Estimation of Dynamic Discrete Choice Models in Continuous Time*

Peter Arcidiacono[1], Patrick Bayer[1],
Jason R. Blevins[2], and Paul B. Ellickson[3]

[1] *Duke University and NBER*
[2] *Duke University*
[3] *Simon Graduate School of Business, University of Rochester*

January 9, 2010

Abstract. This paper provides a method for estimating large-scale dynamic discrete choice models (in both single- and multi-agent settings) within a continuous time framework. The advantage of working in continuous time is that state changes occur sequentially, rather than simultaneously, avoiding a substantial curse of dimensionality that arises in multi-agent settings. Eliminating this computational bottleneck is the key to providing a seamless link between estimating the model and performing post-estimation counterfactuals. While recently developed two-step estimation techniques have made it possible to estimate large-scale problems, solving for equilibria remains computationally challenging. In many cases, the models that applied researchers estimate do not match the models that are then used to perform counterfactuals. By modeling decisions in continuous time, we are able to take advantage of the recent advances in estimation while preserving a tight link between estimation and policy experiments. We also consider estimation in situations with imperfectly sampled data, such as when we do not observe the decision not to move, or when data is aggregated over time, such as when only discrete-time data are available at regularly spaced intervals. We illustrate the power of our framework using several large-scale Monte Carlo experiments.

Keywords: dynamic discrete choice, discrete dynamic games, continuous time.

JEL Classification: C13, C35, L11, L13.

1. Introduction

Empirical models of single-agent dynamic discrete choice (DDC) problems have a rich history in structural applied microeconometrics, starting with the pioneering work of Gotz and McCall (1980), Miller (1984), Wolpin (1984), Pakes (1986), and Rust (1987). These

*We thank the attendees of the 2009 Cowles Foundation conference on structural microeconomics and seminar participants at Chicago Booth and Duke for useful comments and suggestions.

methods have been used to study problems ranging from the choice of college major to the optimal age of retirement. However, due to the inherent complexity of nesting multi-agent DDC problems within iterative estimation routines, these methods were long considered intractable when it came to estimating multi-agent strategic games, despite a growing number of computational methods for solving for their equilibria (Pakes and McGuire, 1994, 2001; Doraszelski and Satterthwaite, 2007). Recently, in a parallel series of papers, Aguirregabiria and Mira (2007), Bajari, Benkard, and Levin (2007), Pesendorfer and Schmidt-Dengler (2007), and Pakes, Ostrovsky, and Berry (2007), have showed how to extend two-step estimation techniques, originally developed by Hotz and Miller (1993) and Hotz, Miller, Sanders, and Smith (1994) in the context of single-agent problems, to more complex multi-agent settings.[1] Ironically, the bottleneck is now on the computational side: while empirical researchers can estimate models with state spaces of ever-expanding complexity, post-estimation counterfactuals and simulations are limited by a curse of dimensionality that arises in simultaneous move games. To solve for optimal policies, players must form expectations over all combinations of actions that each of their rivals can take, a computational problem whose burden increases exponentially in the number of players. In many cases, the model that researchers estimate is far richer than what they are able to use for simulations, leading some to suggest alternatives to the Markov perfect equilibrium concept in which firms condition on long run averages (regarding rivals) instead of current information (Weintraub, Benkard, and Van Roy, 2008). The goal of this paper is exploit the sequential structure of continuous time games to break the computational curse, create a tight link between estimation and counterfactuals, and open the door to more complex and realistic models of strategic interaction.

Ours is not the first paper to tackle these computational issues. Making full use of computing resources, Pakes and McGuire (2001) extend their seminal approach to solving dynamic games (Pakes and McGuire, 1994) by replacing explicit integration with simulation and utilizing an adaptive algorithm that targets the recurrent class of states. Their computational approach is able to alleviate the curse of dimensionality that arises when calculating expectations over future states as well as the increasing size of the state space itself, but does rely on the recurrent class being small. In theoretical work that is closest to ours, Doraszelski and Judd (2008) exploit the structure of continuous time to break the curse of dimensionality associated with the calculation of expectations over rival actions. In particular, because state changes occur only one agent at a time, the dimension

[1] Two-step estimation of dynamic discrete games was originally proposed by Rust (1994). Rust recommended substituting non-parametric estimates of rivals' reaction functions into each player's dynamic optimization problem, turning a complex equilibrium solution into a collection of simpler games against nature.

of these expectations grows linearly in the number of players, rather than exponentially, resulting in computation times that are orders of magnitude faster than those of discrete time. Building on their insights, we seek to connect the computational advantages of continuous time with the empirical tractability of discrete choice models and the power of two-step estimation. To do so, we recast the dynamic discrete choice problem in discrete time as a sequential discrete choice problem in continuous time. In particular, rather than choosing simultaneous continuous actions, as in Doraszelski and Judd (2008), players in our framework make sequential, discrete decisions at random times in a random order that is determined by a collection of competing Poisson processes. Having players make discrete decisions at random times also allows for structural errors in the payoff functions.

The use of Markov jump processes to model the evolution of state variables combined with the discrete nature of the control problem results in a simple, yet flexible mathematical structure that's computationally light enough to make full solution estimation feasible for very large problems. The model also inherits many features of standard discrete choice models, so many of the insights and tools commonly used in discrete time settings, such as two-step CCP (conditional choice probability) estimation, are directly applicable within our continuous time approach as well. Notably, this flexibility allows our framework to accommodate more complex sampling schemes. For example, we show how to handle situations in which certain observations are missing (e.g. passive actions such as the decision not to invest) or where the data are only sampled at discrete intervals (e.g. quarterly or yearly). Both extensions are likely to be empirically relevant given the limitations of publicly-available datasets (most are collected at regular intervals). In the case of time aggregation, we exploit the fact that the game may be *observed* at discrete intervals but *played* in continuous time to first use computationally light two-step estimation techniques to recover the structural parameters and then re-solve the model in continuous time to perform counterfactuals or simulate data. The continuous time framework thus offers a seamless link between estimation and computation, allowing the same underlying model to be used throughout. We demonstrate the power of our approach using several large scale Monte Carlo experiments, many of which would be infeasible using previous methods.

