

Automated Bargaining

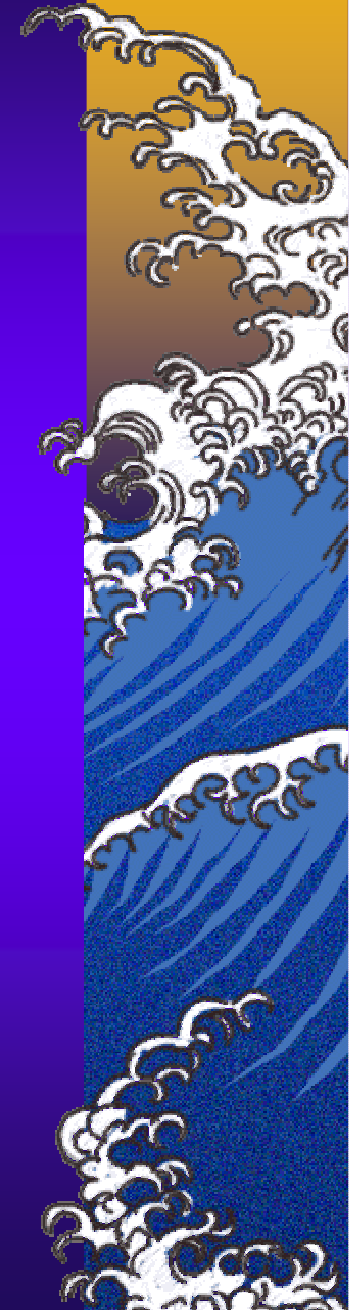
Edward Tsang

Centre for Computational Finance and
Economic Agents (CCFEA)

University of Essex

<http://cswww.essex.ac.uk/CSP/finance>

<http://www.cfea-labs.net>



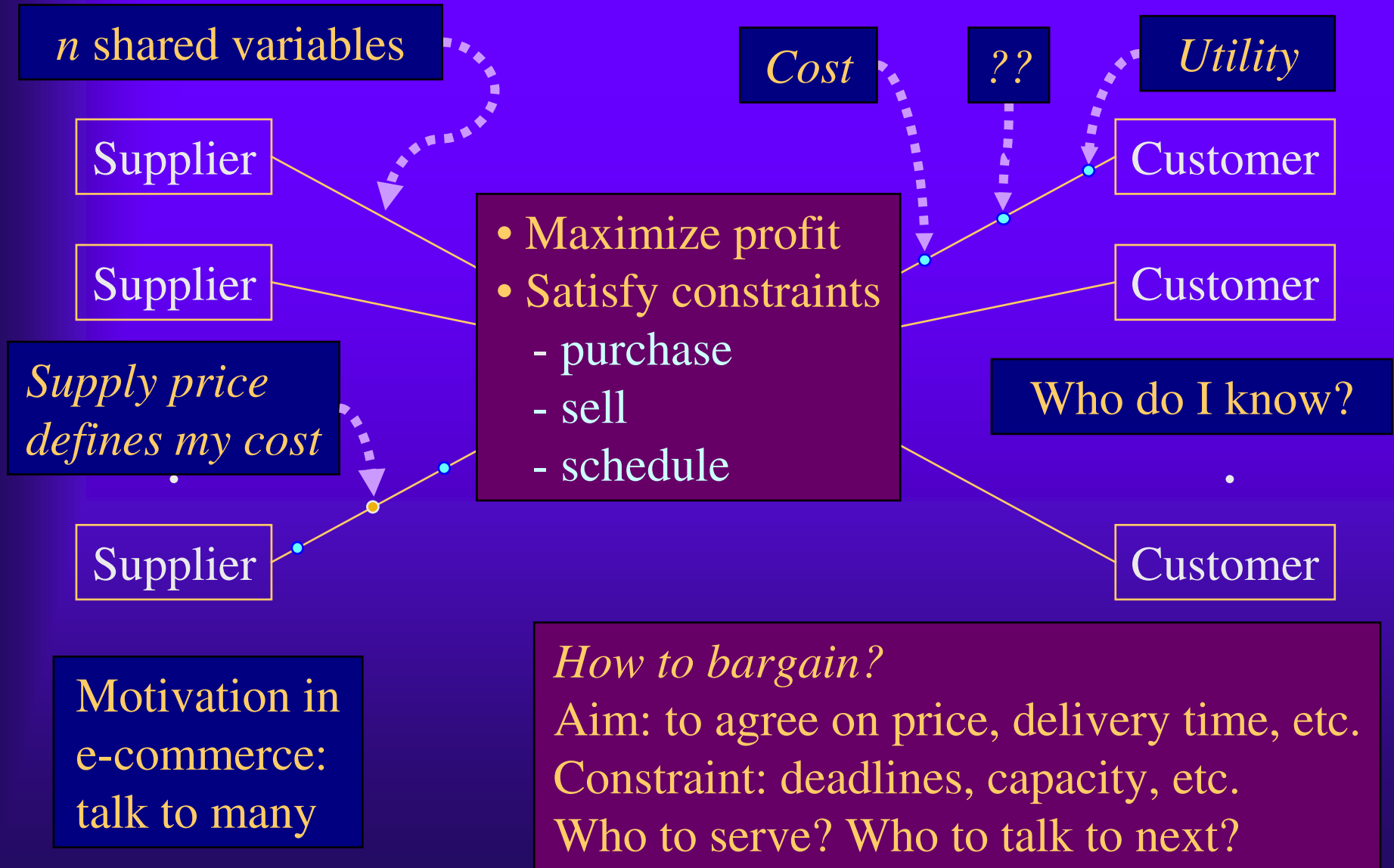


Centre for Computational Finance and Economic Agents (CCFEA)

- ◆ Interdisciplinary centre
- ◆ Director: Sheri Markose (Economics)
- ◆ Deputy: Edward Tsang (Computer Science)
- ◆ Lecturer: Olaf Menkins (CCFEA)
- ◆ City Associates chair: Nick Constantinou HSBC
- ◆ 11 PhD students, 26 Doctoral+Master students
- ◆ Selected Projects:
 - Forecasting, bargaining, payments, herding

