Evolutionary Bargaining

Game theory: Two players alternative offering game
Subgame perfect equilibrium found

What is my share??

Player A ⟷ Player B

Slight game modification →
Laborious work on new solutions
Perfect rationality assumption

Technical details (non-trivial)
Co-evolution
Incentive methods invented

- Proposal: EC for approximating solutions on new games

Two players alternative offering game

Player 1: How about 70% for me 30% for you?

t = 0, Player 1's pay off is 70%

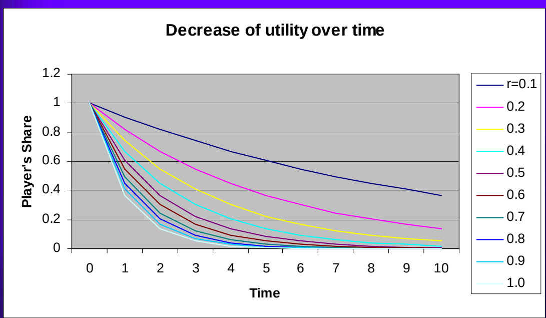
Player 1: No, how about 50-50?

t = 2, Player 1's pay off is 50% × $e^{-0.1 \times 2}$ = 41%

- If neither players have any incentive to compromise, this can go on for ever
- Payoff drops over time – incentive to compromise
- A's Payoff = $x_A \exp(-r_A t\Delta)$

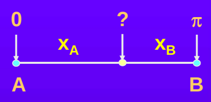
Let r_1 be 0.1, Δ be 1

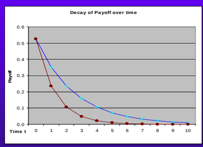
Payoff decreases over time



Bargaining in Game Theory

In reality:
Offer at time $t = f(r_A, r_B, t)$
Is it necessary?
Is it rational? (What *is* rational?)





t = # of rounds, at time Δ per round

- A's payoff x_A drops as time goes by
  A's 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:
$x_A = \mu_A$
at t=0

Notice:
No time *t* here

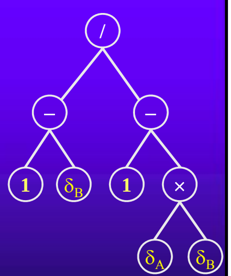
25 November 2009

Evolutionary Computation for Bargaining

Technical Details

Issues Addressed in EC for Bargaining

- Representation
  - Should *t* be in the language?
- 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?

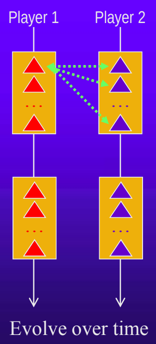


A run through

25 November 2009

Two populations – co-evolution

- We want to deal with asymmetric games
  - E.g. two players may have different information
- One population for training each player's strategies
- Co-evolution, using relative fitness
  - Alternative: use absolute fitness



Representation of Strategies

- A tree represents a mathematical function g
- Terminal set: $\{1, \delta_A, \delta_B\}$
- Functional set: $\{+, -, \times, \div\}$
- Given g, player with discount rate r plays at time t

$$g \times (1 - r)^t$$

- Language can be enriched:
  - Could have included e or time t to terminal set
  - Could have included power ^ to function set
- Richer language → larger search space → harder search problem

Incentive Method: Constrained Fitness Function

- No magic in evolutionary computation
  - Larger search space → less chance to succeed
- Constraints are heuristics to focus a search
  - Focus on space where promising solutions may lie
- Incentives for the following properties in the function returned:
  - The function returns a value in (0, 1)
  - Everything else being equal, lower δ_A → smaller share
  - Everything else being equal, lower δ_B → larger share

  Note: this is the key to our search effectiveness

Incentives for Bargaining

- $F(g_i) =$

$GF(s(g_i)) + B$ $B \geq 3$ (tournament selection is used)	If g_i in (0,1] & $SM_i > 0$ & $SM_j > 0$
$GF(s(g_i)) + ATT(i) + ATT(j)$	If g_i in (0,1] & ($SM_i \leq 0$ or $SM_j \leq 0$)
$ATT(i) + ATT(j) - e^{(-1/\|gi\|)}$	If g_i NOT in (0,1]

