. Jonker

Ongoing collaboration with: Tim Baarslag and Reyhan Aydogan

TU Delft

Contents of these slides

- Recap: Formalizing Negotiations
  - Domain models
  - Preferences and preference elicitation
  - Analysing results
- Protocols
  - bilateral
  - Multi-lateral
- Negotiation strategies
  - BOA framework
  - An overview of important strategies
- Analysis of Negotiation Dynamics
- Using the BOA framework
- Prepare for Tutorial and Test

Why negotiation is difficult for humans?

- Leaving money on the table
  - Sub-optimal outcomes for both sides.
- Bounded rationality
  - Outcome space might be too big.
- Emotions
  - May have negative effect on acting rationally
  - Some people are too shy to ask what they want.

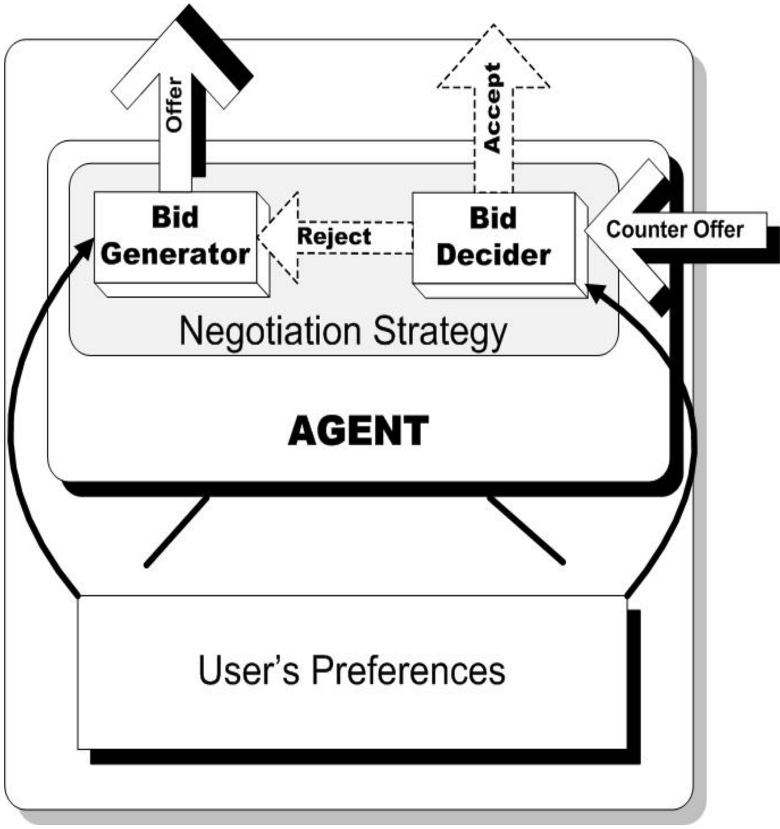
TU Delft

Automated Negotiation

- Intelligent software agents negotiate on behalf of their users.
- Possible application domains: e-commerce, politics, cloud-computing, crisis management system, task allocation, etc.

How can a software agent negotiate its user’s behalf?

- Evaluating bids
  - Reasoning on its user’s **preferences**
- Employing a **negotiation strategy**
  - Which action the agent will take
  - How the agent will generate its offer
  - When the agent will accept the opponent’s counter offer
- Communicating with other agent (s) based on predefined rules (**negotiation protocol**)



Preferences:

Utility Functions

Holiday Domain

- Hotel Location: Sea, Historical Places, Mountain.
- Season: Winter, Summer.
- Duration: One week, Two weeks, Three weeks.

Hotel Location	**Season**	**Duration**	**Utility**
Sea	Summer	1- week	0.95
Historical Places	Summer	1- week	0.87
Mountain	Summer	1-week	0.36
Sea	Summer	2- weeks	0.76
...	...	...	...

Most of the negotiation approaches use utility functions.

TU Delft

Preferences
Additive Utility Functions

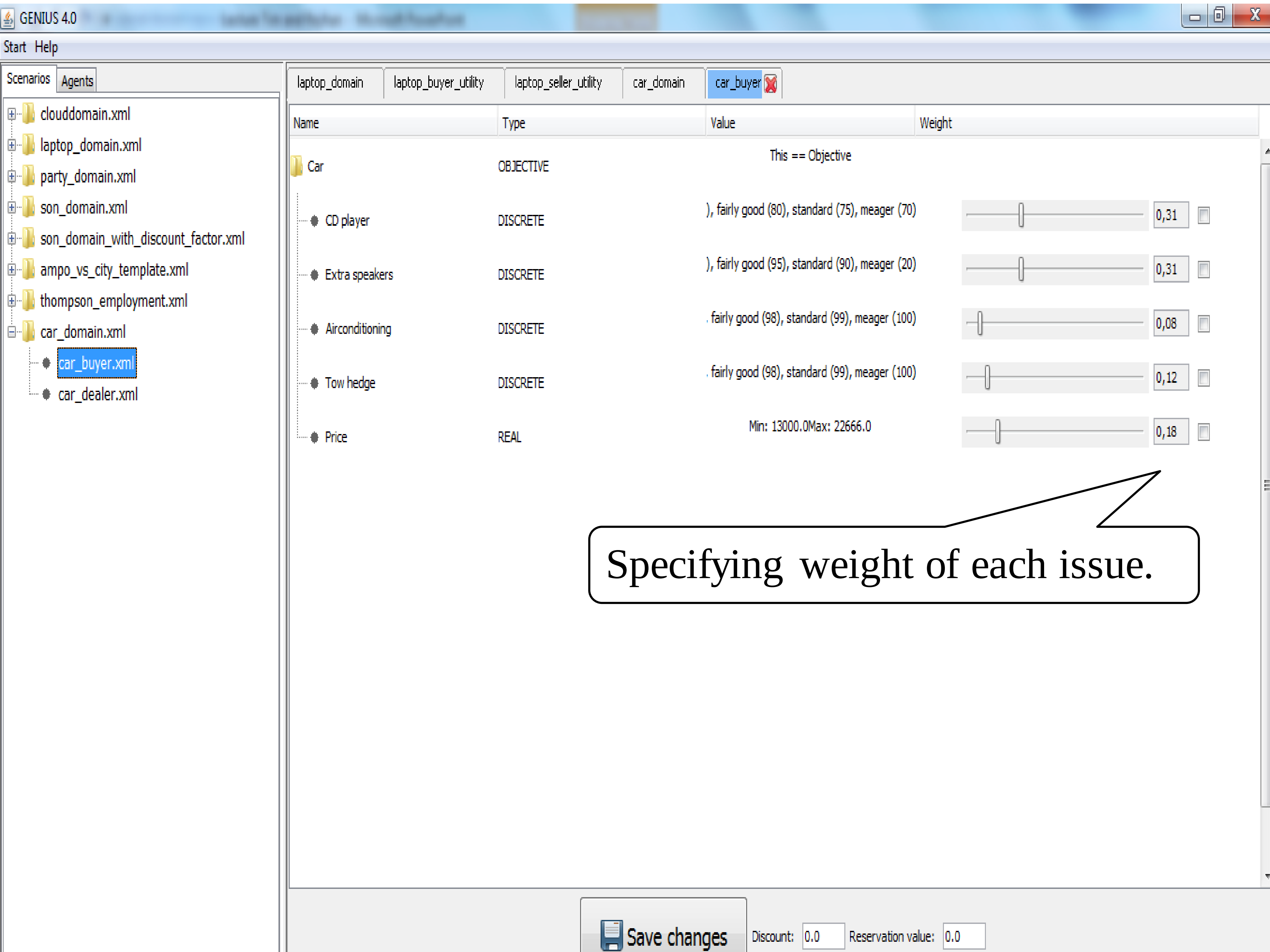
- **Assumption:** Issues are preferentially independent.
- **Utility** is determined as a weighted sum of the evaluation values of each issue.

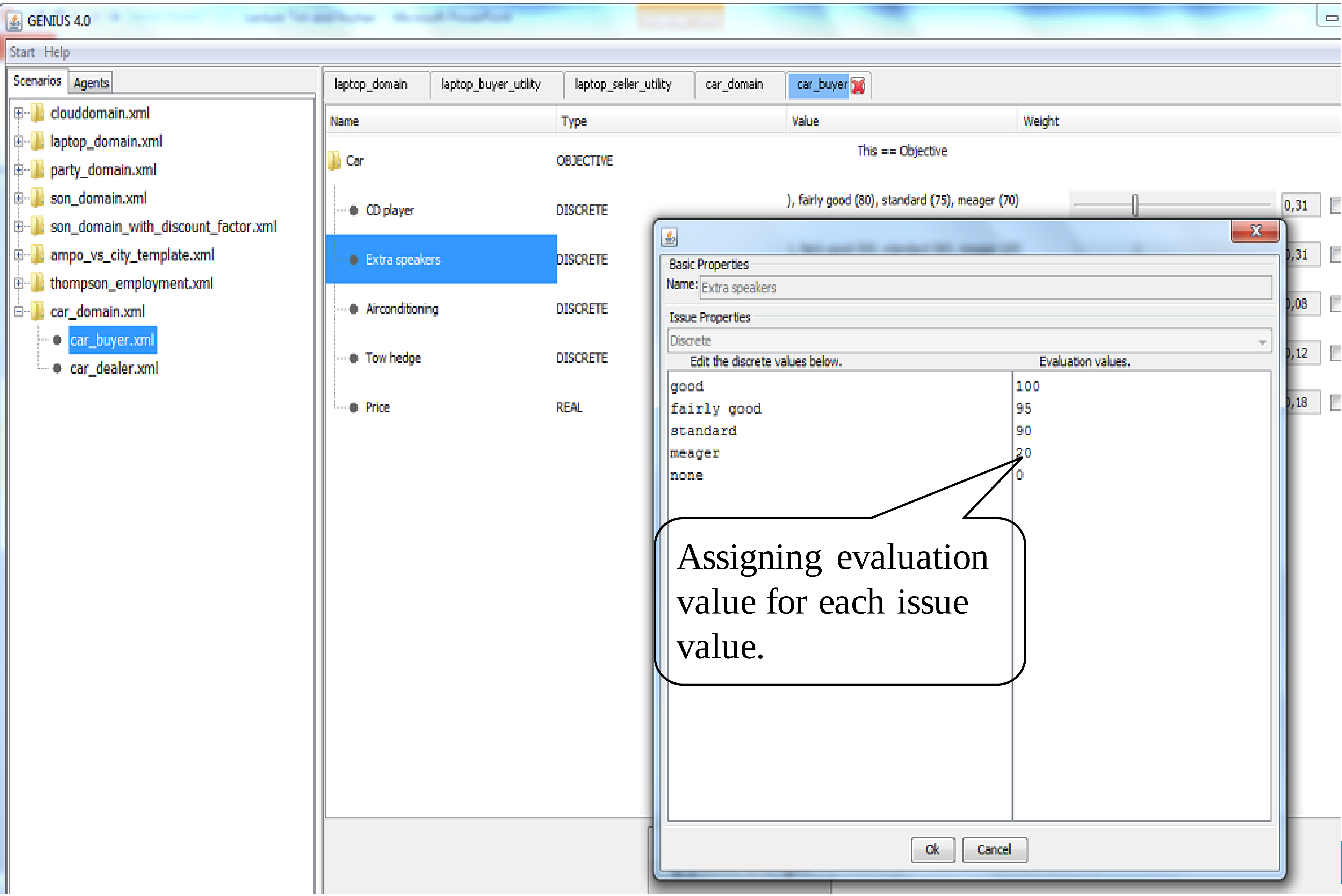
$$U(X_1, X_2, ..., X_n) = \sum_{i=1}^{n} w_i * U_i(X_i)$$

- The utility of <Sea, Summer, One-week>=

  0.5 * U_{1}(Sea) + 0.3 * U_{2} (Summer) +0.2 * U_{3}(One-Week)

  where w_{1} (Location)= 0.5, w_{2} (Season)= 0.3, w_{3}(Duration) =0.2


GENIUS 4.0
Start Help
Scenarios Agents
clouddomain.xml
laptop_domain.xml
party_domain.xml
son_domain.xml
son_domain_with_discount_factor.xml
ampo_vs_city_template.xml
thompson_employment.xml
car_domain.xml
car_buyer.xml
car_dealer.xml
laptop_domain
laptop_buyer_utility
laptop_seller_utility
car_domain
car_buyer
Name
Type
Value
Weight
Car
OBJECTIVE
This == Objective
CD player
DISCRETE
), fairly good (80), standard (75), meager (70)
0,31
Extra speakers
DISCRETE
), fairly good (95), standard (90), meager (20)
0,31
Airconditioning
DISCRETE
fairly good (98), standard (99), meager (100)
0,08
Tow hedge
DISCRETE
fairly good (98), standard (99), meager (100)
0,12
Price
REAL
Min: 13000.0Max: 22666.0
0,18
Specifying weight of each issue.
Save changes
Discount:
0.0
Reservation value:
0.0


GENIUS 4.0
Start Help
Scenarios
Agents
clouddomain.xml
laptop_domain.xml
party_domain.xml
son_domain.xml
son_domain_with_discount_factor.xml
ampo_vs_city_template.xml
thompson_employment.xml
car_domain.xml
car_buyer.xml
car_dealer.xml
laptop_domain
laptop_buyer_utility
laptop_seller_utility
car_domain
car_buyer
Name
Type
Value
Weight
Car
OBJECTIVE
This == Objective
CD player
DISCRETE
), fairly good (80), standard (75), meager (70)
0,31
Extra speakers
DISCRETE
Airconditioning
DISCRETE
Tow hedge
DISCRETE
Price
REAL
Basic Properties
Name: Extra speakers
Issue Properties
Discrete
Edit the discrete values below.
Evaluation values.
good
fairly good
standard
meager
none
100
95
90
20
0
Assigning evaluation value for each issue value.
Ok
Cancel

Reflection on last week: Normalization

- You need first to normalize all evaluation functions (also named valuation functions in last week's slides). I discuss it here for discrete domains:
  - Pick the item of the issue's range that has maximum value, the normalized evaluation function for that issue maps that item to 1.
  - For all other items in the range normalize in proportion to the maximum value to determine it's normalized value.

Normalization (2/3)

- Formally: Given issue j with range D_j, and valuation function v_j
- Let $m=\max(\{v_j(x) \mid x \in D_j\})$,
- We define the *normalized evaluation* function $e_j : D_j \rightarrow [0,1]$ by: $e_j(x)=v_j(x)/m$

- Then to determine the normalized utility of a bid:
  - Multiply the normalized evaluation of the issue wrt the bid with the weight of that issue
  - Do this for all issues
  - Add it all up

Normalization (3/3)

- Formally:
  - Let $D = D_1 \times D_2 \times \ldots \times D_n$ be the cartesian product of the ranges of all issues. D is also the bid space.
  - Let w_j be the weight of issue j, for all j: $1 \leq j \leq n$.
  - Let e_j be the normalized evaluation function of issue j.
  - The *normalized utility* function $u: D \rightarrow [0,1]$ is defined by: $u(b) = \sum_{1 \leq j \leq n} (w_j * e_j(b_j))$, where b_j is the projection of b on issue j.

