Dynamic Discrete Choice Models with 2–Period Finite Dependence

Yu Hao[1], Hiroyuki Kasahara[2], Katsumi Shimotsu[3]

[1] University of Hong Kong · [2] UBC · [3] University of Tokyo

Dec 2025 · DSE Conference Hong Kong

Motivation: Computational Burden

Nested Fixed Point (NFXP) Algorithm (Rust, 1987):

- Outer loop: maximize log-likelihood over parameters θ
- Inner loop: solve for value function $V(x)$ for each candidate θ
- Problem: When state space is large, repeatedly solving Bellman equation is expensive

$$V(x) = \int \max_{d} \left[u_\theta(x,d) + \varepsilon(d) + \beta \sum_{x'} V(x') f(x'|x,d) \right] g(\varepsilon)$$

Solution: Use Finite Dependence to avoid repeated DP solutions

The Breakthrough: Arcidiacono & Miller (2011)

Arcidiacono & Miller discovered that **many structural models exhibit Finite Dependence**

Key Insight: The difference in choice-specific values depends only on **finite periods , ahead**, not the full infinite horizon

$$v(x,d) - v(x,j) = \sum_{\tau=0}^{T} \beta^{\tau} [\text{flow payoffs}] + \text{remainder}$$

where remainder involves only future CCPs and payoffs, not $V(\cdot)$

This eliminates the need to solve for the full value function during estimation

Bus Engine Replacement Model (Rust, 1987)

State: Mileage x_t of bus engine

Choices:

- $d_t = 1$ (Keep): Pay maintenance cost, mileage increases
- $d_t = 2$ (Replace): Pay replacement cost, mileage resets to 0

Rust's finding: Model exhibits **1-period finite dependence**

AM19's contribution: Provided the general theoretical framework to characterize when FD holds

1–Period Finite Dependence in Bus Engine Model

Renewal Action ($d = \text{Replace}$):

$$F(\text{Replace})_{ij} = \pi_j \quad \forall i$$

The replacement resets mileage to 0 regardless of current x_t

Key Property:

$$F(\text{Replace}) \cdot V = \left(\sum_{\ell} \pi_\ell V_\ell \right) \mathbf{1} = C \cdot \mathbf{1}$$

The value function becomes a **constant vector** after replacement

Result: The difference in values collapses in 1 period:

$$v(x, \text{Keep}) - v(x, \text{Replace}) = [u(\text{Keep}) - u(\text{Replace})] + \beta \sum_{x'} [V(x') - C] f(x'|x, \text{Keep})$$

Structural Characterization (AM19 Framework)

AM19's Original Definition of ρ-period Finite Dependence (Eq. 3.10):

For initial choices j and j', there exist choice sequences such that:

$$\kappa^*_{t+\rho}(z_{t+\rho+1}|z_t, j) = \kappa^*_{t+\rho}(z_{t+\rho+1}|z_t, j')$$

where $\kappa^*_\tau(z_{\tau+1}|z_t, j)$ = probability of state $z_{\tau+1}$ following choice path j

Key Insight: Two different initial choices lead to **identical state distributions** after ρ periods

Result: Future value differences collapse after ρ periods:

$$v_j(z_t) - v_{j'}(z_t) = u_j(z_t) - u_{j'}(z_t) + \sum_{\tau=t+1}^{t+\rho} \beta^{\tau-t}[\cdots]$$

In Bus Engine: $\rho = 1$ (replacement resets mileage to 0 regardless of prior choice)

Dynamic Discrete Choice: Setup

- State: $x_t \in \{1, \ldots, X\}$
- Action: $d_t \in \{1, 2\}$
- Shocks: ε_t

Value:

$$V(x) = \mathbb{E}\Big[\max_d \{u(x, d) + \varepsilon(d) + \beta V(x')\} \mid x\Big],$$

CCP: $P(d \mid x) = \Pr[d \text{ chosen at } x]$.

