AAMAS 2013

Saint Paul
Minnesota
USA

6th - 10th May 2013

twelfth international conference on autonomous agents and multiagent systems

T8 – Tutorial on Agent-Mediated Electronic Negotiation

Han La Poutré (CWI, Amsterdam, NL)
Valentin Robu (University of Southampton, UK)
Shaheen Fatima (Loughborough University, UK)
Takayuki Ito (Nagoya Institute of Technology, Japan)
Katsuhide Fujita (University of Tokyo, Japan)

May 7, 2013

Agent-Mediated Electronic Negotiation

AAMAS 2013, St. Paul, Minnesota

Tutorial No. 8, 7 May '13

Han La Poutré
CWI, Amsterdam, NL

Valentin Robu
University of Southampton, UK

Shaheen Fatima
Loughborough University, UK

Takayuki Ito
Nagoya Institute of Technology, Japan

Katsuhide Fujita
University of Tokyo, Japan

Introduction: what is negotiation?

- Negotiation = method of competitive (or partially cooperative) allocation of goods, resources, or tasks between agents
- Applications:
  - Electronic commerce (shopbots etc.)
  - Distributed logistics
  - Hospital scheduling
  - Bandwidth allocation
  - Supply chain management
  - Crisis management

Negotiation (bargaining) versus auctions

- Two main types of allocation mechanisms:
  - **Auctions**
    - Fixed protocol and rules, mainly centralized
    - Possible to design optimal mechanisms that guarantee certain desirable properties – especially in one-shot settings
    - Often target at direct revelation (bidders reveal the prices for preferred combinations), presence of a trusted center
  - **Negotiation (bargaining)** mechanisms:
    - Allows the use of more decentralized, flexible protocols
    - Allows customized and complex agreements
    - Agents can use incomplete information about their opponent (and their own) preferences
    - Focus is on designing agent strategies, not the mechanism itself
  - **This tutorial focuses on bargaining!**

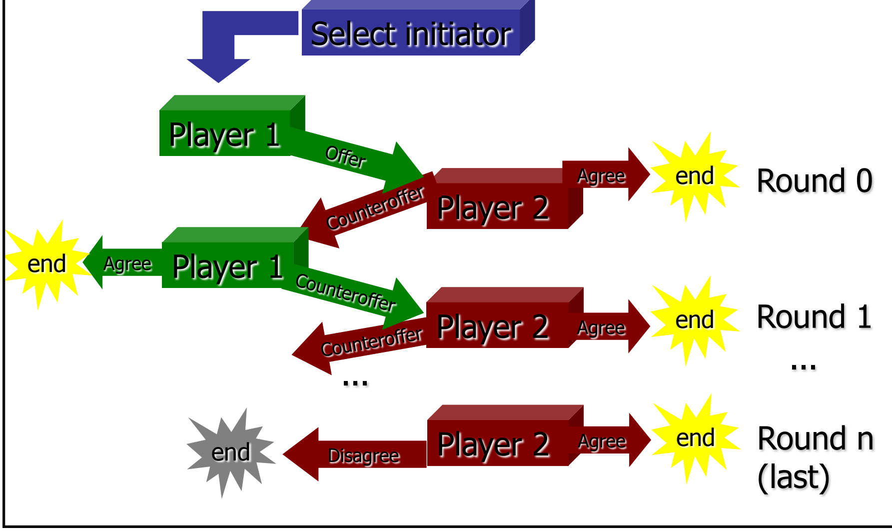
Bargaining: introductory notions

Games

- Game: players that interact with each other
  - And get a payoff at the end
- A game is defined by its rules
  - *Who* can do *What* and *When;*
  - Who *gets what* at the *end* of the game
- Games strategy of a player
  - Description of the actions by that player
- Negotiation: specific type of game
  - Agents want to make a deal
  - 2 players: buyer and seller
  - Usually involves alternating-offers

5

Ex: negotiation



6

Perspectives on the negotiation problem

- Classical game theoretic
  - Assumption 1: Rules of the game, preferences & beliefs of all players are common knowledge
  - A2: Full rationality on the part of all players (=unlimited computation)
  - Preferences encoded in a (limited) set of player **types**
  - Closed systems, predetermined interaction, small sized games
- Heuristic perspective (<= AI, MAS research)
  - No common knowledge or perfect rationality assumptions needed
  - Agent behaviour is modeled directly
  - Suitable for open, dynamic environments
  - Space of possibilities is very large
- Argumentation-based negotiation
  - Based on formal logics of dialogue games

7

Utility functions

- Utility function = function that maps all possible game outcomes in the choice set into …
  - **Cardinal utility -** mapping to a real number (e.g. between 0 - 1)
  - **Ordinal utility** - specifies only an (partial) ordering between outcomes
- Utility functions can be:
  - **Over a single issue** (e.g. only over price)
    - *Note:* In a setting where the utility is simply the expected profit to be obtained (monetary), utility functions are called **quasi-linear.**
  - **Over multiple issues (attributes)**
    - **Discrete-values**
    - **Continuous**

8

Example: Negotiation

- Agents negotiate about the exchange of services or goods and a price
- (Monetary) *Utility* of a possible negotiation outcome *D* for each agent *a* is e.g.:
  - Amount it is willing to pay for the goods/services exchanged in the deal: its valuation $v_a(D)$
    - the deal's/goods/services *value* to it
  - minus the price of the deal itself for it: $p_a(D)$
    - the actual price it has to pay
  - thus $v_a(D) - p_a(D)$
- Each agent wants maximal utility

9

Example Utility

- E.g., a possible negotiation outcome:
  - Agent 1 gives one television to agent 2
    - Agents 1's valuation: 500 euro; agent 2: 300 euro
  - Agent 2 gives two goods to agent 1: car and a bicycle
    - Agent 1's valuation: 2500 euro; agent 2: 1000 euro
  - Valuation of the goods transfer in the deal:
    for agent 1 this is +2000 euro
    for agent 2 this is -700 euro
- If in addition, agent 1 has to pay the price of 1000 euro to agent 2, then for this possible outcome:
  - the deal has a utility value of
    1000 euro for agent 1, and
    300 euro for agent 2

10

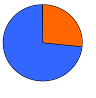
Pareto efficiency

- Pareto efficiency:
  - A game outcome *d* is *pareto efficient* (*pareto optimal)* if there is no game outcome that is better for at least one agent and not worse for the other agent:
    - There is no game outcome *d'* for agents A and B s.t.
      $[\ u_A(d') \geq u_A(d)$ and $u_B(d') \geq u_B(d)\ ]$ and
      $[\ u_A(d') > u_A(d)$ or $u_B(d') > u_B(d)\ ]$

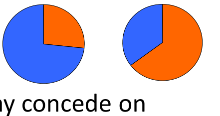
11

Zero vs. non-zero sum games

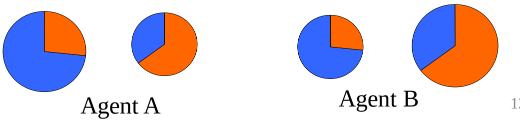
- Zero sum game: one party's gain is always the other party's loss



  - **E.g. - Pie-sharing game**
    - **- Negotiating over a single issue (e.g. price)**
- Non-zero-sum game: trade-offs are possible such that both parties improve their utility



  - Usually occur in the case of **multiple issues**
  - E.g.: in a work contract negotiation, the employer may concede on holiday days and New Year bonus
  - Employees concede on irregular work schedules



Agent A Agent B

12

Negotiation protocols (1)

- Protocol = rules of the game
- Protocols for:
  - **One issue negotiation**
  - **Multi-issue negotiation**
- Classification:
  - One shot (one party makes the offer, the other accepts or rejects)
  - Alternating offers (parties make repeating counter-offers)
  - Simultaneous vs. alternating offers

13

Multi-issue negotiation protocols

- If there are several issues, in which order to deal with them?
  - **"Integrative" (or "Global") negotiation protocols:** all issues negotiated at the same time
  - **Sequential protocols:** one issue at a time
    - **Independent implementation** -> take effect separately
    - **Simultaneous implementation** -> all issues must be agreed upon before the agreement takes effect
    - **Agenda problem:** in which order to negotiate the issues [Fatima et al '04]

14

Equilibrium concepts: Nash equilibrium

- Proposed by John Nash (1951)
- Strategies of all players are said to be in Nash equilibrium if no other party can benefit by unilaterally changing his/her strategy
- Other equilibrium concepts exist
  - **Stronger: dominant strategies, subgame perfect**
  - **Weaker: Bayes-Nash**

Other equilibrium concepts in non-cooperative games

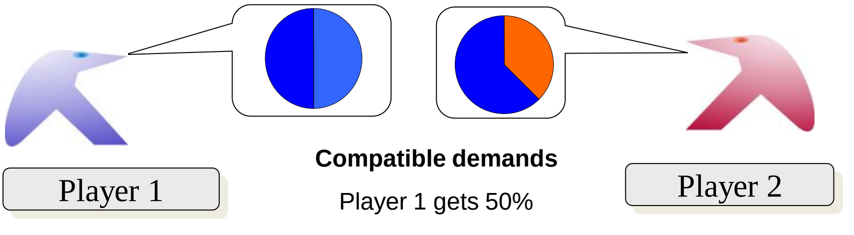
- **Dominant strategy** equilibrium: optimal in all circumstances, regardless of the bid of other players (e.g. bidding your true value in a Vickrey auction)
- **Bayes-Nash** equilibrium: optimal given the known prior probability distribution over the other players' types
- **Subgame perfect** (Selten): optimal strategy for the entire game = optimal in every subgame

Classic bargaining games

17

Nash demand game (1)

- Setting: 2 agents want to share a pie (sum of money etc.), i.e. “bargaining surplus”.
- “One shot” game
  - **Both agents simultaneously demand a fraction of the pie**
  - **If demand are compatible (sum <1), each gets his demand**
  - **If not, each get the disagreement payoff (typically 0)**



Compatible demands

Player 1 gets 50%

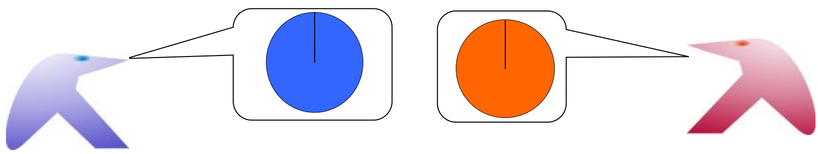
Player 2 gets 33%

Not Pareto efficient and not Nash equilibrium

18

Nash demand game (2)

- **What is a Nash equilibrium in this game?**
- **Is this equilibrium always Pareto-efficient?** (i.e. no piece of the pie gets thrown out)



Player 1

Player 2

Incompatible demands

Each player gets nothing

NOT Pareto efficient

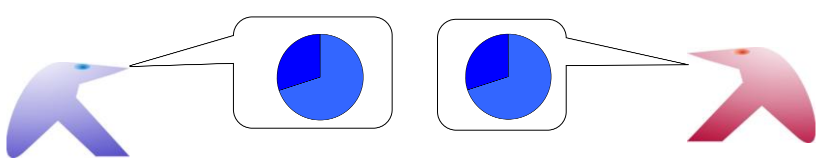
BUT this strategy is a

Nash equilibrium !