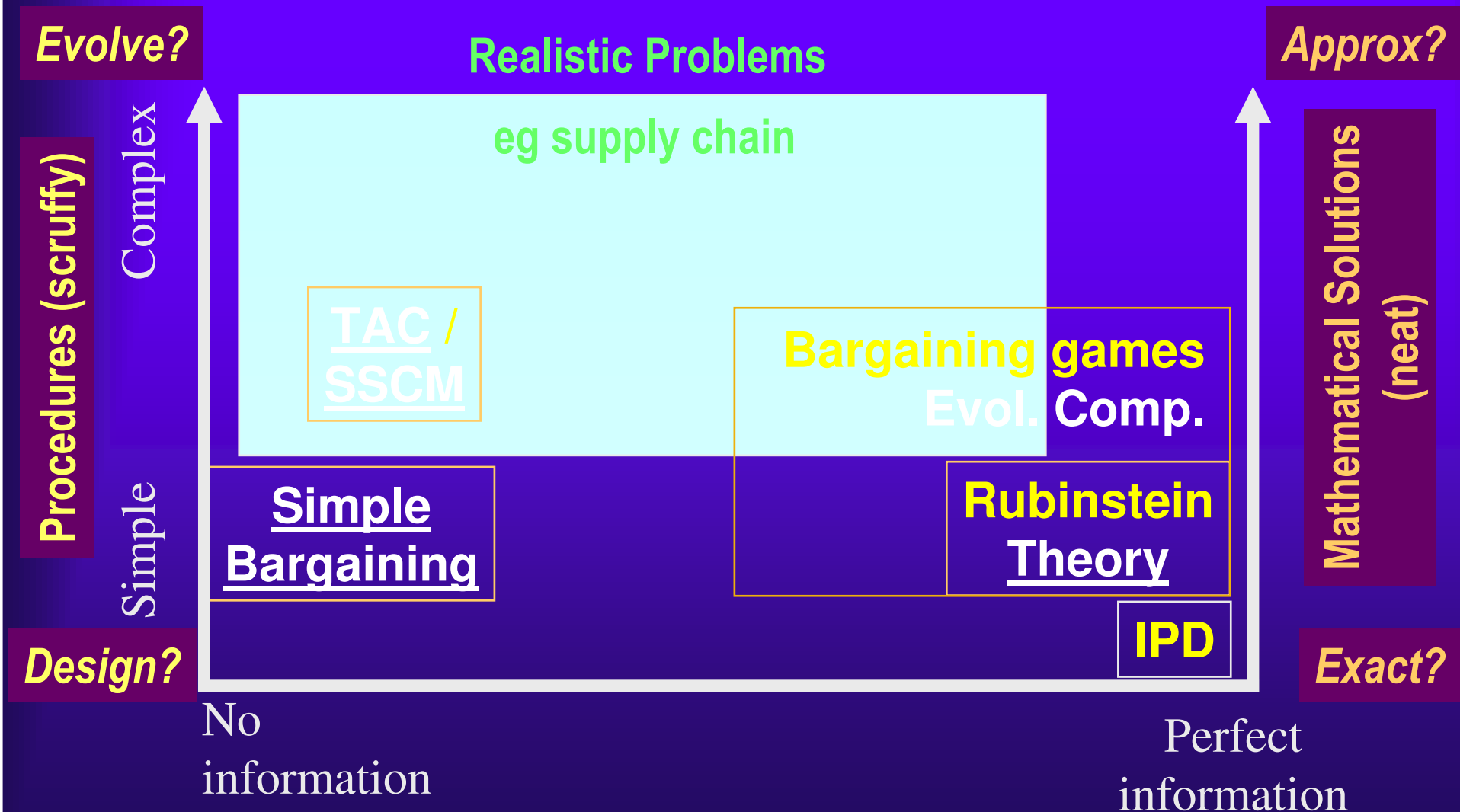


Automatic Bargaining Overview





Bargaining work at CCFEA





The Automatic Bargaining Research Team at Essex



Abhinay Muthoo
Economics
Game Theory



Sheri Markose
Economics/CCFEA
Red Queen Effect



Maria Fasli
Computing
Agent Tech.



Edward Tsang
Computing/CCFEA
Constraints,
Business models



Nanlin Jin
Computing
Extending Rubinstein Model
Evolving strategies



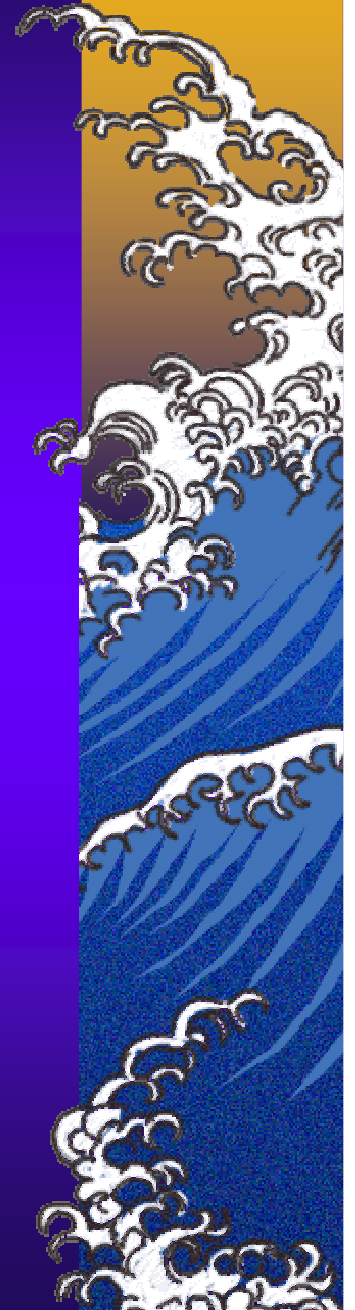
Tim Gosling
Computing/BT
Distributed scheduling
Evolving middlemen



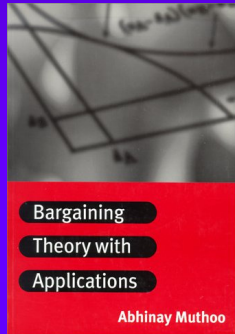
Biliana Alexandrova-Kabadjova
CCFEA/BoMexico
Electronic money
Payment System

Lots of ideas
to be explored

Collaborations
Welcome



Bargaining Theory



Abhinay Muthoo

[http://www.essex.ac.uk/economics/
people/staff/muthoo.shtm](http://www.essex.ac.uk/economics/people/staff/muthoo.shtm)



Bargaining in Game Theory

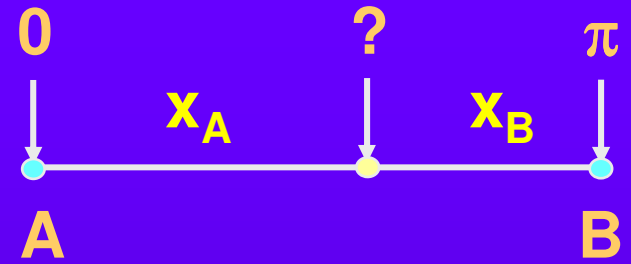
◆ Rubinstein Model:

In reality:

Offer at time $t = f(r_A, r_B, t)$

Is it necessary?

Is it rational? (What is rational?)



◆ A's payoff x_A drops as time goes by

$$A's \text{ Payoff} = x_A \exp(-r_A t\Delta)$$

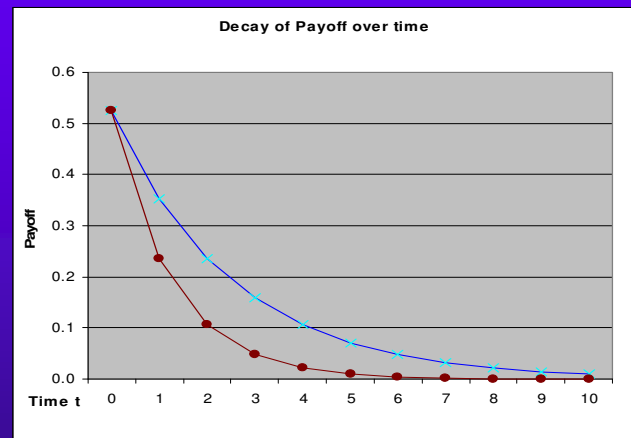
◆ Important Assumptions:

- Both players rational
- Both players know *everything*

◆ Equilibrium solution for A:

$$\mu_A = (1 - \delta_B) / (1 - \delta_A \delta_B)$$

where $\delta_i = \exp(-r_i \Delta)$



Optimal offer:

$$x_A = \mu_A \text{ at } t=0$$

Notice:

No time t here

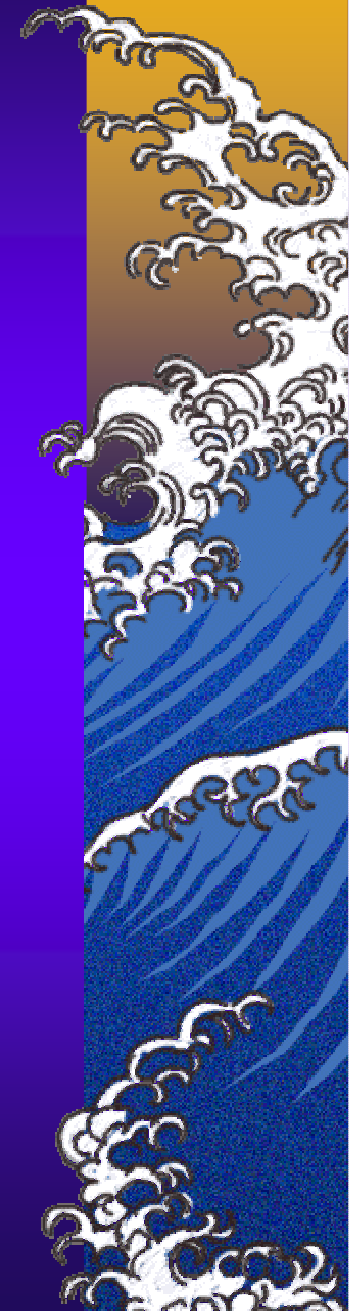




Evolutionary Bargaining Strategies

Nanlin Jin

<http://cswww.essex.ac.uk/CSP/bargain>





Evolutionary Bargaining Strategies

◆ Prisoners' Dilemma

- Co-evolution
- GA vs PBIL

◆ Rubinstein's Model

- Offer at time $t = f(r_A, r_B, t)$
- x_A^* , x_B^* emerged as best results
- Other solutions emerged occasionally

◆ Current work

- Asymmetric information
- Outside options



Evolutionary Rubenstein Bargaining, Overview

- ◆ Game theorists solved Rubenstein bargaining problem
 - Subgame Perfect Equilibrium (SPE)
- ◆ Slight alterations to problem lead to different solutions
 - Outside option
 - Asymmetric information
 - Different time intervals
- ◆ Evolutionary computation
 - Succeeded in solving a wide range of problems
 - EC has found SPE in Rubenstein's problem
 - Can EC find solutions close to unknown SPE?
- ◆ Co-evolution is an *alternative approximation* method to find game theoretical solutions
 - Less time for approximate SPEs
 - Less modifications for new problems



