Some Thought-ettes on Artificial Agent Modeling and Its Uses*

Steven O. Kimbrough
University of Pennsylvania
http://opim-sun.wharton.upenn.edu/∼sok/
kimbrough@wharton.upenn.edu

ECEG, Washington, D.C., 1/26/04

*File: eceg-foils.tex/pdf.

Goals for the Presentation

- Trying to be responsive to the mandate, “Tell us about artificial agent models.”
- Will honor the request by deviating from a strict overview.
- *Agent model*, like many things, e.g., *client-server architecture*, have multiple, conflicting definitions.
- No use agonizing over a definition. Let’s look at examples in hopes of stimulating thought.

Context/background

Four ways to study contexts of strategic interaction (CSIs or games):

1. *A priori* — classical game theory; analytic study and results; rational choice theory presumed

2. *In vivo* — "games in the wild" Natural history of games? (needed)

3. *In vitro* — behavioral game theory; experimental economics; biology, too

4. *In silico* — or algorithmic game theory; This is where agent models come in.

Algorithmic Game Theory

- What happens when well-defined algorithmic agents meet in CSIs? Nash? Pareto?
- Why does what happens happen?
- How do smarts pay off (if at all)? Learning? What sorts of learning? *et cetera*. . .
- Important, essential for fielding artificial agents in CSIs, as in e-business
- My focus: Agents that learn (not merely adapt) in CSIs.

Focus: Repeated Games, Repeated Interactions

- *Supergame* — a game composed of games (called subgames)
- *Repeated game* — a special kind of supergame: subgames all the same; these are called *stage games*
- A lot of basic work: repeated games with 2×2 stage games.

OK, Agents

- In part, a nifty programming idea. Think: rule-based expert systems similarly motivated.

- In part, a response to ideas on distributed computing and problem solving, and ideas on emergence.

- Also, large strain of motivation for ‘bots’; autonomous, often mobile, agents doing work for us. Not central to our concern here.

OK, Agents, con't.

- Recommended: "WHY AGENTS? ON THE VARIED MOTIVATIONS FOR AGENT COMPUTING IN THE SOCIAL SCIENCES" by Robert Axtell, Brookings, Center on Social and Economic Dynamics, Working Paper No. 17, November 2000

- Programming/modeling idea: agents are objects, you set them loose in an environment and see what happens.

From Axtell's Paper

"...an agent based computational model becomes little more than:

```
program typical_agent_model;
        initialize agents;
        repeat:
                agents_interact;
                compute_statistics:
        until done;
end.
```

Code fragment 3: Typical agent-oriented program."

Well, yes and no.

Distinctions and Approaches

- Contests. One on one, etc. Appropriate especially for identity-centric agents.

- Tournaments. Strategy-centric. Axelrod's IPD tournaments. Seminal for agent-based modeling.

- 'Ecological' experiments. Aka: replicator dynamics. Most work has been 'mean field.' Important findings: zones of attraction for strategies. Beyond mean field replicator dynamics: a little correlation and "Evolution does not respect modular rationality." Fixation in a population of strictly dominated strategies. Say it again: major departures from standard economic models.

Distinctions and Approaches

- Spatial games (and societies). Fancy CA (cellular automata) systems. Focus of very much activity. Seminal and delightful work: *Growing Artificial Societies* by Epstein & Axtell.

  Has lead to development of software environments for (this sort of) agent-based modeling. Ascape, in Java.

  http://www.brook.edu/dybdocroot/es/dynamics/models/ascape/

- Network games. Work just getting underway. Natural for modeling power systems, I would think.

Agent-Based Modeling Environments

- Ascape (above). Axtell. Free from Brookings. Important commercial player: NuTech Solutions, Inc. http://www.nutechsolutions.com/
- Swarm. Out of SFI. C++ library. Complaints about difficulty. Very powerful.
- RePast. Java, open-source re-implementation of Swarm. Accessible and powerful. http://repast.sourceforge.net/
- StarLogo, NetLogo. Turtles and patches. A great place to begin.

Strategy-centric agents

- Strategy-centric versus identity-centric agents
- Strategy-centric: naked strategies, put out to play.
- Identity-centric: agents that acquire and modify, learn with strategies.