Our Contribution: Solving for Weights

Beyond AM19's Characterization: Many models don't satisfy 1-period FD via state resets

- Entry/exit with persistent productivity shocks
- Dynamic oligopoly games with strategic interaction
- Human capital accumulation with stochastic returns

Challenge: These exhibit **2-period FD** but AM19's state distribution approach requires solving a **nonlinear bilinear system** in weights

Our Solution: Reformulate 2-period FD as a **linear least-squares problem**

- Sequential regression-based weight solving (not optimization)
- Broadens class of models exhibiting manageable finite dependence
- Maintains computational efficiency of CCP-based estimation

Hadamard Product Representation

The two-period law of motion involves the product of weights and one-step transitions for each intermediate state:

$$\kappa_{t+2}^{\boldsymbol{\omega}_{t+1}}(x_{t+2} \mid x_t, d_t) = \sum_{x_{t+1}} \sum_{k \in \mathcal{D}} \omega_{k,t+1}(x_{t+1} \mid x_t, d_t)\, f_t(x_{t+1} \mid x_t, d_t)\, f_{t+1}(x_{t+2} \mid x_{t+1}, k).$$

Stacking over x_{t+2} yields the **Hadamard form**:

$$\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}(x_t, d_t) = \sum_{k \in \mathcal{D}} \Big(\boldsymbol{\omega}_{k,t+1}(x_t, d_t) \odot \boldsymbol{f}_t(x_t, d_t)\Big)^{\top} \mathbf{F}_{t+1}(k),$$

where $\odot$ denotes elementwise (Hadamard) product.

Key insight: Using $\boldsymbol{a} \odot \boldsymbol{b} = \operatorname{diag}(\boldsymbol{a})\, \boldsymbol{b}$, the mapping becomes **linear** in the scaled vector $\boldsymbol{\omega}_{k,t+1} \odot \boldsymbol{f}_t$, enabling least-squares solution.

2–Period Dependence (AM19)

$$
\begin{aligned}
v_t(x_t, d_t) = u_t(x_t, d_t) &+ \beta\, \mathbb{E}\Big[u_{t+1}^{\boldsymbol{\omega}_{t+1}}(x_{t+1} \mid x_t, d_t) + \psi^{\boldsymbol{\omega}_{t+1}}(\boldsymbol{p}_{t+1}(x_{t+1}) \mid x_t, d_t)\Big] \\
&+ \beta^2\, \mathbb{E}\Big[u_{t+2}^{\boldsymbol{\omega}_{t+2}}(x_{t+2} \mid x_t, d_t) + \psi^{\boldsymbol{\omega}_{t+2}}(\boldsymbol{p}_{t+2}(x_{t+2}) \mid x_t, d_t)\Big] + \beta^3\, \mathbb{E}[V_{t+3}(x_{t+3})],
\end{aligned}
$$

- $\boldsymbol{\omega}_{t+1}$: weights for averaging payoffs and surplus at $t+1$
- $\boldsymbol{\omega}_{t+2}$: weights for averaging at $t+2$
- Two-period FD (Arcidiacono–Miller): holds at (x_t, k, ℓ) if there exist $(\boldsymbol{\omega}_{t+1}, \boldsymbol{\omega}_{t+2})$ such that the state distributions at $t+3$ match for both initial actions

$$
\kappa_{t+3}^{\boldsymbol{\omega}_{t+2}}(x_{t+3} \mid x_t, k) = \kappa_{t+3}^{\boldsymbol{\omega}_{t+2}}(x_{t+3} \mid x_t, \ell) \quad \forall x_{t+3} \in \mathcal{X},
$$

so that value differences depend only on weighted payoffs and CCPs over the next two periods, not on the full future value function.