Rubinstein Solution vs Experimental Results

(δ_A, δ_B)	Rubinstein Solution x_A'	Experimental x_A	
		μ	σ
(0.4, 0.4)	<i>0.7143</i>	0.8973	0.0247
(0.4, 0.6)	<i>0.5263</i>	0.5090	0.0096
(0.4, 0.9)	<i>0.1563</i>	0.1469	0.1467
(0.9, 0.4)	<i>0.9375</i>	0.9107	0.0106
(0.9, 0.6)	<i>0.8696</i>	0.8000	0.1419
(0.9, 0.9)	<i>0.5263</i>	0.5065	0.1097
(0.9, 0.99)	<i>0.0917</i>	0.1474	0.1023

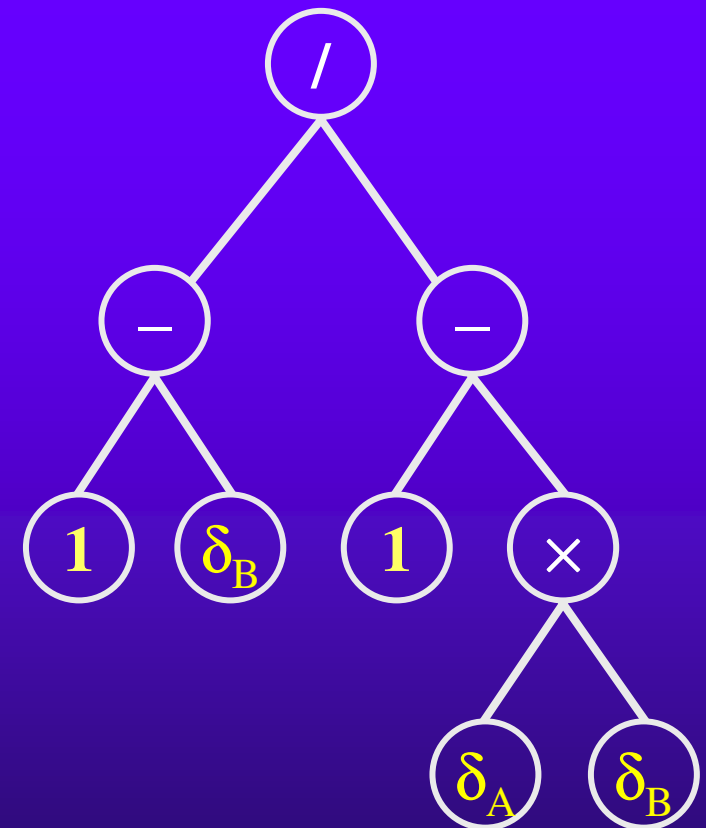
x_A : agreement made by the best strategies in the final (300th) generation

Population size 100; Crossover rates 0 to 0.1; Mutation rates 0.01 to 0.5; Tournament size 3



Issues Addressed, EC for Bargaining

- ◆ Representation
- ◆ One or two population?
- ◆ How to evaluate fitness
 - Fixed or relative fitness?
- ◆ How to contain search space?
- ◆ Discourage irrational strategies:
 - Ask for $x_A > 1$?
 - Ask for more over time?
 - Ask for more when δ_A is low?



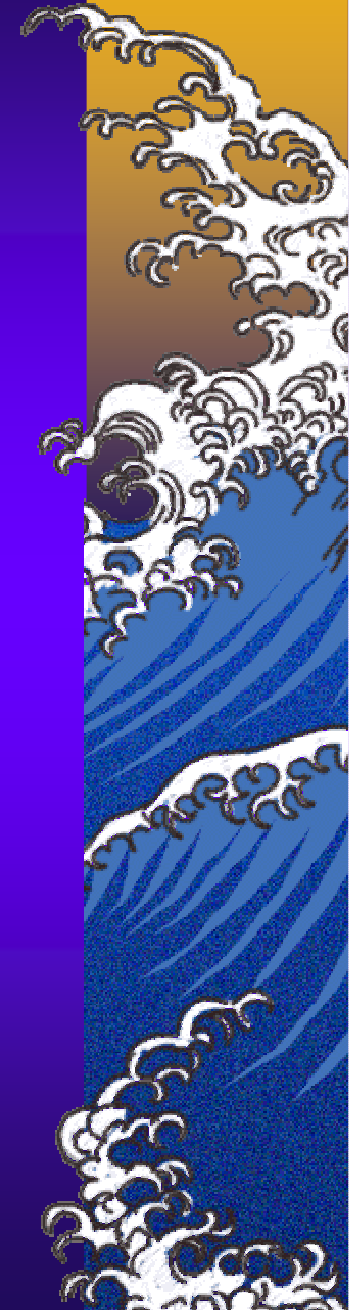
Simple Supply Chain Management Models



Tim Gosling

<http://cswww.essex.ac.uk/CSP/bargain>

BTexact Studentship

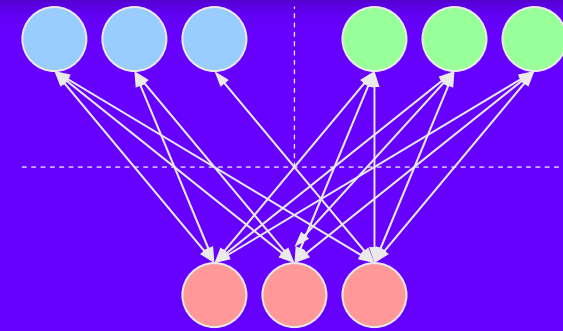




Motivation

- ◆ Humans are very good at:
 - Situation analysis and negotiations
- ◆ Humans are not so good at:
 - Handling large volumes of info & transactions
 - Having several conversations at once
- ◆ Motivated by large electronic supply chains
- ◆ Computer based strategy called for
- ◆ Success is not simply affected by bargaining skills, but also the number of agents it can talk to and the volumes it can handle

The SSCM Mission



- ◆ Provides a simple supply chain trading model
- ◆ Defines three types of participants:
 - Customers
 - Supplier
 - Middlemen (who we are mainly interested in)
- ◆ Middlemen strategy paramelised
- ◆ EC is used for evolving strategies
- ◆ The method is hoped to be general & practical