NEGOTIATION PROTOCOLS

Negotiation Protocol

- A negotiation protocol governs the interaction between negotiating parties by determining
  - how the parties interact/exchange information
    - “ **who can say what and when they can say it** ”
  - when the negotiation ends
- Bilateral Negotiation
  - Alternating Offers Protocol (Rubinstein 1982)
- Multiparty Negotiation
  - Mediated Single Text Protocol

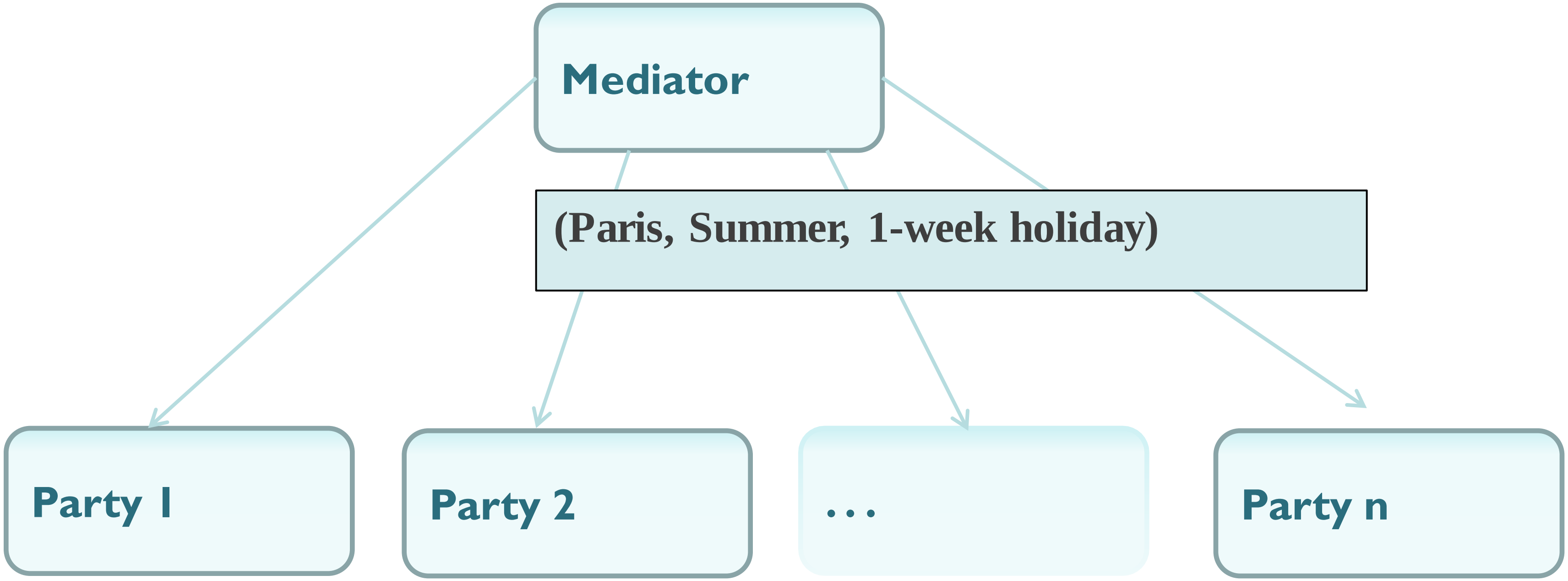
Alternating Offers Protocol (Rubinstein 1982)

- One of the agents initiates negotiation with an offer.
- The agent receiving an offer can
  - accept that offer
  - make a counteroffer
  - end negotiation

- This process continues in a turn-taking fashion until having a consensus or reaching a termination condition such as a deadline.

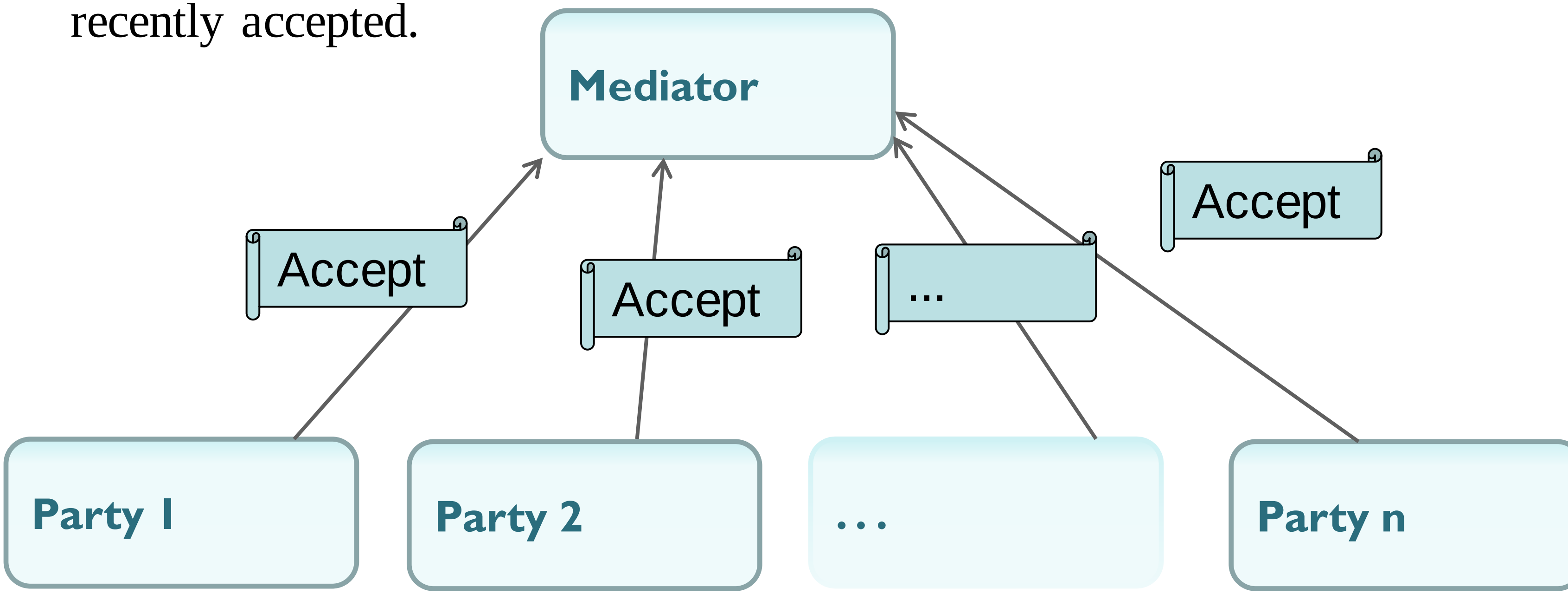
Mediated Single Text Negotiation Protocol
(based on Raiffa 1982)

Mediator generates an offer and asks negotiating agents for their votes either to accept or to reject this offer.



Mediated Single Text Negotiation Protocol

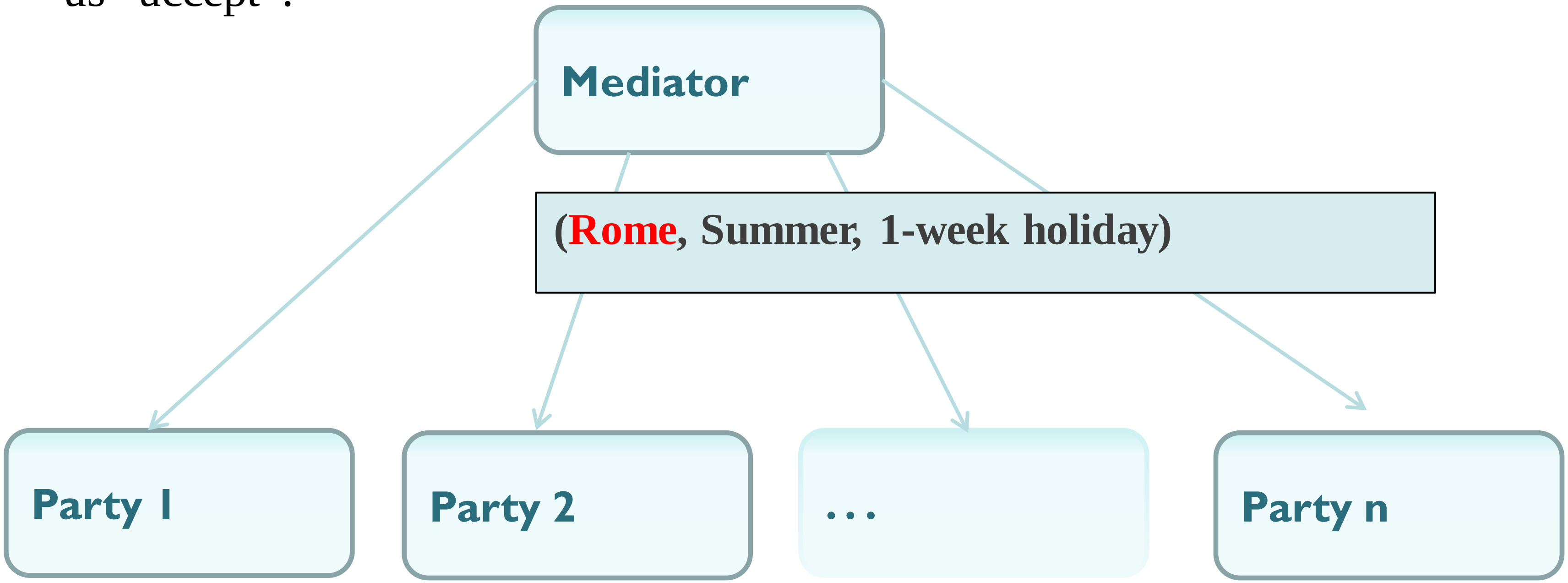
- Negotiating agents send their votes for the current bid according to their acceptance strategy.
- If all negotiating agents vote “accept”, the bid is labeled as the most recently accepted.



E.g. MRA Bid: (Paris, Summer, 1-week holiday)

Mediated Single Text Negotiation Protocol

- Mediator modifies the most recently accepted bid by exchanging one value arbitrary and asks negotiating agents' votes again.
- It updates the most recently accepted bid if all negotiating agents vote as "accept".



This process continues iteratively until reaching a predefined number of bids.

How does a mediator find a solution acceptable to all?

- The role of the mediator is to propose new ideas that are acceptable to all.
- How to find these?
- Trusted mediator might know all profiles (partially)
- What if mediator doesn't know the profiles?
  - Mediated Hill-Climber Agent
  - Mediated Annealer Agent

Mediated Single Text Negotiation Hill-Climber Agent (Klein et al., 2003)

- Accept a bid if its utility is higher than the utility of the most recently accepted bid

  - **MRA Bid= (Antalya, Summer, 1-week),**
  - **Bid_6= (Antalya, Summer, 2-week),**
  - **$U(Bid_6)=0.95 > U(\text{MRA Bid})=0.87 \rightarrow$ ACCEPT**

Note: If the utility of initial bid is quite high for one of the agents, that agent may not accept other bids even though those bids might be better for the majority.

http://ebusiness.mit.edu/research/Briefs/4Klein_Negotiation_Brief_Final.pdf

Mediated Single Text Negotiation: Annealer Agent (Klein et al., 2003)

- Calculates the probability of acceptance for the current bid:

$$\boxed{P(\text{accept}) = \min(1, e^{-\Delta U/T})}$$

T: Virtual temperature gradually declines over time

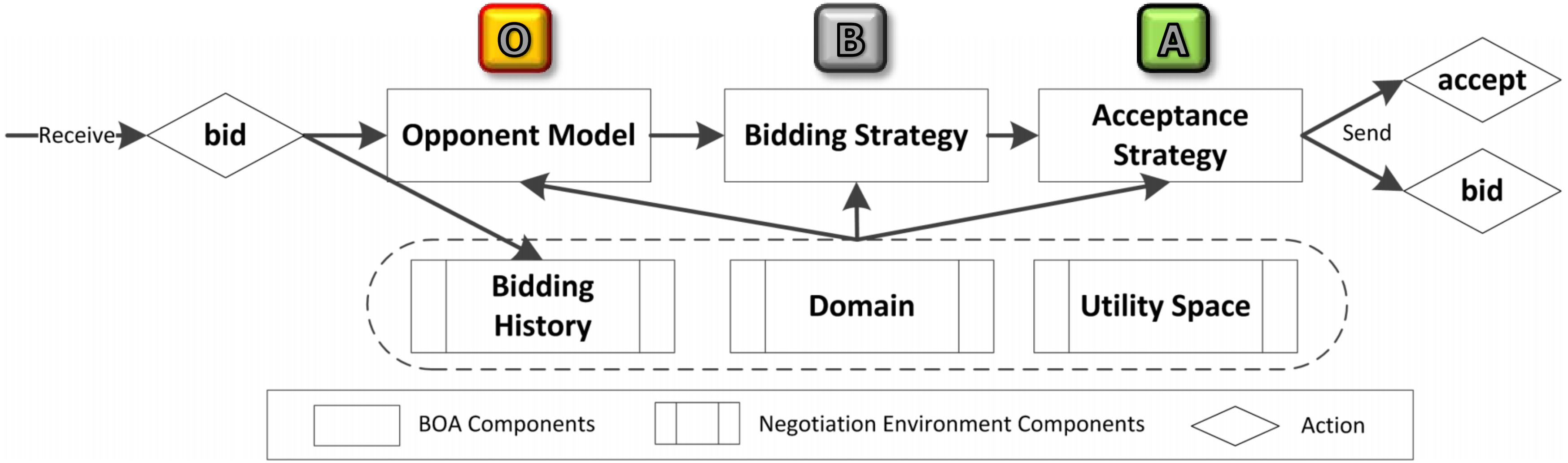
- Higher probability for acceptance
  - The utility difference is small & virtual temperature is high
- Tendency to accept individually worse bids earlier so the agents find win-win bids later

Baarslag et al., for more informationsee phd thesis

NEGOTIATION STRATEGIES

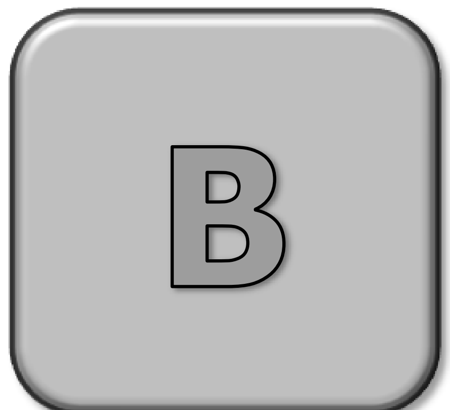
Negotiation Strategy

- Determines
  - which action the agent will take
  - how the agent will generate its offers
  - how the agent decide whether the opponent's counter-offer is acceptable



TUDelft

Bidding strategies



Random Walker

- It generates an offer randomly as follows:
  - Selects values of issues randomly
  - Proposes only those bids whose own utility greater than its reservation utility (RU=0.6).

Time-dependent Concession Strategy [Faratin, Sierra & Jennings, 1998]

- Each agent has a deadline and the agent's behavior changes with respect to the time.
- An offer which is not acceptable at the beginning, may become acceptable over time (conceding while approaching the deadline).
- A function determines how much the agent will concede
  - Remaining negotiation time
  - Parameter related to concession speed (β)
- **Conceder Tactic:**
  - β >1 concedes fast and goes to its reservation value quickly.
- **Boulware Tactic:**
  - β <1 hardly concedes until the deadline

Trade-Off Strategy (1)

- Not only considers its own utility but also take its opponent's utility into account.

- The importance of the issues may be different for the negotiating agents.
  - E.g. delivery time might be more important for the consumer

- The agent may demand more on some issues while concedes on other issues without changing its overall utility as if possible.
  - E.g. higher price in order to have an earlier delivery

Trade-Off Strategy (2)
[Faratin, Sierra, Jennings, AIJ 2002]

- Selects a subset of bids having the same utility with its previous offer (iso-curve)
  - If not possible, it makes minimal concession such as 0.05.
- Among those bids, choose the bids which might be more preferred by its opponent
  - Heuristic: the most similar one to opponent's last bid

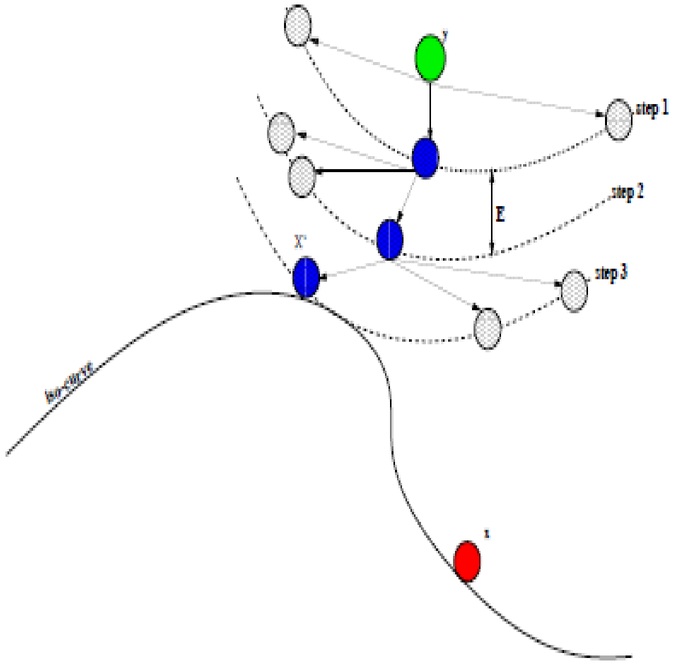


Figure 4.8: Schema of the trade-off algorithm with $N = 3$ and $S = 3$.