Solving Weights Sequentially: Linear System

Step 1: Assume $\boldsymbol{\omega}_{t+1}$ is known

From the 1-period Hadamard representation:

$$\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}(x_t, d_t) = \sum_{k \in \mathcal{D}} \Big(\boldsymbol{\omega}_{k,t+1}(x_t, d_t) \odot \boldsymbol{f}_t(x_t, d_t)\Big)^\top \mathbf{F}_{t+1}(k).$$

This vector is now **treated as fixed data** for the next step.

Step 2: Solve for $\boldsymbol{\omega}_{t+2}$ as a function of $\boldsymbol{\omega}_{t+1}$

Substitute the known $\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}$ into the 2-period recursion:

$$\boldsymbol{\kappa}_{t+3}^{\boldsymbol{\omega}_{t+2}}(x_t, d_t)^\top = \sum_{\ell \in \mathcal{D}} \Big(\boldsymbol{\omega}_{\ell,t+2}(x_t, d_t) \odot \boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}(x_t, d_t)\Big)^\top \mathbf{F}_{t+2}(\ell).$$

Now $\boldsymbol{\omega}_{t+2}$ appears **linearly** (via Hadamard product with fixed $\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}$).

Step 3: 2-period dependence becomes a linear constraint

Two-period finite dependence requires equality across initial choices:

$$\left[\boldsymbol{\omega}_{1,t+2}(x_t,1)\odot\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}(x_t,1)-\boldsymbol{\omega}_{1,t+2}(x_t,2)\odot\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}(x_t,2)\right]^\top\tilde{\mathbf{F}}_{t+2}+\left[\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}(x_t,2)-\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}(x_t,1)\right]^\top\mathbf{F}_{t+2}(2)=0.$$

Key: This is **linear in $\boldsymbol{\omega}_{t+2}$** because $\boldsymbol{\kappa}_{t+2}^{\boldsymbol{\omega}_{t+1}}$ is fixed. Solve via least squares.

Summary: Sequential assumption → known intermediate distribution → linear system → efficient solution.

Finite Dependence: Algebraic Characterization

1-period FD (1FD)

Holds if and only if:

$$\tilde{F}F(k)\Pi_{\tilde{F}} = 0$$

where $\Pi_{\tilde{F}} \equiv I - \tilde{F}^{+}\tilde{F}$.

Intuition: The transition matrix $F(k)$ maps the null space $\mathcal{N}(\tilde{F})$ back into itself. States that have identical values remain indistinguishable after action k.

Special Case: Rank-1 Null Space ⇒ 1FD

When $\tilde{F}$ has **rank deficiency 1**, the null space $\mathcal{N}(\tilde{F})$ is one-dimensional.

The projection onto this space is:

$$\Pi_{\tilde{F}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} / X$$

(all rows identical—every state maps to the same "value-equivalent class")

Key observation: When $F(k)$ acts on this all-ones structure, it cannot create new orthogonal directions. The result $F(k)\Pi_{\tilde{F}}$ still lies in the one-dimensional null space.

Therefore: $\tilde{F}F(k)\Pi_{\tilde{F}} = 0$ automatically ✓

Examples with rank-1 null space:

- Bus engine replacement (renewal resets to one value)
- Terminal exit action (absorbs to one terminal state)
- Entry/exit with exogenous transitions
- Kronecker structure with renewal in endogenous block

2–Period Finite Dependence: Recursive Projection

Many models lack this rank deficiency structure. Yet 2FD often holds via weighted paths over two periods, making them computationally tractable without solving the Bellman equation repeatedly.

2-period FD (2FD) allows value differences to collapse over two periods via weighted action paths.

Main condition (typically satisfied in economic models):

$$\tilde{F}F(k)F(k)\Pi_{\tilde{F}}\left(I-(\tilde{F}F(k)\Pi_{\tilde{F}})^{+}(\tilde{F}F(k)\Pi_{\tilde{F}})\right)=0$$

This condition is satisfied by entry/exit games, dynamic oligopoly models, and human capital accumulation—the standard economic applications.