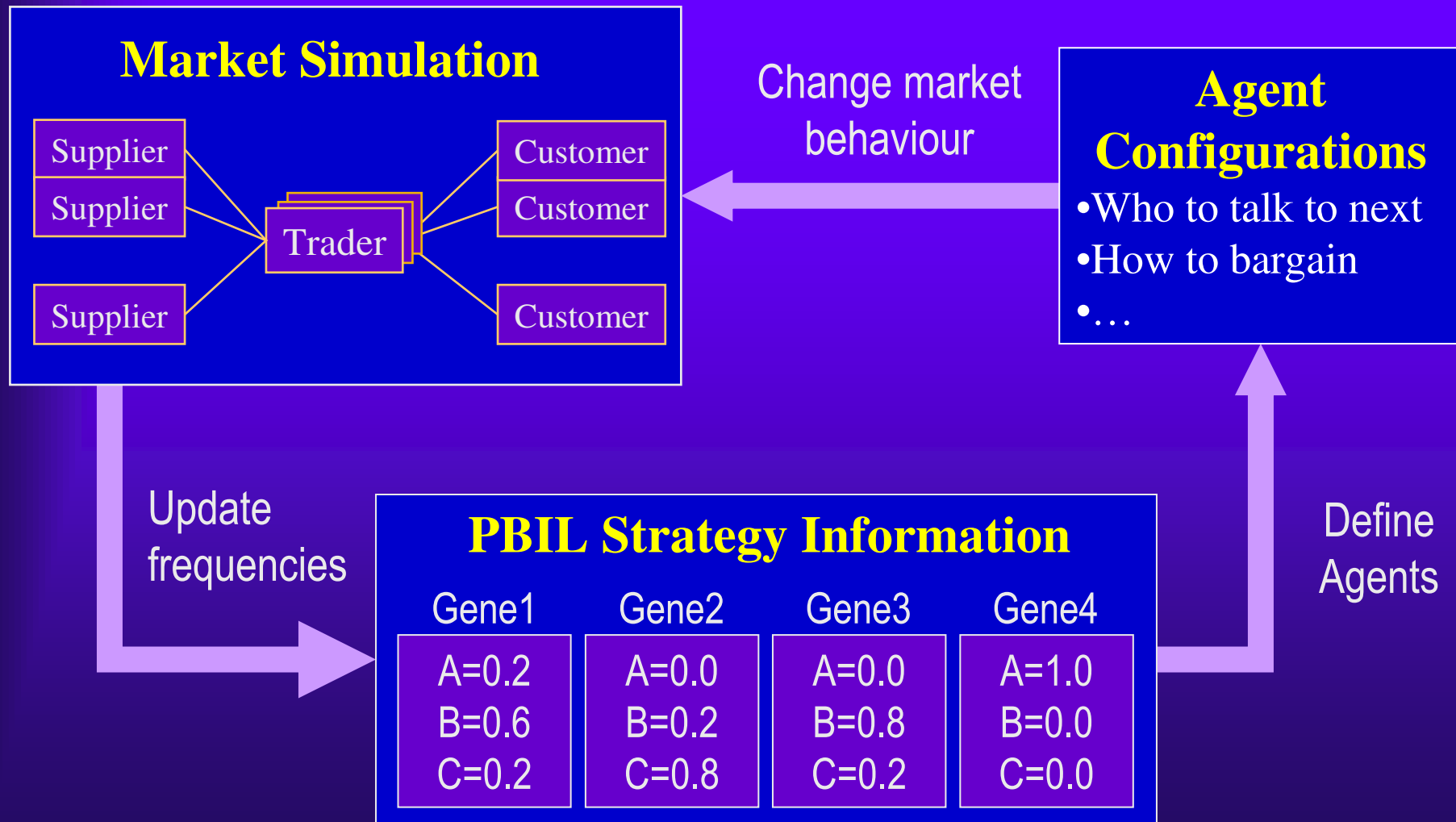


Scenarios Studied

- ◆ Customers requirements specific & non-negotiable
 - Satisfiable S2: Customer requirements are negotiable
- ◆ Dedicated customers for each middleman
- ◆ Customers initiate trade
- ◆ Suppliers have limited supply of resources
- ◆ One supplier per product S3: Multiple suppliers per product
- ◆ Suppliers passive: wait for requests from middlemen
- ◆ Middlemen task is to:
 - Evaluate requirements, reject those it can't fulfil
 - Attempt to fulfil remaining requirements



Population-based Incremental Learning (PBIL) in Simple Supply Chain Management (SSCM)

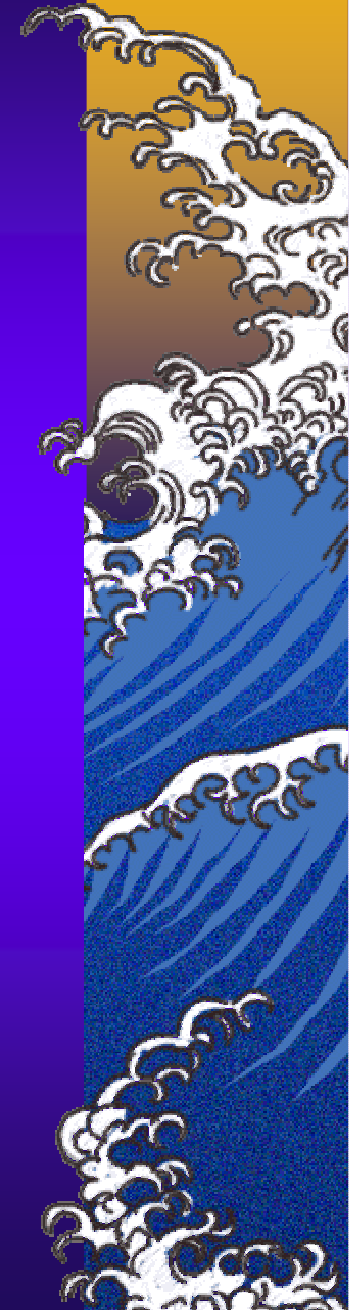


Trading Agents Competition for E-Commerce



Maria Fasli

<http://cswww.essex.ac.uk/staff/mfasli>

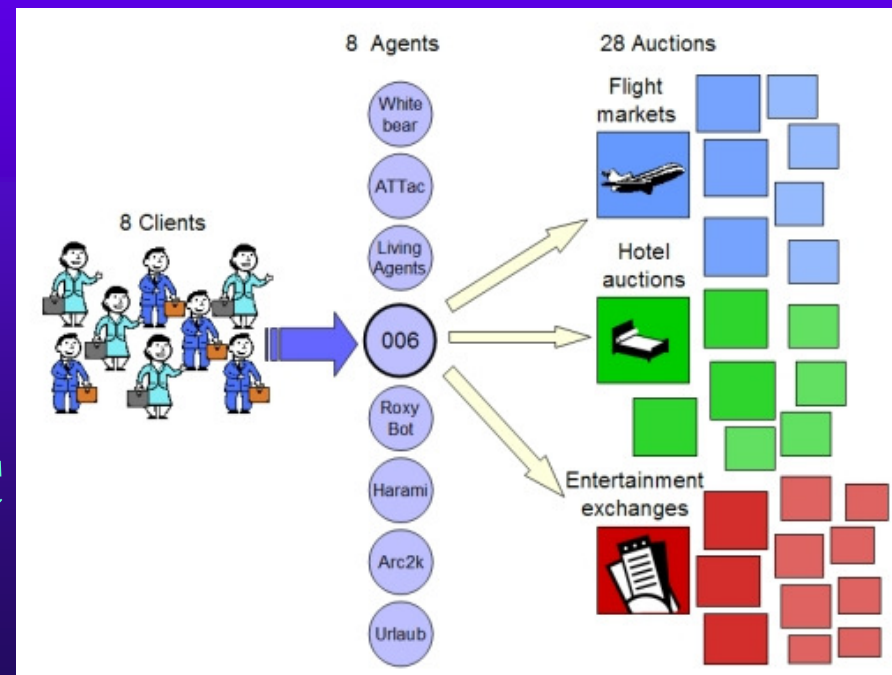




Trading Agents Competition (TAC)

◆ Classic Game (*Thalis*)

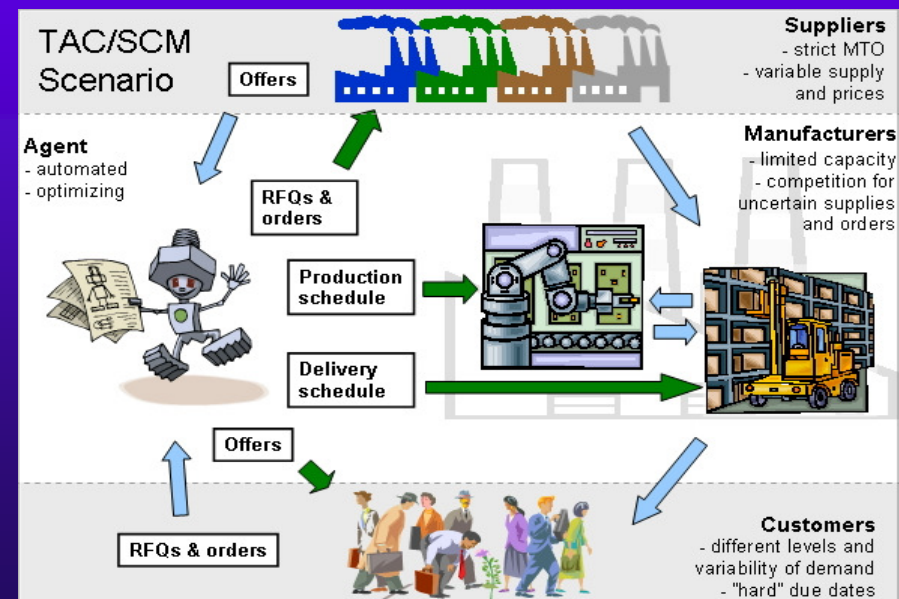
- Simultaneous auctions with substitutable and interrelated goods
- Dynamic bid configuration depending on historical data, current state and projected state
- Application of *Strategic Demand Reduction*
- Domain-specific heuristics
- 3rd and 4th positions in TAC 2003 and 2004 respectively



TAC Work at Essex

◆ Supply Chain Management Game (*Socrates*)

- An agent acts as a reverse auctioneer with the suppliers in multi-attribute auctions with substitutable and interrelated goods. Suppliers use a reputation mechanism and their delivery may be partial or complete
- Dynamic scheduling for production and delivery
- Ordering strategy and factory utilisation are interdependent and crucial
- ICEC-03: 7th position