This figure is taken from Faratin's PhD Thesis.

Behaviour Dependent Strategies:

[Faratin, Sierra & Jennings, 1998]

- The agent imitates its opponent's behaviour.
- The degree of imitation may vary
  - Absolute Tit-For-Tat:
    - E.g. The opponent increases the price by 50 units then the agent will decrease the price by 50 units.
  - Relative (proportionally) Tit-For-Tat:
    - Taking into account the changes of its opponent's behaviour in a number of previous steps.
  - Averaged Tit-For-Tat
    - Taking into account the average changes within a window of size of its opponent history

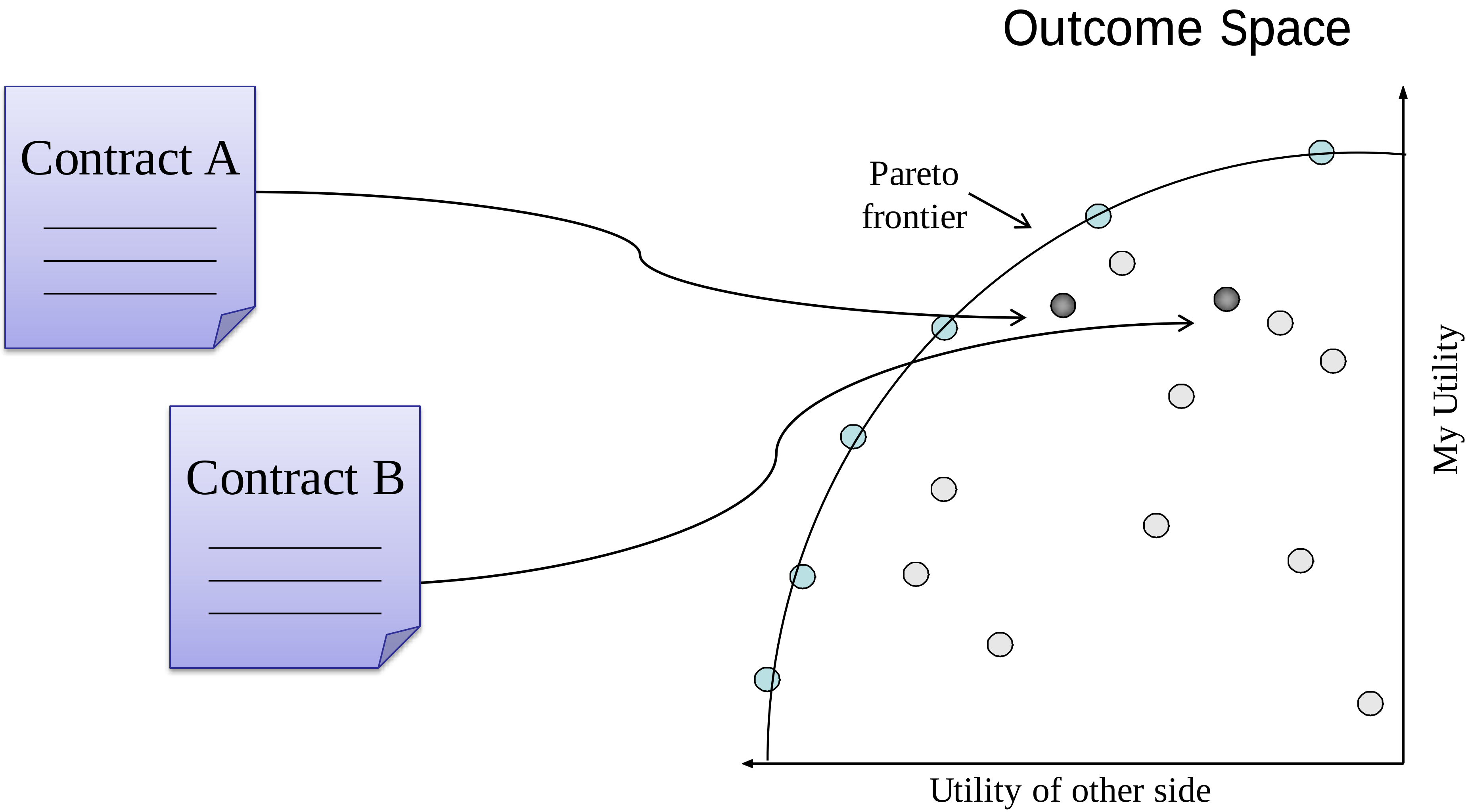
Opponent modelling strategies

O

Opponent Modelling (1) Why?

- Exploit the opponent
- Maximize chance of reaching an agreement
  - Requiring outcome with acceptable utility for opponent, i.e. resolving the conflict of interest.
- Increase the efficiency of a negotiated agreement
  - Searching through the outcome space for outcomes that are mutually beneficial
  - Reaching better/optimal agreements
- Avoid unfortunate moves
  - which is worse for both agents
- Make trade-offs and maximize social welfare
- Reach agreements early
  - Reducing communication cost

Opponent Modeling



TU Delft

Opponent Modelling (2) What?

- Learning which issues are important for the opponent
  - Issue weights
- Learning opponent's preferences
  - Evaluation of issue values
  - Preference ordering of issue values
- Learning about opponent's strategy
  - Predicting the utility of its next offer
- Learning what kind of offers are not acceptable
  - Reservation value
  - Constraints

Some Examples (1):

- Kernel density estimation for estimating the opponent's **issue weights** [Coehoorn and Jennings 2004]
  - Intuition: The opponent has a tendency to concede slowly on important issues.
  - Assumption: Weighted scoring function & Concession based strategy

- Bayesian Learning for predicting **evaluation functions and weights** [Hindriks and Tykhonov, 2008]
  - Hypothesis for evaluation functions: uphill, downhill, triangular
  - Assumption: Linear additive functions & Concession based strategy

Some Examples (2)

- A guessing heuristic for predicting the opponent's **unknown weights** [Jonker, Robu & Treur, 2007]
  - Some of the weights are revealed by the opponent
  - Requiring domain knowledge

- Concept-based Learning (RCEA) for classifying offers regarding **their acceptability** [Aydogan & Yolum 2012]
  - Assumption: Conjunctive & Disjunctive Constraints
  - Intuition: Avoid offering unacceptable offers to opponent

A Simple Example: Frequency Analysis Heuristic

- A heuristic adopted by some of the agents in ANAC competition such as *HardHeaded Agent*

- Based on **how often the value of an issue changed** and **the frequency of appearance of values in offers**

- Learning issue weights: importance of issues
  - Heuristic: If the value is often changed, then the issue gets a low weight.

- Learning issue value weights: evaluations of the issue values
  - Heuristic: A preferred value will appeared more often in agent's offers than a less preferred value.

Frequency Analysis Heuristic (2)

Estimation of issue weights

- Assume that we have two issues (X, Y) and opponent 's first offer is [x1,y1].
  - Take the predicted weights 0.5 and 0.5 for X, Y respectively
- Second offer [x1, y2]
  - W1=0.5 +n since opponent didn't change the value of X
  - W2=0.5
  - If n= 0.1 then new weights will be 0.6, 0.5 respectively
    - W1new= 0.55 W2new= 0.45

TU Delft

Frequency Analysis Heuristic (3)

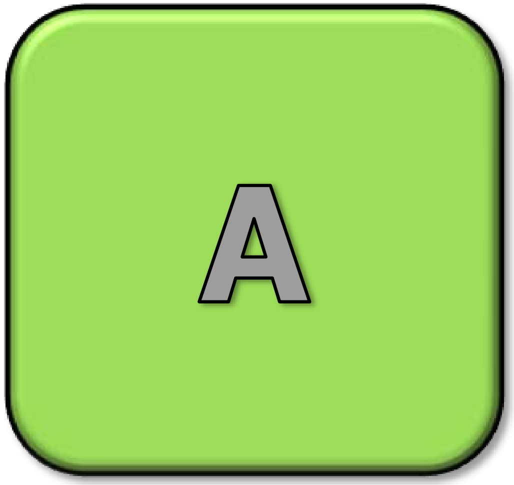
Estimation of evaluation values for issues

- Assume negotiation round=45 and our opponent's offer history
  - "Brand" issue in Laptop domain

Issue Values:	Dell	Mac	HP
# of times appeared in offers	20	15	10
Estimated Evaluation	1.0 (20/20)	0.75 (15/20)	0.5 (10/20)

TU Delft

Acceptance Conditions



Introduction

Why and when should we accept?

- In every negotiation with a deadline, one of the negotiating parties has to accept an offer to avoid a break off.
- A break off is usually an undesirable outcome; therefore, it is important to consider under which conditions to accept.

Introduction

The Acceptance Dilemma

- When designing such conditions one is faced with the acceptance dilemma:
  - Accepting too early may result in suboptimal agreements
  - On the other hand, accepting too late may result in a break off

- We have to find a balance:

$$\begin{aligned}\text{Total average utility} \quad = \quad & \text{Agreement percentage} \\ & \times \\ & \text{Average utility of agreements.}\end{aligned}$$

Acceptance Conditions

Selection of Existing Acceptance Conditions

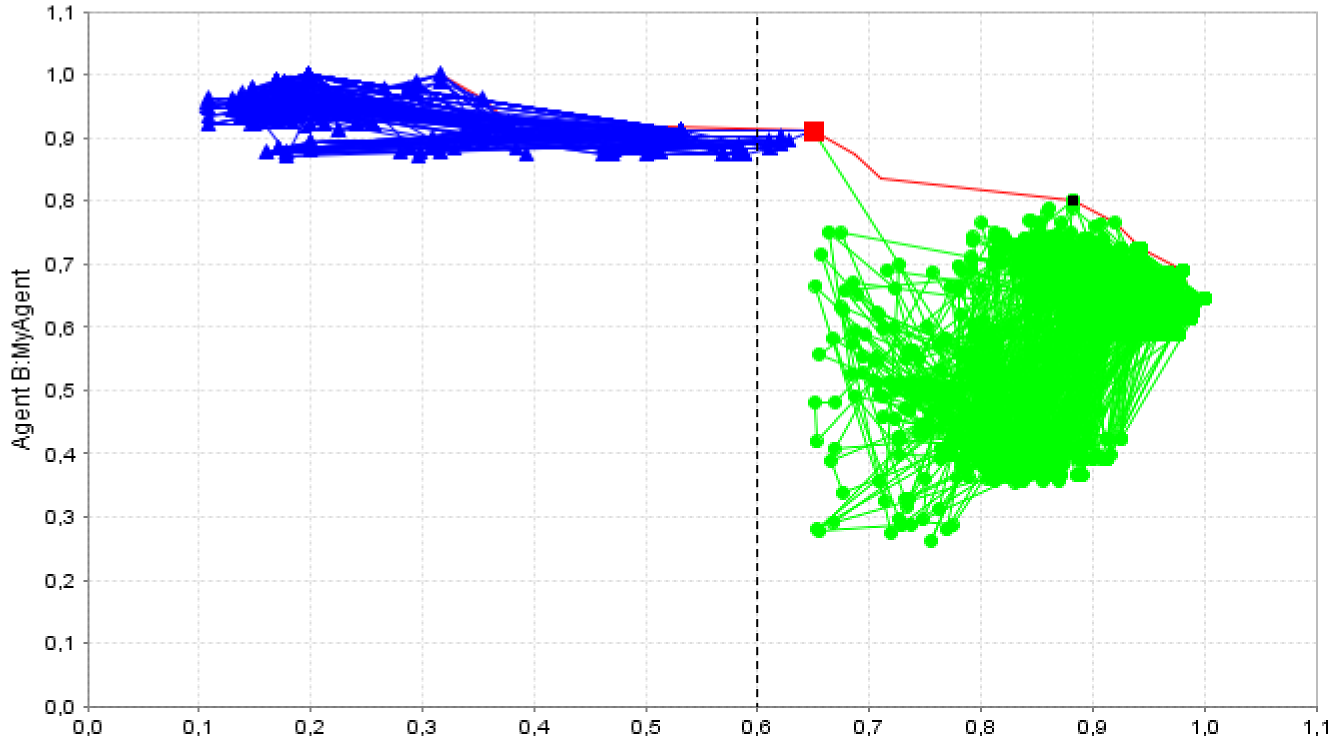
- In literature and current agent implementations, we see the following recurring acceptance conditions:

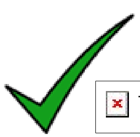
- $\mathbf{AC}_{\mathrm{const}}(\alpha)$ Accept when the opponent's bid is better than α
- $\mathbf{AC}_{\mathrm{next}}$ Accept when the opponent's bid is better than our upcoming bid
- $\mathbf{AC}_{\mathrm{time}}(T)$ Accept when time $T \in [0, 1]$ has passed

Acceptance Conditions

t = 0.7

Example





This image cannot currently be displayed.

This image cannot currently be displayed.

This image cannot currently be displayed.

This image cannot currently be displayed.

AC_{next}

TU Delft

Acceptance Conditions

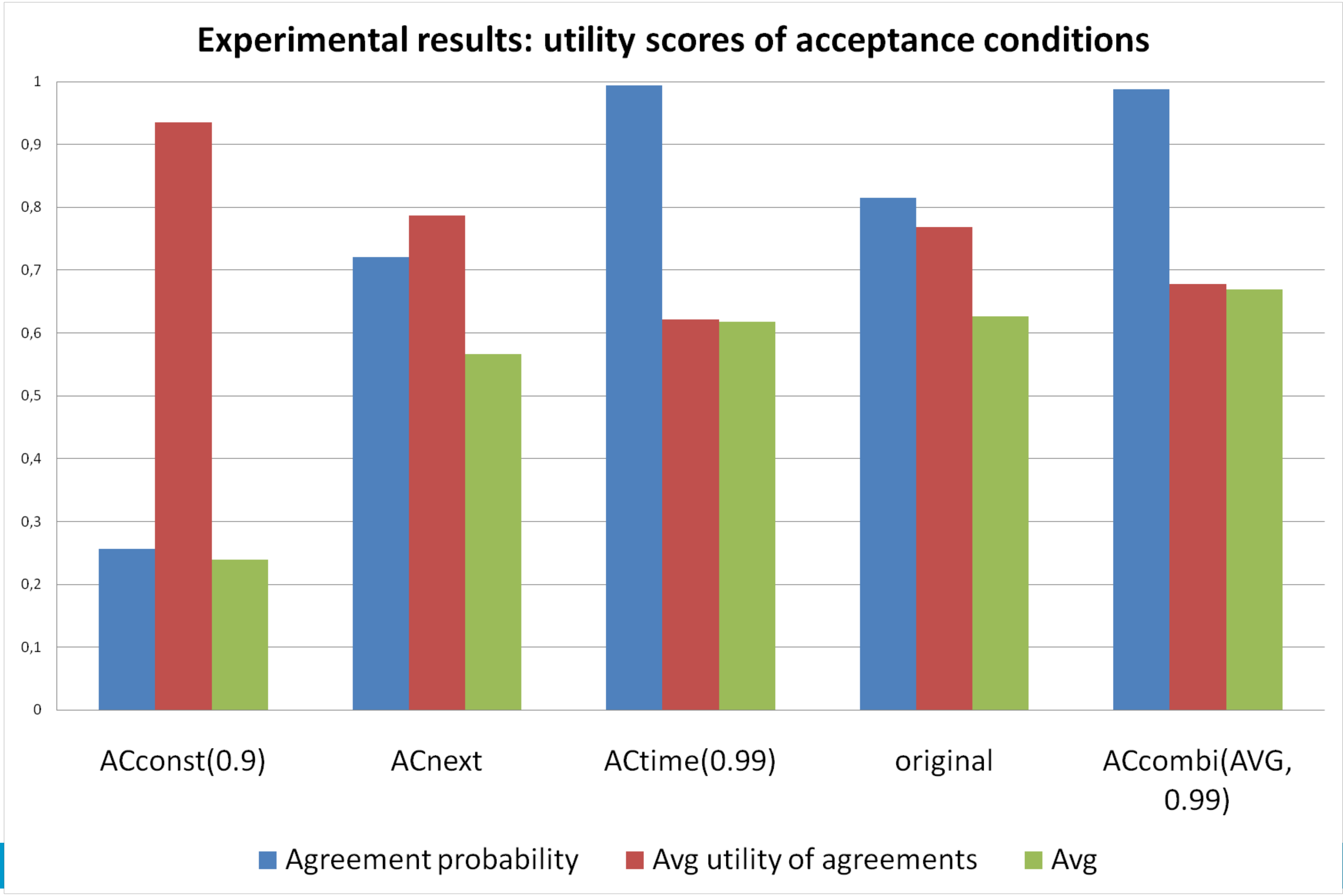
Combining Acceptance Conditions

- We can also combine acceptance conditions, e.g.:

$$\mathbf{AC}_{\text{combi}}(T, \alpha) \overset{\text{def}}{\iff} \mathbf{AC}_{\text{next}} \vee \mathbf{AC}_{\text{time}}(T) \wedge \mathbf{AC}_{\text{const}}(\alpha).$$

- $\mathbf{AC}_{\text{combi}}(T, \alpha)$ splits the negotiation time into two phases: $[0, T)$ and $[T, 1]$

- We can also choose non-constant values for α such as average utility so far received (AVG), or maximum utility (MAX).