2FD: Geometric Intuition via Row Space

Define:

- $A \equiv \tilde{F} F(k) \Pi_{\tilde{F}}$ (1-step mapped difference)
- $B \equiv \tilde{F} F(k) F(k) \Pi_{\tilde{F}}$ (2-step mapped difference)

2FD condition:

$$B\big(I - A^{+} A\big) = 0$$

Equivalent:

$$\mathrm{Row}(B) \subseteq \mathrm{Row}(A)$$

Intuition: Walking two steps forward does NOT create new directions that $\tilde{F}$ can see. The value-relevant information at $t+2$ is already captured at $t+1$.

Case 1: Renewal Structure

Model: Maintenance firm chooses **Replace** or **Keep** (Rust 1987)

$$F(\text{Replace})_{ij} = \pi_j \quad \forall i$$

Replacement resets mileage to 0.

Result: **1FD holds ✓**

Rank-deficiency 1 in $\tilde{F} \rightarrow$ value differences annihilated in 1 period.

Case 2: Kronecker with Renewal

Structure: $F(k) = F_Y(k) \otimes F_Z$

- $F_Y(k)$: endogenous (active/exit)
- F_Z: exogenous (market shocks)

Result: **1FD holds ✓**

If F_Y has renewal, Kronecker structure preserves null space.

Examples: Entry/exit with demand shocks; dynamic oligopoly (Oblivious Equilibrium).

Case 3: Triple Kronecker (Persistent Shocks)

Structure: $F(k) = F_Y(k) \otimes F_W(k) \otimes F_Z$

- $F_W(k)$: choice-dependent AR(1) (productivity)

Result: **1FD fails ✗ | 2FD holds ✓**

Persistence prevents 1-period alignment; 2-period weights reconcile paths.

Examples: Firm dynamics with persistent productivity (Hotz & Miller 1988); schooling vs. work (Keane & Wolpin 1997).

Case 4: Block-Hadamard (Oligopoly)

Structure: Discrete state (incumbent/entrant) + continuous (productivity)

$$F(k) = \begin{pmatrix} P_{00}^{(k)} F_Z & P_{01}^{(k)} F_Z \\ P_{10}^{(k)} F_Z & P_{11}^{(k)} F_Z \end{pmatrix}$$

Result: **1FD fails ✗ | 2FD holds ✓**

Continuous states overlap; 2-period weights synchronize joint distribution.

Examples: Entry/exit dynamics (Erickson & Pakes 1995); collusion (Asker & Ljungqvist 2010).

When does this hold?

1. **Renewal** (F_Y is rank-1): A already spans everything $\to \mathrm{Row}(B) \subseteq \mathrm{Row}(A)$ trivially. ✓
2. **Kronecker with renewal**: Null space preserved by product structure. ✓
3. **Persistent AR(1) shocks**: After 1 period, distribution spreads. But after 2 periods with **weighted paths**, the conditional expectation of future value function stays within the same span. ✓
4. **Overlapping continuous states**: Discrete component resets; continuous states have common support. Two periods allow alignment without new "information directions." ✓

Counter-example: Irreversible absorbing states with divergent payoff paths (e.g., bankruptcy vs. survival) – $\mathrm{Row}(B) \not\subseteq \mathrm{Row}(A)$. ✗

Comparison: AM19 vs. Our Approach

Aspect	AM19 (2019)	Our Method (2025)
Dependence	1-period or 2-period	1-period or 2-period
Weight System	Bilinear in $(\omega_{t+1}, \omega_{t+2})$	Linear in weights
Degrees of Freedom	$4(D-1)X$ parameters	Reduced structure
Solver	Non-linear optimization (hard)	Least-squares (easy)
Broader Models	Limited scope	Covers entry/exit, games, etc.