- $GF(s(g_i))$ is the game fitness (GF) of a strategy (s) based on the function g_i, which is generated by genetic programming
- When g_i is outside (0, 1], the strategy does not enter the games

25 November 2009

C1: Incentive for Feasible Solutions

- The function returns a value in (0, 1]
- The function participates in game plays
- The game fitness (GF) is measured
- A bonus (B) incentive is added to GF
  - B is set to 3
  - Since tournament is used in selection, the absolute value of B does not matter (as long as B>3)

25 November 2009

C2: Incentive for Rational δ_A

- Everything else being equal, lower δ_A → smaller share for A

Given a function g_i:

- The sensitive measure $SM_i(\delta_i, \delta_j, \alpha)$ measures how much g_i decreases when δ_i increases by α

Attribute ATT(i) =

1	If $SM_i(\delta_i, \delta_j, \alpha) > 1$
$-e^{(1/SMi(\delta i, \delta j, \alpha))}$	If $SM_i(\delta_i, \delta_j, \alpha) \leq 1$

$0 < ATT(i) \leq 1$

25 November 2009

C3: Incentive for Rational δ_B

- Everything else being equal, lower δ_B → larger share for A

Given a function g_i:

- The sensitive measure $SM_j(\delta_i, \delta_j, \alpha)$ measures how much g_i increases when δ_j increases by α

Attribute ATT(j) =

1	If $SM_j(\delta_i, \delta_j, \alpha) > 1$
$-e^{(1/SM_j(\delta_i, \delta_j, \alpha))}$	If $SM_j(\delta_i, \delta_j, \alpha) \le 1$

$0 < ATT(j) \le 1$

Sensitivity Measure (SM) for δ_i

- $SM_i(\delta_i, \delta_j, \alpha)$ =

$[g_i(\delta_i \times (1 + \alpha), \delta_j) - g_i(\delta_i, \delta_j)] \div g_i(\delta_i, \delta_j)$	If $\delta_i \times (1 + \alpha) < 1$
$[g_i(\delta_i, \delta_j) - g_i(\delta_i \times (1 - \alpha), \delta_j)] \div g_i(\delta_i, \delta_j)$	If $\delta_i \times (1 + \alpha) \ge 1$

- SM_i measures how much g_i decreases when δ_i increases by α

  (g_i is the function which fitness is to be measured)

Sensitivity Measure (SM) for δ_j

- $SM_j(\delta_i, \delta_j, \alpha)$ =

$[g_i(\delta_i, \delta_j) - g_i(\delta_i, \delta_j \times (1 + \alpha))] \div g_i(\delta_i, \delta_j)$	If $\delta_j \times (1 + \alpha) < 1$
$[g_i(\delta_i, \delta_j \times (1 - \alpha)) - g_i(\delta_i, \delta_j)] \div g_i(\delta_i, \delta_j)$	If $\delta_j \times (1 + \alpha) \ge 1$

- SM_j measures how much g_i increases when δ_j increases by α

  (g_i is the function which fitness is to be measured)

Bargaining Models Tackled

Determinants	Complete Information	Uncertainty	
		1-sided	2-sided
Discount Factors	* Rubinstein 82	* Rubinstein 85 x *Imprecise info* *Ignorance*	x *Bilateral ignorance*
+ Outside Options	* Binmore 85	x *Uncertainty + Outside Options*	More could be done easily

* = Game theoretical solutions known
x = *game theoretic solutions unknown*

25 November 2009

Models with known equilibriums

Complete Information

- Rubinstein 82 model:
  - Alternative offering, both A and B know δ_A & δ_B
- Binmore 85 model, outside options:
  - As above, but each player has an outside offer, w_A and w_B

Incomplete Information

- Rubinstein 85 model:
  - B knows δ_A & δ_B
  - A knows δ_A
  - A knows δ_B is δ_w with probability w_0, δ_s ($> \delta_w$) otherwise

25 November 2009

Models with unknown equilibriums

Modified Rubinstein 85 / Binmore 85 models:

- 1-sided Imprecise information
  - B knows δ_A & δ_B; A knows δ_A and a normal distribution of δ_B
- 1-sided Ignorance
  - B knows both δ_A and δ_B; A knows δ_A but not δ_B
- 2-sided Ignorance
  - B knows δ_B but not δ_A; A knows δ_A but not δ_B
- Rubinstein 85 + 1-sided outside option

25 November 2009

Equilibrium with Outside Option

x_A^*	Conditions	
μ_A	$w_A \le \delta_A \mu_A$	$w_B \le \delta_B \mu_B$
$1 - w_B$	$w_A \le \delta_A(1-w_B)$	$w_B > \delta_B \mu_B$
$\delta_B w_A + (1-\delta_B)$	$w_A > \delta_A \mu_A$	$w_B \le \delta_B(1-w_A)$
$1 - w_B$	$w_A > \delta_A(1-w_B)$	$w_B > \delta_B(1-w_A)$
w_A	$w_A + w_A > 1$	–

25 November 2009

Equilibrium in Uncertainty – Rub85

$V_s = \frac{1-\delta_s}{1-\delta_1\delta_s}$	$\delta_2 = \delta_w$		$\delta_2 = \delta_s$	
	x_1^*	t^*	x_1^*	t^*
$w_0 < w^*$	V_s	0	V_s	0
$w_0 > w^*$	x^{w0}	0	$1 - ((1 - x^{w0}) / \delta_w)$	1

$$w^* = \frac{V_s - \delta_1^2 V_s}{1 - \delta_w + \delta_1 V_s(\delta_w - \delta_1)}$$

$$x^{w_0} = \frac{(1-\delta_w)(1-\delta_1^2(1-w_0))}{1-\delta_1^2(1-w_0)-\delta_1\delta_w w_0}$$

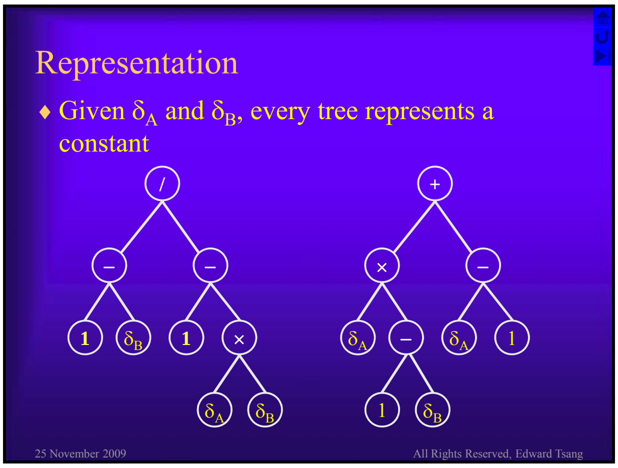
25 November 2009

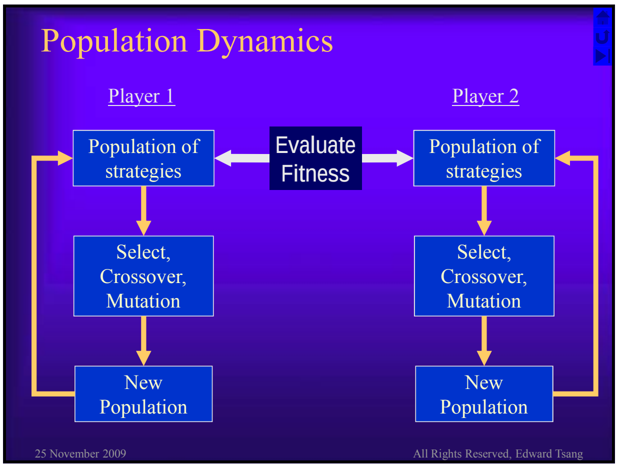
Evolutionary Bargaining Conclusions

- Demonstrated GP's flexibility
  - Models with known and unknown solutions
  - Outside option
  - Incomplete, asymmetric and limited information
- Co-evolution is an *alternative approximation* method to find game theoretical solutions
  - Perfect rationality assumption relaxed
  - Relatively quick for approximate solutions
  - Relatively easy to modify for new models
- Genetic Programming with incentive / constraints
  - Constraints helped to focus the search in promising spaces
- Lots remain to be done…