This paper is structured as follows. Section 2 reviews some basic properties of continuous time Markov jump processes. Section 3 introduces our model in a simple single-agent context in order to build intuition. Section 4 extends the model to the multi-agent setting. Concrete and canonical examples are provided in both cases. Section 5 develops our estimators, including both full-solution and two-step approaches, and discusses issues associated with partial observation and time aggregation. Section 6 contains the results of several Monte Carlo studies relating to full-solution and two-step estimation in both

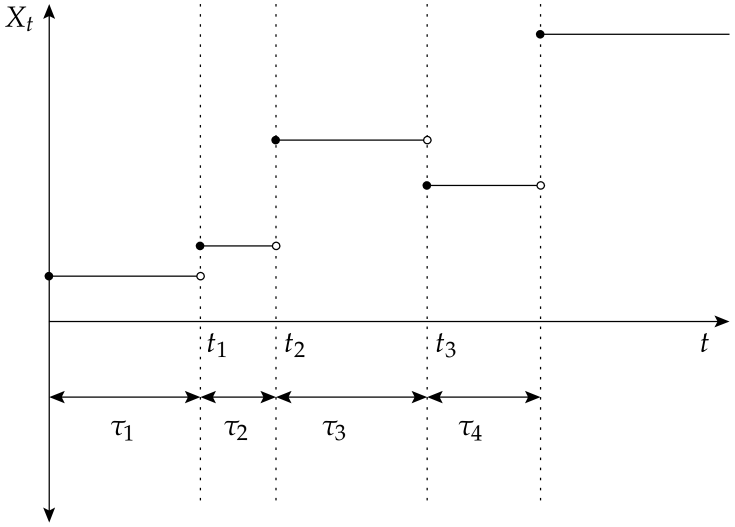

FIGURE 1. Markov jump process: a representative sample path with jumps at times t_n and inter-arrival times τ_n.

settings, including time aggregation, unobserved passive moves, and computational time comparisons. Section 7 concludes.

2. Background

The models we describe below are based on Markov jump processes and we briefly review their properties here. A Markov jump process is a stochastic process X_t indexed by $t \in [0, \infty)$ taking values in some discrete state space $\mathcal{X}$. If we begin observing this process at some arbitrary time t and state X_t, it will remain in this state for a duration of random length τ before transitioning to some other state $X_{t+\tau}$. The trajectory of such a process is a piecewise-constant, right-continuous function of time. This is illustrated in Figure 1 where a single realization x_t is plotted along with corresponding jump times t_n and inter-arrival times τ_n, where n denotes the n-th jump.

Jumps occur according to a Poisson process and the length of time between jumps is therefore exponentially distributed. The probability density function (pdf) of the Exponential distribution with rate parameter $\lambda > 0$ is

$$f(x; \lambda) = \begin{cases} \lambda \, \mathrm{e}^{-\lambda x}, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

and the cumulative distribution function (cdf) is

$$F(x;\lambda) = \begin{cases} 1 - \mathrm{e}^{-\lambda x}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

The mean is $1/\lambda$, the inverse of the rate parameter or frequency, and the variance is $1/\lambda^2$.

We consider stationary processes with finite state spaces $\mathcal{X} = \{1, \ldots, K\}$. Before proceeding, we first review some fundamental properties of Markov Jump Processes, presented without proof. For details see, for example, Karlin and Taylor (1975, section 4.8).

A finite Markov jump process can be summarized by it's *intensity matrix*

$$Q = \begin{bmatrix} -q_{11} & q_{12} & \cdots & q_{1K} \\ q_{21} & -q_{22} & \cdots & q_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ q_{K1} & q_{K2} & \cdots & -q_{KK} \end{bmatrix}$$

where for $i \neq j$

$$q_{ij} = \lim_{h \to 0} \frac{\Pr\left(X_{t+h} = j | X_t = i\right)}{h}$$

represents the probability per unit of time that the system transitions from i to j and

$$q_{ii} = \sum_{j \neq i} q_{ij}$$

denotes the rate at which the system transitions out of state i. Thus, transitions out of i follow an exponential distribution with rate parameter q_{ii} and, conditional on leaving state i, the system transitions to $j \neq i$ with probability

$$(1) \qquad P_{ij} = \frac{q_{ij}}{\sum_{k \neq i} q_{ik}}.$$

Finally, let $P_{ij}(t) = \Pr\left(X_{t+s} = j \,|\, X_s = i\right)$ denote the probability that the system has transitioned to state j after a period of length t given that it was initially in state i. Let $P(t) = (P_{ij}(t))$ denote the corresponding matrix of these probabilities, the *transition matrix*. $P(t)$ can be found as the unique solution to the system of ordinary differential equations

$$\begin{aligned} P'(t) &= P(t)Q, \\ P(0) &= I. \end{aligned}$$

commonly referred to as the forward equations. It follows that

$$(2) \qquad P(t) = \mathrm{e}^{tQ} = \sum_{k=0}^{\infty} \frac{(tQ)^k}{k!}.$$

In some cases, we will need to calculate $P(t)$ for estimation, but in practice we cannot calculate the the infinite sum (2) directly. For a general intensity matrix Q, we compute e^{tQ} using routines from Expokit, a Fortran package for calculating matrix exponentials (Sidje, 1998). We employ algorithms for dense intensity matrices in the case of single agent problems with small state spaces, and we use sparse matrix algorithms for more efficient computation in the case of dynamic games, which tend to have large state spaces and sparse intensity matrices.