Key Contributions

1. **Theoretical**: Generalize AM19 framework with linear weight structure
   - Clear null space conditions
   - Sequential linear solution method
2. **Computational**: Efficient weight-solving via regression
   - Replaces non-linear optimization
   - Scales to large state spaces
3. **Applicability**: Broader class of models satisfy 2-period FD
   - Entry/exit games with persistent shocks
   - Dynamic oligopoly
   - Human capital accumulation

Empirical Example: Entry–Exit Model

State: Productivity ω_t (AR(1)), Operating Status $y_t \in \{0, 1\}$

Transitions (Tauchen discretization):

- ω is persistent (like AM19's case 3)
- Operating status resets if firm exits

Structure: $F(k) = F_y(k) \otimes F_\omega(k) \otimes F_z$

Finding: 1-period FD fails (because ω is persistent)

But **2-period FD holds** via our linear weight method ✓

Optimal weights found via regression → value differences align at $t + 3$

Empirical Example: Duopoly Entry–Exit Game

State Vector: $(n_t, \omega_t^1, \omega_t^2, z_t)$ where:

- $n_t \in \{0, 1, 2\}$: Number of active firms
- ω_t^j (discretized, 5 grid points): Firm j's productivity (AR(1), **persistent**)
- z_t (2 states): Market demand shock (i.i.d.)

Actions: $a_j \in \{\text{Exit}, \text{Enter}/\text{Continue}\}$ per firm

Timing:

1. Firms observe (ω^1, ω^2, z) and choose entry/exit
2. Active firms compete (Bertrand/Cournot)
3. Productivity transitions $\omega_{t+1}^j \sim F_\omega(\omega_t^j)$, $z_{t+1} \sim F_z$ (i.i.d.)

Transition Matrix: Player 1's Perspective (Expected)

Expected transition given Rival (P2) follows equilibrium CCP σ_2

$$\mathcal{T}(a_1) = \mathbb{E}_{a_2 \sim \sigma_2(s)} \left[F(a_1, a_2) \right]$$

Structure for $a_1 = \text{Continue}$ (starting from Duopoly):

$$\mathcal{T}(Enter) = \begin{bmatrix} 0 & \underbrace{\text{diag}(\sigma_2^{exit}(2)) \cdot F_\omega}_{\text{Rival Stays } (2\to2)} & \underbrace{\text{diag}(\sigma_2^{stay}(2)) \cdot F_\omega}_{\text{Rival Exits } (2\to1)} \\ 0 & \underbrace{\text{diag}(\sigma_2^{exit}(1)) \cdot F_\omega}_{\text{Rival Stays } (2\to2)} & \underbrace{\text{diag}(\sigma_2^{stay}(1)) \cdot F_\omega}_{\text{Rival Exits } (2\to1)} \\ 0 & \underbrace{\text{diag}(\sigma_2^{exit}(0)) \cdot F_\omega}_{\text{Rival Stays } (2\to2)} & \underbrace{\text{diag}(\sigma_2^{stay}(0)) \cdot F_\omega}_{\text{Rival Exits } (2\to1)} \end{bmatrix} \otimes F_z \quad \mathcal{T}(Exit) = \begin{bmatrix} \underbrace{\text{diag}(\sigma_2^{exit}(2)) \cdot F_\omega}_{\text{Rival Stays } (2\to2)} & \underbrace{\text{diag}(\sigma_2^{stay}(2)) \cdot F_\omega}_{\text{Rival Exits } (2\to1)} & 0 \\ \underbrace{\text{diag}(\sigma_2^{exit}(1)) \cdot F_\omega}_{\text{Rival Stays } (2\to2)} & \underbrace{\text{diag}(\sigma_2^{stay}(1)) \cdot F_\omega}_{\text{Rival Exits } (2\to1)} & 0 \\ \underbrace{\text{diag}(\sigma_2^{exit}(0)) \cdot F_\omega}_{\text{Rival Stays } (2\to2)} & \underbrace{\text{diag}(\sigma_2^{stay}(0)) \cdot F_\omega}_{\text{Rival Exits } (2\to1)} & 0 \end{bmatrix} \otimes F_z$$