25 November 2009


Running GP in Bargaining
Representation, Evaluation
Selection, Crossover, Mutation


Representation
♦ Given δ_A and δ_B, every tree represents a constant
/
–
–
1
δ_B
1
×
δ_A
δ_B
+
×
–
δ_A
–
δ_A
1
1
δ_B
25 November 2009
All Rights Reserved, Edward Tsang


Population Dynamics
Player 1
Player 2
Population of strategies
Evaluate Fitness
Population of strategies
Select, Crossover, Mutation
Select, Crossover, Mutation
New Population
New Population
25 November 2009
All Rights Reserved, Edward Tsang

Evaluation

- Given the discount factors, each tree is translated into a constant x
  - It represents the demand represented by the tree.
- All trees where x < 0 or x > 1 are evaluated using rules defined by the incentive method
- All trees where 0 ≤ x ≤ 1 enter game playing
- Every tree for Player 1 is played against every tree for Player 2

25 November 2009

Evaluation Through Bargaining

Demands by Player 2's strategies

Player 1 Demands	.46	.31	.65	.20	Player 1 Fitness
.75	0	0	0	.75	0.75
.24	.24	.24	.24	.24	0.96
.36	.36	.36	0	.36	1.08
.59	0	.59	0	.59	1.18

- Incentive method ignored here for simplicity

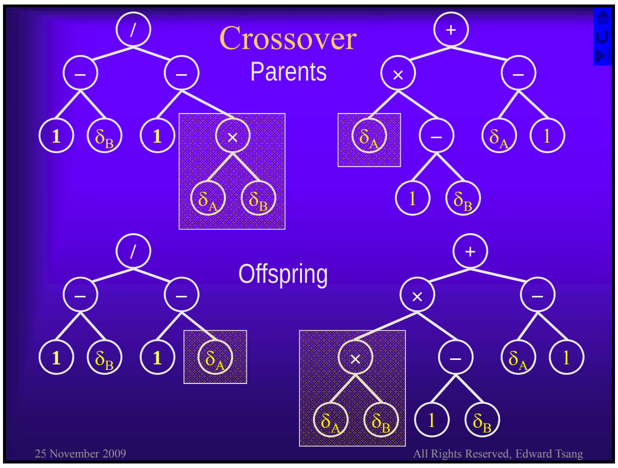
25 November 2009

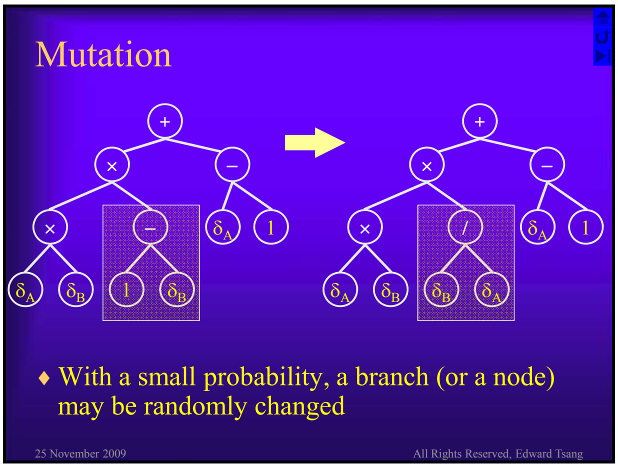
Selection

Rule (Demand)	Fitness	Normalized	Accumulated
R1 (0.75)	0.75	0.19	0.19
R2 (0.96)	0.96	0.24	0.43
R3 (1.08)	1.08	0.27	0.70
R4 (1.18)	1.18	0.30	1.00
Sum:	3.97	1	

- A random number r between 0 and 1 is generated
- If, say, r=0.48 (which is >0.43), then rule R3 is selected

25 November 2009


Crossover
Parents
Offspring
25 November 2009
All Rights Reserved, Edward Tsang


Mutation
♦ With a small probability, a branch (or a node) may be randomly changed
25 November 2009
All Rights Reserved, Edward Tsang