2×2 Games

- Previous work: "Simple Reinforcement Learning Agents: Pareto Beats Nash in an Algorithmic Game Theory Study" by Kimbrough and Lu, forthcoming in *Information Systems and e-Business Management*.

- Subsequent work (in NetLogo; in Java (thanks to Ann Kuo)).

Learning Regime

1. Alternative (or consideration) set of strategies: $\mathcal{A} = \{0, 1\}$ for each player.

2. Attractiveness estimation: linear updating rule for $A^i, \ \ i \in \mathcal{A}$:
   $A^i_{t+1} = A^i_t + \alpha\{r^i_t - A^i_t\}$

   NewEstimate = CurrentEstimate + StepSize$\{$reward - CurrentEstimate$\}$

3. Choice/exploration policies.

   Softmax. ϵ–greedy

Softmax

$$\Pr(A_t^i) = \frac{e^{A_t^i/\tau}}{\sum_j e^{A_t^j/\tau}}$$

$\tau \longrightarrow 0$ as $n \longrightarrow \infty$

Essential Findings

- When the numbers are right, agents tend to find Pareto outcomes, even at the expense of Nash outcomes, in terms of the stage game. (E.g. in Prisoner's Dilemma, chicken)

- When Nash and Pareto outcomes coincide and multiple Nash, agents tend (when the numbers are right) to find Pareto-optimal Nash outcomes. (E.g., Stag Hunt)

- Results sensitive to actual payoffs (in contravention to classical game theory)

- In any event, players tend to extract more wealth than would otherwise be predicted.