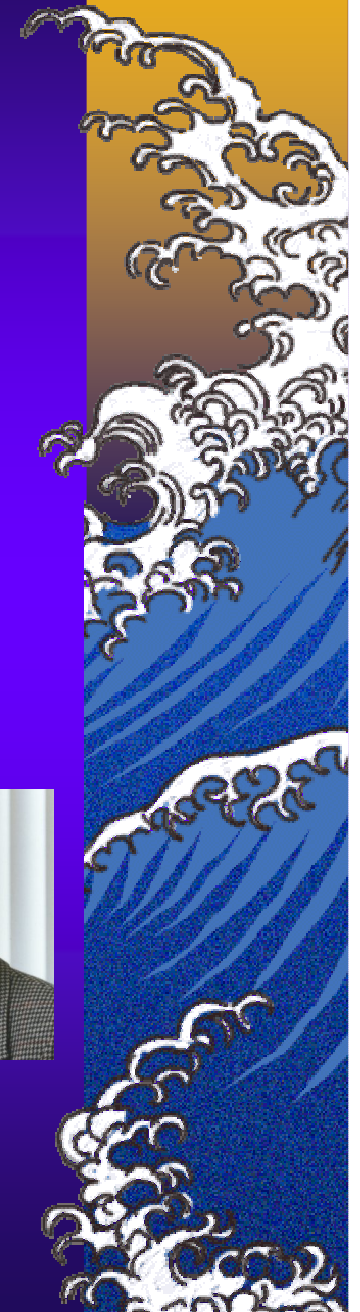


Bargaining as Constraint Satisfaction

Simple Bargaining Game

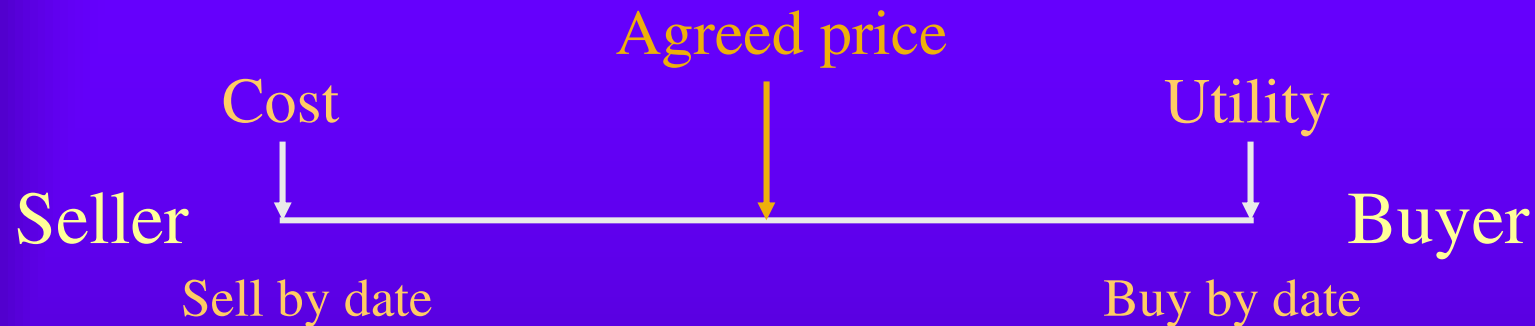
Edward Tsang

<http://cswww.essex.ac.uk/CSP/edward>





Local Constraint Optimisation



- ◆ Every agent is self-centred
- ◆ Agents constrain each other
- ◆ The simplest form of local constraint satisfaction / optimisation above
 - All deeper research depends on strategy in this problem



Information Available, Tournament 3.1

- ◆ No information on others' constraints
 - No information about the range of costs and utilities were available
- ◆ Bid history available within each game
 - e.g. [+45, -80, +40, -90]
- ◆ No information on previous games
 - i.e. no knowledge on identity of opponent



Tournaments

- ◆ Tournament 3.1 (2002)
 - No information about opponents
- ◆ Tournament 3.2 (2003)
 - Ranges of cost/utilities/SBD/BBD known
- ◆ Tournament 6.0 (2003)
 - Chain bargaining
- ◆ Tournament 5.1 (2004)
 - No SBD/BBD, each round costs £k to the player



Simple Chain-Bargaining Game

Chain completes iff
all adjacent players
agree on deals



- ◆ More information \rightarrow more mathematical solutions
- ◆ Less information \rightarrow procedural (messy) strategies





Meet the Sellers

- ◆ **The Jacob-Seller (dgiaco_s)**
 - Drop price linearly, make obvious drop in penultimate move
- ◆ **The Keen-Seller-2 (keen_s2)**
 - Half price each turn, keen to accept deals
- ◆ **The Stubbings-Seller (pmstub_s)**
 - Reduce price at increasing rate, try to recognize deadlines
- ◆ **The Stacey-Seller (rpstac_s)**
 - Complex rules for various situations, hard-bargaining
- ◆ **The Smart-Seller-4 (smart_s4)**
 - Estimate buyer's bottom line based on bid-history





Jacob-Seller (dgjaco_s)

- ◆ Accepts bids that are above the cost by a predefined margin,
- ◆ Or when it judges (based on the bids history) that the buyer has reached its limit.
- ◆ Start offer: cost plus a predefined premium
- ◆ General rule: This offer is reduced linearly until 4th final day. It then offers cost plus a target profit (parameter to the program) for one move. The penultimate move makes an obvious drop in price to tempt the buyer. A minimum profit is demanded in the final offer.





Keen-Seller-2 (keen_s2)

- ◆ Relatively simple
- ◆ Keen to make deals as soon as the bid is above its cost, but...
- ◆ When time is available, attempt to get a better deal by delaying commitment by one round.
- ◆ Start by a very high offer
- ◆ General strategy: reduce price by half towards cost in each round.





Stubbings-Seller (pmstub_s)

- ◆ Special cases carefully checked and responded to
 - such as the buyer has bid below the cost of pmstub_s,
- ◆ General rule: offer $MC \times (1+r)^d$
 - where MC (minimum price) is 60% above cost,
 - $1+r$ is the given number of days to sell
 - d is the number of days gone.

i.e. reduce offer price at an increasing rate
- ◆ Attempt to judge whether buyer has reached deadline
 - Check if $(1+(b_1 - b_2)) / (1+(b_2 - b_3))$ is below 10%,
 - where b_1 , b_2 and b_3 are the last, last but one and two bids





Stacey-Seller (rpstac_s)

- ◆ Complex seller: 18 rules for various situations
- ◆ Drive hard bargains by various sensible means.
- ◆ When the bid is above cost, the bid is accepted if
 - (i) the last two bids are 50% above cost;
 - (ii) the last three bids are 25% above cost; or
 - (iii) the last four bids are 15% above cost.
- ◆ Final 2 days' strategies fine tuned with 7 rules
 - depending on its predetermined margin thresholds
 - and the buyer's latest offer.
- ◆ General rule: reduce offer by 7.5% of the cost per round, as long as the offer is above cost.





Smart-Seller-4 (smart_s4)

- ◆ *A Target is worked out*
 - principally based on an estimation of the pattern of the buyer's previous bids.
- ◆ Up to three bids are used to project the buyer's next bid.
- ◆ Haggle until it runs out of time, or
 - it believes the buyer has reached its bottom line
 - and the bid is above its cost.





Meet the Buyers

- ◆ **Keen-Buyer (keen_b)**
 - Simple buyer that accepts any offer below its utility
- ◆ **Progressive-Buyer-2 (progress_b2)**
 - Linearly increase bid towards utility
- ◆ **Tryhorn-Buyer (mjtryh_b)**
 - Complex rules to predict target and drive hard bargain
- ◆ **Sourtzinos-Tsang-Buyer (psourt_b)**
 - Increase bids reflecting $utility^2 \div seller's\ offer$
- ◆ **Stacey-Buyer (rpstac_b)**
 - Complex rules for various situations, hard-bargaining





Keen-Buyer (keen_b)

- ◆ Simple buyer
- ◆ Keen to make deals
- ◆ Accept any offer that is below its utility
- ◆ Start: bid a low price
- ◆ General strategy: increase price by half towards the utility in each round





Progressive-Buyer-2 (progress_b2)

- ◆ Increasing the bid linearly towards utility
- ◆ This gives the seller a chance to chart its progress and predict its bottom line
- ◆ Philosophy: give the seller a chance to cooperate should the seller wants to
- ◆ When the offer is below the utility, it is accepted if:
 - (a) there are less than 3 days left; or
 - (b) the latest offer is within 95 and 100% of the previous offer (this is seen to be a sign of the seller reaching its limit).





Tryhorn-Buyer (mjtryh_b)

- ◆ Built upon two important modules:
 - (a) a predictor that estimates the bottom line of the seller and
 - (b) a purchase-adviser that decides whether an offer is acceptable.
- ◆ Attempt to compute seller's arithmetic progress
- ◆ Complex rules were used to compute the next bid
- ◆ In general, drive a hard bargain by not raising its bids very much until late in the negotiation
- ◆ An offer is acceptable if it is the buyer's last day to buy
- ◆ Whether an offer is acceptable depends on
 - (a) the offer/utility ratio (the lower the better) and
 - (b) the length of the negotiation (the longer the negotiation, the keener it is to accept the offer).