19

Nash demand game (3)

- **What is a Nash equilibrium in this game?**
- **Other equilibriums are indeed Pareto-efficient?** (i.e. no piece of the pie gets thrown out)



Player 1

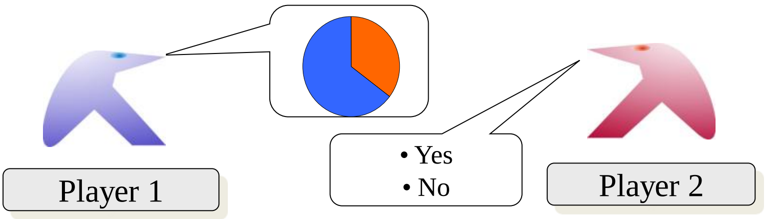
Player 2

Perfect splits of the pie
Player 1/2 gets e.g. 70%
Player 2 gets e.g. 30%
Pareto efficient
And Nash equilibrium

20

The ultimatum game (1)

- Less “competitive” than the Nash demand game
- Two stages:
  - **Agent 1 proposes a split of the pie**
  - **Agent 2 can either accept or refuse**
  - **If agent 2 refuses, both get nothing**

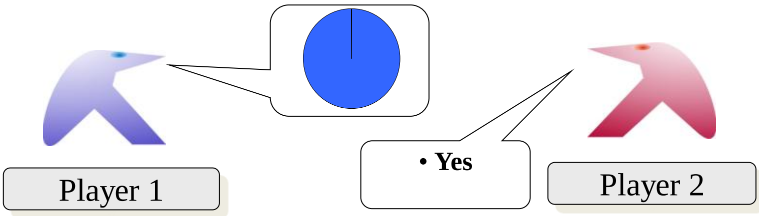


The ultimatum game (2)

- Game theory: what is the expected outcome in case of “rational” behaviour
  - Look forward and reasoning backwards: what will player 2 do with my offer x?
  - Player 2 accepts any offer > 0
  - Player 1 offers (almost) 0 and gets (almost) all the added values
- “Take-it-or-leave-it” offer of player 1

The ultimatum game (3)

– Only one **subgame perfect equilibrium**: (c.f. Binmore):
  - **Agent 1 demands the whole surplus (or –ε where ε↓0)**
  - **The other agent ACCEPTS**



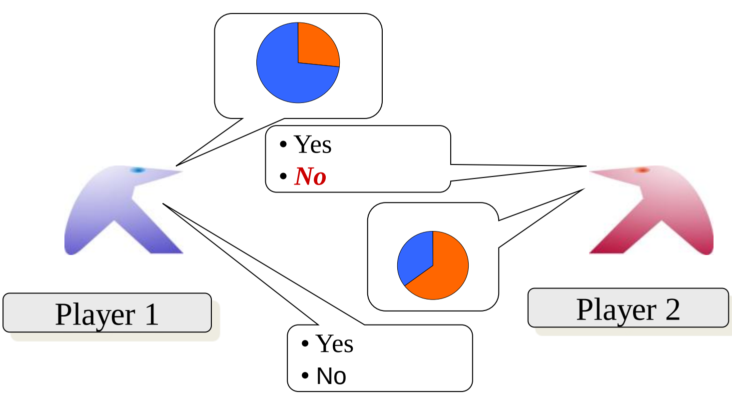
- Does this equilibrium model *real* human behaviour?
- Why not ? (-> perfect rationality assumption)

23

The alternating offers game (1)

– Each party has a right to make a (counter-offer) offering the other x the pie, while he/she keeps 1-x

– Game continues until one party accepts the other's offer or a stop criterium is reached, e.g. fixed deadline



24

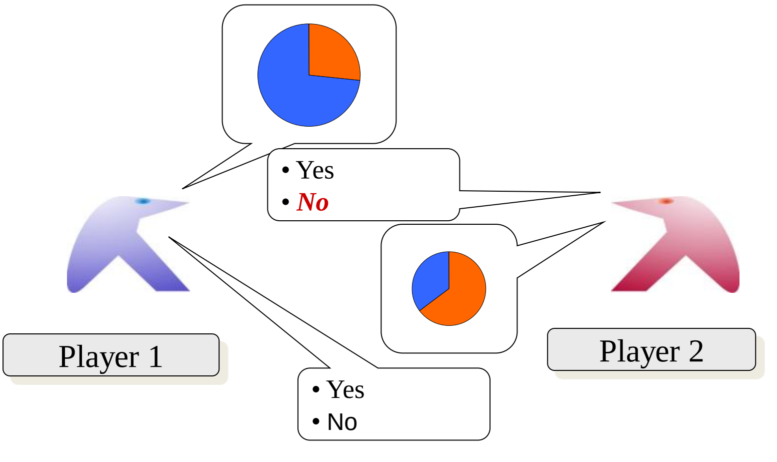
Alternating offers (2): Finite time horizon

- Several ways to model time pressure:
  - **Fixed deadlines**
  - **Fixed bargaining costs per round**
  - **Discount factors**
  - **Break-off probability**

- Case of discount factors:
  "Melting" ice cake
  - Utility of agent i at time t, given initial utility X: $\mathbf{X^*(\delta_i)^t}$
- Agents can perform "backward reasoning" starting from possible outcomes

25

Alternating offers (3): Finite time horizon



26

Alternating offers (4): Finite time horizon

- Example: cake halves each round
  - Player 2 can now only get half in the second round
  - Look forward and reason backwards: player 1 offers half of the cake in the first round
  - Infinite negotiation game for a melting cake: always a 2/3 vs. 1/3 division

Alternating offers game: the Rubinstein solution

- Players 1, 2 have discount factors: δ_1 , δ_2
- Both players have complete information about the other's δ
- Rubinstein proved that the only SPE is one in which player 1 gets: $(1-\delta_2)/(1-\delta_1\delta_2)$ and player 2 gets the rest
  - **In case of same factors $\delta_1 = \delta_2$:** $(1-\delta)/(1-\delta^2)$
- Result: players don't actually need to bargain
- In SPE: Player 1 proposes $(1-\delta_2)/(1-\delta_1\delta_2)$ and player 2 accepts

Negotiation concession tactics

Reference:
[Faratin, Sierra & Jennings, '98]

29

Monotonic concession protocol

- [Rosenschein & Zlotkin '94: Rules of Encounter]
- Players are not allowed to make offers which have a lower utility for their opponent than their last offer
- The minimum concession per round can be fixed above 0 => guarantee to terminate
- Question: how to make concessions?
  - If I do not know the opponents preferences
  - If there are multiple issues
- Note: In multi-issue negotiations with unknown opponent preferences, it is not always possible to make monotonic concessions

30

Time-dependent concession tactics (1)

- Suppose we have a buyer (the case of the seller is symmetrical) which desires to buy a good for an aspiration price Pmin and reservation price Pmax (highest he is willing to pay); deadline is a time T_{max}
- Price offered at time t will be:

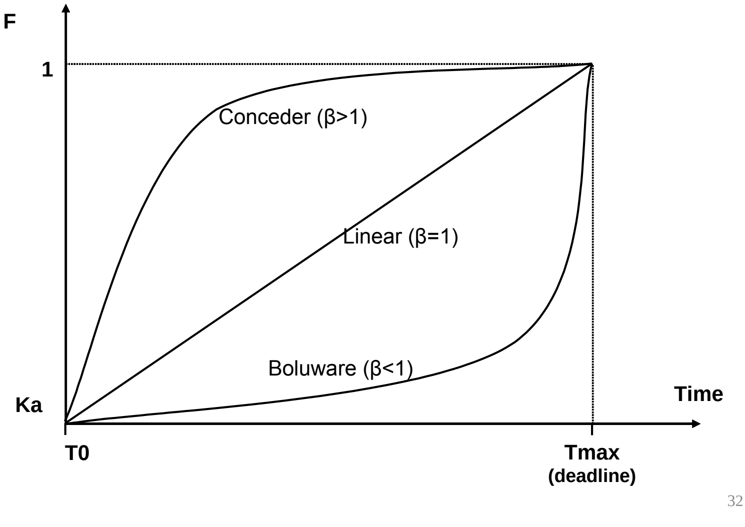
$$P(t) = P_{\min} + F(t)(P_{\max} - P_{\min})$$

- F(t) gives the fraction of the distance left between the first (best) offer and the reservation value

$$F(t) = k_a + (1 - k_a)\left(\frac{\min(t, T_{\max})}{T_{\max}}\right)^{1/\beta}$$

31

Time-dependent concession tactics (3)



32

Time-dependent concession tactics (4)

- Hard-headed (β->0)
  - No concessions, sticks to the initial offer throughout (the opponent may concede, though)
- Linear time-dependent concession (β=1)
  - Concession is linear in the time remaining until the deadline
- Boulware (β<1)
  - Concedes very slowly; initial offer is maintained until just before the deadline
- Conceder (β>1)
  - Concedes to the reservation value very quickly
- Tit-for-tat (discussed on next slide)

Tit-for-tat (TFT) concession strategy

- The agent detects the concession the opponent makes during the previous negotiation round, in terms of increase in its own utility function
- The concession the agent makes in the next round is equal to the concession made by the opponent in the previous round,
- Next offer must fall in the acceptable region (e.g. for price only negotiation, above the reservation price of the seller and below that of the buyer)

$$U_{own}(b_{own}, t+1) - U_{own}(b_{own}, t) \leq U_{own}(b_{other}, t) - U_{own}(b_{other}, t-1)$$

(The equal sign is sometimes used, though in practice it can be less)

Negotiating Agents and Learning

- Negotiating agents in agent systems
  - Frequent negotiations
- Fixed negotiation strategies could be exploited
  - Smart agents could learn what can be achieved, against various types of fixed strategies
  - (we return to this issue in the multi-issue negotiation section)
- Bounded rationality
  - Incomplete information about opponent agents
- Adaptivity and learning is necessary for agents
  - For large numbers of negotiations

Modelling time pressure: firm deadlines vs. time discounting

- There are two main mechanisms in which agent owners can specify time pressure to their agents:
  - **Firm deadline** (=time until which agreement must be reached, otherwise the agent gets disagreement payoff)
  - **Time discounting function**, i.e. an outcome reached later has smaller utility than an outcome reached now
  - Note: [Sandholm & Vulkan, ’02]: the game-theoretic equilibrium strategy in deadline bargaining is to wait until the very last moment to concede

