Players as Serial or Parallel Random Access Machines

(*EXPLORATORY REMARKS*)

Timothy Van Zandt

⟨tvz@insead.edu⟩
INSEAD (France)

Presentation at DIMACS Workshop on Bounded Rationality
January 31 – February 1

Outline

1. General remarks on modeling bounded rationality in games

2. General remarks on modeling players as serial or parallel RAMs.

3. Some examples.

Bounded rationality in games

(Def'n: Bounded rationality = any deviation from the standard fully rational "consistent and infinitely smart" model.)

Starting point:

- a model of bounded rationality for single-agent decision problems.

Then

- extend to multi-player games.

Typically requires a modification of, or at least careful thought about, "standard" game theory equilibrium and non-equilibrium tools.

Three approaches to bounded rationality

1. Simple behavioral rules + dynamic model of adaptation or evolution

   Examples: Evolutionary game theory, multi-agent systems.

2. Optimization with non-standard preferences.

   Examples: Intertemporal inconsistency; game theory with non-additive priors.

3. Complexity + Constrained optimal

   Examples: Modeling players as finite automata; requiring computability of strategies.

Complexity approach for single-agent decision problems

What it means to model complexity:

1. Specify the **procedures** by which decisions are made;
2. model the capabilities and resources of the agent in making the decisions.

What this introduces:

1. restricts the set of feasible decision rules or strategies;
2. overall performance of a decision procedure reflects both "material payoff" of the decision rule and a "complexity cost".

What is gained by modeling complexity

Less ad hoc than other approaches; tries to model why people are not fully rational (because it is too hard!). Can see how complexity affects behavior.

Closing the model: which decision procedure?

Ways to specify which decision procedure the player uses:

1. Ad hoc: pick one based on e.g. empirical evidence.

2. Evolution and adaptation.

3. Constrained-optimal.

4. Deciding how to decide … (but doesn't close model).

Why are economists averse to constrained-optimal approach?

Reduces the problem to constrained-optimization which is solved correctly.

What happened to the bounded rationality?

Why is this aversion misguided?

There is no logical contradiction: The modeler is solving the optimization problem, not the agent.

The bounded rationality is present in the restrictions on the set of feasible procedures, no matter how the model is closed.

Advantages to constrained optimality:

1. Characterizing constrained-optimal procedures delineates the constraint set and hence shows the effects of the complexity constraints.

2. Useful normative exercise.

3. Constrained optimality may have empirical relevance for stationary environments where selection of decision procedure is on much longer time scale than the daily use of the decision procedure.

4. **Most importantly: A good way to capture fact that decision makes are goal oriented and do try to make the best use of their limited information processing capabilities and this affects the patters of behavior and the functioning of organizations.**

And this aversion has distorted researchers away from explicitly modeling complexity.

(*Behavior economics may be the fastest way to see empirical implications of some observed behavior, but is not a good theory about why people behave the way they do and how complexity tradeoffs affect behavior.*)

Complexity in static Bayesian games

(*Note: From now on, limiting attention to computation complexity rather than communication complexity.*)

In a static Bayesian game $(N, (A_i, T_i, u_i)_{i \in N}, \mu)$, we can model complexity of computing a strategy $\sigma_i : T_i \to A_i$.

(Or in mechanism design, model the complexity of computing the outcome function.)

This uses standard "batch" or "off-line" complexity theory.

Serial complexity: mainly how long the computation takes; measures both delay (costly because?) and time as a resource.

Parallel complexity (particularly useful if "player" is an organization): resources and delay become separate.

Complexity in dynamic games

Each player faces a “real-time” or “on-line” control problem.

Captures essential feature of real life: the problem is to keep up with a changing environment, with flows of new information all the time; flows of new decisions all the time.

“Computability” takes on a new meaning. E.g., even some linear policies are not computable.

Dynamic games are inherently data rich even if there is no information about the environment: players observe and react to each others′ actions.

Some pitfalls (for both static and dynamic games)

Theorems?