Experimental results: utility scores of acceptance conditions
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
ACconst(0.9)
ACnext
ACtime(0.99)
original
ACcombi(AVG, 0.99)
Agreement probability
Avg utility of agreements
Avg

Conclusion

- $\mathbf{AC}_{\text{next}}$ is often used, but does not always give the best results.
- $\mathbf{AC}_{\text{const}}(\alpha)$ performs worst of all AC's, as a good value for α is highly domain-dependent.
- $\mathbf{AC}_{\text{time}}(T)$ always reaches an agreement, but of relatively low utility.
- We need combinations of different approaches.

Conclusion:

Challenges in Automated Negotiation

- Designing negotiation protocols & strategy
- Representing and Reasoning on Preferences in Negotiation
- Predicting Other Agent's Preferences during Negotiation
- Acceptance Strategies

TU Delft

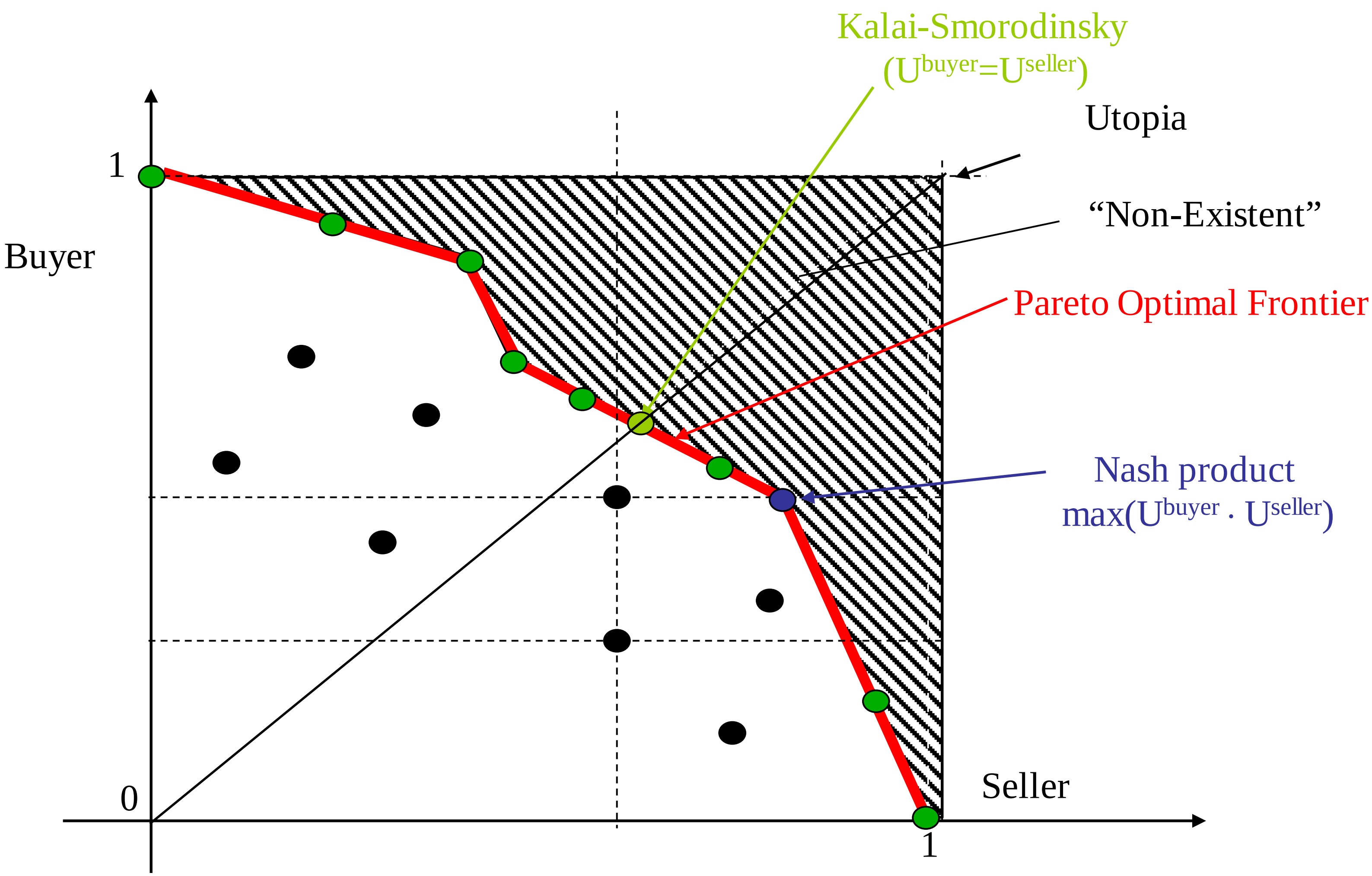
Hindriks, Jonker, Tykhonov, 2011

ANALYSIS OF NEGOTIATION DYNAMICS

Analysis of negotiation strategies

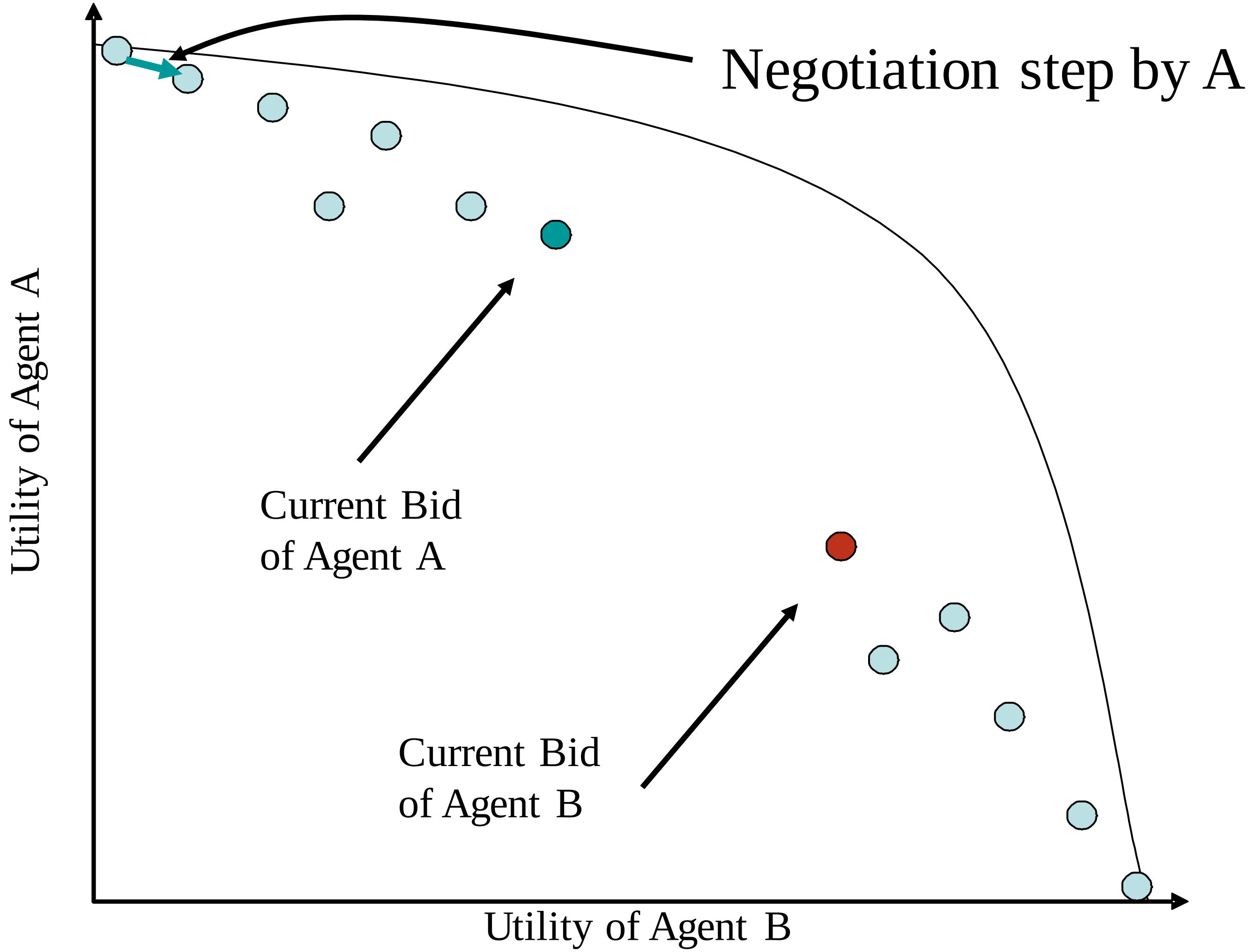
- What kind of bids to **make**:
- Process analysis
  - Step analysis
  - Dynamic properties
- What kind of bids to **accept**:
- Outcome analysis
  - Nash product
  - Kalai-Smorodinsky
  - Pareto-optimal

Outcome Analysis



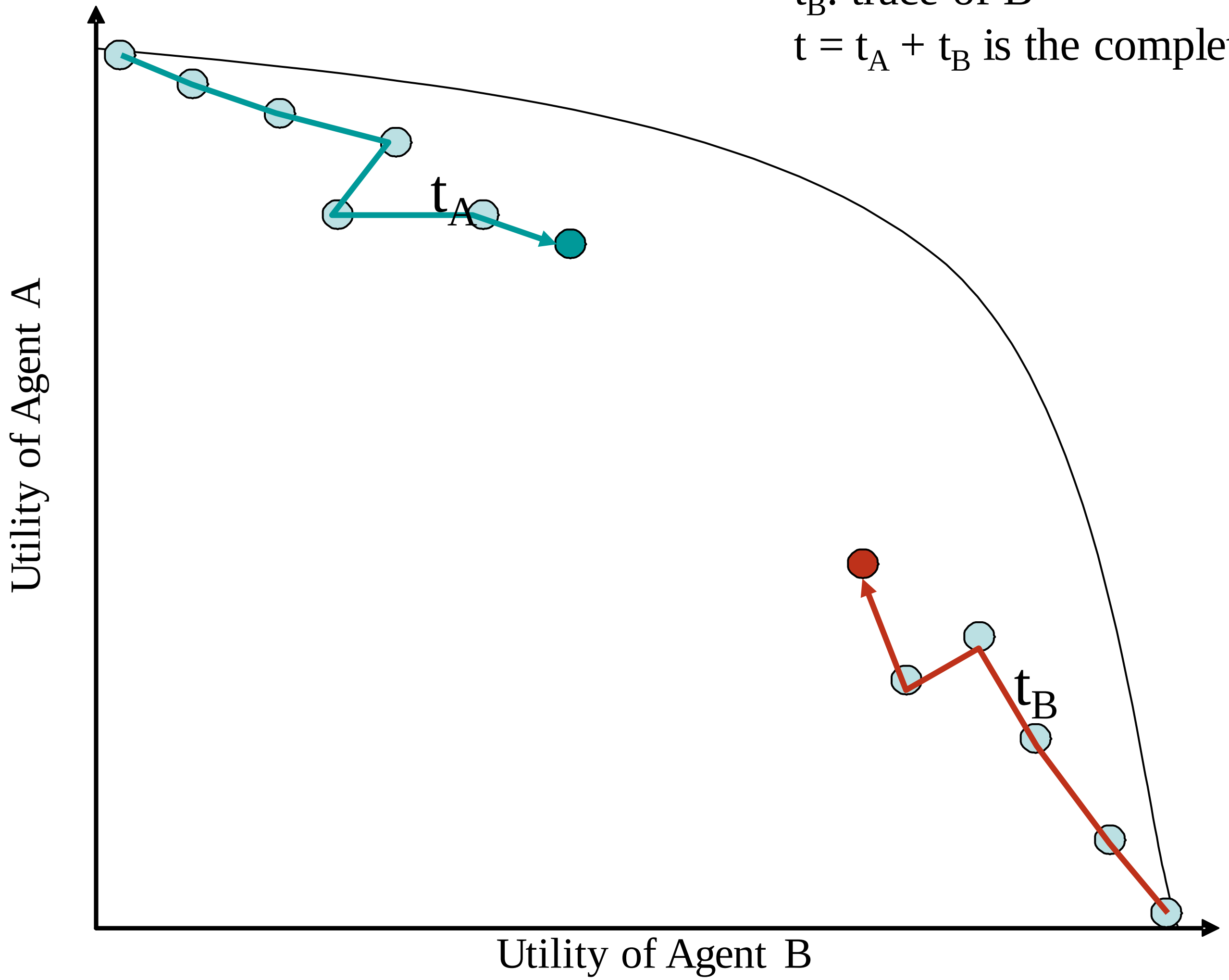
TU Delft

Negotiation traces



Negotiation traces

t_A: trace of A
t_B: trace of B
$t = t_A + t_B$ is the complete nego trace



TU Delft

Utility, negotiation steps, and traces

$U_S(b)$: utility of “Self” for bid b

$U_O(b)$: utility of “Other” for b

$\Delta_a(b, b') = U_a(b') - U_a(b),\ a \in \{S,O\}$

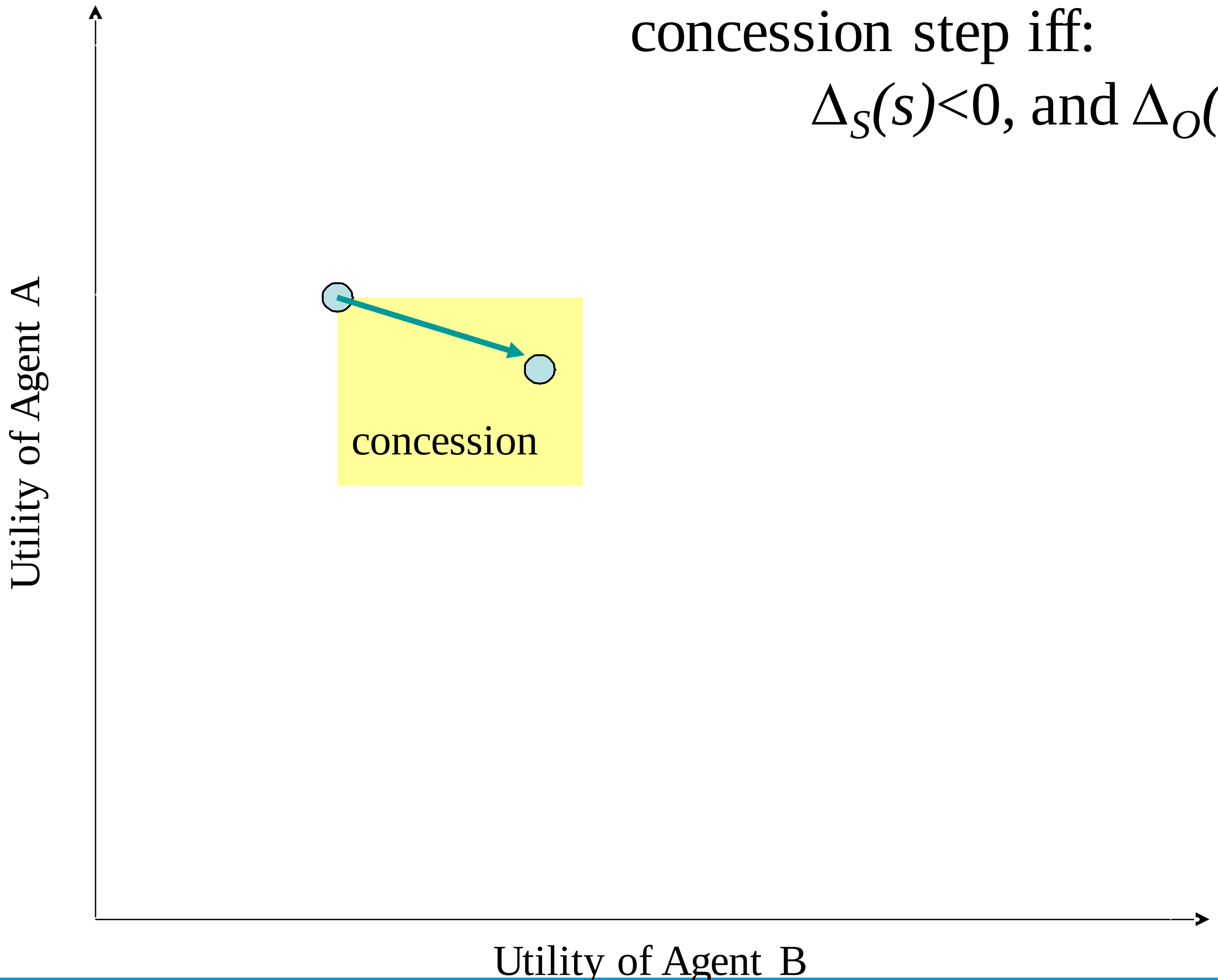
$\Delta_a(s)$: $\Delta_a(b, b')$ for a step $s = b \rightarrow b'$.