Multi-issue (and multi-attribute) negotiations

References:

[Gerding, Bragt, La Poutré, ’02]

[Jonker & Robu, AAMAS ’04]

37

Single issue vs. multi-issue negotiation

- Single issue negotiations
  - Represent zero sum-games: one party’s win is the other one’s loss
  - Example: seller and buyer negotiating over a price
- Multi-issue negotiations
  - Non-zero sum games
  - An agent can make concessions in one or more issues in order to extract concessions in other issues preferable to him/her

38

Cooperative bargaining theory

- Seminal work: H. Raiffa – Art and science of negotiation, 1982
- FOTE (Fully Open Truthful exchange) assumption: both parties reveal their preferences to a central "mediator" agent, who computes optimal outcomes
- MAS research removes the need of a mediator, by allowing agents to work with incomplete information

39

Example set-up: sale of a car

- Four attributes (CD player, Extra speakers, Tow hedge (Drawing hook), Air conditioning) have value labels, and each party assigns to them an evaluation.
- The evaluation of price is described by a linear function (ascending or descending)

Value labels	Buyer	Seller
Good	100	30
Fairly good	85	65
Standard	70	80
Meager	20	65
None	0	100

40

Example set-up (2)

- Each attribute is given a preference weight coefficient.
- $$U_{contract} = \sum w_i * U_i$$ for all items i
- Symmetrical vs. asymmetrical preferences

	Buyer	Seller
Airco	90 (18%)	15 (3%)
Drawing hook	90 (18%)	15 (3%)
CD player	15 (3%)	90 (18%)
Speakers	15 (3%)	90 (18%)
Price	300 (59%)	-

41

EXAMPLE TRACE

BUYER'S INTERFACE

round	price	drawing hook	airco	extra speakers	cd_player	utility own bid	utility others
1	18000	good	good	good	good	1	0.740741
2	17450	fairly good	standard	meager	meager	0.92037	0.829185
3	18222	fairly good	standard	none	standard	0.909481	0.839926
..9	18583	fairly good	standard	none	standard	0.882741	0.867407

SELLLER'S INTERFACE

round	price	drawing hook	airco	extra speakers	cd_player	utility own bid	utility others
1	16900	none	none	none	none	1	0.316667
2	19306	fairly good	standard	none	standard	0.938269	0.595321
3	19161	fairly good	standard	none	standard	0.919679	0.799295
..9	18790	fairly good	standard	none	standard	0.872115	0.845577

42

Eficiency in multi-attribute negotiation

- "Intelligent" trade-offs between issues
    - Buyer wants both GPRS system and leather seats, but he only really cares about getting a GPRS system.
    - Dealer prefers to give neither (extra work), but leather seats are much more difficult to install
    - **Result:** Seller concedes on GPRS and the buyer on the quality of the leather
- How far can you go with such trade-offs?
- Pareto-optimal contract: A contract is said to be Pareto-optimal if no further improvement is possible in the utility of one agent, without reducing the utility of the other agents
- Pareto frontier: Set of all Pareto-optimal contracts

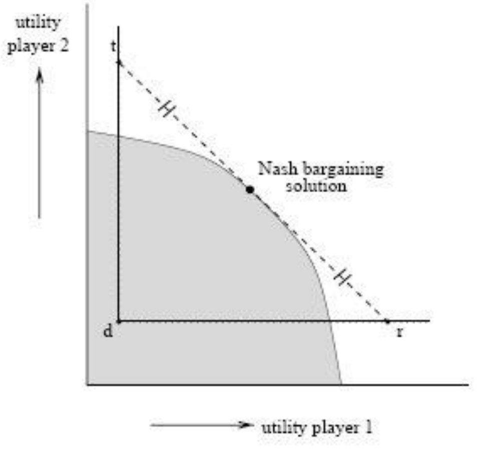
43

"Fairness" in cooperative negotiation

- Among the points on the Pareto-efficient frontier - a few so-called "solution concepts"
- Utilitarian:
    - Contract combination which maximizes the sum of utilities of the agents
- Egalitarian (Kalai-Smorodinsky):
    - Maximizes the MINIMUM of the two utilities
- Nash point:
    - Maximizes the PRODUCT of the utility functions of both agents

44

Pareto-efficient frontier: continuous case



- Pareto-efficient frontier for a game with 2 players
- Nash bargaining solution maximizes the product of the utilities
- J. Nash showed this agreement simultaneously satisfies 4 important axioms

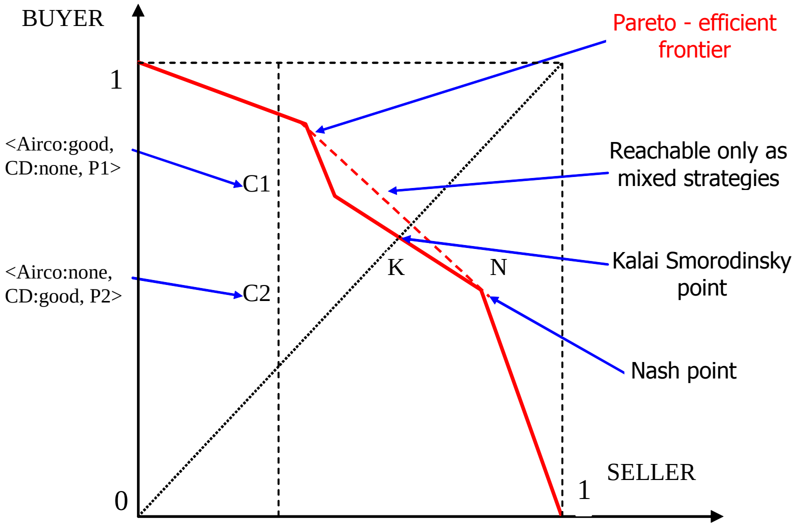
45

Domination. “Pure” vs. mixed strategies

- A contract B1 strictly dominates another contract B2 if both (or all parties) prefer B1 to B2
- Pareto-efficient contracts = not strictly dominated by any other contracts
- Pure strategies: Fixed outcomes
- Mixed strategies: Allocation is the result of a lottery
- **Example:**
  - Two people negotiate when to go and see a soccer game
  - Tom slightly prefers to go on Friday, Bob slightly prefers Monday
  - Pure strategy: firm agreement on one day
  - Mixed strategy: they toss a coin
  - **What is Pareto-efficient to do?**

46

Multi-issue contract space (pure strategies setting)



Guessing model

- BUYER

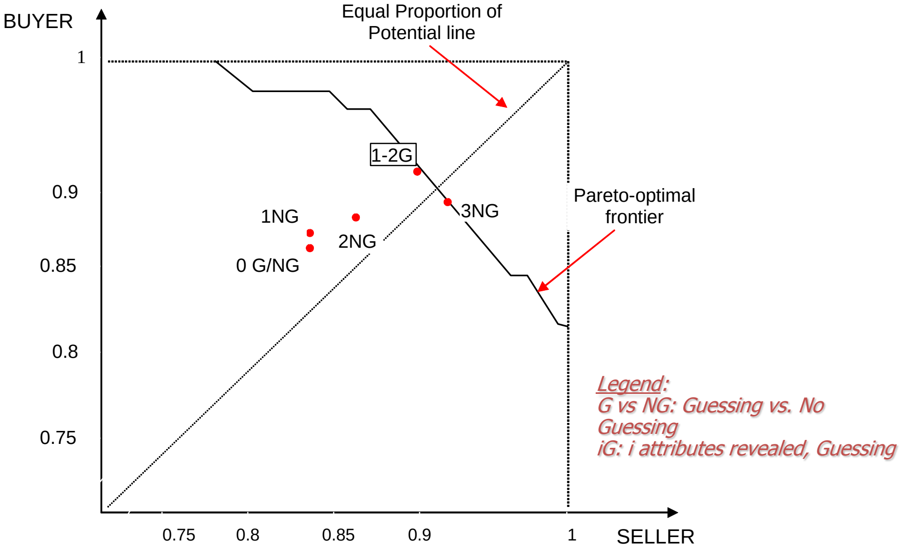
Drawing hook	Airco	Extra speakers	CD player	Price
Good	Good	Good	Good	18000
Fairly good	Standard	Meager	Meager	17450
Fairly good	Standard	None	None	17968

RWDC(Airco)>RWDC(CD_player)

Seller uses:

Good > Fairly good> Standard > Meager > None

Example experimental results



49

Multi-issue negotiation solutions: case of *linear* utility functions

- Fuzzy logic
  – [Faratin et. al, 2002]
- Kernel density estimation
  – [Coehoorn & Jennings, 2004]
- Bayesian methods
  – [Sycara '98] [Raz & Kraus '06]
- Based on ordinal ranking heuristics
  – [Jonker & Robu, 2004]
- Constraint based methods
  – [Ehtamo & Hamalaianen, '01]

50

Multi-issue negotiation: solutions for *non-linear* utility functions

- Simulated annealing
  - [Klein et. al. 2002]
- Evolutionary computing
  - [Lin, '04] [Gerding, Bragt, La Poutre, '02]
- Utility graphs
  - [Robu, Somefun, La Poutre '05]
- Heuristics based ISO-utility curves
  - [Somefun, Gerding, La Poutre, '04], [Lai, Sycara & Li, '06]
- Auction-based methods
  - [Ito, Klein & Hattori, 2006]

51

Negotiation over multiple continuous issues using ISO-utility lines

References:

[Gerding, Somefun, La Poutré, '04]

[Lai, Li & Sycara, '06]

52

Negotiation Strategies for Continuous Items

- Dealing with multiple items
  - Continuous values for each item
  - Trade-offs between items
- Find negotiation strategy
  - 2 items, say i_1 and i_2
  - Utility functions for agents A and B:
    $u_A(i_1, i_2)$ and $u_B(i_1, i_2)$

53

Negotiation Strategies for Continuous Items

- Competitive aspect: `tug-of-war'
  - *Concession Strategy on bids* (i_1, i_2)
  - *Determines the utility of the bids* $u_A(i_1, i_2)$
  - *See above for 1-item concession strategies* on $u_A(i_1, i_2)$
    - So, now for *utility values* of 2 items*:* $u_A(i_1, i_2)$