Example: Prisoner's Dilemma

	C	D
C	(3,3)**	(0, 3+δ)*
D	(3+δ, 0)*	(δ, δ)#

Table 1: #=Nash; *=Pareto

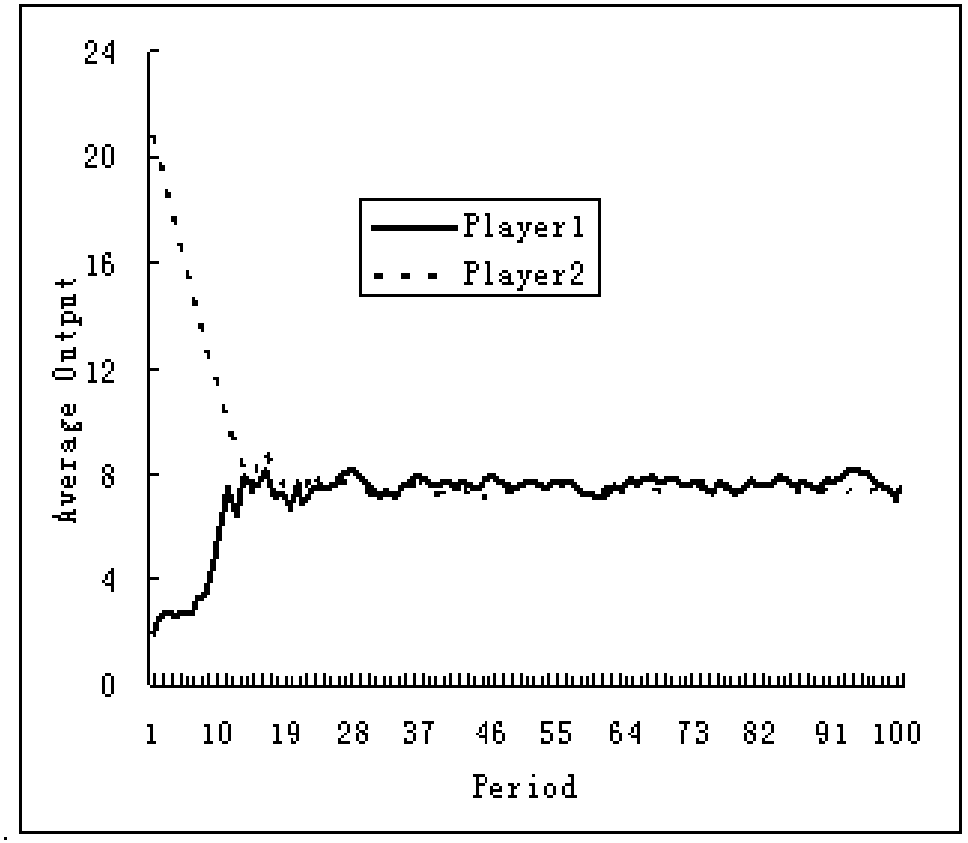
Summary of Results

$\varepsilon-$greedy action selection					Softmax action selection			
CC	CD	DC	DD	δ	CC	CD	DC	DD
9422	218	183	177	0.05	9334	302	285	79
9036	399	388	150	0.5	9346	294	220	140
5691	738	678	2693	1	7537	954	1267	242
3506	179	275	6040	1.25	8203	542	994	261
1181	184	116	8519	1.5	7818	767	775	640
2	98	103	9797	1.75	4685	270	422	4623
97	114	91	9698	2	1820	217	220	7743
0	100	92	9808	2.5	0	77	117	9806
2	96	94	9808	2.95	0	90	114	9796

WeB2003 Paper: Holt's Cournot Game

- $\pi(x, y) = (12 - 0.5(x + y))x$ and similaryly for $\pi(y, x)$
- Competitive outcome: $x + y = 12/0.5 = 24$, 12 each for a profit each of 0.
- Monopoly outcome: $x + y = 12$, 6 each for a profit each of 36.
- Cournot/Nash outcome: $x + y = (2 \cdot 12)/(3 \cdot 0.5) = 16$, 8 each for a profit each of 32.
- Holt's findings: human subjects produce slightly less than 8 each on average.

Our Agents in the Holt/Cournot Supergame



Atomic versus Molecular Strategies

- Heretofore (us and others): agents only learn strategies for the stage game. Atomic strategies.
- These are m-0, memory of 0, strategies.
- What if they learned strategies defined over more than one subgame? Molecular strategies.
- We looked at m-1, memory of 1 previous subgame, molecular strategies.

Molecular Learning Regime in 2×2 Games

1. Alternative (or consideration) set of strategies: $\mathcal{A} = \{00, 01, 10, 11\}$ for each player. Form: ab: play a if last time counter-player played 0, play b if last time counter-player played 1. E.g., in Prisoner's Dilemma, 01 is TIT FOR TAT.

   All else the same:

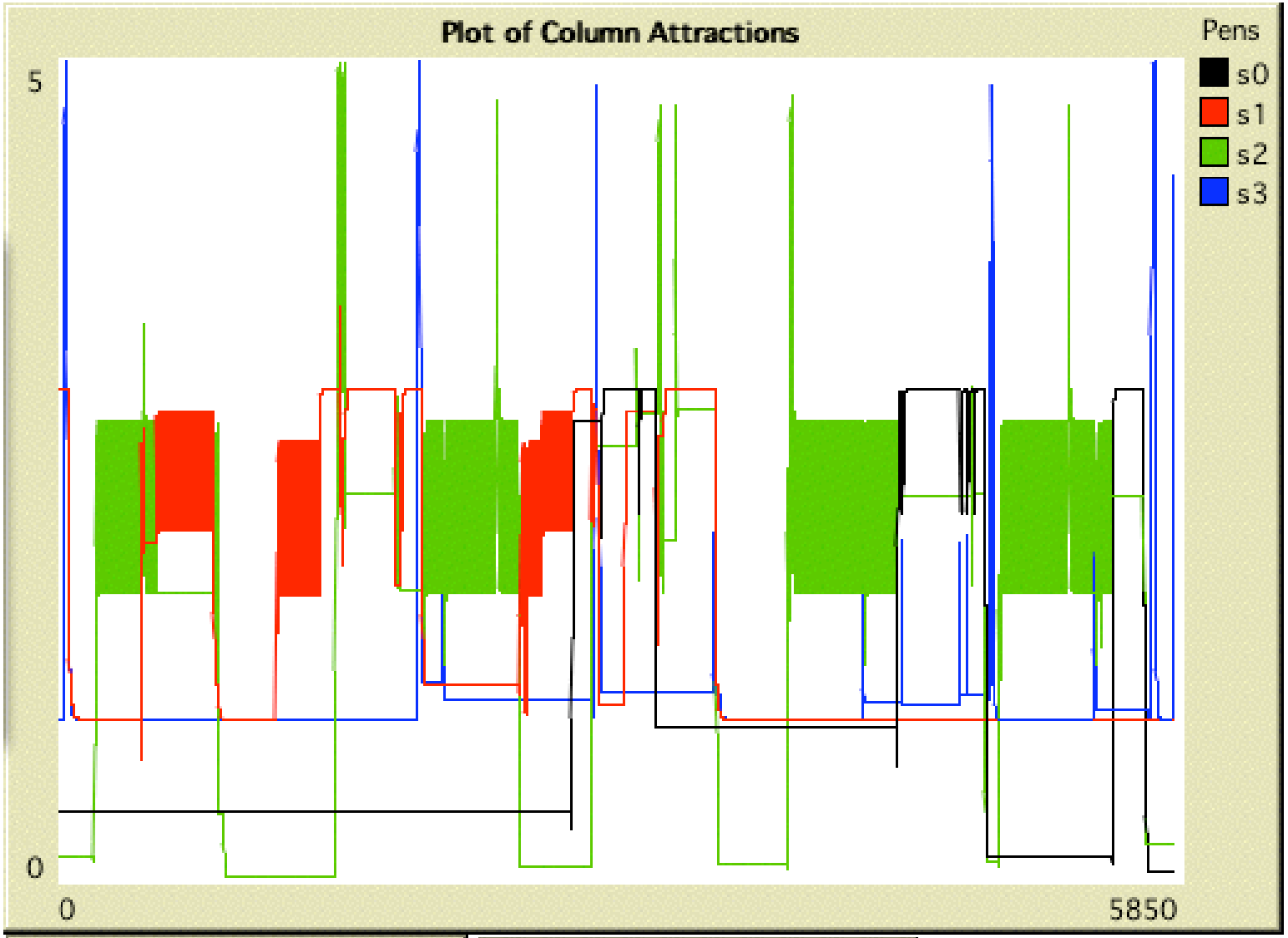
2. Attractiveness estimation: linear updating rule for $A^i, \ \ i \in \mathcal{A}$:
   $A^i_{t+1} = A^i_t + \alpha\{r^i_t - A^i_t\}$