Finally, we review some properties of the Exponential distribution which will be required for constructing the value function later. In particular, we note that if there are n competing Poisson processes (or Exponential distributions) with rates λ_i for $i = 1, \dots, n$, then distribution of the minimum wait time is Exponential with rate $\sum_{i=1}^n \lambda_i$ and, furthermore, conditional on an arrival the probability that it is due to process i is $\lambda_i / \sum_{j=1}^n \lambda_j$. These propositions are well known, but we present them here for completeness.

Proposition 1. *Suppose $\tau_i \sim \mathrm{Expo}(\lambda_i)$, for $i = 1, \dots, n$, are independent and define $\tau \equiv \min_i \tau_i$. Then*

$$\tau \sim \mathrm{Expo}(\lambda_1 + \dots + \lambda_n).$$

Proof. See Appendix A. ■

Proposition 2. *Let $\tau_1, \dots, \tau_n$ be independent random variables and let $\iota = \arg\min_i \tau_i$ be the index of the minimum. If $\tau_i \sim \mathrm{Expo}(\lambda_i)$, then*

$$\Pr(\iota = i) = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}.$$

Proof. See Appendix A. ■

These two propositions allow us to treat the n competing Poisson processes $(\tau_1, \dots, \tau_n)$ as a single joint process (τ, ι) where the joint distribution is given above.

3. Single-Agent Dynamic Discrete Choice Models

Consider a single agent decision problem in which time is a continuous variable $t \in [0, \infty)$. The state of the model at any time t can be summarized by a member x of some finite state space $\mathcal{X}$. Two competing Poisson processes drive the dynamics of the model. First, a continuous-time Markov jump process on $\mathcal{X}$ with intensity matrix Q_0 represents moves by nature—state changes that aren't a direct result of actions by the agent. At each time t, if a jump occurs next, the state jumps immediately to the new value. The agent may not influence this process. Second, a Poisson arrival process with rate λ governs when the agent can move. When the agent has an opportunity to move, the agent chooses an action a from the discrete choice set $\mathcal{A} = \{1, \dots, J\}$, conditional on the current state $k \in \mathcal{X}$. The set $\mathcal{A}$ contains all possible actions the agent can take when given the opportunity to move.

The agent is forward looking and discounts future payoffs at a rate ρ. While the model is in state k, the agent receives flow utility u_k. Thus, if the model remains in state k over some interval $[0, \tau)$, the present discounted value of the payoff obtained over this period from the perspective of time 0 is $\int_0^\tau \mathrm{e}^{-\rho t} u_k \, dt$.

Upon receiving a move arrival when the current state is k, the agent chooses an action $j \in \mathcal{A}$. The agent then receives an instantaneous payoff $\psi_{jk} + \varepsilon_j$ associated with making choice j in state k, where ε_j is a choice-specific payoff shock that is iid over time and across choices. Let σ_{jk} denote the probability that the agent optimally chooses choice j in state k. Let v_{jk} denote the continuation value received by the agent after making choice j in state k. In most cases, v_{jk} will consist of a particular element of the value function, for example, if the state is unchanged after the action then we might have $v_{jk} = V_k$, where V_k denotes the value function at state k (defined below). On the other hand, if there is a terminal action after which the agent is no longer active, then we might have $v_{jk} = 0$.[2]

The instantaneous payoffs ψ_{jk} represent one-time changes to the agent's utility incurred as a direct result of action j. For example, in an entry-exit model ψ_{jk} might represent the cost of entry or a scrap value accrued upon exit. We typically assume, as is standard in the discrete choice literature, that for each j, ε_j follows the standard type I extreme value distribution, or Gumbel distribution, with cdf $F(x) = \mathrm{e}^{-e^{-x}}$ and that $\varepsilon_j \perp\!\!\!\perp \varepsilon_k$ for all $k \neq j$.

A representative sample path generated by this model is shown in Figure 2. Moves by the agent are indicated by dashed lines while state changes are represented by dotted lines. Inter-arrival times are indicated by τ_{in} where i denotes the identity of the player (with $i = 0$ denoting nature) and n denotes the event number. The agent's decisions (a_{t_n}) are indicated at each decision time. For example, at time t_1, the agent chooses action 0 which does not result in a state change, while at time t_4, the agent chooses action 1 which causes the state to jump to zero.

We can now write the Bellman equation, a recursive expression for the value function V_k which gives the present discounted value of all future payoffs obtained from starting in some state k and behaving optimally in future periods. Without loss of generality, we use time 0 as the initial time. Let τ denote the time until the next event: either an exogenous

[2] There might also be uncertainty about the resulting state. In such cases we let ϕ_{jkl} denote the probability with which the model transitions to state l after the agent takes action j in state k, where for each j and k we have $\sum_{l=1}^K \phi_{jkl} = 1$. In many cases, such as an exit decision, these probabilities will be degenerate. In this notation, for example, one might express the future value term as $v_{jk} = \sum_{l=1}^K \phi_{jkl} V_l$. Since there are many possible scenarios, we use the notation v_{jk} for generality.