A trace t is a series of negotiation steps, i.e., transitions $b \rightarrow b'$ with b, b' offers.

Concession step

denoted by (S-, O≥), s is a concession step iff:

$\Delta_S(s) < 0$, and $\Delta_O(s) \geq 0$.

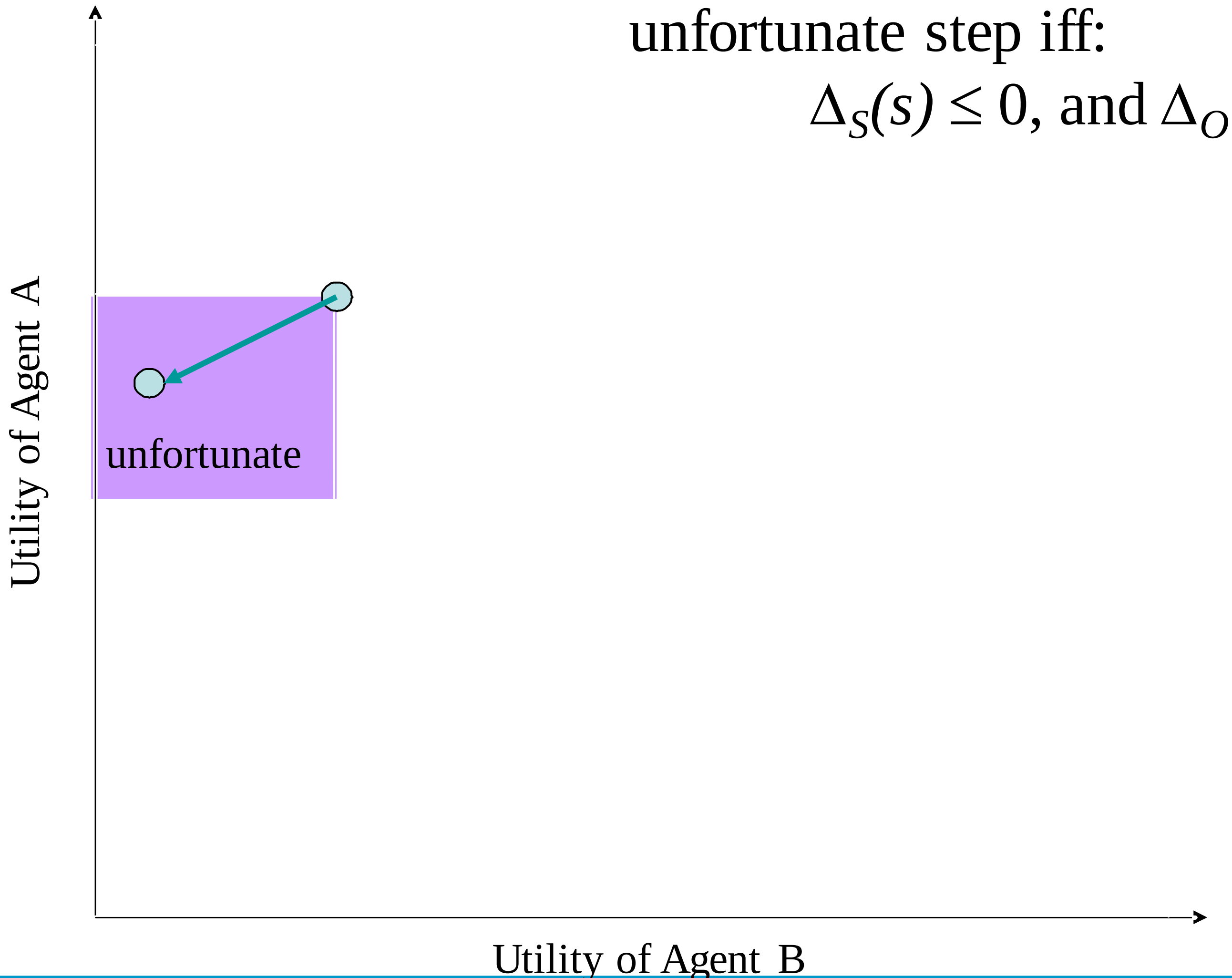


Bosse, Jonker, RRS 2005

Unfortunate step

denoted by (S≤, O-), s is an unfortunate step iff:

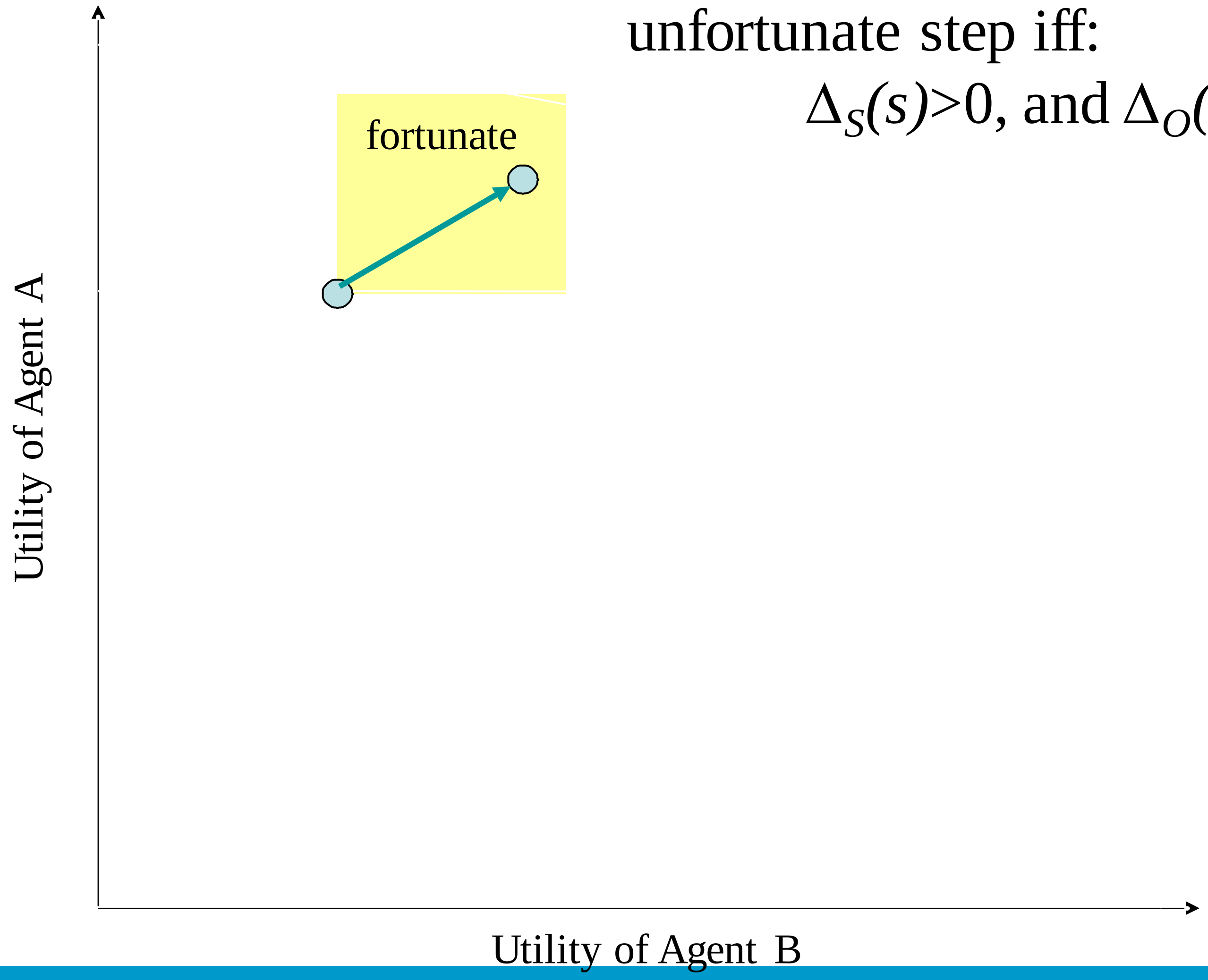
$\Delta_S(s) \leq 0$, and $\Delta_O(s) < 0$.



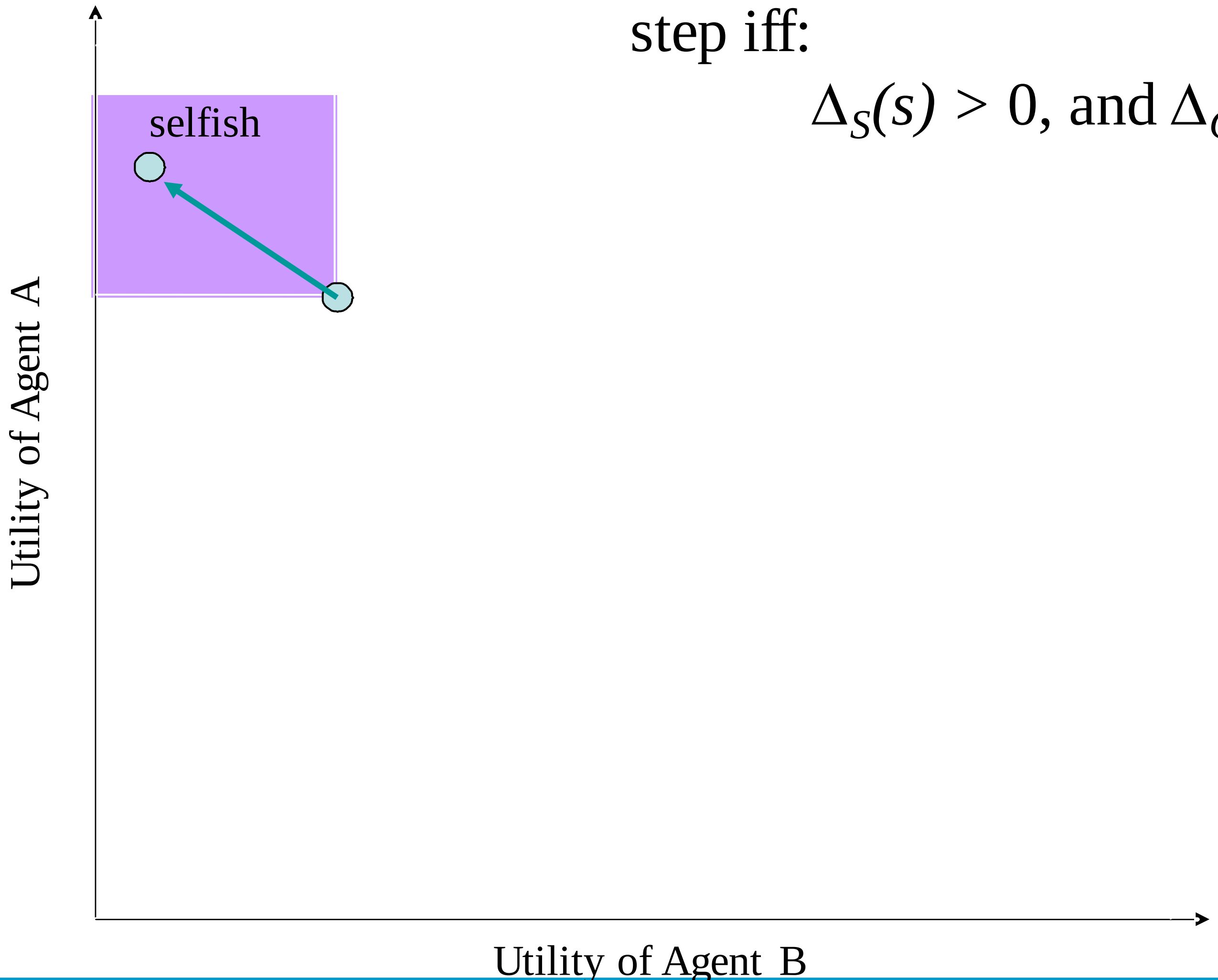
Fortunate step

denoted by (S+, O+), s is an unfortunate step iff:

$\Delta_S(s)>0$, and $\Delta_O(s)>0$.



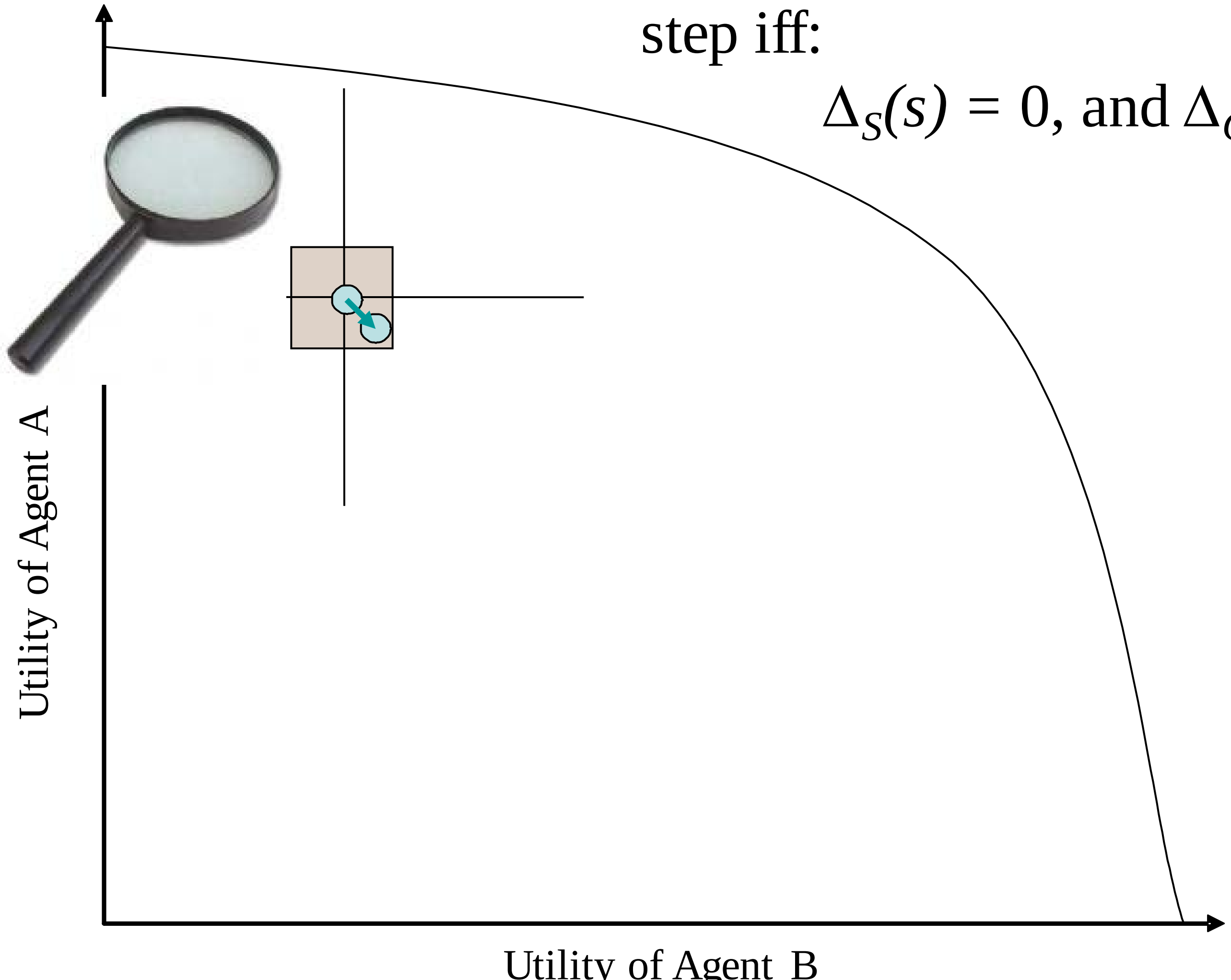
Selfish step



denoted by (S+, O≤), s is a selfish step iff:

$$\Delta_S(s) > 0, \text{ and } \Delta_O(s) \leq 0.$$

Silent step

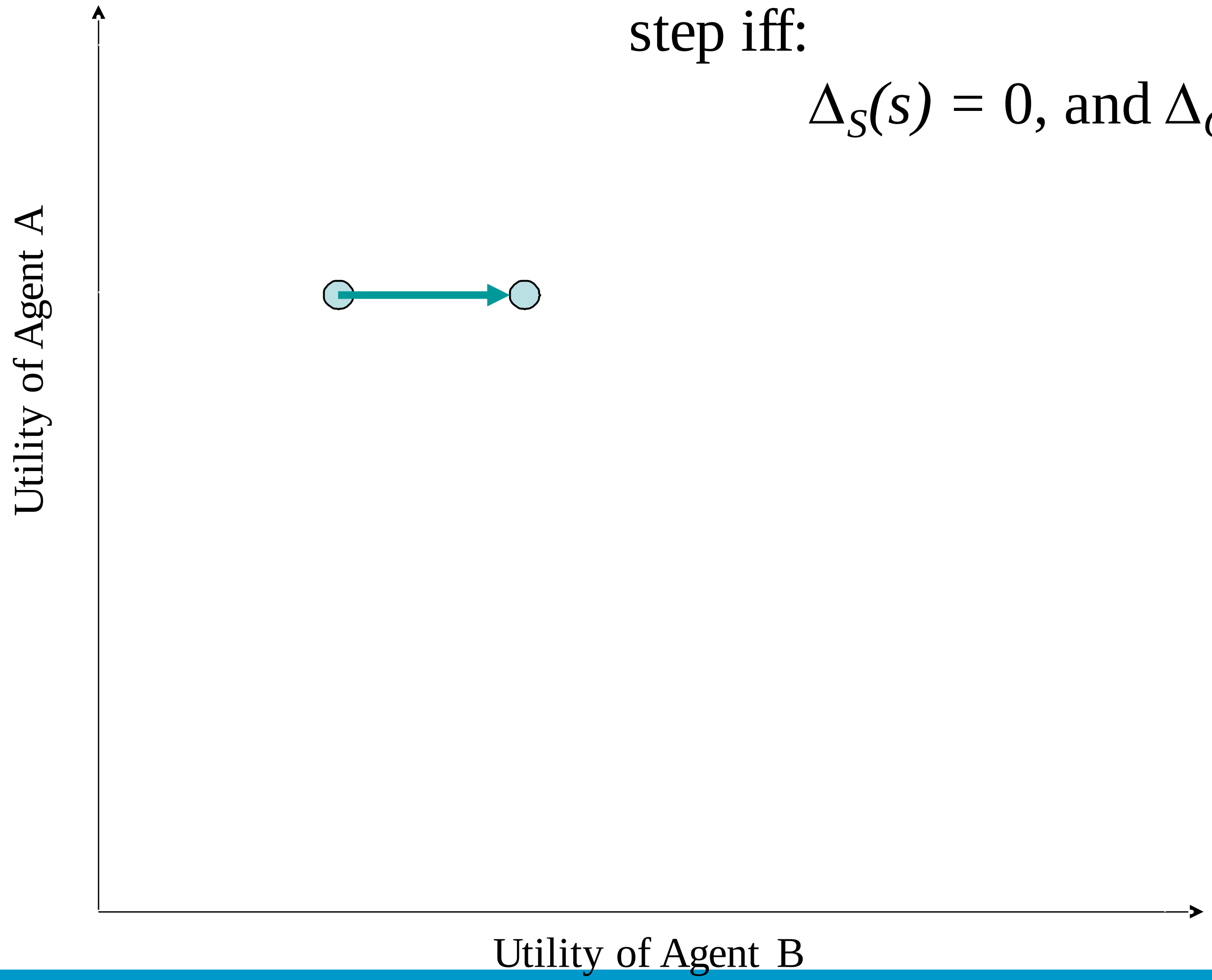


denoted by (S=, O=), s is a silent step iff:

$$\Delta_S(s) = 0, \text{ and } \Delta_O(s) = 0.$$

TU Delft

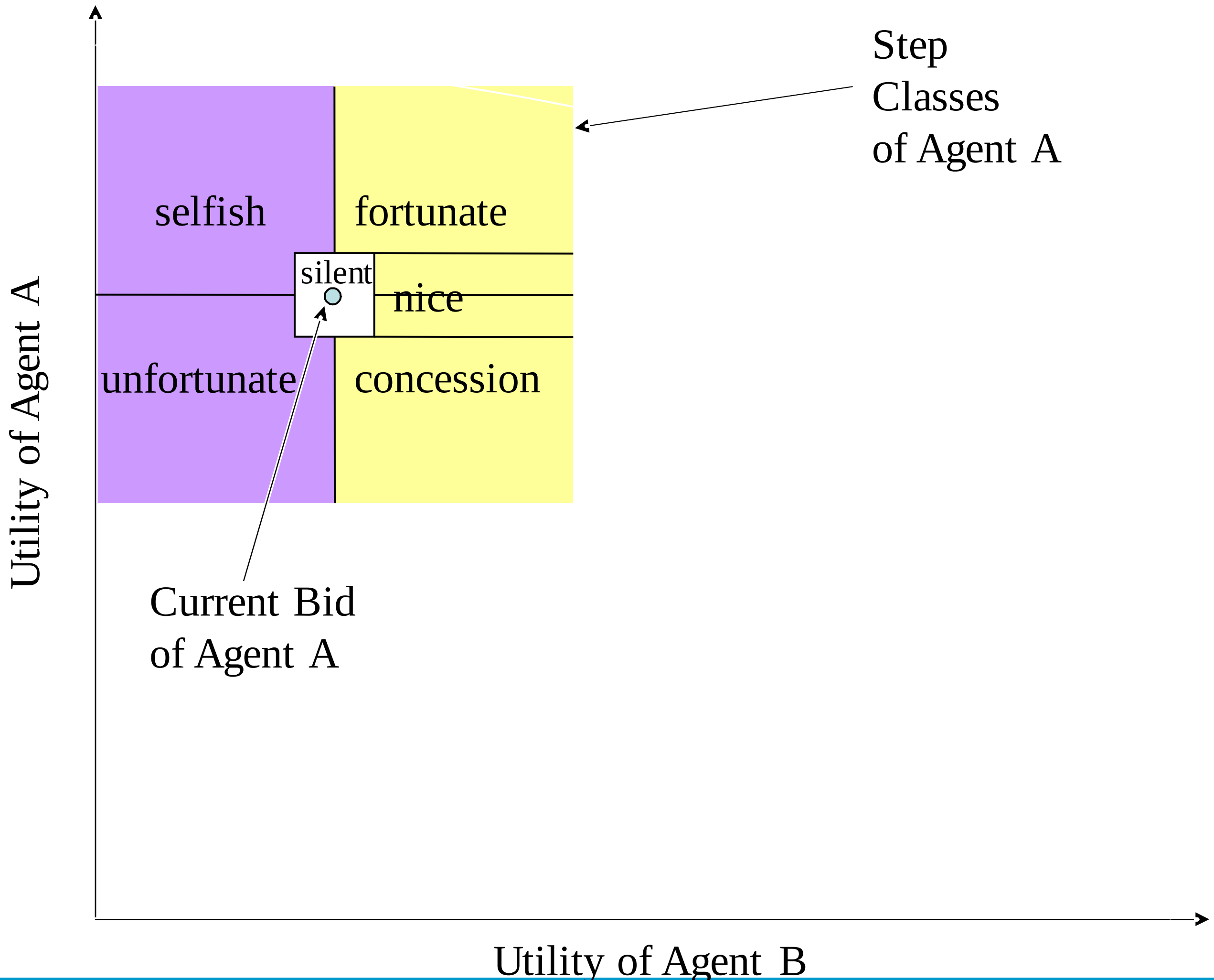
Nice step



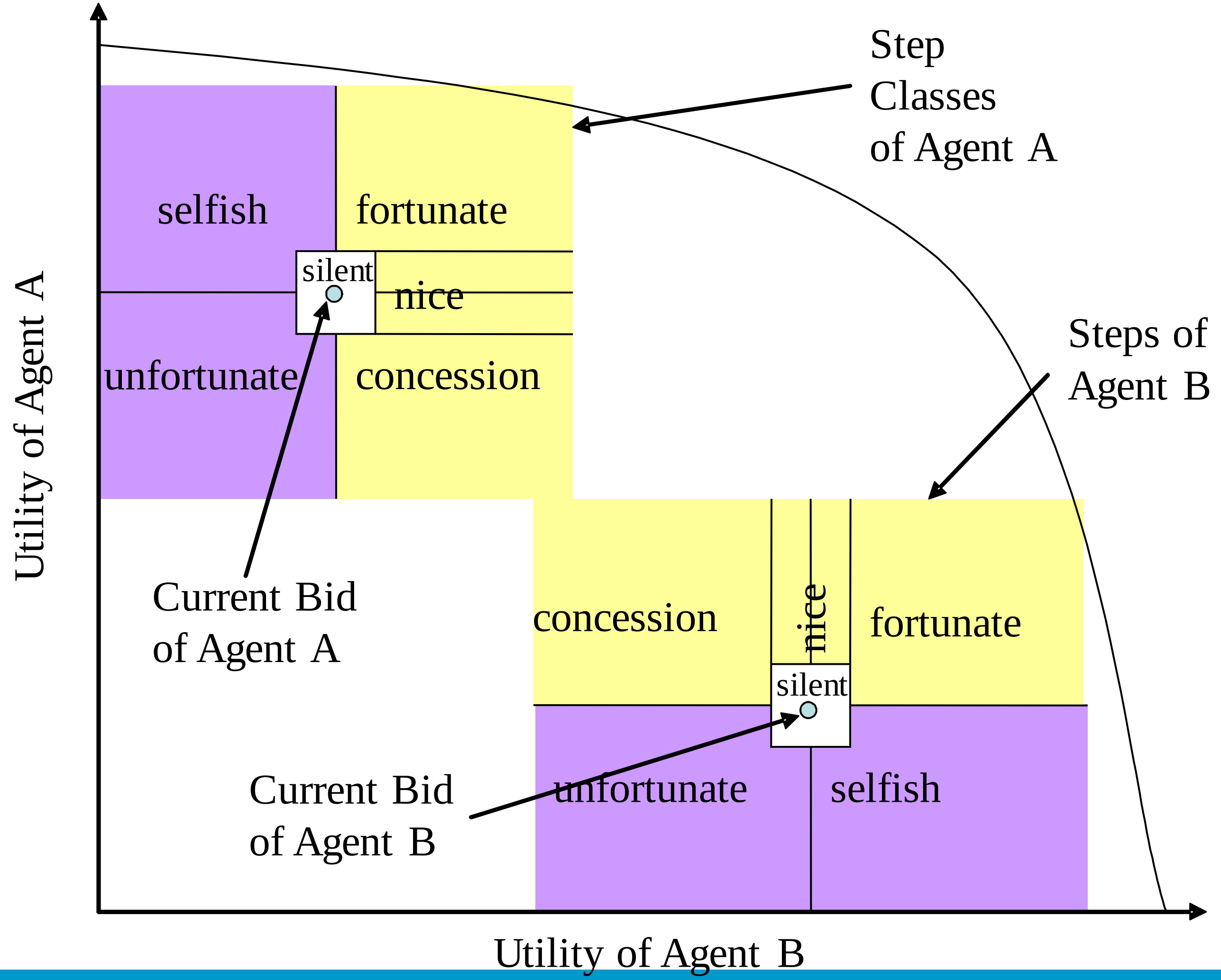
denoted by (S=, O+), s is a nice step iff:

$$\Delta_S(s) = 0, \text{ and } \Delta_O(s) > 0.$$

Classification of negotiation steps



Classification of negotiation steps



TU Delft

Sensitivity to Opponent Preferences

A rational negotiator would try to make *fortunate, nice, or concession steps.*

$$\text{sensitivity}_a(t) = \frac{\%_{Fortunate}(t_a) + \%_{Nice}(t_a) + \%_{Concession}(t_a)}{\%_{Selfish}(t_a) + \%_{Unfortunate}(t_a) + \%_{Silent}(t_a)}$$

- In case no selfish, unfortunate or silent steps are made we stipulate that $sensitivity_a(t)=\infty$.
- If $sensitivity_a(t)<1$, then an agent is more or less insensitive to opponent preferences;
- If $sensitivity_a(t)>1$, then an agent is more or less sensitive to the opponent's preferences, with complete sensitivity for $sensitivity_a(t)=\infty$.

TU Delft

The Three Strategies

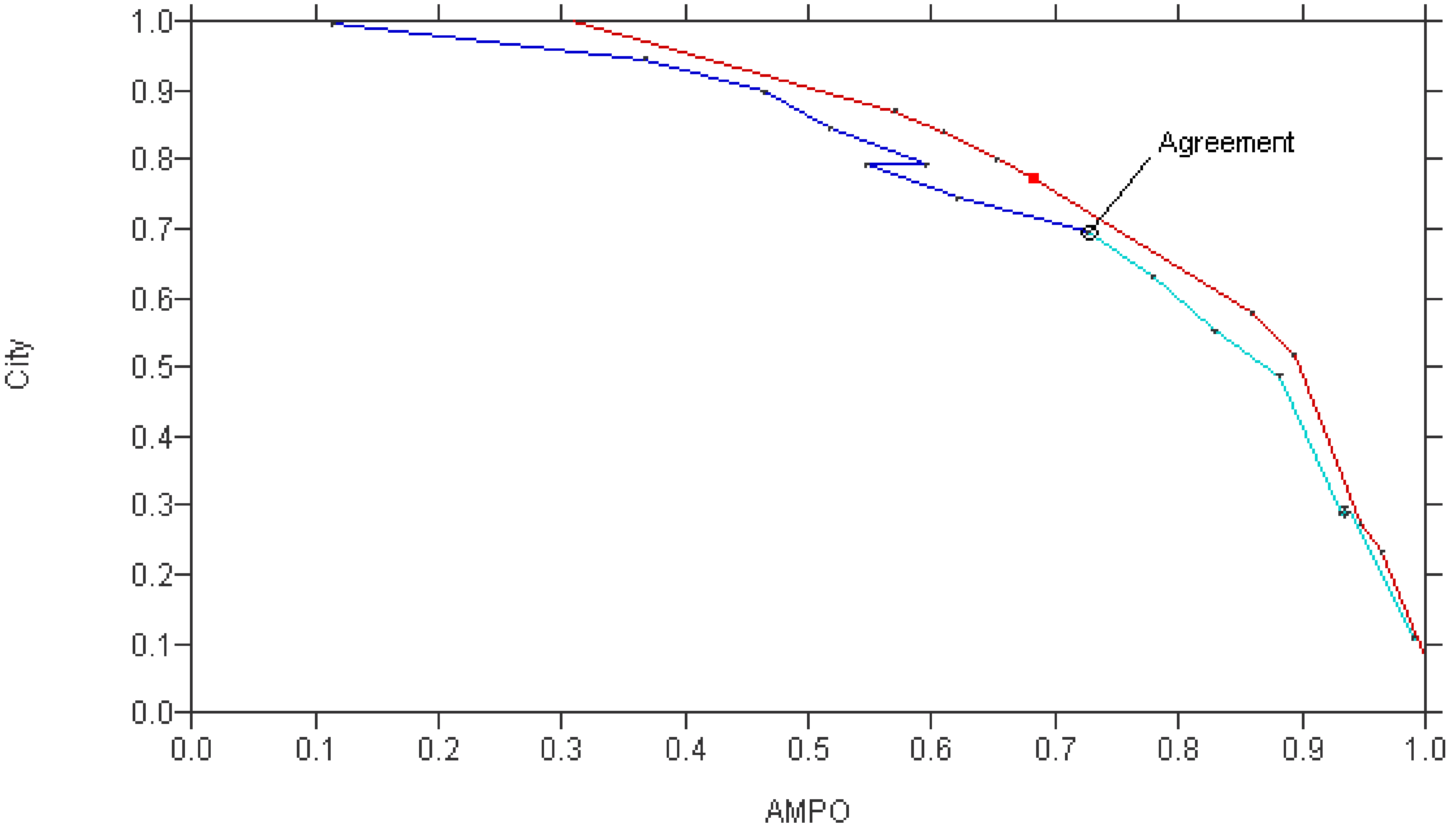
- ABMP [Jonker, Treur, 2001]
  - does not use any knowledge about opponent;
  - calculates concession step on every round of negotiation;
  - always make concession on every issue;
- Trade-off [Faratin, Sierra, Jennings, AIJ 2002]
  - uses domain knowledge;
  - tries to find bids on the same iso-level of own utility function that is closer to the current opponent's bid, makes concession of 0.05 if stuck;
  - uses opponent's bid to make trade-offs;
- Random-Walker [Hindriks, Jonker, Tykhonov, IAT 2007]
  - Selects values of issues randomly
  - Proposes only those bids that have own utility >0.6