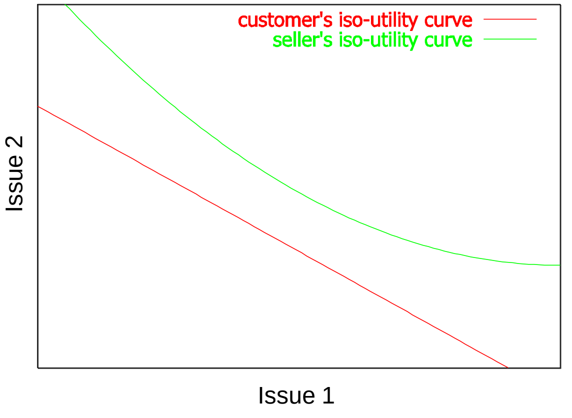
  Cooperative aspect: multi-issue trade-off
  - Find Pareto-efficient outcomes by making trade-offs in (i_1, i_2)
  - Beneficial for both agents (win-win)
  - *Pareto-search Strategy on* $u_A(i_1, i_2)$ and $u_B(i_1, i_2)$

54

Example

Iso-utility curves for given bundle

All points on one curve same utility: $u_A(i_1, i_2)$ = *constant* = *c*

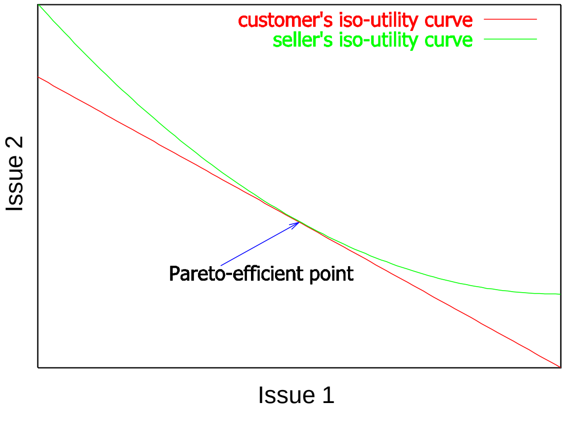


Agents:
A: customer
B: seller

Example

Concession strategies: iso-utility curves move (they approach)

- "Constant" c is changed (in $u_A(i_1, i_2) = c$)

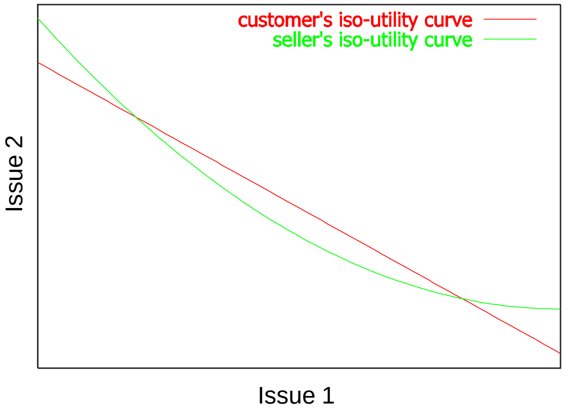


Agents:
A: customer
B: seller

Example

Concession strategies: iso-utility curves move (they approach)

- "Constant" c is changed (in $u_A(i_1, i_2) = c$)



Agents:
A: customer
B: seller

57

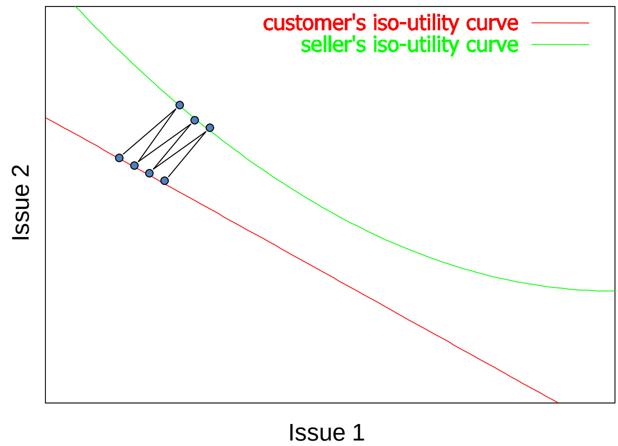
Pareto-search Strategy

- Find Pareto-efficient point without *knowing* opponent's curve
  - Approach the Pareto-efficient solutions during concession

- Solution: "Orthogonal Strategy"
  - For a given bid B_{opp} of the opponent, determine your counter bid B_{new} on your own iso-utility curve that is closest to B_{opp}
    - Can be enhanced with Derivative Follower (not here)

58

Orthogonal Strategy



Negotiation Strategies for Continuous Items

- Thus:
  Negotiation strategy is decomposed into two parts
  - Pareto-search strategies
    - Orthogonal search
  - Concession strategies
    - As before
  - Intertwined / “active in parallel”

References: Books

- Howard Raiffa – The art and science of negotiation, Harvard Univ. Press, 1982
- M.J. Osborne, A. Rubinstein – Bargaining and Markets, Academic Press, 1990.
- J.S. Rosenschein, G. Zlotkin – Rules of encounter, MIT Press, 1994.
- Roger B. Myerson – Game Theory: Analysis of Conflict, Harvard University Press, 1997.
- Sarit Kraus – Strategic Negotiation in Multi-Agent Environments, MIT Press, 2001.

References (1)

- P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. Int. Journal of Robotics and Autonomous Systems, 24(3-4):159-182, 1998.
- A. Rubinstein. A bargaining model with incomplete information about time preferences. Econometrica, 53, January 1985.
- T. Sandholm and N. Vulkan. Bargaining with deadlines. In AAAI-99, Orlando, FL, 1999
- Winoto P., McCalla G. and Vassileva J. (2004) Non-Monotonic-Offers Bargaining Protocol, Proc of Autonomous Agents and Multi-Agent Systems (AAMAS'04), New York 2004
- Yann Chevaleyre, Ulle Endriss, Jerome Lang, and Nicolas Maudet. Negotiating over small bundles of resources. In Proc. of the 4th AAMAS, Utrecht, 2005.
- Somefun, E. Gerding, S.M. Bohté, and J.A. La Poutré, Automated Negotiation and Bundling of Information Goods, In: Agent Mediated Electronic Commerce V (AMEC-V), Melbourne, Australia, Springer LNAI, 3048, pages 1 - 17, Springer Verlag, 2004.
- Catholijn Jonker and Valentin Robu. Automated multi-attribute negotiation with efficient use of incomplete preference information. In 3rd Int. Conf. on Autonomous Agents & Multi Agent Systems (AAMAS), New York, pages 1056-1063, 2004

References (2)

- Guoming Lai, Katia Sycara, and Cuihong Li. A decentralized model for multi-attribute negotiations with incomplete information and general utility functions. In Proc. of RRS'06, Hakodate, Japan, 2006.
- R. M. Coehoorn N. R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. In Proc. 6th Int Conf. on E-Commerce, Delft, 2004.
- S. van der Putten, V. Robu, H. La Poutré, A. Jorritsma, and M. Gal, Automating Supply Chain Negotiations using Autonomous Agents: a Case Study in Transportation Logistics, Proceedings AAMAS 2006, Hakodate, Japan, ACM Press, 2006.
- V. Robu and J. A. La Poutré. Retrieving the structure of utility graphs used in multi-item negotiation through collaborative filtering of aggregate buyer preferences. In Proc. of RRS'06, Hakodate, Japan
- V. Robu, D.J.A. Somefun, and J. A. La Poutré. Modeling complex multi-issue negotiations using utility graphs. In Proc. of the 4th Int. Conf. on Autonomous Agents & Multi Agent Systems (AAMAS), Utrecht, 2005,
- D.J.A. Somefun, E.H. Gerding, and J.A. La Poutré. Efficient methods for automated multi-issue negotiation: Negotiating over a two-part tariff. International Journal of Intelligent Systems, 21(1):99-119, 2006

Complex Negotiations for Intractable Problems

S.S. Fatima

Department of Computer Science

Loughborough University, UK

S.S.Fatima@lboro.ac.uk

Overview

- The overall objective: *automate negotiation* – develop software agents that negotiate on our behalf
- Focus on scenarios with:
  - multiple issues
  - time constraints (deadline, discount factor)
  - incomplete information
  - competitive agents (game theoretic)
- These features make negotiation complex – hard to automate
- Hence we will look at some computationally feasible methods for overcoming this complexity

Outline

1. Single issue negotiation
2. Methods for multi-issue negotiation
3. Complexity of negotiating multiple issues
4. Devising computationally feasible solutions
5. Summary

Negotiation

- A means for agents to communicate and compromise to reach mutually beneficial agreements; the agents have different preferences over the possible agreements and so must find one that is acceptable to all negotiators

  Agent 1: O1 > O2 > O3　　**Agent 2**: O2 > O3 > O1

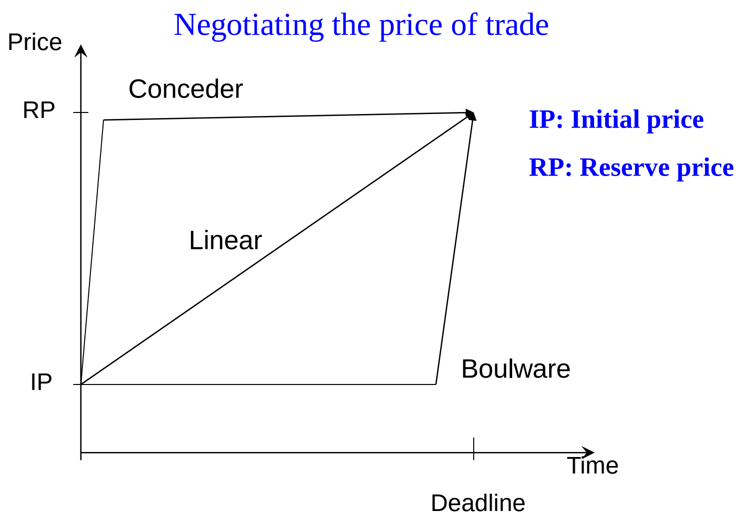
- A number of applications require agents to negotiate:
  - data allocation in information servers
  - task allocation
  - resource allocation
  - e-commerce

Negotiation protocol and strategy

- Protocol: Gives the rules of encounter
- Strategy: An agent's strategy is a specification of the sequence of actions (usually offers or responses) the agent plans to make during negotiation – depends on the protocol
- Optimal strategy: An agent's optimal strategy is one that maximizes its *individual utility*

Finding the optimal strategy is a key problem for the negotiators

Trading agents scenario - Buyer strategies



Time dependent strategies

$$\mathrm{Price}(t) = IP^{a} + F^{a}(t) \times (RP^{a} - IP^{a}) \qquad \textit{For the buyer}$$