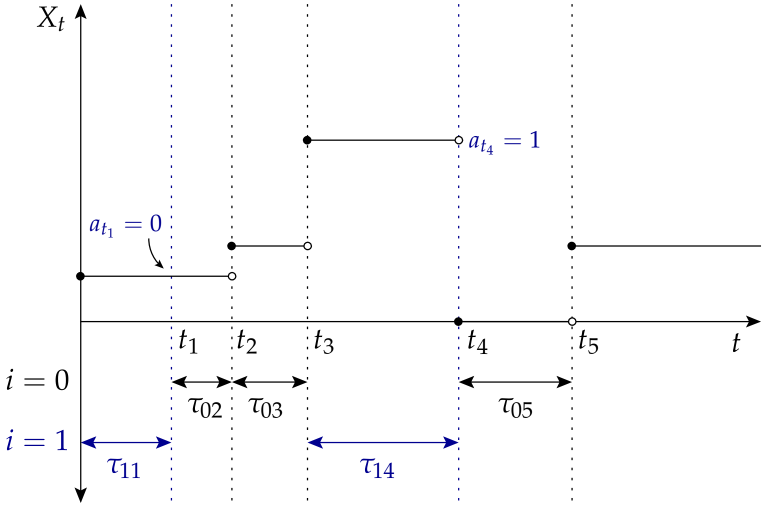

FIGURE 2. Single agent model: a representative sample path where t_n, τ_{in}, and a_{in} denote, respectively, the time, inter-arrival time, and action corresponding to n-th event. Moves by the agent are denoted by $i = 1$ while $i = 0$ denotes a state change (a move by nature).

change or an opportunity for the agent to move. In state k we have

$$V_k = \mathrm{E}\left[\int_0^\tau \mathrm{e}^{-\rho t}\, u_k\, dt + \mathrm{e}^{-\rho\tau}\,\frac{1}{\lambda + q_{kk}}\left(\sum_{l\neq k} q_{kl} v_l + \lambda \max_j \left\{\psi_{jk} + \varepsilon_j + v_{jk}\right\}\right)\right]. \tag{3}$$

Here we have used Propositions 1 and 2 to express the expectation over the joint distribution over (τ, ι) as the expectation with respect to τ of the expectation with respect to ι conditional on τ.

The first term in (3) represents the flow utility obtained in state k from the initial time until the next event (a move or jump), at time τ. The second term represents the discounted expected future value obtained from the time of the event onward, where $\lambda/(\lambda + q_{kk})$ is the probability that the event is a move opportunity and $q_{kl}/(\lambda + q_{kk})$ is the probability that the event is a jump to state $l \neq k$. The expectation is taken with respect to both τ and ε.

A policy rule is a function $\delta : \mathcal{X} \times \mathcal{E} \to \mathcal{A}$ which assigns to each state k and vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_J)$ an action from $\mathcal{A}$. The optimal policy rule satisfies the following inequality condition:

$$\delta(k, \varepsilon) = j \iff \psi_{jk} + \varepsilon_j + v_{jk} \geq \psi_{lk} + \varepsilon_l + v_{lk} \quad \forall l \in \mathcal{A}.$$

That is, when given the opportunity to choose an action, δ assigns the action that maximizes the agent's expected future discounted payoff. Thus, under the optimal policy rule, the conditional choice probabilities σ_{jk} satisfy

$$\sigma_{jk} = \Pr[\delta(k, \varepsilon) = j \,|\, k].$$

Note that the move arrival rate, λ, and the choice probabilities of the agent, σ_{jk}, also imply a Markov jump process on $\mathcal{X}$ with intensity matrix Q_1, where Q_1 is a function of both λ and σ_{jk} for all j and k. Summing the intensity matrices Q_0 and Q_1 yields the intensity matrix of the combined process. This simple and intuitive structure is especially important in extending the model to include multiple agents, and in estimation with discrete time data.

3.1. Example: A Single Agent Renewal Model

Our first example is a simple single-agent renewal model, based on the bus engine (capital) replacement problem analyzed by Rust (1987). The state space represents accumulated mileage and is indexed by the finite set $\mathcal{X} = \{1, \dots, K\}$. The agent has a binary choice set $\mathcal{A} = \{0, 1\}$, which represents the choice over whether or not to replace the engine, which resets the mileage to its baseline level. The agent faces a cost minimization problem where the flow cost incurred in mileage state k is $u_k = -\beta k$ where $\beta > 0$. The action $j = 0$ represents continuation, where the state remains unchanged, and the choice $j = 1$ causes the state to reset to $k = 1$.

The $K \times K$ intensity matrix for the jump process on $\mathcal{X}$ is

$$Q_0 = \begin{bmatrix} -q_1 - q_2 & q_1 & q_2 & 0 & \dots & 0 \\ 0 & -q_1 - q_2 & q_1 & q_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -q_1 - q_2 & q_1 & q_2 \\ 0 & 0 & \dots & 0 & -q_1 - q_2 & q_1 + q_2 \\ 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}.$$

Thus, the state can only move forward until it reaches the final state K at which point it remains there until it is reset to state 1 by the agent. For any state $1 \leq k < K - 1$ the state may jump forward either one state or two (and only one at state $K - 1$). Conditional on jumping, the probabilities of moving forward one state or two are $q_1/(q_1 + q_2)$ and $q_2/(q_1 + q_2)$ respectively.

In the notation of the general model above, the continuation values are

$$v_{jk} = \begin{cases} V_k, & \text{if } j = 0, \\ V_1, & \text{if } j = 1. \end{cases}$$

That is, when the model is in state k and the agent chooses to continue, $j = 0$, the state is unchanged and the continuation value is simply V_k. On the other hand when the agent chooses to reset the state, $j = 1$, the continuation value is V_1, the present discounted value of being in state 1. Although no cost is incurred from continuation, the agent incurs a one-time cost of c when choosing to reset the state to the initial value:

$$\psi_{jk} = \begin{cases} 0, & \text{if } j = 0, \\ -c, & \text{if } j = 1. \end{cases}$$

The value function for this model can thus be represented recursively as

$$V_k = \mathrm{E}\Bigg[\int_0^\tau \mathrm{e}^{-\rho t}\, u_k\, dt + \mathrm{e}^{-\rho\tau}\Big(\frac{q_1}{\lambda + q_1 + q_2} V_{k+1} + \frac{q_2}{\lambda + q_1 + q_2} V_{k+2} + \frac{\lambda}{\lambda + q_1 + q_2} \max\{\varepsilon_0 + V_k, -c + \varepsilon_1 + V_1\}\Big)\Bigg]$$

for $k \leq k - 2$. It is similar for $k - 1 \leq k \leq K$, with the appropriate adjustments being made at the state space boundary.