   NewEstimate = CurrentEstimate + StepSize{reward - CurrentEstimate}

3. Choice/exploration policies. Softmax. ϵ–greedy

Example: Standard Prisoner's Dilemma

	D	C
D	1,1	5,0
C	0,5	3,3


Plot of Column Attractions
Pens
s0
s1
s2
s3
5
0
0
5850
Row Points Per
2.197
Col Points Per
2.255

Essential Findings on 2×2 Games

As before, but the results are not very sensitive to the actual payoffs.

- Agents tend to find Pareto outcomes, even at the expense of Nash outcomes, in terms of the stage game. (E.g. in Prisoner's Dilemma, chicken)
- When Nash and Pareto outcomes coincide and multiple Nash, agents tend to find Pareto-optimal Nash outcomes. (E.g., Stag Hunt)
- Actual payoffs do matter (in contravention to classical game theory)
- In any event, players tend to extract much more wealth than would otherwise be predicted.

4 Molecular Strategies in Holt's Cournot Game

1. G-TFT. If $y_{t-1} > y_{t-2}$, then $x_t = x_{t-1} + \delta$; else $x_t = x_{t-1} - \delta$

2. BESTRESPONSE. $x_t = 12 - 0.5y_{t-1}$.

3. S-TFT.

   (a) If $x_{t-1} < y_{t-1}$ and $y_{t-2} \leq y_{t-1}$, then $x_t = x_{t-1} + \delta$.
   (b) If $x_{t-1} > y_{t-1}$ or $x_{t-2} = x_{t-1} = y_{t-1} = y_{t-2}$, then $x_t = x_{t-1} - \delta$.
   (c) Else, $x_t = x_{t-1}$.

4. COPYCAT. $x_t = y_{t-1}$.

Paired Strategies: Profits

	G-TFT	BESTRESPONSE	S-TFT	COPYCAT
G-TFT	(36,36)	(33.138,28.165)	(36,36)	(36,36)
BR	(28.165, 33.138)	(32,32)	(32,32)	(32,32)
S-TFT	(36,36)	(32,32)	(36,36)	(36,36)
CC	(36,36)	(32,32)	(36,36)	(23.166,23.166)

Play with all 4 Strategies

What will happen if the agents engage all four strategies simultaneously, under the above learning regime?