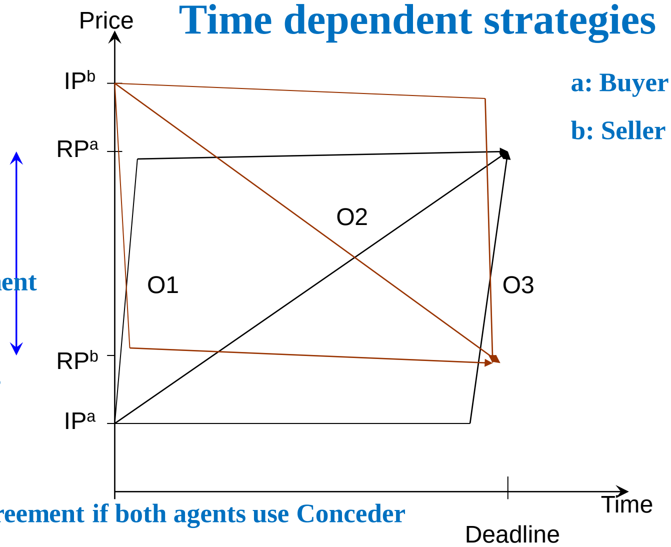
$$\mathrm{Price}(t) = RP^{b} + (1 - F^{b}(t)) \times (IP^{b} - RP^{b}) \qquad \textit{For the seller}$$

$F^a(t)$ and $F^b(t)$ lie between 0 and 1

For ($t = 0$): $F^a(t) = 0$ **and** $F^b(t) = 0$

For ($t = Deadline$): $F^a(t) = 1$ **and** $F^b(t) = 1$

Time dependent strategies



a: Buyer

b: Seller

Zone of Agreement OR Surplus

O1: Agreement if both agents use Conceder

O2: Agreement if both agents use Linear

O3: Agreement if both agents use Boulware

Single issue negotiation

- Agents *a* and *b* negotiate over a pie of size 1
- Deadline: *n* and Discount factor: δ
- Utility: $U(x, t) = \begin{cases} x\,\delta^{t-1} & \text{if } t \leq n \\ 0 & \text{otherwise} \end{cases}$
- The agents negotiate using Rubinstein's *alternating offer's protocol*

Alternating offers protocol

Time	Offer
1	a □ b (accept/reject)
2	b □ a (accept/reject)
-	
-	
n	

Optimal Offers

How much should an agent offer in the first time period?

Let n=1 and *a* be the first mover

Agent *a*'s optimal offer:

Propose to keep the whole pie; agent *b* will accept this

Optimal strategies for n = 2

$\delta = 1/4$ first mover: *a*

Offer: (x, y) x: *a*'s share; y: *b*'s share

Optimal offers obtained using backward induction

Agreement

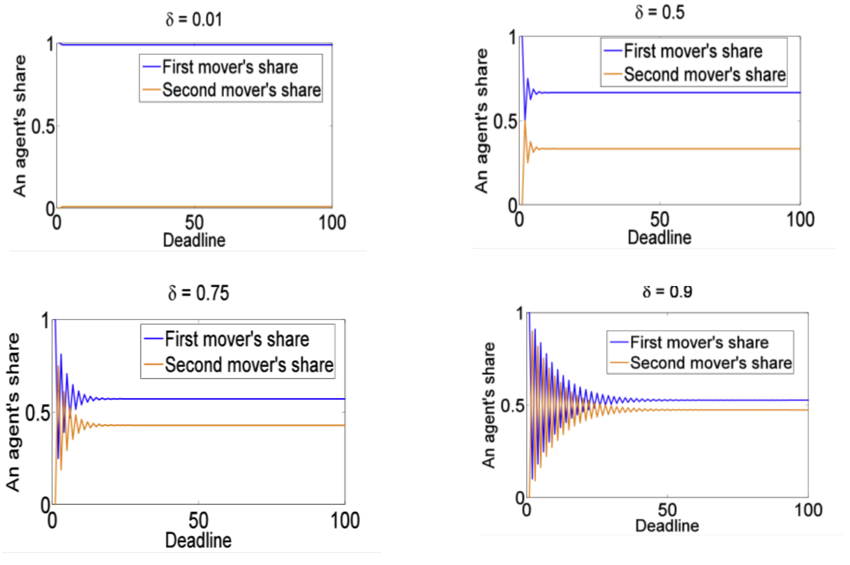
Time	Size of pie	Offering agent	Offer
1	1	a → b	(3/4, 1/4)
2	1/4	b → a	(0, 1/4)

The offer (3/4, 1/4) forms a Nash equilibrium

Effect of discount factor and deadline on the equilibrium outcome

- What happens to first mover's share as δ increases?
- What happens to second mover's share as δ increases?
- As deadline increases, what happens to first mover's share?
- Likewise for second mover?

Effect of δ and deadline on the agents' shares



Multiple issues

- Set of issues: S = {1, 2, …, m}. Each issue is a pie of size 1
- The issues are divisible
- Deadline: n (for all the issues)
- Discount factor: δ_c for issue c
- Utility: $U(\mathbf{x}, t) = \sum_c U(\mathbf{x}_c, t)$

Multi-issue procedures

- Package deal procedure: The issues are bundled and discussed together as a package
- Simultaneous procedure: The issues are negotiated in parallel but independently of each other
- Sequential procedure: The issues are negotiated sequentially one after another

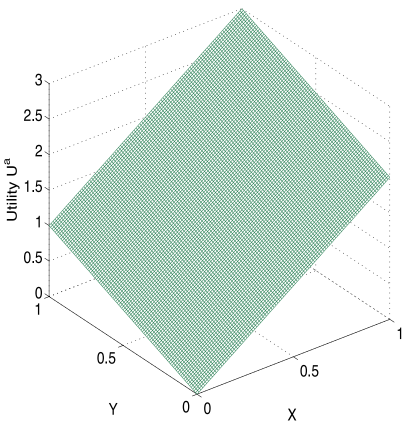
Package deal procedure

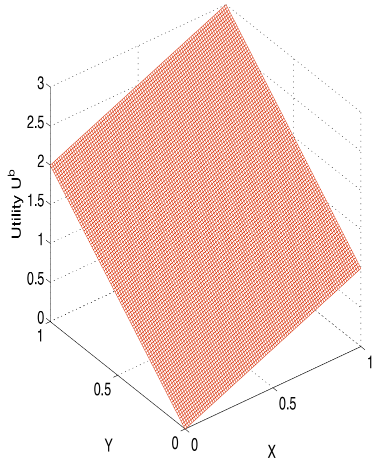
- Issues negotiated using alternating offer's protocol
- An offer specifies a division for each of the *m* issue
- The agents are allowed to accept/reject a complete offer
- The agents may have different preferences over the issues
- The agents can make tradeoffs across the issues to maximize their utility – this leads to Pareto optimal outcome

Utility for two issues

$U^a = 2X + Y$

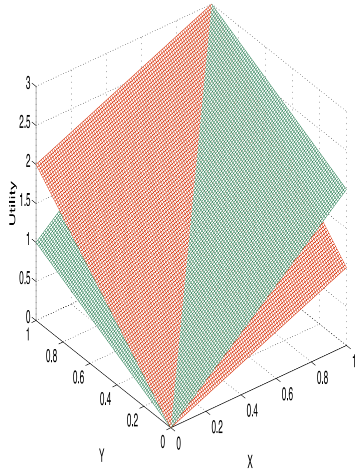
$U^b = X + 2Y$

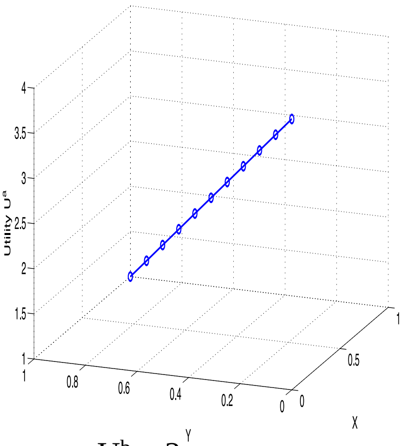




Making tradeoffs

What is *a*'s utility for $U^b = 2$





$U^b = 2$

Example for two issues

DEADLINE: n = 2

DISCOUNT FACTORS: $\delta_1 = \delta_2 = 1/2$

UTILITIES: $U^a = x_1 + 2x_2$; $U^b = 2y_1 + y_2$

Agreement

Time	Size of pie	Offering agent	Package Offer
1	1, 1	a → b	[(1/4, 3/4); (1, 0)] OR [(3/4, 1/4); (0, 1)]
2	1/2, 1/2	b → a	[(0, 1/2); (0, 1/2)] $U^b = 1.5$

The outcome is not symmetric

Nash equilibrium strategies

For $t = n$

The offering agent takes 100 percent of all the issues

The receiving agent accepts

For $t < n$ (for agent a):

OFFER $[x, y]$	RECEIVE $[x, y]$
s.t. $U^b(y, t) = EQ_{UB}(t+1)$ If more then one such $[x, y]$ perform trade-offs across issues to find best offer	If $U^a(x, t) \geq EQ_{UA}(t+1)$ ACCEPT else REJECT

$EQ_{UA}(t+1)$ is a's equilibrium utility for $t+1$

$EQ_{UB}(t+1)$ is b's equilibrium utility for $t+1$

Making trade-offs – divisible issues

Agent *a*'s trade-off problem at time *t*:

TR: Find a package [x, y] to

Maximize $\sum_{c=1}^{m} k^{a}_{c}\, x_c$

Subject to $\sum_{c=1}^{m} k^{b}_{c}\, y_c \geq EQ_{UB}(t+1)$ $\quad 0 \leq x_c \leq 1;\ 0 \leq y_c \leq 1$

This is the fractional knapsack problem

Fractional knapsack problem

- A thief robbing a store finds *m* items
- The i^{th} item is worth v_i dollars and weighs w_i pounds
- He wants to take as valuable a load as possible but he can carry at most *W* pounds in the knapsack
- He can take fractions of items
- Which items should he take?