If we assume that the ε_j are iid with $\varepsilon_j \sim \text{TIEV}(0, 1)$ then we can simplify this expression further using the closed form representation of the expected future value (cf. McFadden, 1984) and the law of iterated expectations (replacing $\mathrm{E}_{\tau,\varepsilon}$ with $\mathrm{E}_\tau\, \mathrm{E}_{\varepsilon|\tau}$) to obtain:

$$\mathrm{E}\left[\max\left\{V_k + \varepsilon_0, V_1 - c + \varepsilon_1\right\}\right] = \ln\left[\exp(V_k) + \exp(V_1 - c)\right],$$

and thus,

$$V_k = \mathrm{E}\Bigg[\int_0^\tau \mathrm{e}^{-\rho t}\, u_k\, dt + \mathrm{e}^{-\rho\tau}\Big(\frac{q_1}{\lambda + q_1 + q_2} V_{k+1} + \frac{q_2}{\lambda + q_1 + q_2} V_{k+2} + \frac{\lambda}{\lambda + q_1 + q_2} \ln\left(\exp(V_k) + \exp(V_1 - c)\right)\Big)\Bigg]. \tag{4}$$

The value function summarizes the present discounted value of all future cost flows from the perspective of an arbitrary point in time, without loss of generality taken to be time 0, and at an arbitrary state $k \in \mathcal{X}$. Here, τ represents the length of time until the arrival of the next event. At each point in time, the agent makes a decision based on an expected future utility comparison, with the expectation taken with respect to the next event time τ, and ε. Inside the expectation, the first term provides the expected flow utility accumulated over the time interval $[0, \tau)$. Since the agent does not move during this time, the state evolves undeterred according the Markov jump process defined by the intensity matrix Q_0, resulting in a cost flow u_k at each instant. The second term is the present discounted value of future utility from time τ onward, after the next event occurs. At the arrival time τ, the state jumps to $k+l$, $l \in \{1,2\}$ with probability $q_l/(\lambda+q_1+q_2)$ and with probability $\lambda/(\lambda+q_1+q_2)$, the agent gets to move and makes an expected future utility maximizing choice of $j \in \{0,1\}$. The agent may choose $j=0$ and simply continue accumulating the flow cost until the next arrival, or choose or $j=1$ and reset the state to 1 by paying a cost c. The type I extreme value assumption also yields closed forms for the associated CCPs:

$$(5) \quad \sigma_{jk} = \begin{cases} \frac{\exp(V_k - V_1 + c)}{\exp(V_k - V_1 + c) + 1}, & \text{if } j = 0, \\ \frac{1}{\exp(V_k - V_1 + c) + 1}, & \text{if } j = 1. \end{cases}$$

4. Multi-Agent Dynamic Discrete Games

Extending the single-agent model of Section 3 to the case of dynamic discrete games with N players is simply a matter of modifying the intensity matrix governing the market-wide state vector to incorporate players' beliefs regarding the actions of their rivals. Following Harsanyi (1973), we treat the dynamic discrete game as a collection of single-agent games against nature, in which moves by rival agents are distributed in accordance with players' beliefs.

Suppose there are N players indexed by $i = 1, \dots, N$. The state space $\mathcal{X}$ is now a set of vectors of length N, where each component corresponds to the state of player i. Player i's discount rate is ρ_i. We shall simplify the notation later by imposing symmetry and anonymity, but for generality we index all other quantities by i, including the flow utility in state k, u_{ik}, the choice probabilities, σ_{ijk}, instantaneous payoffs, ψ_{ijk}, and transition probabilities resulting from the action of a player, ϕ_{ijkm}.

Although it is still sufficient to have only a single state jump process on $\mathcal{X}$ (with some intensity matrix Q_0) to capture moves by nature, there are now N competing Poisson processes with rates λ_i generating move arrivals for each of the N players. The next event in the model is determined by the earliest arrival of one of these $N+1$ processes.

By assuming that the iid shocks to the instantaneous payoffs are private information of the individual players, we can re-interpret the multi-agent model as a collection of games against nature, and incorporate into the intensity matrix the uncertainty about the moves of rival firms, allowing us to construct the value function for the multi-agent model in much the same way as in the single-agent case. Let τ denote the time of the next event, a state jump or a move opportunity for any player, which is the minimum of a collection of competing Poisson processes with rates given by the intensity matrix Q_0 and the move arrival rates λ_i for $i = 1, \dots, N$.

A representative sample path from a two-player game is displayed in Figure 3. Moves

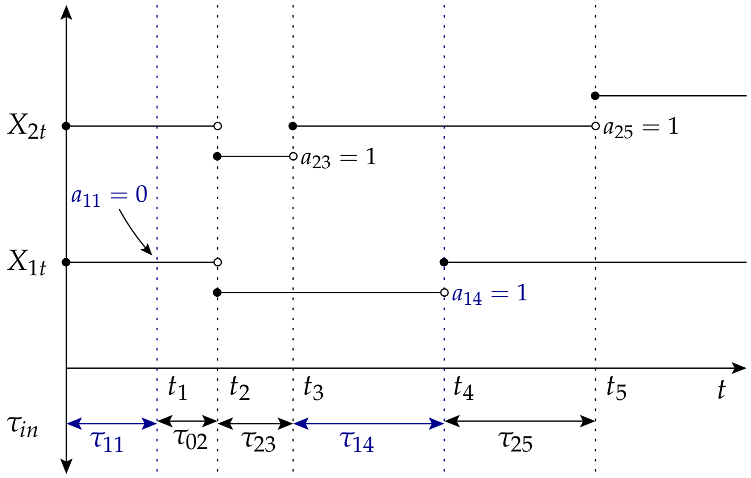

FIGURE 3. Multi-agent model: a representative sample path for two players ($i = 1, 2$) and nature ($i = 0$). Event times are denoted by t_n, inter-arrival times are denoted τ_{in}, and actions are denoted a_{in}. Here, at t_1, player 1 chooses $a_{11} = 0$ which has no effect on the state and at t_2 an exogenous state change decreases both players' states. The final three events are moves by players 1 and 2 where action 1 is chosen by both, increasing the player-specific state variables in each case.

by nature are indicated by $i = 0$. The moves and inter-arrival times at the n-th event are denoted by a_{it} and τ_{in} respectively. Here, for example, player 1 moves at time t_1 and chooses action 0 which does not change the state. The move by nature (an exogenous state change) at t_2 decreases both players' states. Player 2 then moves at t_3, choosing action 1 which increases player 2's state. Similar moves by players 1 and 2 follow at times t_4 and t_5.