TU Delft

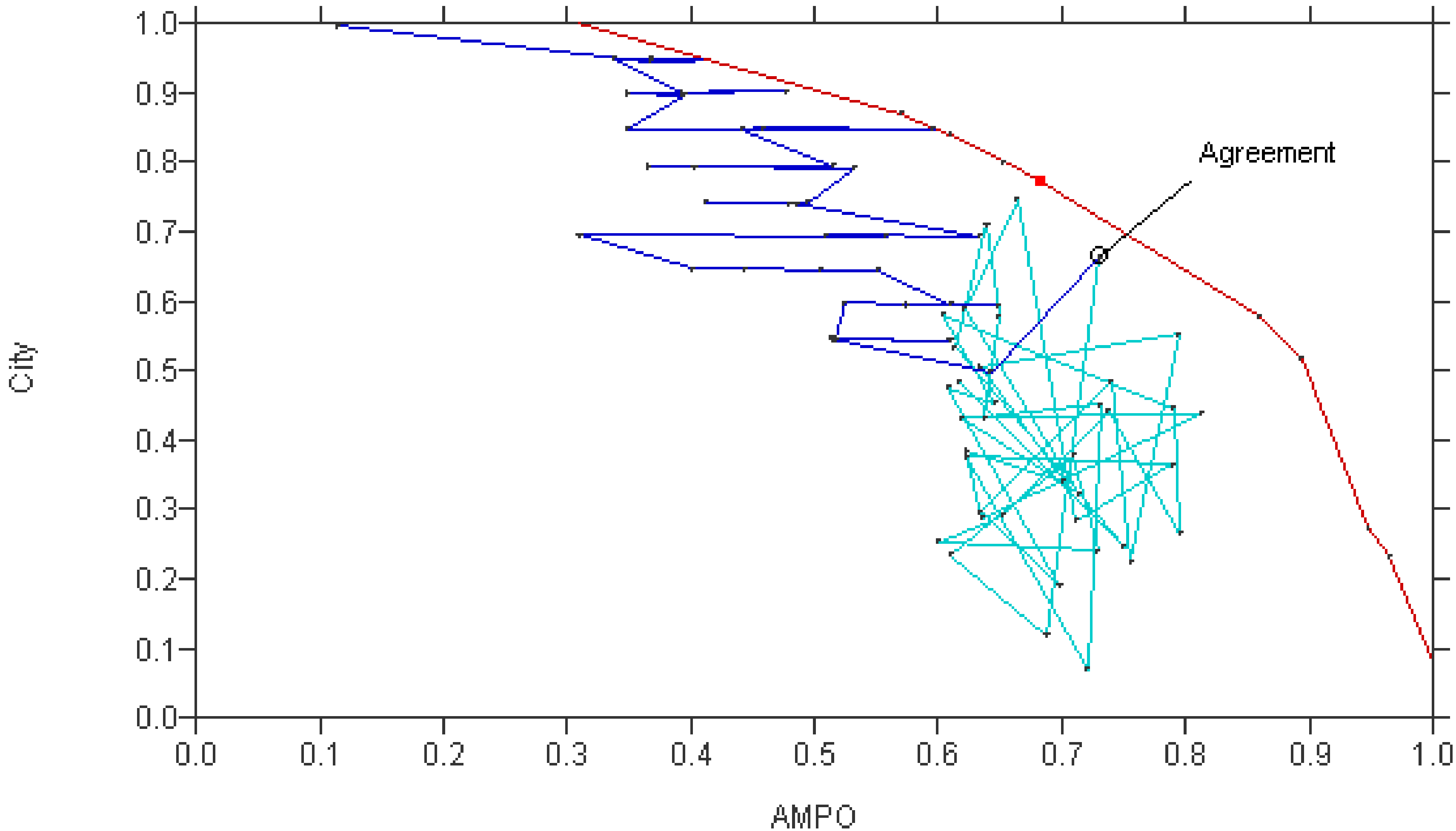
The Three Domains

- Second hand car selling domain:
  - 5 issues (4 discrete issues and price issue),
  - only the buyer's preferences and the price issue are predictable
- Service-oriented negotiation (SON):
  - 4 continues issues;
  - all issues are predictable;
- AMPO vs City
  - 10 issues;
  - only 8 issues are predictable;

Trade-Off (City) vs Trade-Off strategy (AMPO)



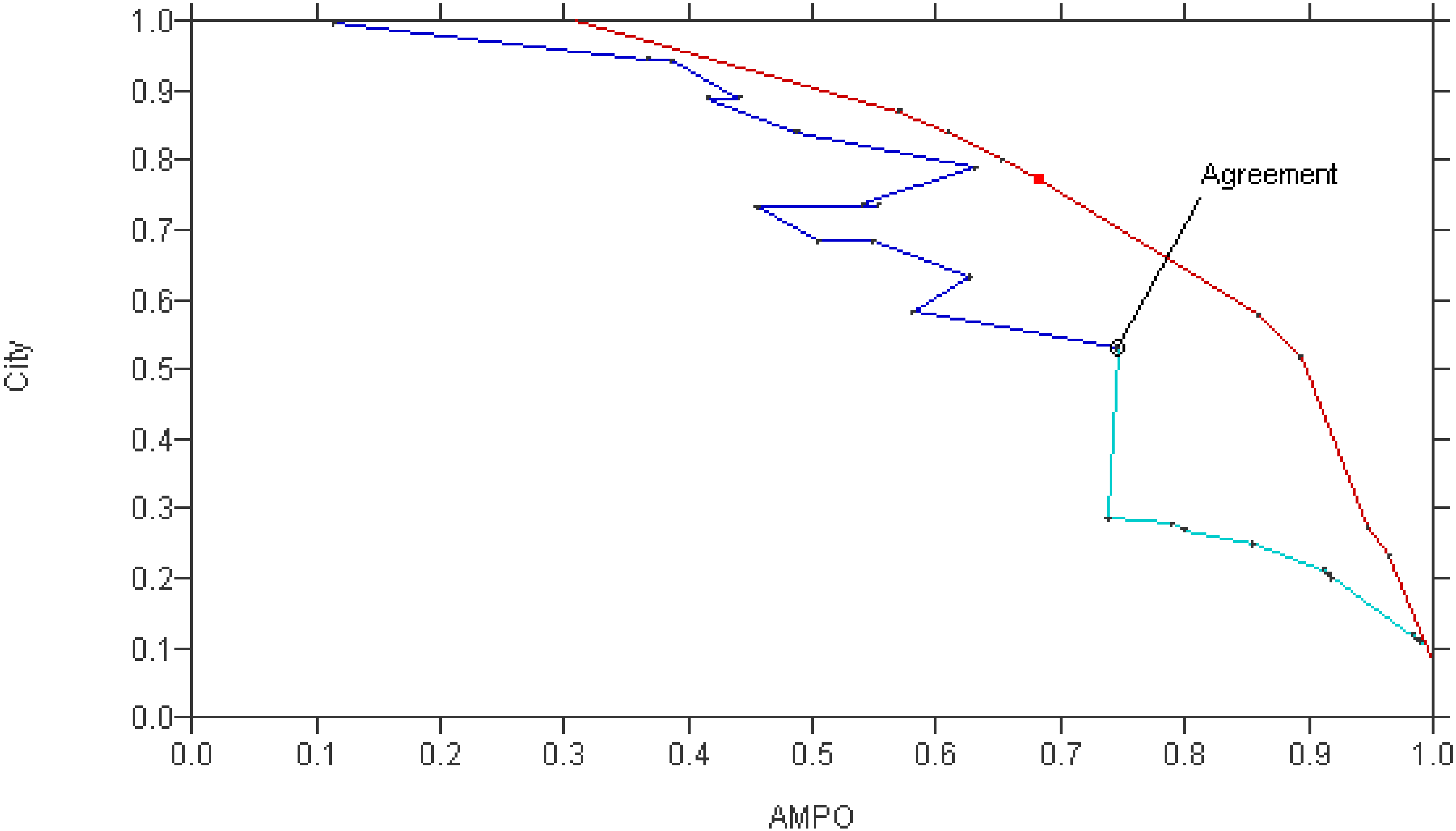
Trade-Off (City) vs Random Walker (AMPO)



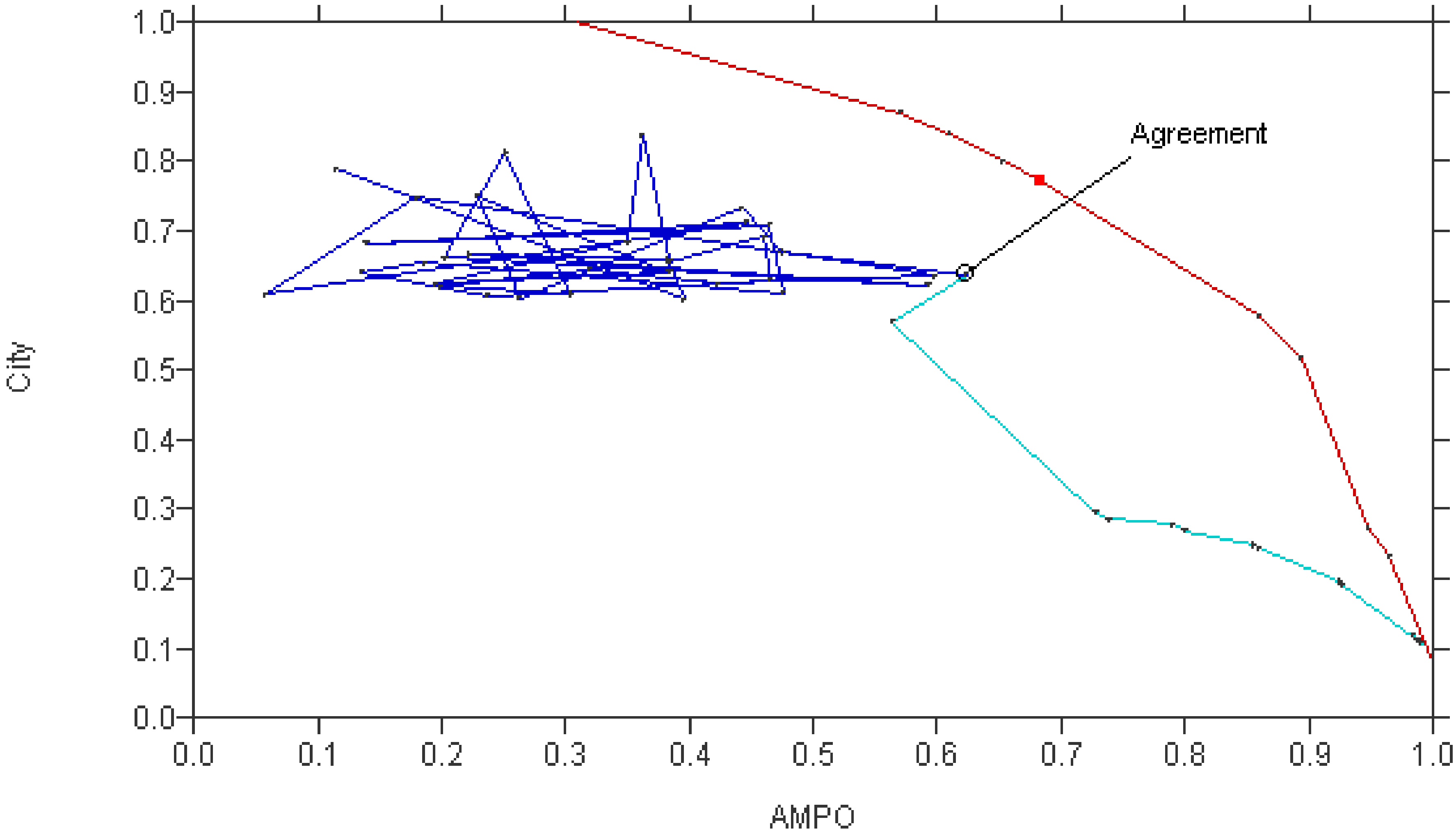
©Hindriks, Jonker, Tykhonov, IAT 2007

TU Delft

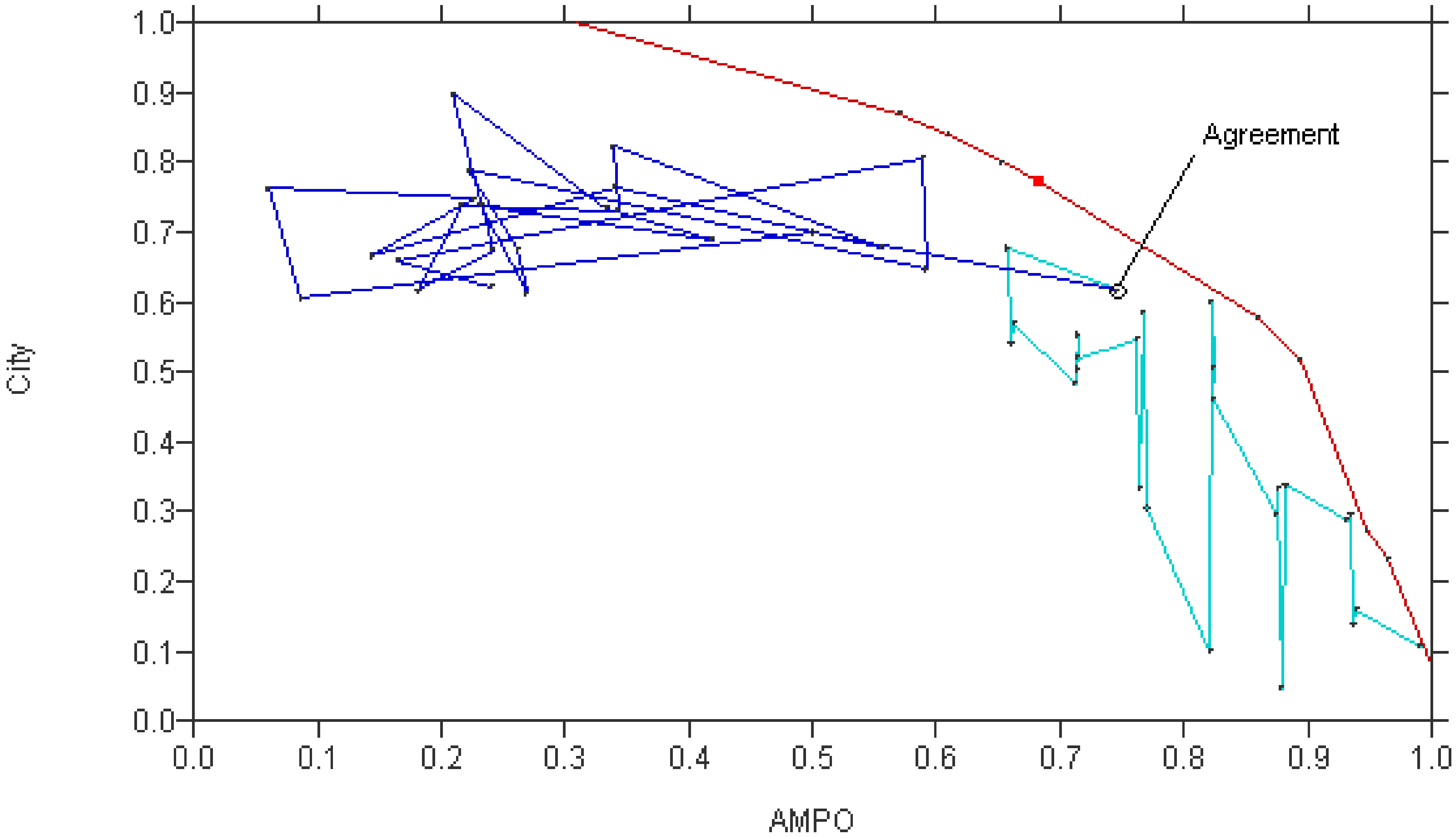
Trade-Off (City) vs ABMP (AMPO)



Random Walker (City) vs ABMP (AMPO)



Random Walker (City) vs Trade-Off strategy (AMPO)



Outcome Utility

- Overall utility:
  - ABMP 0.72,
  - Trade-Off 0.74, and
  - Random Walker 0.69.
- Trade-Off:
  - Outperforms ABMP on the SON domain with complete information and on the AMPOvsCity domain;
  - Underperforms wrt ABMP on the second hand car domain due to wrong weights and unpredictable issues;
- ABMP:
  - Strong on the second hand car domain;
  - Underperforms on the SON domain.