Maximize $\sum_{c=1}^{m} v_c\, x_c$

subject to $\sum_{c=1}^{m} w_c\, x_c \leq W$ $\quad 0 \leq x_c \leq 1$

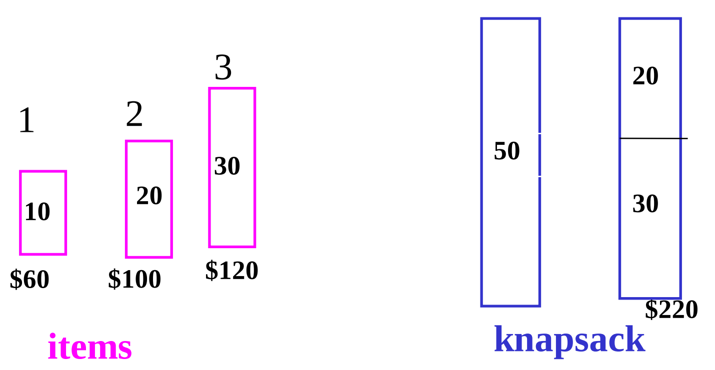
Greedy strategy

- A greedy algorithm obtains an optimal solution to a problem by making a sequence of choices
- For each decision point in the algorithm, the choice that seems best at the moment is chosen (future consequences not considered)
- This heuristic strategy does not always produce an optimal solution, but sometimes it does

Greedy strategy 1

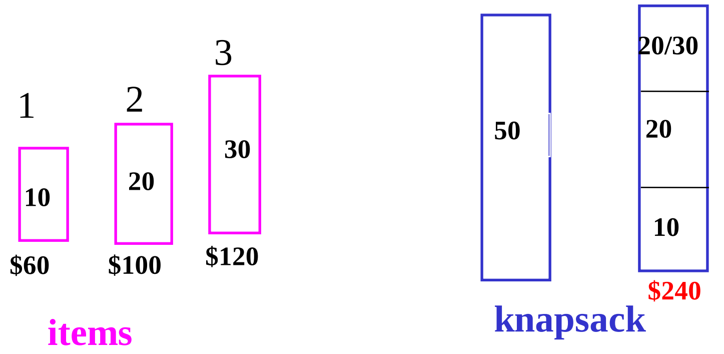
Select the items in the order of greatest value

Example:



Greedy strategy 2

Select the items in the order of greatest value per unit weight – this is the optimal strategy



Making trade-offs – divisible issues

Agent *a*'s perspective (time *t*)

Agent *a* considers the *m* issues in the increasing order of k^a/k^b and assigns to *b* the maximum possible share for each of them until *b*'s cumulative utility equals $EQ_{UB}(t+1)$

Equilibrium strategies

For $t = n$
The offering agent takes 100 percent of all the issues
The receiving agent accepts

For $t < n$ (for agent a)

OFFER [x, y]	RECEIVE [x, y]
s.t. $U^b(y, t) = EQ_{UB}(t+1)$ If more then one such [x, y] perform trade-offs across issues to find best offer	If $U^a(x, t) \geq EQ_{UA}(t+1)$ ACCEPT else REJECT

Equilibrium solution

- An agreement on all the *m* issues occurs in the first time period
- Time to compute the equilibrium offer for the first time period is $O(mn)$
- The equilibrium solution is Pareto-optimal (an outcome is Pareto optimal if it is impossible to improve the utility of both agents simultaneously)
- The equilibrium solution is not unique, it is not symmetric

Making trade-offs – indivisible issues

Agent *a*'s trade-off problem at time t is to find a package [x, y] that

$$\textit{Maximize} \quad \sum_{c=1}^{m} k_c^a x_c$$

$$S.t. \quad \sum_{c=1}^{m} k_c^b y_c \geq EQ_{UB}(+1) \quad x_c : 0 \text{ or } 1; \quad y_c : 0 \text{ or } 1$$

For indivisible issues, this is the integer knapsack problem

Dynamic programming

- Integer knapsack problem cannot be solved using the greedy strategy (previous example)
- Integer knapsack problem has optimal sub-structure property

$$A(i,c) = \begin{cases} A(i-1,c) & \text{if } w_i > c \\ \max\left(A(i-1,c), v_i + A(i-1,c-w_i)\right) & \text{if } w_i \leq c \end{cases}$$

Complexity and approximations

- IKP is NP-hard
- The problem of finding the *optimal offers* for *indivisible issues* is also NP hard
- Hence the need to devise approximation algorithms

Approximation algorithms are computationally efficient

They give solutions that are close to the optimum

Performance guarantees

Quality of approximation is measured in terms of:

- Absolute performance guarantee
- Relative performance guarantee

Absolute performance guarantee

- Given an optimization problem P, for any instance x and any approximate solution y of x, the absolute error of y with respect to x is defined as:

  $D(x,y) = |v^*(x) - v(x,y)|$ where $v^*(x)$ is the optimal solution

- Given an optimization problem P, and an approximation algorithm A for P, we say that A is an absolute approximation algorithm if there exists a constant k such that, for every instance x of P, $D(x,A(x)) \leq k$

Relative performance

- The quality of an approximation algorithm is measured in terms of the ratio of the approximate and optimal solutions

- Given an optimization problem P, for any instance x and any approximate solution y of x, the performance ratio of y with respect to x is defined as

  $R(x,y) = max(v(x,y)/v^*(x), v^*(x)/v(x,y))$

- The *ratio* $R(x,y)$ is at least one

Relative performance guarantee

Given an optimization problem *P*, and an approximation algorithm *A* for *P*, we say that *A* is an *r*-approximation algorithm for P if given any input instance *x* of *P*, $R(x,A(x)) \leq r$

Polynomial time approximation schemes

An algorithm A is said to be a polynomial time approximation scheme (PTAS) for a problem P if, for any instance *x* of P and any rational value $r > 1$, A when applied to input (x, r) returns an *r*-approximate solution of *x* in time polynomial in $|x|$

Fully polynomial time approximation schemes

An algorithm A is said to be a fully polynomial time approximation scheme (FPTAS) for a problem P if, for any instance *x* of P and any rational value $r > 1$, A when applied to input (x,r) returns an *r*-approximate solution of *x* in time polynomial both in $|x|$ and in $1/r$

Approximate solution for knapsack problem

There is an FPTAS for the integer knapsack problem

Time complexity: $O(m/\varepsilon^2)$

z: approximate solution   z*: optimal solution

Relative error of approximation

$$(z - z^*) / z^* \leq \varepsilon$$

Equilibrium for indivisible issues

- There is an ε-approximate Nash equilibrium
- A strategy profile is said to be an ε-Nash equilibrium if it is not possible for any player to gain more than ε in payoff by unilaterally deviating from his strategy
- Every Nash equilibrium is equivalent to a ε-equilibrium where $\varepsilon = 0$.

Approximate equilibrium solution

- An agreement on all the *m* issues occurs in the first time period
- The equilibrium solution is Pareto-optimal (in approximation)
- The equilibrium solution is not unique, it is not symmetric
- Time to compute the equilibrium offer for the first time period is $O(nm/\varepsilon^2)$

Key points

- Single issue:
  - Time to compute equilibrium is O(n)
  - The equilibrium is not unique, it is not symmetric
- Multiple divisible issues: (exact solution)
  - Time to compute equilibrium for $t=1$ is O(mn)
  - The equilibrium is Pareto optimal, it is not unique, it is not symmetric
- Multiple indivisible issues: (approx. solution)
  - There is an FPTAS to compute approximate equilibrium
  - The equilibrium is Pareto optimal (in approximation), it is not unique, it is not symmetric

Simultaneous procedure

- Equilibrium for each individual issue is the same as that for single issue negotiation
- Computationally easy: equilibrium can be computed in polynomial time – O(mn)
- Outcome may not be Pareto optimal (no tradeoffs)
- Additional question arises: choice of first mover for each individual issue

Sequential procedure

- Equilibrium for each individual issue is the same as that for single issue negotiation
- Computationally easy: equilibrium can be computed in polynomial time – O(mn)
- Outcome may not be Pareto optimal (no tradeoffs)
- Is slower than the PDP and the simultaneous procedures since only one issue is negotiated at a time (*m* time periods)
- The equilibrium outcome depends on the order in which the issues are negotiated – this ordering is called agenda

Negotiation agenda

- The agents may have different preferences over the possible agendas
- The agents must therefore find a way of settling the agenda before they negotiate the issues
- Possible ways of deciding the agenda:
  - Exogeneously: The agenda is decided before the agents negotiate the issues, the agenda is part of protocol; the issues are then negotiated as per the agenda
  - Endogeneously: The agents decide what issue they will negotiate next during the process of negotiation – this is useful when set of issues is not known in advance

Sequential procedure

Some additional questions arise:

- Choice of first mover for each issue
- Choice of the order (agenda) in which the issues will be negotiated
- Choice of deciding the agenda before or during negotiation

Summary - I

For the PDP:

- Multi-issue negotiation for divisible issues is computationally easy
- Multi-issue negotiation for indivisible issues is NP-hard
- For indivisible issues, there is an ε-approximate Nash equilibrium (computable in polynomial time) with
  - only one copy of each issue
- The outcome for the PDP is Pareto optimal, it is not unique, it is not symmetric
- An agreement on all the issues occurs in the first time period

Summary - II

For the simultaneous procedure:

- For divisible and indivisible issues the equilibrium is computatable in polynomial time
- An agreement on all the issues occurs in the first time period
- The outcome is not Pareto optimal, it is not unique, it is not symmetric
- Additional issue: deciding the first mover for each individual issue

For the sequential procedure:

- For divisible and indivisible issues the equilibrium is computatable in polynomial time
- An agreement on issues *i* occurs in time period *i*
- The outcome is not Pareto optimal, it is not unique, it is not symmetric
- Negotiation takes longer than the PDP and the simultaneous procedures
- Additional factors: deciding the first mover for each individual issue, and deciding the negotiation agenda

References

1. O.H. Ibarra and C.E. Kim, Fast approximation algorithms for the knapsack and sum of subsets problems, *Jnl of ACM* 22: 463-468, 1975.
2. H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack Problems, Springer, 2004
3. A. Marchetti-Spaccamela and C. Vercellis, Stochastic online knapsack problems, *Mathematical Programming*, 68: 73-104, 1995.
4. M.J. Osborne and A. Rubinstein, A course in game theory, The MIT press, 1994.
5. J.S. Rosenschein and G. Zlotkin, Rules of encounter, The MIT press, 1994.
6. S. Kraus, Strategic negotiation in multiagent environments, The MIT press, 2001.
7. S.S. Fatima, M. Wooldridge, and N.R. Jennings, Multi-issue negotiation with deadlines, *Jnl of AI Research*, 21: 381-471, 2006.
8. S.S. Fatima, M. Wooldridge, and N.R. Jennings, Approximate and online multi-issue negotiation, In *Proceedings of AAMAS-07*, 947-954, 2007.
9. R. Keeney and H. Riffa, Decisions with multiple objectives: Preferences and value trade-offs, John Wiley, 1976.
10. M.J. Osborne and A. Rubinstein, Bargaining and Markets, Academic Press, 1990.