Returning to the model, note that in the interval between the previous event time and τ, no other events may take place since by definition τ is the time of the *next* event. In some state k, the probability that the event is a move by player i is proportional to λ_i and the probability that the state jumps from k to $l \neq k$ is proportional to q_{kl}. The denominator of these probabilities is the sum of all of the rates involved, so that the probability that the next event in state k is a move opportunity for player i is

$$\frac{\lambda_i}{\sum_{l=1}^{N} \lambda_l + q_{kk}},$$

where $q_{kk} = \sum_{l \neq k} q_{kl}$, and the probability that the state jumps from k to m is

$$\frac{q_{km}}{\sum_{l=1}^{N} \lambda_l + q_{kk}}.$$

As before, let σ_{ijk} denote the probability that action j is chosen optimally by player i in state k. These choice probabilities are determined endogenously in the model. The continuation values are denoted v_{ijk}, and ϕ_{ijkl} denotes the probability that immediately after player i takes action j, the state jumps to another state l.

Given the above notation, the value function for player i in state k is

$$
\begin{aligned}
(6) \quad V_{ik} = \mathrm{E}\Big[\int_0^\tau \mathrm{e}^{-\rho_i t}\, u_{ik}\, dt + \mathrm{e}^{-\rho_i \tau}\, \frac{1}{\sum_{i=1}^N \lambda_i + q_{kk}} \Big(\sum_{l \neq k} q_{kl} V_{il} \\
+ \sum_{l \neq i} \lambda_l \sum_{j=1}^{J} \sigma_{ljk} \sum_{m=1}^{K} \phi_{ljkm} V_{im} + \lambda_i \max_j \left\{ \psi_{ijk} + \varepsilon_{ij} + v_{ijk} \right\} \Big) \Big].
\end{aligned}
$$

This expression is complicated only for the sake of generality. In many applications, it will be the case that the ϕ_{ljkm} terms are degenerate, with deterministic state transitions following moves. Further simplifications are also possible when players are symmetric.

A policy rule in this model is a function $\delta_i : \mathcal{X} \times \mathcal{E}_i \to \mathcal{A}_i$ which maps each state k and vector $\varepsilon_i = (\varepsilon_{i1}, \dots, \varepsilon_{iJ})$ to an action from $\mathcal{A}_i$. Given a set of beliefs σ_{ljk} for each rival $l \neq i$ about the probability that player l chooses j in state k (which enter Q_{-i}), the optimal policy rule satisfies the following condition:

$$
(7) \quad \delta_i(k, \varepsilon_i) = j \iff \psi_{ijk} + \varepsilon_{ij} + v_{ijk} \geq \psi_{ilk} + \varepsilon_{il} + v_{ilk} \quad \forall l \in \mathcal{A}_i.
$$

That is, when given the opportunity to choose an action, δ_i assigns the action that maximizes the agent's expected future discounted payoff given the specified beliefs. Then, under a given policy rule, the conditional choice probabilities of player i, σ_{ijk}, satisfy

$$
(8) \quad \sigma_{ijk} = \Pr[\delta_i(k, \varepsilon_i) = j \,|\, k].
$$

A *Markov perfect equilibrium* is a collection of policy rules $(\delta_1, \dots, \delta_N)$ and a set of beliefs $\{\sigma_{ijk} : i = 1, \dots, N;\ j = 1, \dots, J;\ k = 1, \dots, K\}$ such that both (7) and (8) hold for all i.

4.1. Example: A Quality Ladder Model

To illustrate the application to dynamic games we consider the quality ladder model of Ericson and Pakes (1995). This model is widely used in industrial organization and has been examined extensively by Pakes and McGuire (1994, 2001), Doraszelski and Satterthwaite (2007), Doraszelski and Pakes (2007), and several others. The model consists of N players who compete in a single product market. The products are differentiated in that the product of firm i has some quality level $\omega_i \in \Omega$ where $\Omega = \{1, 2, \dots, \bar{\omega}, \bar{\omega} + 1\}$ is the finite set of possible quality levels, with $\bar{\omega} + 1$ denoting the "quality" of inactive firms. Firms with $\omega_i < \bar{\omega} + 1$ are incumbents.

We consider the particular case of price competition with a single differentiated product where firms make entry, exit, and investment decisions, however, the quality ladder framework is quite general and can be easily adapted to other settings. For example, Doraszelski and Markovich (2007) use this framework in a model of advertising where, as above, firms compete in a differentiated product market by setting prices, but where the state ω_i is the share of consumers who are aware of firm i's product. Gowrisankaran (1999a) develops a model of endogenous horizontal mergers where ω_i is a capacity level and the product market stage game is Cournot with a given demand curve and cost functions that enforce capacity constraints depending on each firm's ω_i.