Sourtzinos-Tsang-Buyer (psourt_b)

- ◆ Use a combination of bidding rules
- ◆ Bid 1000th of the seller's first offer
- ◆ Then bid 100th of seller's second offer
 - As long as the bids are below its utility.
- ◆ General rule: $\text{bid} = \text{Utility}^2 / \text{Last_offer}$
 - i.e. the fraction of the utility that reflects the ratio between the utility and the seller's last offer





Stacey-Buyer (rpstac_b)

- ◆ Complex buyer: 20 rules to handle various situations
- ◆ Drive hard bargains by various sensible means
 - Even when offer is below its utility, delay acceptance
 - Refuse to raise its bid if seller has not lowered its price for three rounds
- ◆ Final 2 days' strategies fine tuned with 6 rules
 - Depending on its predetermined margin thresholds (35%)
 - and seller's latest offer
- ◆ General rule: increase offer by 7.5% per round, as long as bid is below utility

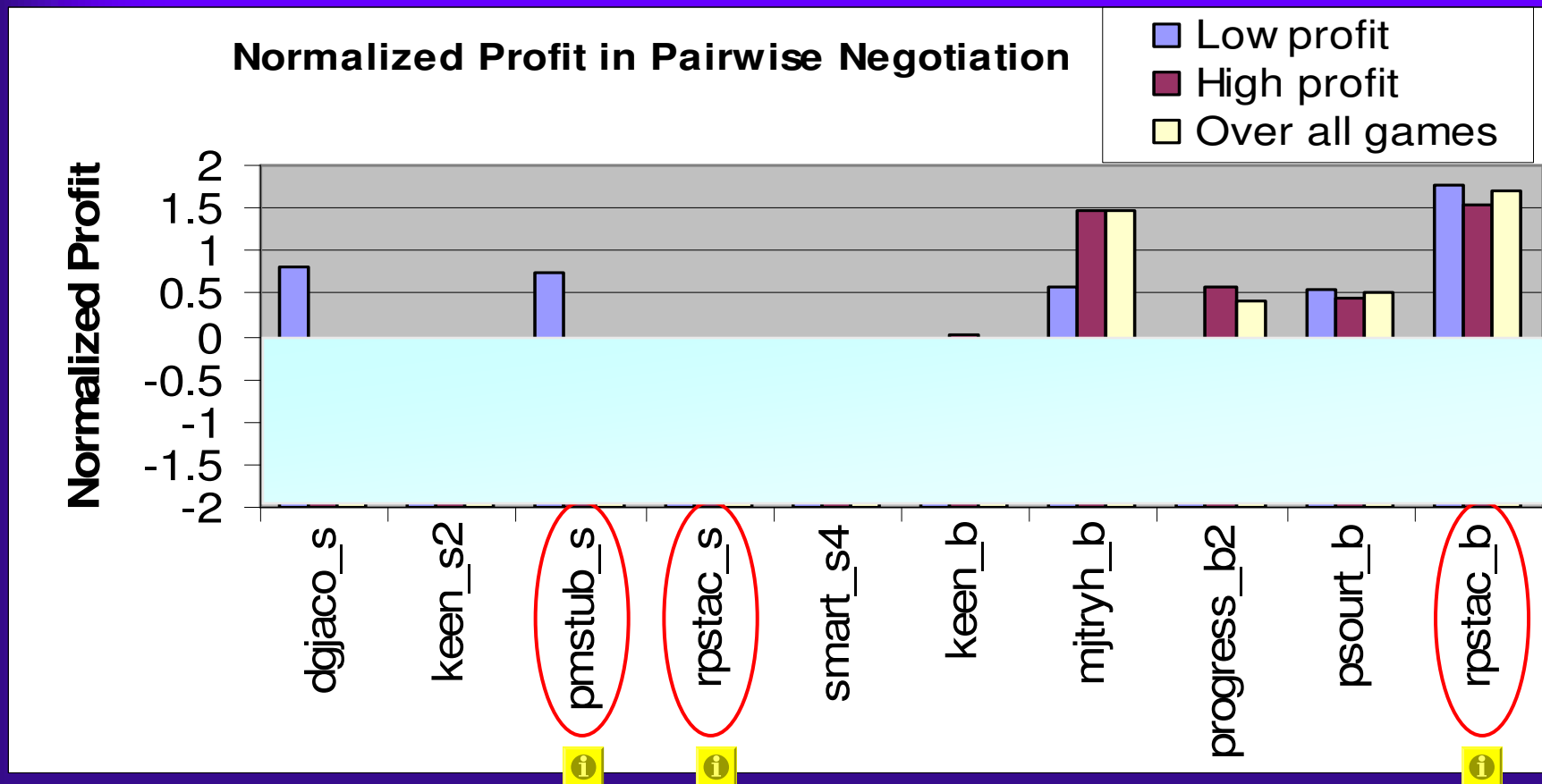




Experiment 1: No Middleman

- ◆ Every seller plays every buyer
- ◆ 1,000 randomly generated problems per pair
- ◆ Days to sell & Days to buy: 3..20
- ◆ Cost range: 101..300
- ◆ Utility range:
 - Low profit: 301..500
 - Medium profit: 1001..1300
 - High profit: 5101..5300

Individual seller/buyer Performance



- ◆ Buyers generally do better
- ◆ Aggressive sellers/buyers generally do better

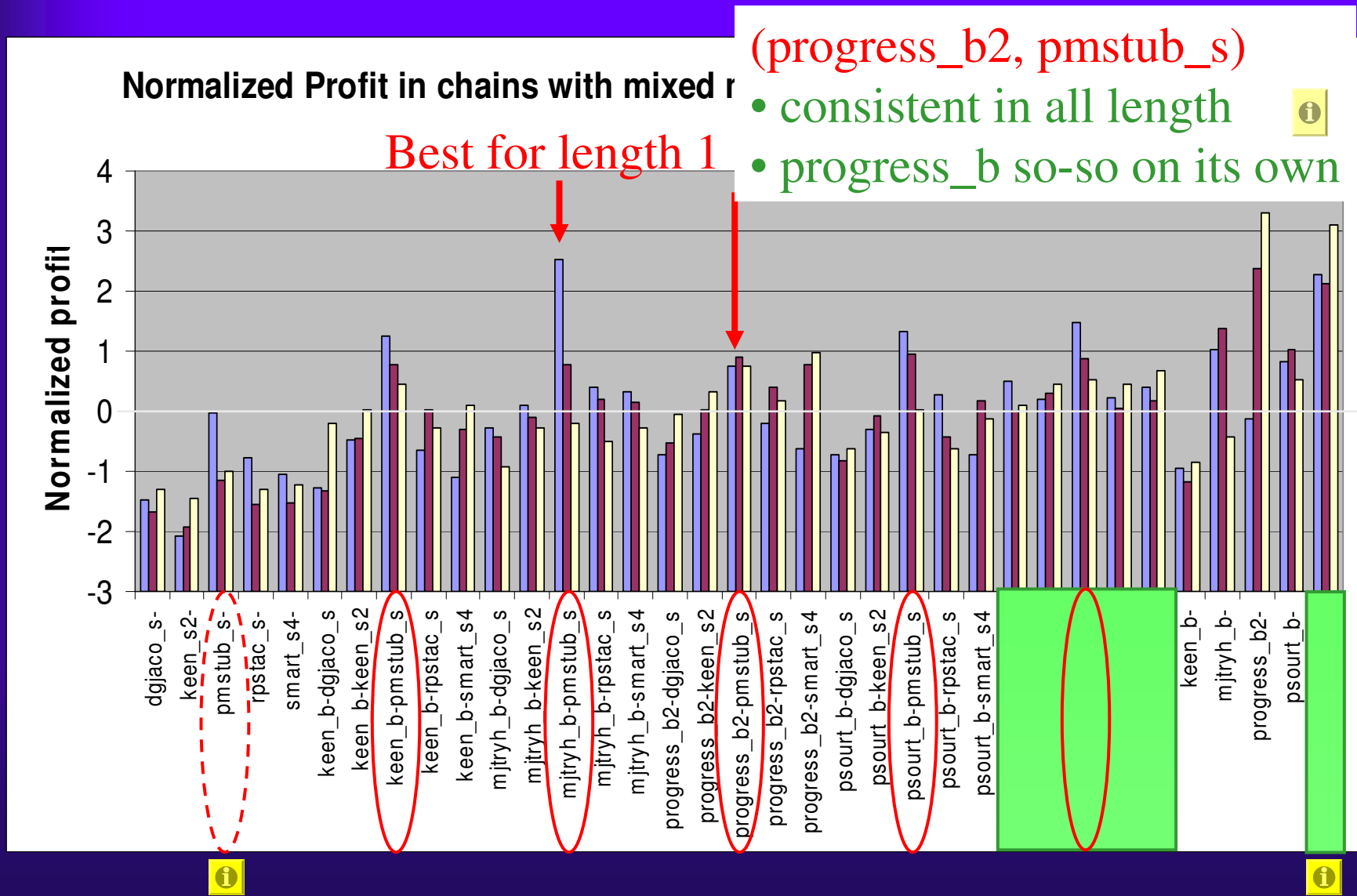


Experiment 2: Mixed Middlemen

- ◆ 1,000 randomly generated chains
 - With 1, 5 and 10 middlemen per chain
 - Each middleman = (random seller, random buyer)
- ◆ Days to sell & Days to buy: 3..20 (as before)
- ◆ Cost range: 101..300
- ◆ Utility range: low & high profit
- ◆ Utility range and # of games varied over chain length