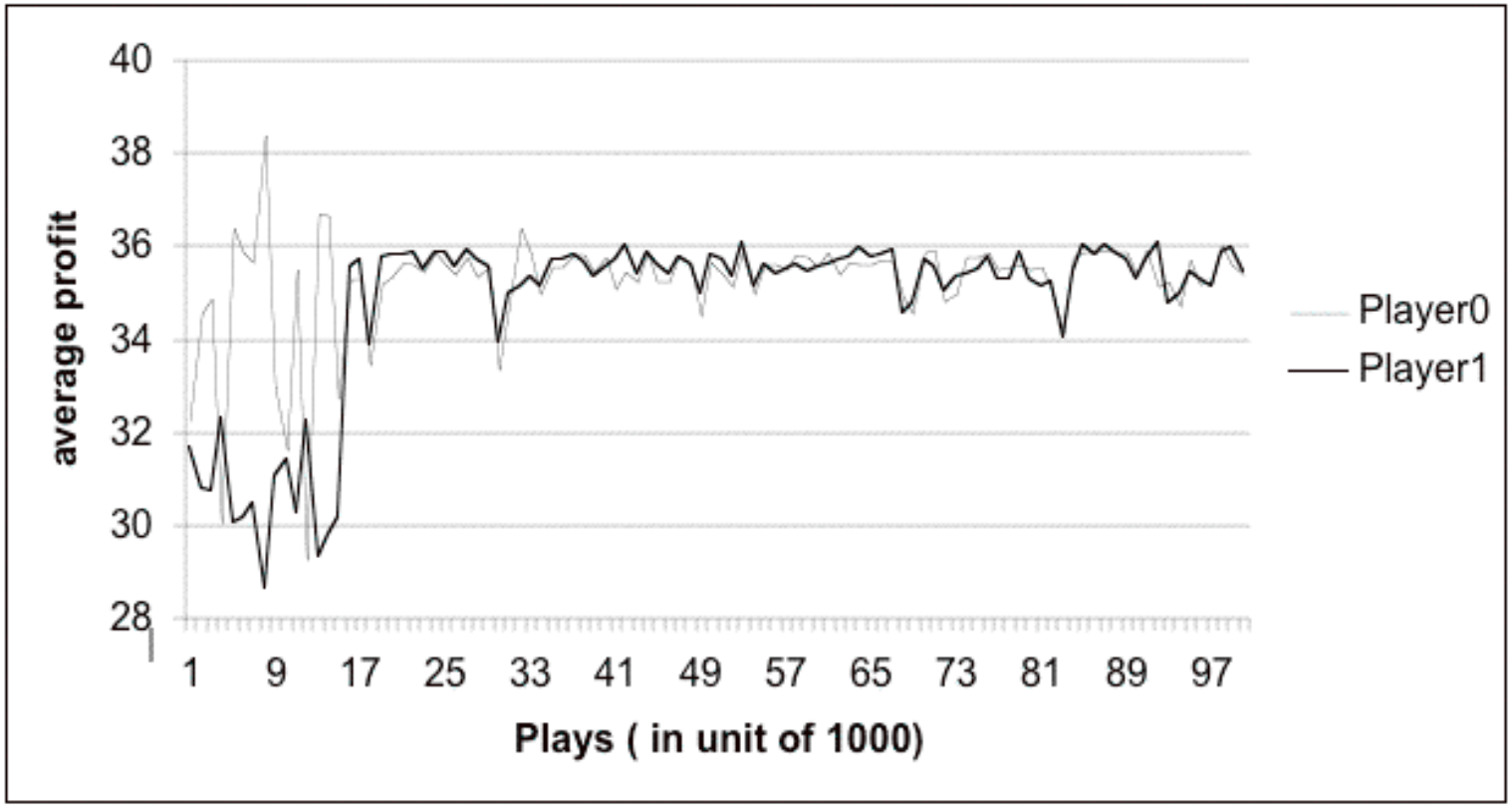
$\mathcal{A} = \{\text{G-TFT}, \text{BR}, \text{S-TFT}, \text{CC}\}$

Results over 100 runs of 100,000 (rounds) plays, averaged over the last 1000 rounds of play.