4.1.1. State Space Representation

We make the usual assumption that firms are symmetric and anonymous. That is, the primitives of the model are the same for each firm and only the distribution of firms across states, not the identities of those firms, is payoff-relevant. We also assume players share the same discount rate, $\rho_i = \rho$ for all i, and arrival rate, $\lambda_i = \lambda$, for all i. By imposing symmetry and anonymity, the size of the state space can be reduced from the total number of distinct market structures, $(\bar{\omega}+1)^N$, to the number of possible distributions of N firms across $\bar{\omega}+1$ states. The set of payoff-relevant states is thus the set of ordered tuples of length $\bar{\omega}+1$ whose elements sum to N:[3]

$$\mathcal{S} = \{(s_1, \dots, s_{\bar{\omega}+1}) : \sum_j s_j = N\}.$$

In this notation, each vector $\omega = (\omega_1, \dots, \omega_N) \in \Omega^N$ maps to an element $s = (s_1, \dots, s_{\bar{\omega}+1}) \in \mathcal{S}$ with $s_j = \sum_{i=1}^N 1\{\omega_i = j\}$ for each j.

In practice we map the multidimensional space $\mathcal{S}$ to an equivalent one-dimensional state space $\mathcal{X} \subset \mathbb{N}$ consisting of the integers $\{1, \dots, |\mathcal{S}|\}$. We use the same state-space encoding algorithm as Pakes and McGuire (1994) and Doraszelski and Judd (2008) to convert market structure tuples $s \in \mathcal{S}$ to integers $x \in \mathcal{X}$. The state of the market from the perspective of firm i is uniquely described by two integers (x, ω_i) where $x \in \mathcal{X}$ denotes the market structure and ω_i is firm i's own quality level. This algorithm was studied in detail by Gowrisankaran (1999b).

4.1.2. Product Market Competition

Again, we make the same product market assumptions as Pakes and McGuire (1994), namely that there is a continuum of consumers with measure $M > 0$ and that consumer j's utility from choosing the good produced by firm i is given by $g(\omega_i) - p_i + \varepsilon_i$, where ε_i

[3] This representation is space-efficient if N is large relative to $\bar{\omega}+1$. Otherwise, the algorithm used by Pakes and McGuire (1994), as described in Pakes, Gowrisankaran, and McGuire (1993), will be more parsimonious. See Gowrisankaran (1999b) for details.

is iid across firms and consumers and follows a type I extreme value distribution. Pakes and McGuire (1994) introduce the g function to enforce an upper bound on profits. As in Pakes et al. (1993), for some constant ω^* we use the function

$$g(\omega_i) = \begin{cases} \omega_i & \text{if } \omega_i \leq \omega^*, \\ \omega_i - \ln(2 - \exp(\omega^* - \omega_i)) & \text{if } \omega_i > \omega^*. \end{cases}$$

Let $\varsigma_i(\omega, p)$ denote firm i's market share given the state ω and prices p. From McFadden (1974), we know that the share of consumers purchasing good i is

$$\varsigma_i(\omega, p) = \frac{\exp(g(\omega_i) - p_i)}{1 + \sum_{j=1}^{N} \exp(g(\omega_j) - p_j)}.$$

In a market of size M, firm i's demand is $q_i(\omega, p) = M\varsigma_i$.

All firms have the same marginal cost $c \geq 0$. Taking as given the prices of other firms p_{-i}, the profit maximization problem of firm i is

$$\max_{p_i \geq 0} q_i(p, \omega)(p_i - c).$$

Caplin and Nalebuff (1991) show that there is a unique Bertrand-Nash equilibrium given by the solution to the first order conditions of the firm's problem:

$$\frac{\partial q_i}{\partial p_i}(p, \omega)(p_i - c) + q_i(p, \omega) = 0.$$

Given the functional forms above, the first order conditions become

$$-(p_j - c)(1 - \varsigma_j) + 1 = 0.$$

We solve this nonlinear system of equations numerically using the Newton-Raphson method to obtain the equilibrium prices and the implied profits $\pi(\omega_i, \omega_{-i}) = q_i(p, \omega)(p_i - c)$ earned by each firm i in each state (ω_i, ω_{-i}).

4.1.3. Incumbent Firms

We consider a simple model in which incumbent firms have three choices upon receiving a move arrival. Firms may continue without investing at no cost, they may invest an amount κ in order to increase the quality of their product from ω_i to $\omega_i' = \max\{\omega_i + 1, \bar{\omega}\}$, or they may exit the market and receive the scrap value η. We denote these choices, respectively, by the choice set $\mathcal{A}_i = \{0, 1, 2\}$. When an incumbent firm exits the market, ω_i jumps deterministically to $\bar{\omega} + 1$. Associated with each choice j is a private shock ε_{ijt}. These shocks are iid over firms, choices, and time and follow a type I extreme value distribution.

Given the future value associated with each choice, the resulting choice probabilities are defined by a logit system.

Due to the complexity of the state space, we now introduce some simplifying notation. For any market-wide state $k \in \mathcal{X}$, let $\omega_k = (\omega_{1k}, \dots, \omega_{Nk})$ denote its counterpart in Ω^N. In the general notation introduced above, the instantaneous payoff ψ_{ijk} to firm i from choosing choice j in state k is

$$\psi_{ijk} = \begin{cases} 0 & \text{if } j = 0, \\ -\kappa & \text{if } j = 1, \\ \eta & \text{if } j = 2. \end{cases}$$

Similarly, the continuation values are

$$v_{ijk} = \begin{cases} V_{ijk} & \text{if } j = 0, \\ V_{ijk'} & \text{if } j = 1, \\ 0 & \text{if } j = 2, \end{cases}$$

where state k' is the element of $\mathcal{X}$ such that $\omega_{k'i} = \max\{\omega_{ki} + 1, \bar{\omega}\}$ and $\omega_{k'j} = \omega_{kj}$ for all $j \neq i$. Note that we are considering only incumbent firms with $\omega_{ki} < \bar{\omega} + 1$.