Conclusions

- Want to negotiate efficiently? Know your partner!
- It is impossible to avoid unfortunate steps without sufficient domain knowledge or opponent knowledge.
- In the analysis of negotiation strategies, not only the outcome of a negotiation is relevant, but also the bidding process itself is important.
- When developing a general negotiation strategy test against many opponents and in many domains.

(Baarslag 2014): Tim Baarslag, Alexander Dirkzwager, Koen Hindriks, and Catholijn Jonker. The significance of bidding, accepting and opponent modeling in automated negotiation. In: 21st European Conference on Artificial Intelligence, 2014.

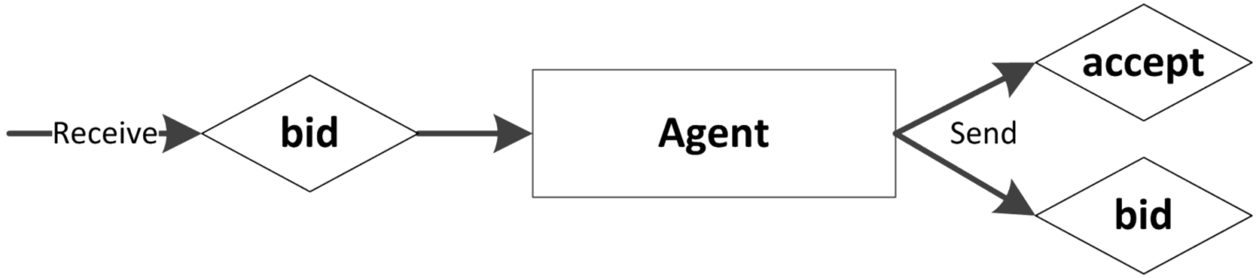
THE BOA FRAMEWORK

TUDelft

Introduction

In search of an efficient automated negotiator

- Challenge: from the outside, agent architectures are essentially a ‘black box’.



- A negotiation strategy is a result of complex interaction between various components, of which the individual performance may vary significantly.
- Overall performance measures make it hard to pinpoint reasons for success.

TU Delft

Introduction

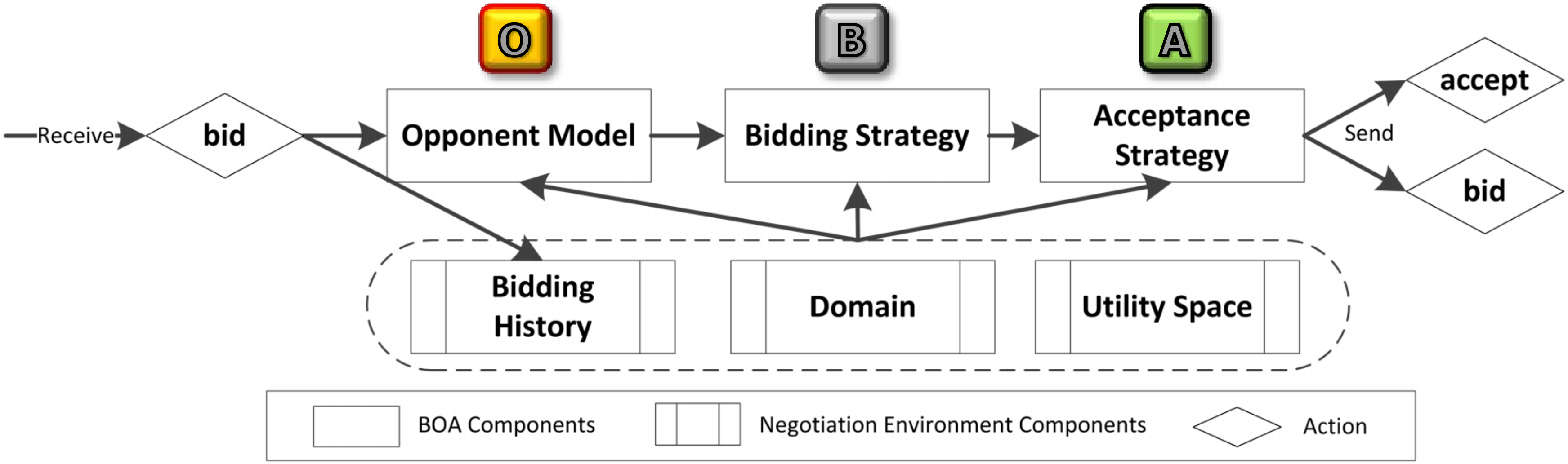
Component-based negotiation

- Many agent strategies are comprised of a fixed set of modules; generally, a distinction is made between three different modules:

  - One that decides which set of bids could be proposed next;
  - One that tries to guess the opponent’s preferences and takes this into account when selecting an offer to send out.
  - One module that decides whether the opponent’s bid is acceptable;

The BOA framework

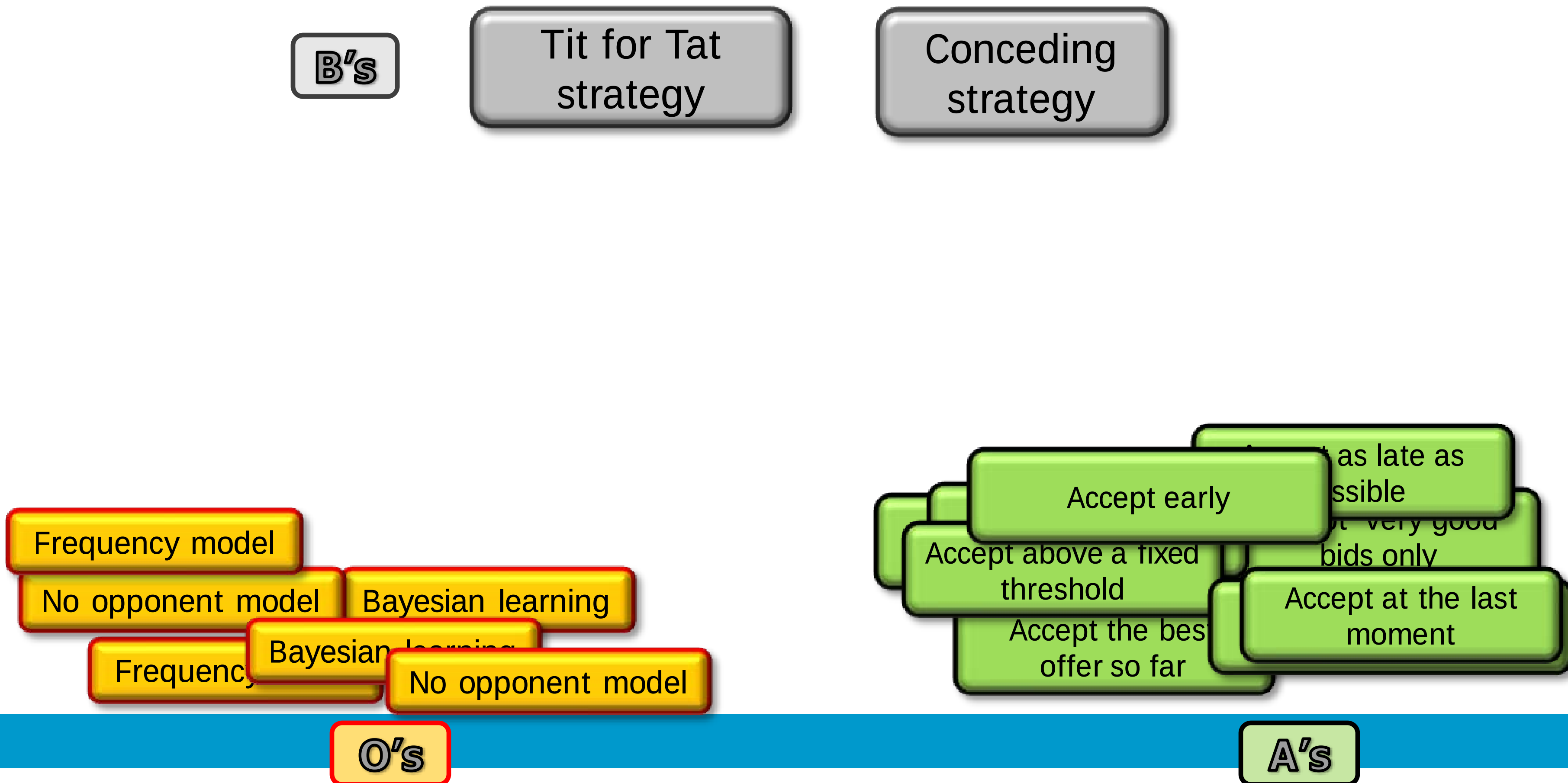
Negotiation flow

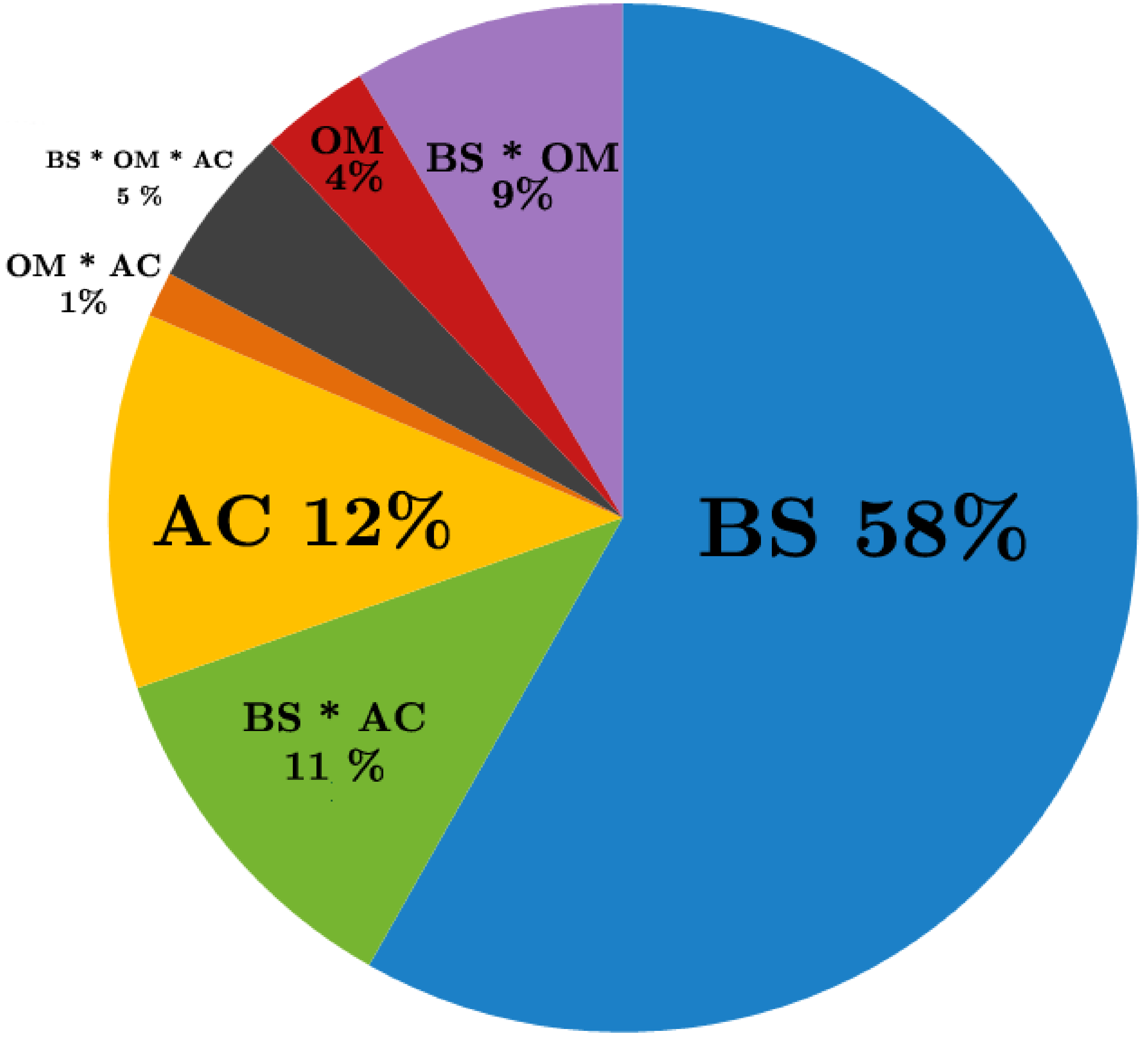


TUDelft

Applying the BOA Framework

Combining components




BS * OM * AC
5 %
OM
4%
BS * OM
9%
OM * AC
1%
AC 12%
BS 58%
BS * AC
11 %

Exam material

- Klein et al., 2003
- Hindriks et al., 2011
- These slides

References (1)

- Baarslag, T., Aydogan, R., Hindriks, K.V., Jonker, C.M., Fujita, K., & Ito, T., (2015). The Automated Negotiating Agents Competition, 2010-2015, *AI Magazine*, 36, pp:115-118.
- Tim Baarslag, Alexander Dirkzwager, Koen Hindriks, and Catholijn Jonker. The significance of bidding, accepting and opponent modeling in automated negotiation. In 21st European Conference on Artificial Intelligence, 2014.
- Baarslag T, Fujita K, Gerding EH, Hindriks K, Ito T, Jennings NR, Jonker CM, Kraus S, Lin R, Robu V, Williams CR, (2013). Evaluating Practical Negotiating Agents: Results and Analysis of the 2011 International Competition, Artificial Intelligence, 198, pages:73 - 103, issn: 0004-3702, doi: 10.1016/j.artint.2012.09.004. http://www.sciencedirect.com/science/article/pii/S0004370212001105?v=s5
- Jonker CM, Hindriks K, Wiggers P, Broekens JD, (2012). Negotiating Agents, AI Magazine, 33. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2421
- Luo, Xudong, Miao, C, Jennings, Nick, He, Minghua, Shen, Z and Zhang, M (2012) KEMNAD: A Knowledge Engineering Methodology for Negotiating Agent Development. Computational Intelligence, 28, (1), 51-105.

References (2)

Klein, M., et al., Protocols for Negotiating Complex Contracts. IEEE Intelligent Systems, 2003. 18(6): p. 32 - 38. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1249167&filter%3DAND%28p_IS_Number%3A27968%29 but easily accessible: http://ebusiness.mit.edu/research/Briefs/4Klein_Negotiation_Brief_Final.pdf

Rosenschein, J.S., and G. Zlotkin. Rules of Encounter. MIT Press, Cambridge, MA, 1994. 16.

Rubinstein, A., Perfect equilibrium in a bargaining model. Econometrica, 50:97–109, 1982. http://arielrubinstein.tau.ac.il/papers/11.pdf

Raiffa, H., The art and science of negotiation. 1982, Cambridge, Mass.: Belknap Press of Harvard University Press. x, 373.

Baarslag T. 2014. What to Bid and When to Stop. Delft University of Technology. http://mmi.tudelft.nl/sites/default/files/thesis.pdf

Hindriks KV, Jonker CM, Tykhonov D, (2011). Let's dans! An analytic framework of negotiation dynamics and strategies, Web Intelligence and Agent Systems, 9, pages:319-335 (see reading material on blackboard for the paper)