How does classic asymptotic complexity theory mesh with game theory?

Finite automata

The first(?) formal model of complexity in dynamic games had two features:

1. Class of games: infinite repetition of finite stage games.

2. Computation model for each player: finite automoton.

A beautiful marriage:

- Representation of a strategy as a finite automoton ended up useful for classic “who-cares-about-complexity” study of repeated games.

- Repeated games are “just complicated enough” to make the finite automoton model interesting.

Turing machines

A strategy in an infinitely repeated game with a finite stage game turns sequences from a finite alphabet into sequences.

That is exactly what a Turing machine does (one tape in; another tape out).

Can impose Turing computability as a constraint.

Random Access Machines

But Turing machines are good for off-line computability, but awkward for real-time computation, especially with data-rich environments.

An example of a single-agent decision problem

Stochastic enviroment n i.i.d. stochastic processes, each following stationary AR(1):

$$x_{it} = \beta x_{i,t-1} + \varepsilon_{it}.$$

Normalize variance of x_{it} to 1. Then $0 < \beta < 1$ measures inversely the speed at which the environment is changing.

Reduced form decision problem:

- Let $X_t = \sum_{i=1}^{n} x_{it}$.
- Action is $A_t \in \mathbb{R}$.
- Payoff: $u(A_t, X_t) = n - (A_t - X_t)^2$.

Thus, problem is to estimate X_t in order to minimize MSE. Available data are past observations of processes.

Statistical optimality

Complexity restricts information on which decisions depend and also the functions that are computed from this information.

Can be "OK" to restrict attention to decision procedures that are **statistically optimal**: Maximize expected payoff conditional on information used.

In this case, given information set Φ:

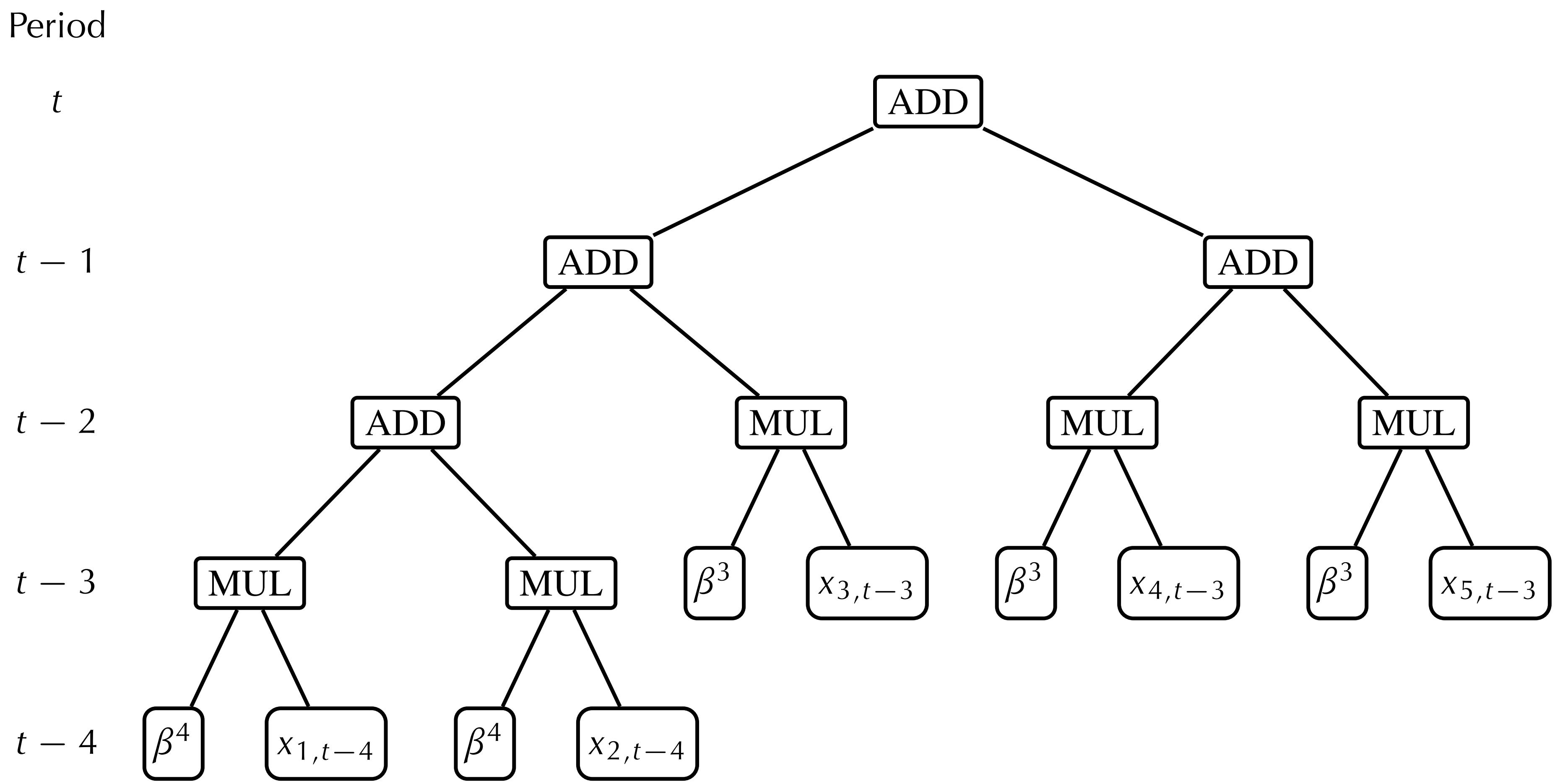
- Given quadratic loss, set $A_t = E\left[X_t \mid \Phi_t\right]$.
- Since processes are i.i.d. $E[X_t|\Phi_t] = \sum_{i=1}^n E[x_{it}|\Phi_t]$.
- Since processes are Markovian, $E[x_{it}|\Phi_t] = \beta^d x_{i,t-L_i}$, where $x_{i,t-d}$ is most recent observation of process i in Φ_t.
- $u(X_t, A_t) = \sum_{i=1}^n b^d$, where $b \equiv \beta^2$