H. Raiffa, The Art and Science of Negotiation, Harvard University Press, 1982.

Complex Automated Negotiation and its Application to Collective Collaboration System

- **Takayuki Ito**
- ito.takayuki@nitech.ac.jp
  - School of Techno-Business Administration, **Nagoya Institute of Technology**
  - PREST "**Sakigake**" **Japan Science and Technology Agency(JST)**
  - Policy Alternatives Research Institute, **University of Tokyo**

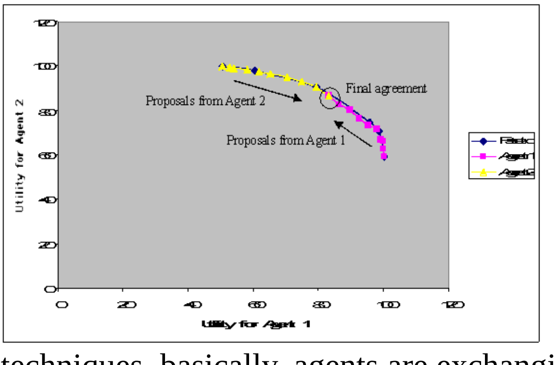
Automated Negotiation

- Work to date on negotiation protocols has addressed mainly **simple** contracts
  - A **single** issue (e.g. price)
  - OR several **independent** issues
    - I want the bike for as cheap as possible
    - I also want the book for as cheap as possible
    - .. etc …
- Basically simple utility functions assume "independence" among issue
  - Multi-attributed Utility Theory
    - Keeney and Raiffa, "Decisions with Multiple Objective". They assume independency among issues.

Complex contracts

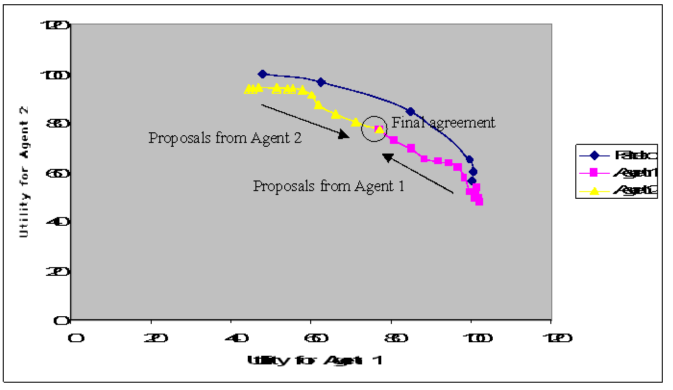
- Complex contracts include **dependencies** among issues - the choice for one issue impacts the value of the choices made for other issues
  - The value of this stereo depends on the speakers I buy with it
- Complex contracts are ubiquitous
  - Products composed from multiple parts
  - Operations composed from multiple missions
  - Supply chains with multiple participants
  - …

Standard techniques (e.g. proposal exchange) work **fine** for **simple** contracts



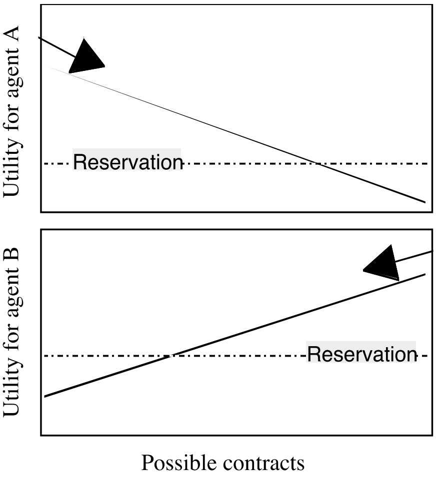
In standard techniques, basically, agents are exchanging their proposals so that they reduce the distance between proposals while minimizing their utility decrement.

Proposal exchange - complex contracts



Simple exchanging proposals do not work well

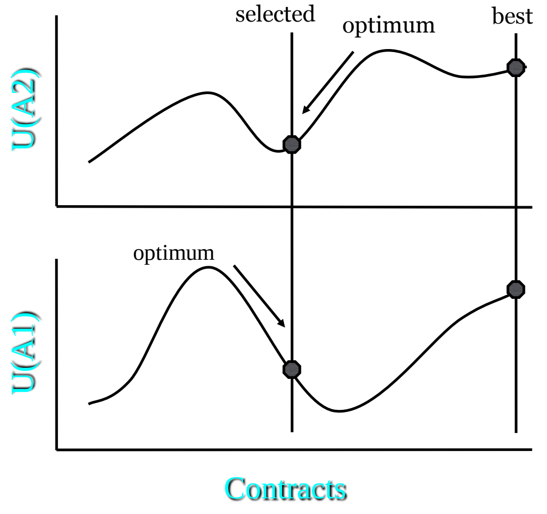
Simple contracts have linear utilities



In general, utility is represented by a simple weighted *sum* of utilities for each individual issue in agreement

Conceding as slowly as possible towards the middle produces optimal outcomes

Complex contracts have “bumpy” utilities

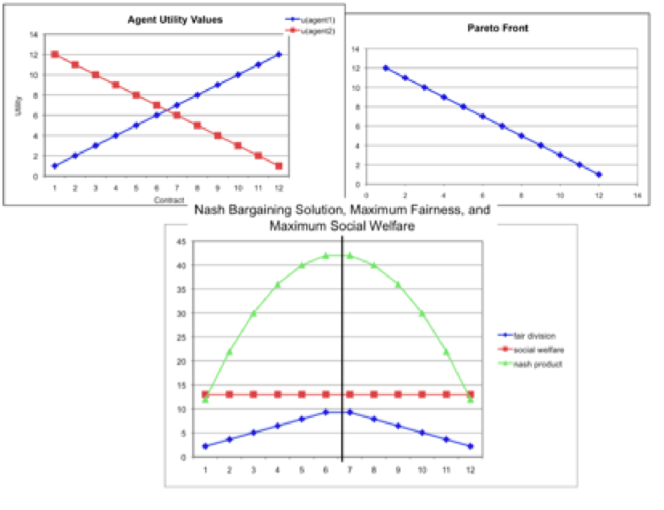


Utility(agreement) is a complex (not just additive) function of the selections for each issue

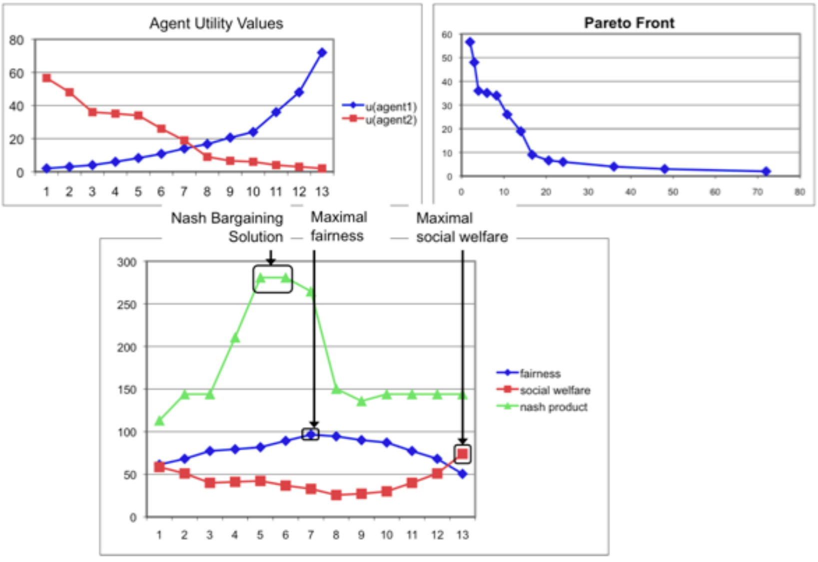
-> multi-optima utility functions

Conceding towards the middle can easily result in lose-lose outcomes

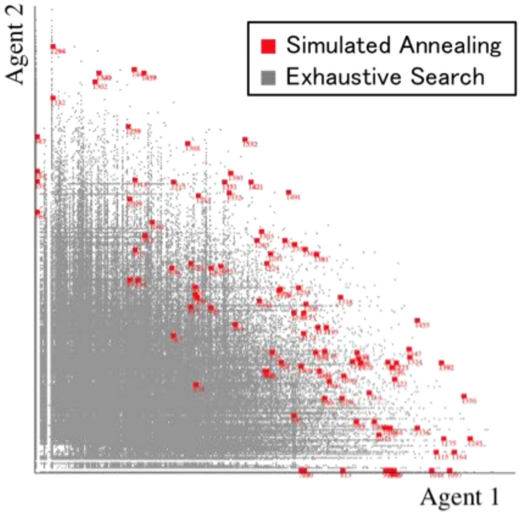
Nash Bargaining solution



Easy to achieve the Nash bargaining solution in the ***linear*** utility case



Difficult to achieve the Nash bargaining solution in the ***non-linear*** utility case



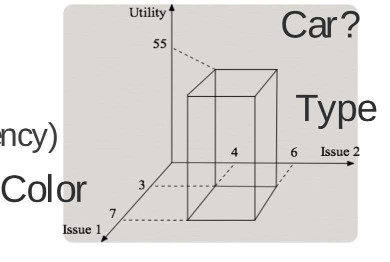
Sparseness makes difficult to find the Nash solution

complex v.s. linear

- Linear
  - Simple exchanging protocol work well
  - It is easy to achieve the Nash bargaining solution
- Complex
  - Simple exchanging protocol can not achieve better agreements
  - Sometime it fails to achieve the Nash bargaining solution

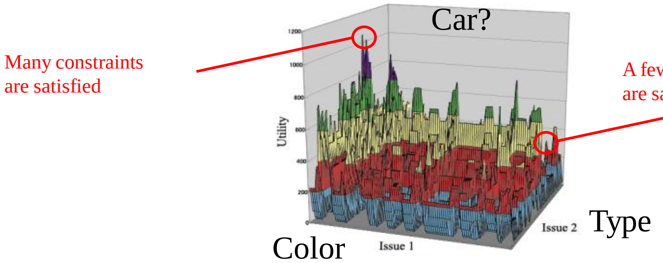
Automated Negotiation Mechanism based on Complex Non-linear Utilities [Ito, IJCAI07]

- Non-linear utility space model
  - m issues with the domain of integers [0, X]
    - Issues are common for agents.
  - A contract is a vector of issue values $s = (s_1, \ldots, s_m)$.
- Agent's utility function
  - The function is represented in terms of constraints.
  - A constraint represents an acceptable region (interdependency) and its value (utility).



Negotiation with Non-linear Utilities

- The utility is a summation of satisfied contracts' values
  - Bumpy non-linear utility space



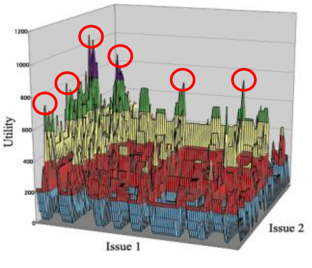
Existing protocols assuming linear utility functions are not effective.

How to obtain a solution with high social welfare for non-linear utility function ?

Takayuki Ito (ito.takayuki@nitech.ac.jp)

Bidding-based Negotiation Protocol

- Sampling -> Bid-generation -> Winner determination
  - An agent submits bids to an mediator for high individual utility.