Chains with Mixed Middlemen





Lessons from Mixed Middlemen Chains

- ◆ Recognizing others' constraints is key to completion
- ◆ Middlemen that allow others to estimate their bottom-line performed reasonably well
 - E.g. (progress_b2, pmstub_s) & (keen_b, pmstub_s)
- ◆ Presence of hard-bargainers maintain high prices in the chain
 - With high prices, chains cannot complete even when constraints are recognized
 - When a chain failed to complete, everyone suffers
 - So the hard-bargainers performed reasonably well
- ◆ Long chains are less likely to complete



Experiment 3: Uniform Chains

- ◆ Chains with the same middleman repeated:
(Seller, (B,S), (B,S), ..., (B,S), Buyer)
- ◆ Useful to assess evolutionary stable middlemen
- ◆ 5 sellers x 5 buyers \rightarrow 25 possible middlemen
- ◆ Chains with 1, 5 and 10 middlemen
- ◆ Same set of problems for each of the 25 chains

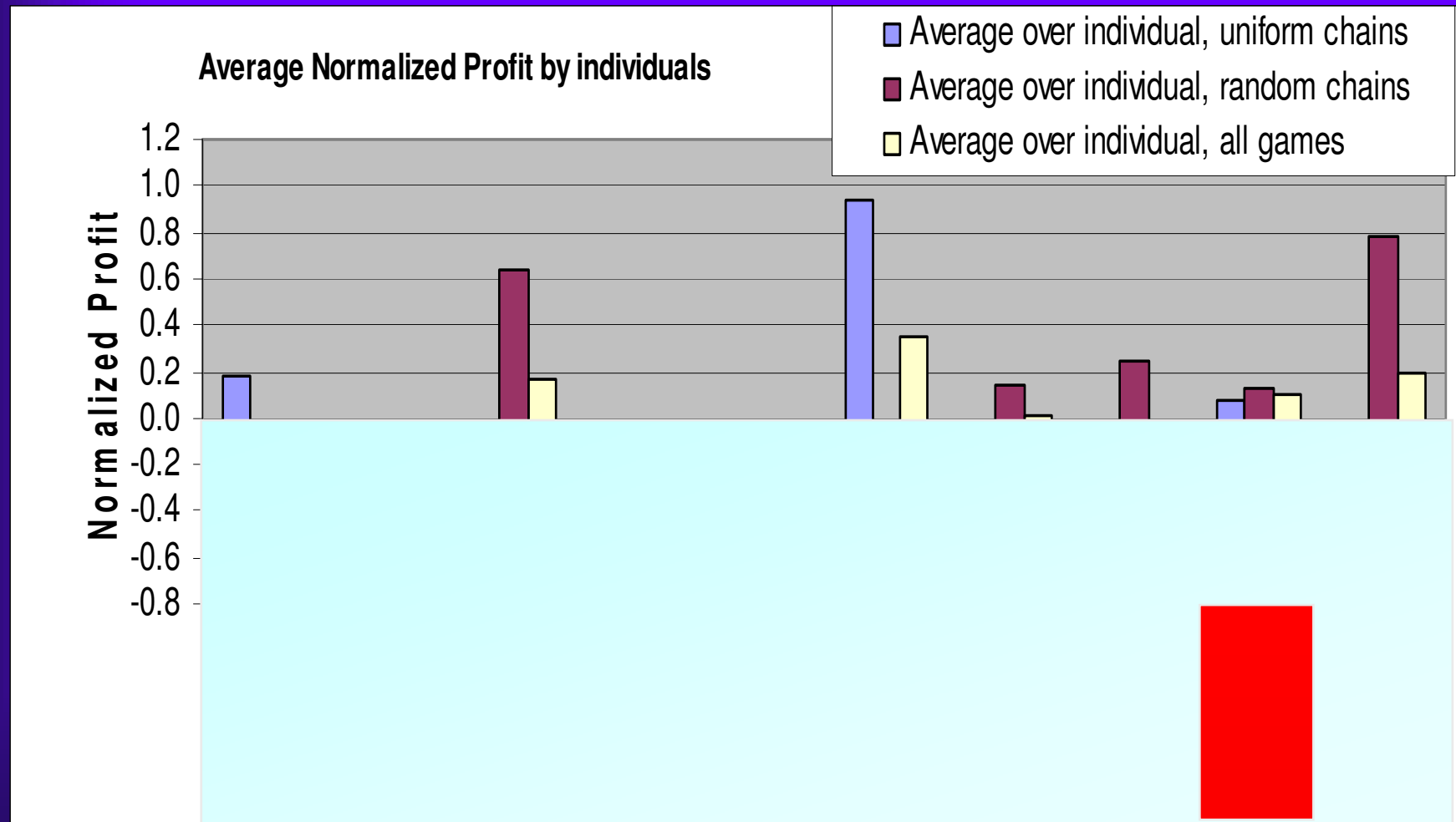


Lessons from Uniform Chains

- ◆ Consistent performers:
 - (keen_b, keen_s2), (keen_b, pmstub_s), (keen_b, rpstac_s), (keen_b, smart_s4), (psourt_b, pmstub_s)
- ◆ All but one involves easy-going players, keen_b or keen_s2
- ◆ Hard-bargainers rpstac_b and rpstac_s scored badly; they compromised too late
 - For any chain to complete, one buyer and one seller must initiate compromises



What are good components?





Summary: Lessons Learned

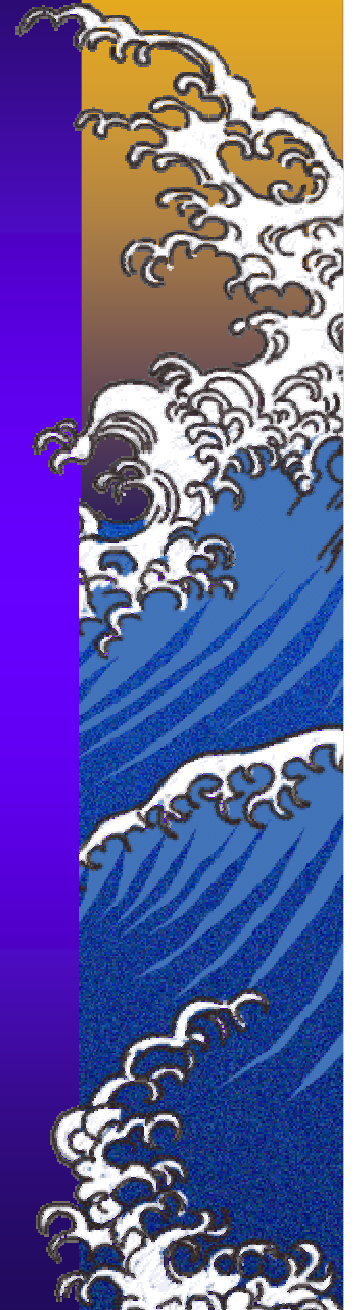
- ◆ No evolutionary stable strategy in our sample
- ◆ It pays to drive hard bargains in mixed chains
 - When a chain breaks down, everyone suffers
- ◆ Recognizing others' constraints is important
 - Revealing one's bottom line may not be too bad
- ◆ Performance depends on profit margin, chain length and chain formation
 - Adaptation is the only chance to succeed