Possible parallel procedures

Use PRAM model. Elementary operations: Add and Mul. Each takes on period.

Even if the information processing is free, there are interesting trade-offs.

An example of a procedure that uses one observation of each process for each decision.


Period
t
t − 1
t − 2
t − 3
t − 4
ADD
ADD
ADD
ADD
MUL
MUL
MUL
MUL
MUL
β^3
x_{3,t−3}
β^3
x_{4,t−3}
β^3
x_{5,t−3}
β^4
x_{1,t−4}
β^4
x_{2,t−4}

Heterogeneous lags

If T is “aggregation tree” (binary tree with leaves $\{1, \ldots, n\}$ isomorphic to subgraph of DAG corresponding to aggregation), then $L_i = 1 + \delta(i, T)$, where $\delta(i, T)$ is depth of leaf i in tree T.

$$U = b \sum_{i=1}^{n} b^{\delta(i,T)} .$$

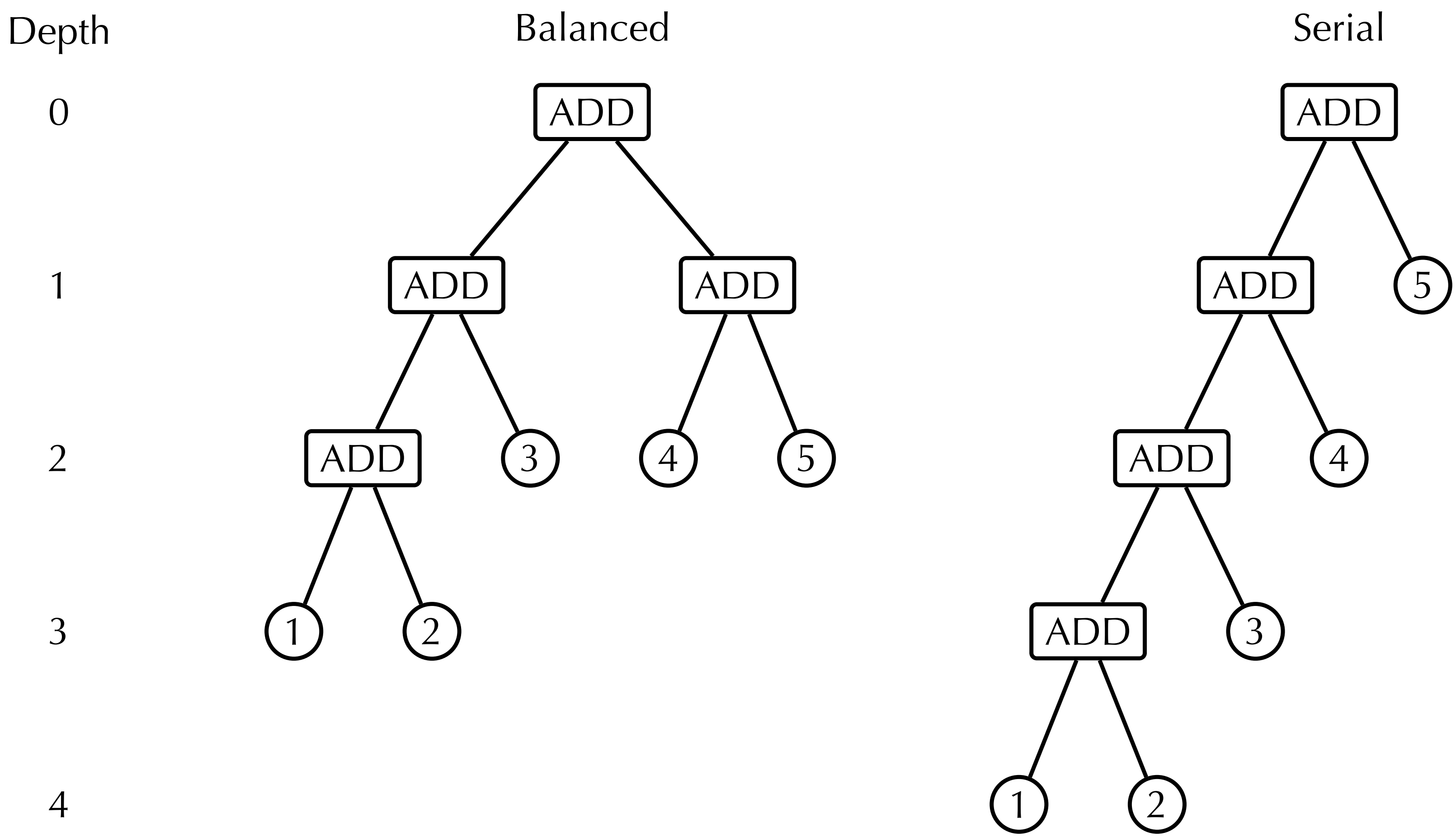
Which is the best aggregation tree?

Abstractly, if Θ is a finite set and $\mathcal{T}$ is the set of binary trees with leaves Θ, what kind of trees solve

$$\max_{T \in \mathcal{T}} \sum_{k \in \Theta} b^{\delta(k,T)} \ ?$$

Two aggregation trees for four data.

If $b < 1/2$, then serial trees; if $b > 1/2$, then balanced tree.



A simple Cournot duopoly game

Linear demand with stochastic intercept: $p(q) = z - q$.

Zero marginal cost; zero computation cost.

Firm i maximizes $E[(z - q_j - q_i)q_i]$.