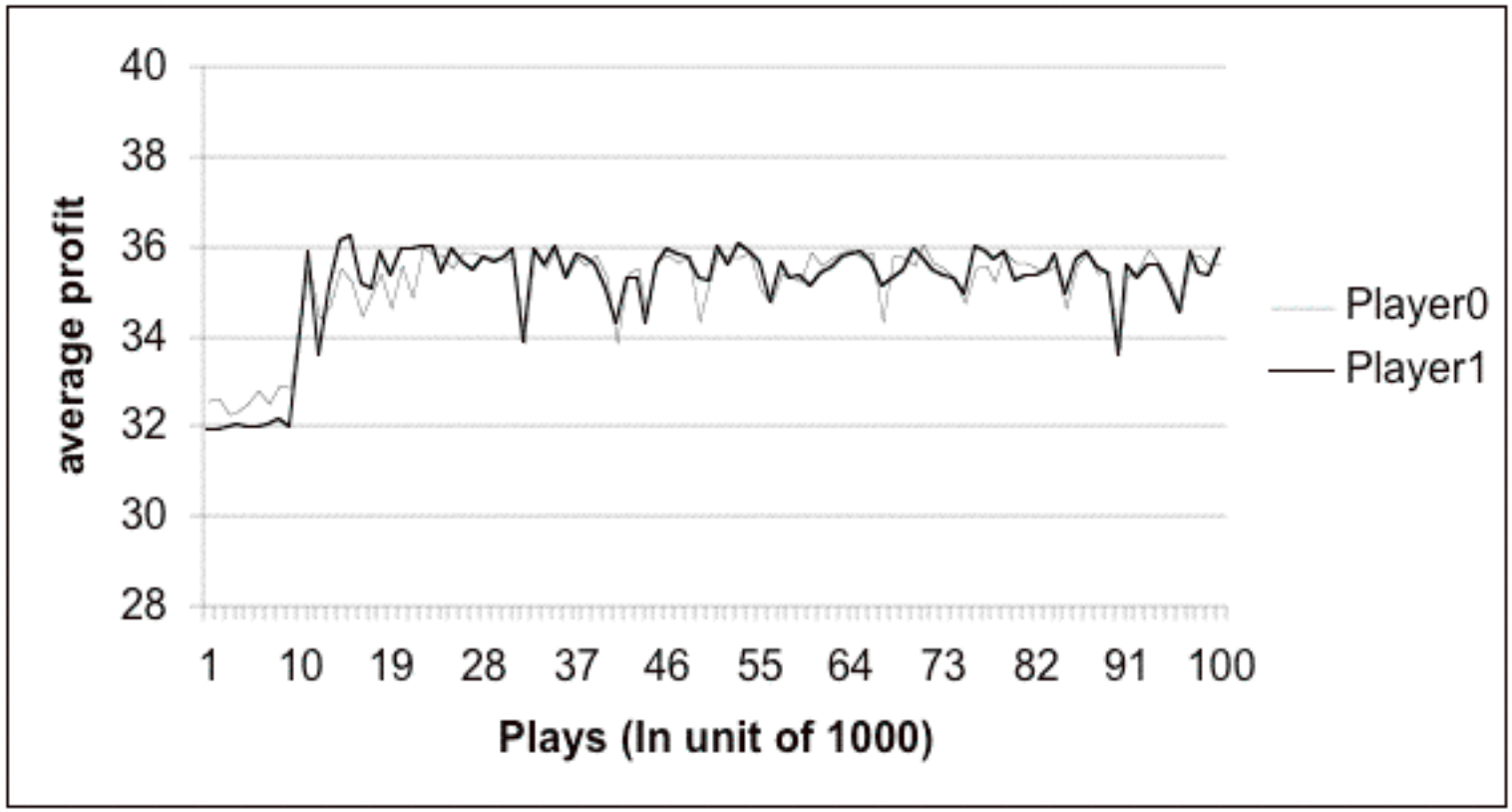
	Average Profit
Softmax action selection	(35.348, 35.323)
ϵ–greedy action selection	(35.487, 35.507)

Two typical runs follow. . .

Molecular Strategies with Holt's Cournot Game: A



Molecular Strategies with Holt's Cournot Game: B



From recent paper by Kimbrough, Lu, and Murphy

"Learning and Tacit Collusion by Artificial Agents in Cournot Duopoly Games."