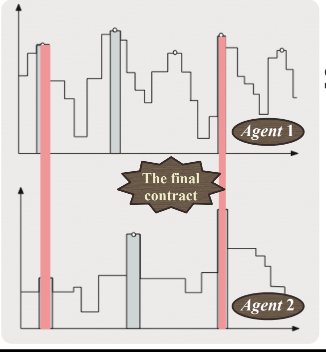


- An agent samples its utility space to find high-utility region.
- Trade-off between high-utility and the limit of # of samples.

Takayuki Ito (ito.takayuki@nitech.ac.jp)

Winner Determination

- The mediator identifies the combinations of bids as the final contract.
  - The final contract is a consistent bids with the highest social welfare.
  - Only one bid from each agent is included.



1. Find mutually consistent bids

Specifying overlapping contract region

2. Select the best contracts

Comparing the summed bid values

Experiments

- Setting
  - Constraints satisfying many issues could have larger weights.
    - The maximum value for a constraint: 100 x # of issues *e.g.*, the possible value for a binary constraint is 200.
  - Agents have the same issues and domain for each issue value.
    - Domain for issue value is [0,9]
  - Approximate search-based winner determination
    - The final contract is searched by the simulated annealing.

Experiments

- **Linear** utility function case
  - Comparison between the optimal result and the result of "Independent Hill Climbing protocol"

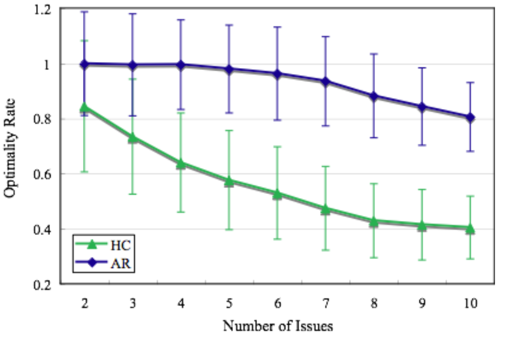
Issues	1	2	3	4	5	6	7	8	9	10
HC	0.973	0.991	0.998	0.989	0.986	0.987	0.986	0.996	0.988	0.991

Optimality with linear utility function (4 agents)

- The simple Independent HC protocol can produce nearly optimal results even for a large space.
  - The mediator can find the best value for issue 1, then issue 2, then issue 3, ...

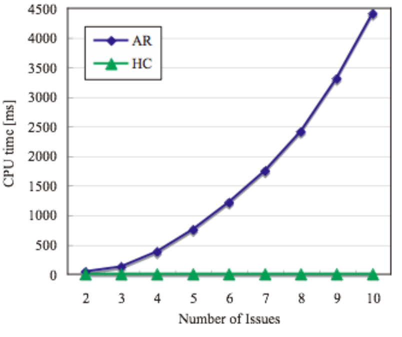
Experiments

- Independent Hill Climbing / Bidding-based method for non-linear utility function
  - HC mediator tends to converge to a local optimum.
  - AR mediator has more chances to find better contract because agents can generate bids covering multiple optima.



Experiments

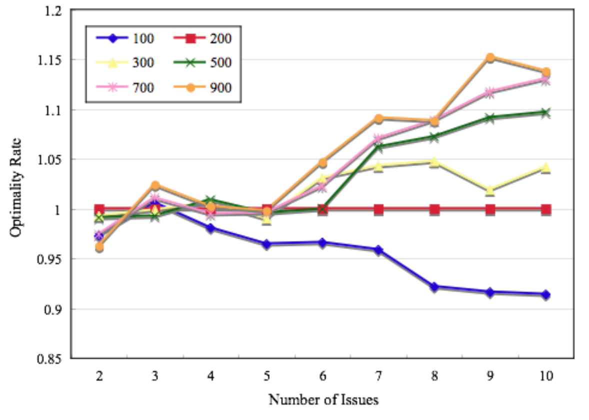
- Independent Hill Climbing / Bidding-based method for non-linear utility function
  - HC mediator can quickly obtain the final contract.
  - AR mediator takes more time, but the final contract is calculated within a practical time.



Scalability

- The impact of the scaling-up of the problem space
  - 90%+ optimality for up to 8 issues

Optimality v.s. Sampling Rate



Sampling Rates v.s. Agreement

- **The Failure rate, % of negotiations that do not lead to an agreement, is higher when there are more sampling points**

- Reason:
  - –When there are many sampling points, each agent has a better chance of finding good local optima in its utility space.
  - –However, the num. of bids is limited for computation time.
  - –This increases a risk of not finding an overlap between the bids

Table 3: Failure rate [%]

	100	**200**	300	500	700	900
2	1	0	0	2	0	0
3	0	0	0	0	0	0
4	2	1	0	1	2	3
5	4	5	6	4	10	9
6	4	11	5	13	13	19
7	5	7	10	20	14	20
8	9	8	10	13	27	19
9	6	15	16	17	30	31
10	7	11	18	26	19	31

Discussion

- The number of bids is ...
- The winner determination computation grows exponentially as (# of bids per agent)$^{\text{\# of agents}}$
  - If we use an exhaustive search method (with branch cutting),
    the problem size is limited to the small one.
- The trade-off between the computation time and the optimality

# of issues	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of bids	54	200	461	758	1074	1341	1636	1746	1972	2086	2238	2326	2491	2648

Extensions

- Grouping Issues

  Hiromitsu Hattori, Mark Klein, and Takayuki Ito, "Using Iterative Narrowing to Enable Multi-Party Negotiations with Multiple Interdependent Issues," in Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007), 2007, pp. 1043-1045.

- Adjusting Thresholds

Fujita, Ito, Klein, "The Effect of Grouping Issues in Multiple Interdependent Issues Negotiation based on Cone-Constraints", ACAN 2010.

Katsuhide Fujita, Takayuki Ito, Mark Klein, "A Preliminary result on a representative-based multi-round protocol for multi-issue negotiations", The Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2008), pp. 1573-1576,2008.

Other approaches

- Secure Centralized Simulated Annealing
  - Katsuhide Fujita, Takayuki Ito, and Mark Klein, "A Secure and Fair Protocol that Addresses Weaknesses of the Nash Bargaining Solution in Nonlinear Negotiation", Group Decision and Negotiation Journal, 2010 (to appear).Katsuhide Fujita, Takayuki Ito,and Mark Klein, "Secure and Efficient Protocols for Multiple Interdependent Issues Negotiation," Journal of Intelligent and Fuzzy Systems, 2009
- Graph Representation of Constraints Block
  - Marsa -Maestre, I., Lopez - Carmona, M. A., Velasco, J. R., Ito, T., Fujita, K., and Klein, M.: Balancing Utility and Deal Probability for Negotiations in Highly Nonlinear Utility Spaces, in IJCAI-09, pp.214-219(2009)

Iterative narrowing possible contract space

Takayuki Ito[#*]

Collaborative work with
Mark Klain*, Hiromitsu Hattori+, and Katsuhide Fujita#

[#]Nagoya Institute of Technology, JAPAN

[*]Sloan School of Management,
Massachusetts Institute of Technology, USA

+Kyoto University, JAPAN

Problem

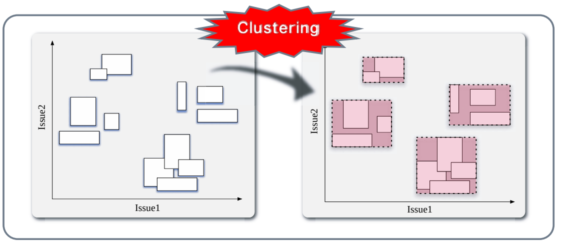
- The bid limit for each agent becomes severe as the number of agents are increased.
  - The winner determination computation grows exponentially as (# of bids per agent) $^{\text{\# of agents}}$, so...
  - I set the limit of the # of bids per agents to $6400000^{1/2}$
    - *e.g.* the bid limit is 5 for 10 agents case.
- The region covered by a high-utility bid tends to be small.
  - ➡The failure rate is getting high!

Approach

- Iterative narrowing of the possible contract space.
  - Multi-round bidding and deal identification to gradually make the contract space smaller.
    - Cluster Bidding
    - Widest-Constraint Bidding
    - Peak Bidding

Cluster Bidding (1)

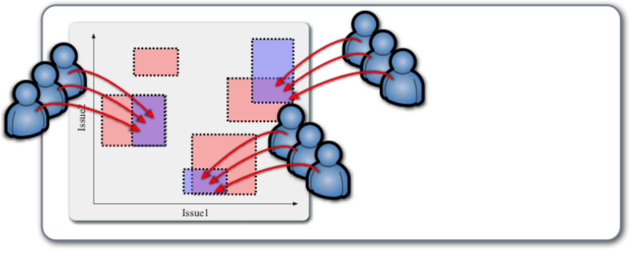
- Extraction of all the promising regions in each utility space
  - The region of a cluster bid consists of the regions of several bids.



- The cluster-bids cover all promising regions while cutting-out “no-interest” regions.
  - The true topology of the utility function is still NOT unveiled.

Cluster Bidding (2)

- Agents can catch all promising regions for all of them.

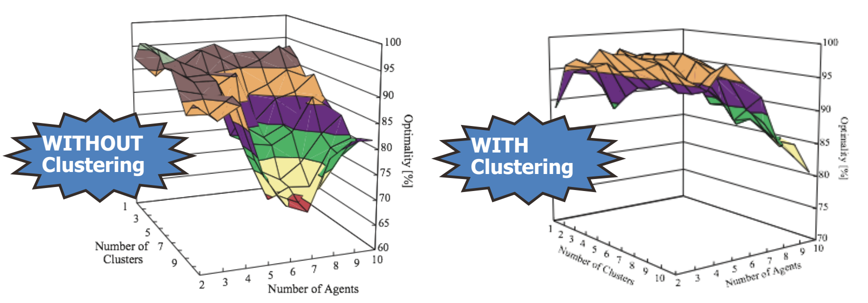


- The total size of overlapping regions, which are available regions for the next round, could become drastically small.
  - The overlaps are guaranteed to be disjoint from each other.
  - ➡ In the next round, an agent can bid for one overlap independently of other overlaps.

Experiments: setting

- Constraints satisfying many issues could have larger weights.
  - The maximum value for a constraint: 100 x # of issues *e.g.*, the possible value for a binary constraint is 200.
- Agents have the same issues and domain for each issue value.
  - Domain for issue value is [0,9]
- Utility functions consists of 50 10-dimensional constraints.
  - Constraints are grouped into clusters
    - 10 clusters per each agent's utility function
    - 5 constraints per cluster

Experiments: optimality



- The optimality WITHOUT the clustering drops off rapidly as the agent utility becomes more correlated.
  - With many overlaps, agents must bid on a few of overlaps.
- The optimality WITH the clustering hovers about 90% except for the toughest problems.