Survival of the Fittest in Chain Bargaining

Fitter strategies make more copies

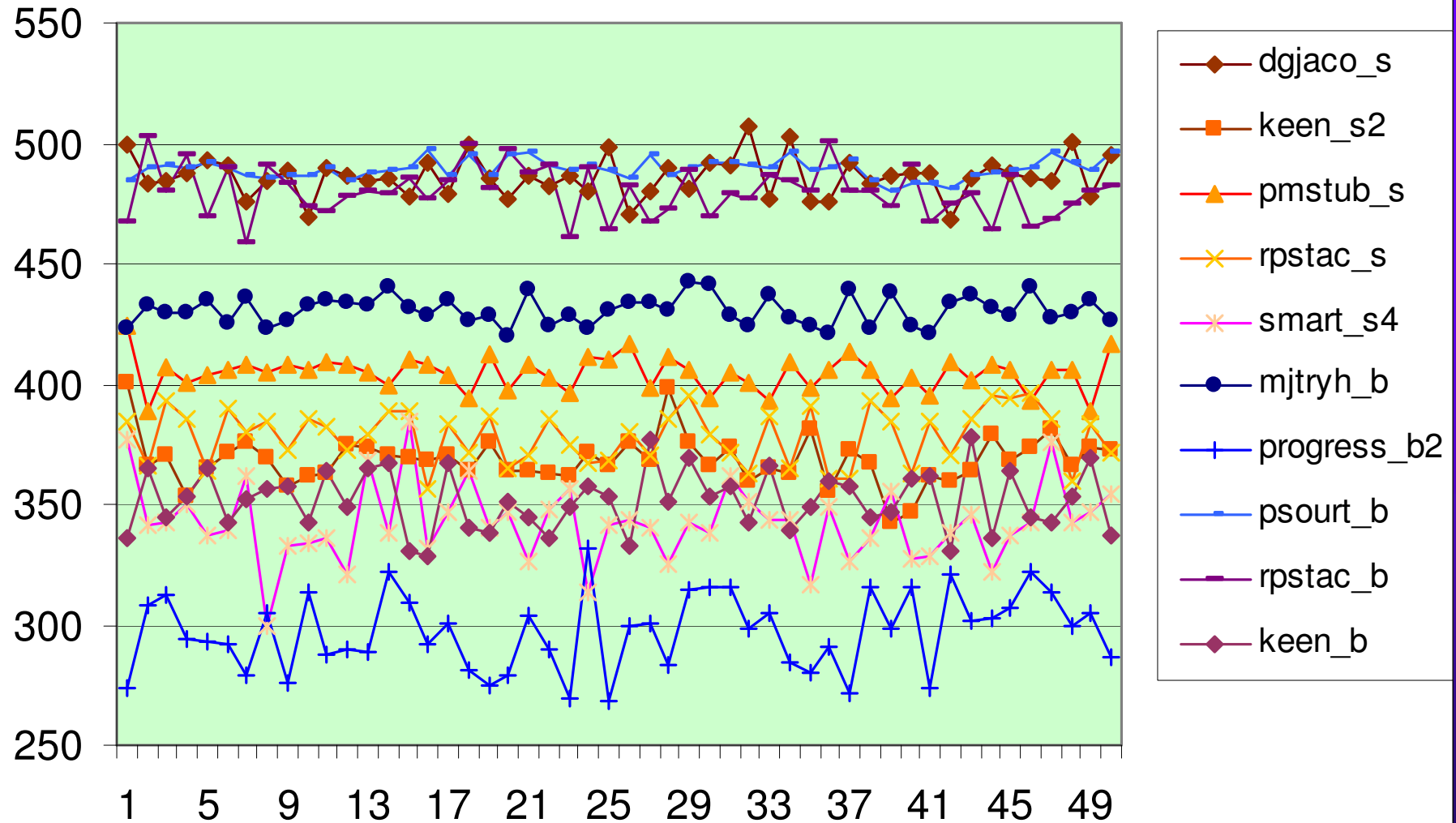
Will the population converge?

If so, converged to what?



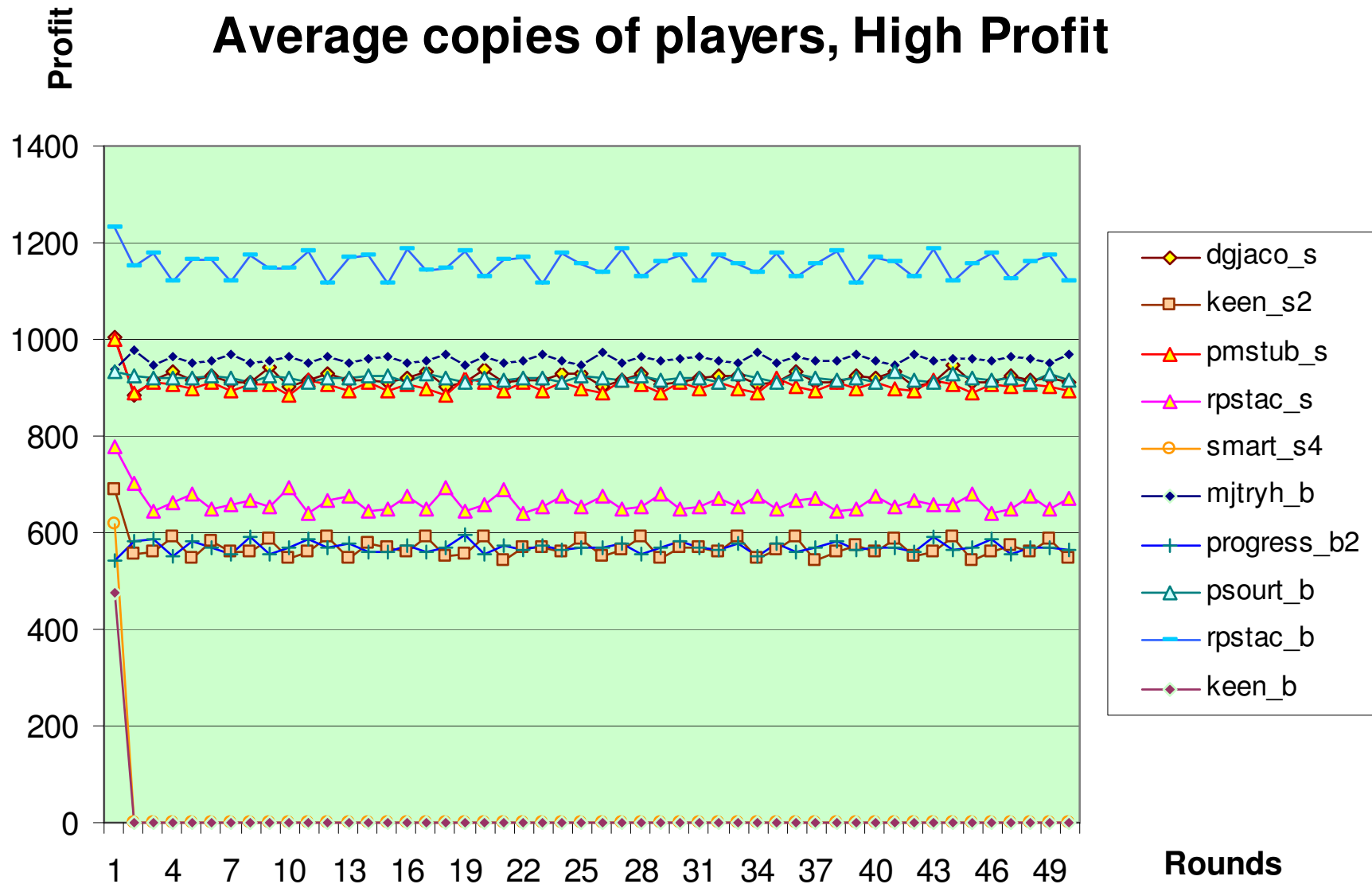


Average copies of players, Simple Bargaining Game





Average copies of players, High Profit





Observations, evolutionary bargaining

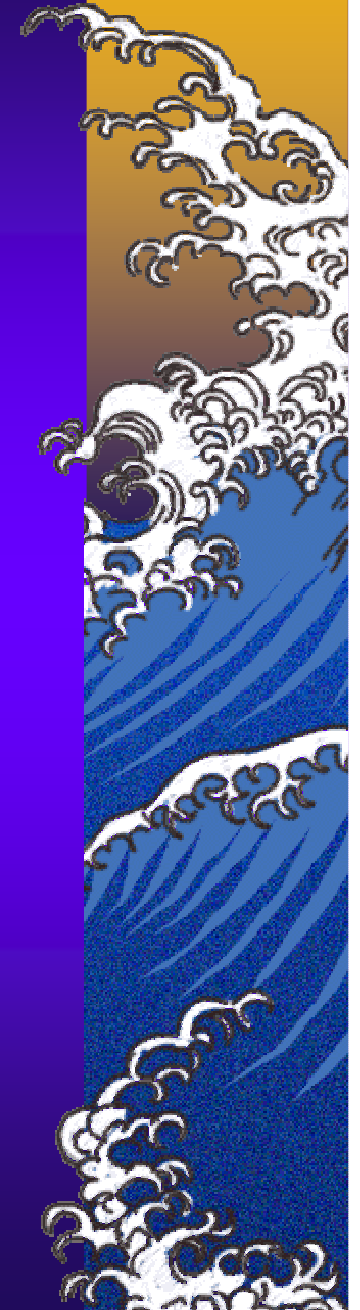
- ◆ Even the weakest player species survive in some settings; the weakest players died in others
- ◆ Consistent results
- ◆ Any correlation between
 - copies of player x in time t
 - to copies of player y in time $t+1$?

Nothing significant observed



Discussions on Bargaining

<http://cswww.essex.ac.uk/CSP/bargain>





What is Rationality?

- ◆ Are we all logical?
- ◆ What if *Computation* is involved?
- ◆ Does *Consequential Closure* hold?
 - If we know P is true and $P \rightarrow Q$, then we know Q is true
 - We know all the rules in Chess, but not the optimal moves
- ◆ “Rationality” depends on computation power!
 - Think faster \rightarrow “more rational”