- **The Hadamard (Element-wise) Twist:**
  The transition matrix is not a simple Kronecker product—each block is weighted by the rival's **Conditional Choice Probabilities (CCPs)**, $\sigma_2(\omega)$.
- **Components:**
  - F_ω: Exogenous AR(1) productivity transition (persistence $\rho \approx 0.95$)
  - σ_2^{stay}: Vector of probabilities that the rival stays, given state ω
  - **Structure:** *Block-Hadamard form*. The rival's decision induces state-dependent weighting of the exogenous transition.

Testing 1-Period FD: Why It Fails

Claim: 1-period FD **does NOT hold** for this oligopoly game.

Definition: 1-period FD requires that $\text{rank}(\mathcal{T}(a) - \mathcal{T}(a'))$ equals the number of transient states.

Why it fails here:

- Compare Action A (Player 1 **Continues**) vs. Action B (Player 1 **Exits**)
- The productivity evolution F_ω is persistent and present in both actions.
- Even with the rival's CCP weighting, the underlying AR(1) kernel F_ω is **rank-deficient** relative to the state space size.
- **Result:** Residual matrix ≈ 0.15–0.20 (non-zero) → The value differences cannot be spanned by 1-period variation.

Finite Dependence: Model Taxonomy

Case	Structure	1FD	2FD	Example
Renewal	$F_Y(k)$ rank-1 reset	✓	✓	Bus engine, job search exit
Kronecker	$F_Y(k) \otimes F_Z$, renewal in Y	✓	✓	Entry/exit + exog. shocks
Triple Kronecker	$F_Y \otimes F_W(k) \otimes F_Z$, W AR(1)	✗	✓	Persistent productivity + exit
Block-Hadamard	Oligopolistic Game	✗	✓	Entry/exit

Key: Cases 1–2 have 1FD automatically. Cases 3–4 need 2FD weights.

The Counter–Example Setup

Finding: Among 45,000 randomly generated stochastic matrices, ~0.004% violate 2-period finite dependence (2FD).

Economic Context: Structural conflict between **irreversible absorption** (bankruptcy) and **safe-haven dynamics** (protected reorganization).

Transition Matrices: Action 1 vs Action 2

$$F(1) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0.3946 & 0.5127 & 0.0926 & 0 \\ 0 & 0 & 0.4771 & 0.5229 \end{bmatrix} \quad F(2) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.192 & 0.3339 & 0 & 0.4742 \end{bmatrix}$$

Action 1: Gamble for Resurrection

- **Row 1 (Safe Haven):** Deterministic transition $[0, 0, 0, 1]$ to State 4 (Operational)—**0% bankruptcy risk**.
- **Row 2 (Liquidation):** Absorbing state—once entered, firm exits permanently.
- **Row 3 (Crisis→Decision Node):** High liquidation risk (**51.27%** to Row 2); survivors funneled to Row 1 (Safe Haven) with **39.46%** probability.

Action 2: Conservative Restructuring

- **Row 1 (Safe Haven):** Deterministic transition $[0, 0, 0, 1]$ to State 4 (Operational)—**0% bankruptcy risk** (identical to Action 1).
- **Row 2 (Liquidation):** Absorbing state—once entered, firm exits permanently.
- **Row 3 (Crisis→Decision Node):** Lower liquidation risk (**25%** to Row 2); survivors distributed uniformly across all states (25% each).

Thank You!

Questions?

Contact:

- Yu Hao: haoyu@hku.hk
- Hiroyuki Kasahara: hkasahar@mail.ubc.ca
- Katsumi Shimotsu: shimotsu@e.u-tokyo.ac.jp

Building on AM19's foundation with efficient linear weight solutions ✓