	Average Profit
Softmax strategy selection	(35.243, 35.255)
$\varepsilon-$greedy strategy selection	(35.172, 35.20)

Table 2: Profit realized by selection policy over the last 1000 of 100,000 plays, averaged over 100 runs; $\gamma = \mathbf{0}$; strategies 0–4 available

Conclusion (almost)

- Major reward from algorithmic game theory studies: results from molecular strategies.

- Have to ask: If our rather dumb agents can figure out how to get monopoly profits in a Cournot game, why believe the Cournot analysis?

- Much remains to be done. Exciting stuff. See update of this last work in paper by Kimbrough, Lu, and Murphy, “Learning and Tacit Collusion by Artificial Agents in Cournot Duopoly Games.”

One more thing

- Not unrelated: metaheuristics for solving constrained optimization problems
- Mention in particular: two-population GA for constrained optimization (by yours truly).
- Impressive results so far.
- Example. . .

Test Problem: Yuan

Objective function

$$\min_{\mathbf{x},\mathbf{y}} z = (y_1-1)^2+(y_2-2)^2+(y_3-1)^2 - \ln(y_4+1)+(x_1-2)^2+(x_2-2)^2+(x_3-3)^2 \quad (1)$$

Constraints:

$$y_1+y_2+y_3+x_1+x_2+x_3 \leq 5 \quad (2)$$

$$y_3^2+x_1^2+x_2^2+x_3^2 \leq 5.5 \quad (3)$$

$$y_1 + x_1 \leq 1.2 \quad (4)$$

$$y_2 + x_2 \leq 1.8 \quad (5)$$

$$y_3 + x_3 \leq 2.5 \quad (6)$$

$$y_4 + x_1 \leq 1.2 \quad (7)$$

$$y_2^2 + x_2^2 \leq 1.64 \quad (8)$$

$$y_3^2 + x_3^2 \leq 4.25 \quad (9)$$

$$y_2^2 + x_3^2 \leq 4.64 \quad (10)$$

Variable bounds

$$x_1, x_2, x_3 \geq 0 \quad (11)$$

$$y_1, y_2, y_3, y_4 \in \{0, 1\} \quad (12)$$

As reported ... $z^* = 4.5796$ at $\mathbf{x} = (0.2, 0.8, 1.908)^T$ and $\mathbf{y} = (1, 1, 0, 1)^T$.

With 2-pop GA (5,000 half-generations):

z^+= 4.579588292413069. $x_1 = 0.199998178908325$, $x_2 = 0.799999776184869$, $x_3 = 1.90787728616851$,

$y_1 = 1, y_2 = 1, y_3 = 0, y_4 = 1$.

Summary Data

Generations	Infeas	InF-out	Fea-out	z^+	InF-med z	σ^2_{z-}Fea	σ^2_{z-}InF
0–99	-0.2824	3.5400	7.3000	5.222503	7.123	2.302	6.839
900–999	-0.2005	3.4100	6.6200	4.594130	6.577	0.840	8.928
1900–1999	-0.0453	3.3100	6.4000	4.581232	9.468	1.015	7.713
2900–2999	-0.0858	3.0400	6.4800	4.579938	5.926	0.426	3.302
3900–3999	-0.0501	2.7000	6.3300	4.579845	5.103	0.251	1.775
4900–4999	-0.0126	3.2900	4.8200	4.579653	5.245	0.253	0.948

Table 3: Yuan Results: Averages over 100 generations. Infeasibility=$-1\cdot$sum of absolute violations of constraints. InF-out=number of feasible offspring from the infeasible population. Fea-out=number of infeasible offspring from the infeasible population. z^+=best solution found in the feasible population. InF-med z=median objective function value in the infeasible population. σ^2_{z-}Fea=variance of objective function values in the feasible population. σ^2_{z-}InF=variance of objective function values in the infeasible population.

Other Uses of an EP/GA

Approximations to Shadow Prices: At generation 236, there appears in the infeasible population a solution with z = 4.4700260560943 8, much lower than z^+, the best feasible solution found during the run. This infeasible solution is at: $x_1 = 0.19546290880964 6, x_2 = 0.79575224702674 6, x_3 = 1.96768190221611, y_1 = y_2 = y_4 = 1, y_3 = 0$. All the variable values in this solution are close to their corresondents in z^+, except x_3. Further, only one constraint is violated, (10), which comes in at 4.871772068, instead of 4.64.

'Model Busting,' John Miller.

That’s All Folks

```
$Id$
```