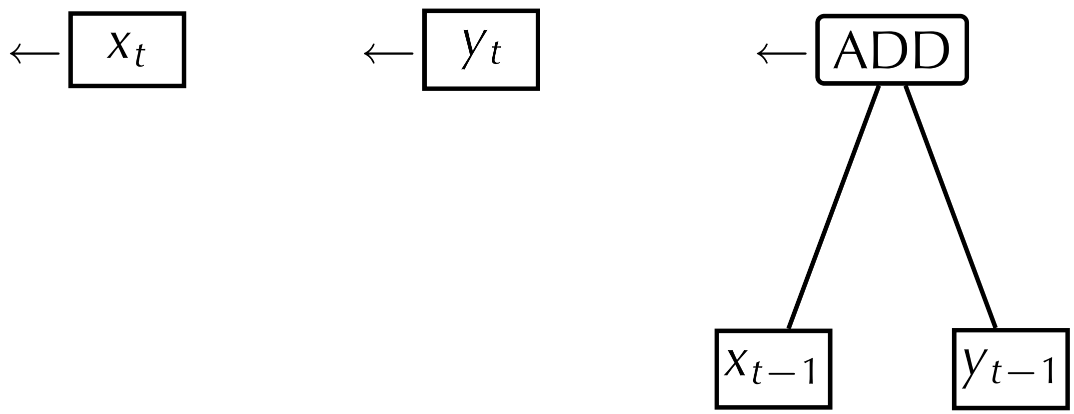
Needs to predict $z - q_j$.

Suppose $z = x + y$, where x and y are AR(1) stochastic processes.

Linear computation in which multiplication by constants is free and addition takes one period (so that it takes one period to aggregate two data as opposed to just using one datum): elementary operation is to compute $\alpha_1 w_1 + \alpha_2 w_2$, where α_1, α_2 are constants and w_1, w_2 are data.

Processes are Markovian; no need to use more than one observation of each process.

Leaves three possible information sets:



Fixing information sets, equilibria are linear.

Classic result for such Cournot competition: firms prefer to have same information in order to track each others actions.

Nash equilibria:

- firms always choose same information sets; perhaps multiple equilibria.
- if environment is changing very quickly, only equilibria are that firms use one datum;
- if environment is changing very slowly, only equilibrium is that firms use two data.

Introduce a costly ability to speed up information processing. Such decisions are strategic complements. If one firm's cost function decreases, equilibrium investments in speed-up of information processing increase for both firms.

A filtering problem (single-agent)

Stochastic environment: scalar process $\{x_t\}_{t=0}^{\infty}$.

Filtering problem: calculate an estimate $\hat{x}_t$ of x_t for all t.

Performance inversely measured by MSE: $L_t \equiv E\left[(\hat{x}_t - x_t)^2\right]$.

Decision procedures: Estimate calculated from data about the environment by a managerial staff whose cost in period t is denoted W_t.

The total cost $C_t \equiv L_t + W_t$.

Stochastic environment

Random walk:

$$x_{t+1} = x_t + v_t$$

Innovation v_t: mean 0 and variance σ_v^2 (="volatility").

Data: Samples of imperfect observations:

$$y_{it} = x_t + \varepsilon_{it}$$

Measurement error ε_{it}: mean 0 and variance σ_ε^2.
(Uncorrelated with ε_{ks} and $\{x_t\}$.)

Limiting case: unconditional variance of $x_t \to \infty$.

Decision procedures (the abstract overview)

Decision procedure: specificies policy and how it is computed.

Parameterized classes of cyclic procedures. Parameters:

- n = sample size
- d = delay between sample and when incorporated into decisions
- k = sampling frequency

Each decision procedure results in long-run average …

- loss $\mathscr{L}(n, d, k)$
- managerial cost $\mathscr{W}(n, d, k)$
- total cost $\mathscr{C}(n, d, k) = \mathscr{L}(n, d, k) + \mathscr{W}(n, d, k)$

Policies

Restrict attention to linear estimators $\hat{x}_t = \hat{E}\left[x_t \mid \Phi_t\right]$ because:

1. otherwise set of decision procedures is too vast and complicated
2. computation may then have recognizable structure
3. nice formulae for estimates (eg Kalman filter) and resulting MSE.

Sampling policies:

- Gather sample $\phi_t = \{y_{1t}, y_{2t}, \ldots, y_{nt}\}$ of size n every k periods
- Compute an estimate from each sample in d periods
- Use to update old estimate.
- Use estimate for k periods

Benchmark of contemporaneous estimation

$\hat{x}_t$ equals projection of x_t on last sample from which computed **and on all preceding samples**. Requires only a simple updating rule (Kalman filter).

Benchmark of contemporaneous estimation.

t = a period in which a sample is taken

ϕ_t^n = sample

ϕ_t = all previous samples

$\hat{x}_t^n = \hat{E}\left[x_t \,|\, \phi_t\right]$ and $\hat{x}_t^p = \hat{E}\left[x_t \;\middle|\; \phi_t^p\right]$

Σ^n and Σ^p = respective MSEs of these estimates.

$$\hat{E}\left[x_t \;\middle|\; \phi_t, \phi_t^p\right] = \frac{\Sigma^n}{\Sigma^p + \Sigma^n}\hat{x}_t^p + \frac{\Sigma^p}{\Sigma^p + \Sigma^n}\hat{x}_t^n\ ,$$

MSE = $(\Sigma^p \Sigma^n)/(\Sigma^p + \Sigma^n)$.

Rewrite as $\hat{x}_t = (1-\alpha)\hat{x}_{t-k} + \alpha y_t$ where $\alpha = \Sigma^p/(\Sigma^p + \Sigma^n)$.

Let Σ^* be steady-state value of MSE in period when sample is gathered.

$$\Sigma^* = (-k\sigma_v^2 + \sqrt{(k\sigma_v^2)^2 + 4k\sigma_v^2\Sigma^n})/2$$

This expresses the steady-state loss Σ^* of the zero-delay policy **with** recall as a function of the loss Σ^n of the zero-delay policy **without** recall

Adding in delay and time between samples:

$$\begin{aligned}
&\mathscr{L}^{\mathrm{r}}(n,d,k) \\
&\equiv \frac{1}{k}\sum_{j=0}^{k-1}\left(\left(-k\sigma_v^2 + \sqrt{(k\sigma_v^2)^2 + 4k\sigma_v^2\sigma_\varepsilon^2/n}\right)/2 + (d+j)\sigma_v^2\right) \\
&= \left(\sqrt{(k^2/4 + k\sigma_\varepsilon^2/(n\sigma_v^2)} + d - 1/2\right)\sigma_v^2 .
\end{aligned}$$

Some results

Proposition Properties of $\mathscr{C}^{\mathrm{nr}}(n, k)$ **without recall (updating)**:

- submodular in (n, k)
- decreasing differences in (n, k) and σ_ε^2
- increasing differences in (n, k) and σ_v^2.

$\Rightarrow$ optimal (n, k) is increasing in σ_ε^2 and decreasing in σ_v^2.

Similar properties with recall and $w = 0$.

An in all cases: mapping from environmental volatility to managerial size n/k has inverted U shape.

An example of a game with filtering

1. Same statistical filtering problem.

2. But x_t is the intercept of inverse demand curve in a Cournot duopoly.

Once again, all Nash equilibria are symmetric. Comparative statics for the previous single-agent problem (which is similar to but not same as the monopoly problem) hold for the equilibria of this game. E.g., beyond a threshold, as environment changes more quickly, in equilibrium both firms sample more frequently and use smaller samples.

Numerical tests indicate: If one firm's processing speed improves, then it processes more data but with smaller managerial size; other firm processes less data and also ends up with smaller managerial size.

Big picture

"Classic" computation models + dynamic games $\Longrightarrow$ interesting real-time learning in games.

Simplest (but interesting) game: stationary ergodic stochastic environment; stationary or cyclic decision procedures. Problem of keeping up with a changing environment.

Theorems?

1. Probably some interesting asymptotic results for abstract game theory;

2. Immediate results for applied games using reduced forms.