The value function for an incumbent firm in state k is thus

$$V_{ik} = \mathrm{E}\Bigg[\int_0^\tau \mathrm{e}^{-\rho_i t}\, \pi_{ik}\, dt + \mathrm{e}^{-\rho_i \tau}\, \frac{1}{n_k\lambda + q_{kk}} \Bigg(\sum_{l \neq k} q_{kl} V_{il} + \sum_{l \neq i} \lambda \sum_{j=1}^{J} \sigma_{ljk} \sum_{m=1}^{K} \phi_{ljkm} V_{im} \\ + \lambda\, (\sigma_{i0l}(V_{ik} + \varepsilon_{i0}) + \sigma_{i1l}(V_{ik'} - \kappa + \varepsilon_{i1}) + \sigma_{i2l}(\eta + \varepsilon_{i2}))\Bigg)\Bigg]$$

where π represents the flow profit accruing from product market competition, n_k denotes the number of active incumbents and potential entrants in state k, and the expectation is with respect to τ and ε_{ij} for all i and j. Conditional upon moving while in state k, incumbent firms face the following maximization problem

$$\max\{V_{ik} + \varepsilon_{i0}, -\kappa + V_{ik'} + \varepsilon_{i1}, \eta + \varepsilon_{i2}\}$$

resulting in the corresponding choice probabilities

$$\sigma_{i0k} = \frac{\exp(V_{ik})}{\exp(V_{ik}) + \exp(-\kappa + V_{ik'}) + \exp(\eta)},$$
$$\sigma_{i1k} = \frac{\exp(-\kappa + V_{ik'})}{\exp(V_{ik}) + \exp(-\kappa + V_{ik'}) + \exp(\eta)},$$
$$\sigma_{i2k} = 1 - \sigma_{i0k} - \sigma_{i1k},$$

where, as before, k' denotes the resulting state after investment.

4.1.4. Potential Entrants

Whenever the number of active firms n is smaller than N, potential entrants receive the opportunity to enter at a rate λ. Thus, if there are n active firms the rate at which incumbents receive the opportunity to move is $n\lambda$ but the rate at which *any* type of move opportunity occurs is $(n+1)\lambda$, the additional λ being for potential entrants. If firm i is a potential entrant with the opportunity to move it has two choices: it can choose to enter ($a_i = 1$), paying a setup cost η^e and entering the market immediately in a predetermined entry state $\omega^e \in \Omega$ (we choose $\omega^e = \lfloor \frac{\bar{\omega}}{2} \rfloor$) or it can choose not to enter ($a_i = 0$) at no cost. Associated with each choice j is a stochastic private payoff shock ε^e_{ijt}. These shocks are iid across firms, choices, and time and are distributed according to the type I extreme value distribution.

In the general notation of Section 4, for entrants ($j = 1$) in state k, the instantaneous payoff is $\psi_{i1k} = -\eta^e$ and the continuation value is $v_{i1k} = V_{ik'}$ where k' is the element of $\mathcal{X}$ with $\omega_{k'i} = \omega^e$ and $\omega_{k'j} = \omega_{kj}$ for all $j \neq i$. For firms that choose not to enter ($j = 0$) in state k, we have $\psi_{ijk} = V_{i0k} = 0$. Thus, conditional upon moving in state k, a potential entrant faces the problem

$$\max\{\varepsilon^e_{i0}, -\eta^e + V_{ik'} + \varepsilon^e_{i1}\}$$

yielding the conditional entry-choice probabilities

$$\sigma_{i1k} = \frac{\exp(V_{ik'} - \eta^e)}{1 + \exp(V_{ik'} - \eta^e)}.$$

4.1.5. State Transitions

In addition to state transitions that result directly from entry, exit, or investment decisions, the overall state of the market follows a jump process where at some rate γ, the quality of each firm i jumps from ω_i to $\omega'_i = \min\{\omega_i - 1, 1\}$. This process represents an industry-wide (negative) demand shock, interpreted as an improvement in the outside alternative.

Being a discrete-time model, Pakes and McGuire (1994) assume that each period this industry-wide quality deprecation happens with some probability δ, implying that the quality of all firms falls on average every $1/\delta$ periods. Our assumption of a rate γ is also a statement about this frequency in that $1/\gamma$ is the average length of time until the outside good improves.

We construct the corresponding intensity matrix Q_0 as follows. We map each market structure s to an integer k and map the resulting structure after deprecation s' to an integer k'. The (k, k) element of Q_0 for each eligible state k is $-\gamma$ while the corresponding (k, k') element is γ. Note that player i's state can never enter or leave $\omega_i + 1$ as a result of a move by nature. This is only possible when a firm enters or exits.

5. Estimation

Methods that solve for the value function v directly and use it to obtain the implied choice probabilities for estimation are referred to as full-solution methods. The nested-fixed point (NFXP) algorithm of Rust (1987), which uses value function iteration to compute v inside of an optimization routine which maximizes the likelihood, is the classic example of a full-solution method. Su and Judd (2008) provide an alternative MPEC (mathematical program with equilibrium constraints) approach which solves the constrained optimization problem directly, bypassing the repeated solution of the dynamic programming problem.

CCP-based estimation methods, on the other hand, are two-step methods pioneered by Hotz and Miller (1993) and Hotz et al. (1994) and later extended by Aguirregabiria and Mira (2007), Bajari et al. (2007), Pesendorfer and Schmidt-Dengler (2007), Pakes et al. (2007), and Arcidiacono and Miller (2008). The CCPs are estimated in a first step and used to approximate the value function in a closed-form inversion or simulation step. The approximate value function is then used in the likelihood function to estimate the structural parameters of the model using a maximum pseudo-likelihood procedure.

Full-solution methods have the advantage that the exact CCPs are known once the value function is found—they do not have to be estimated—and thus the model can be estimated using full-information maximum likelihood. Although these methods are efficient in the statistical sense, they can become quite costly computationally for complex models with many players or a large state space. Many candidate parameter vectors must be evaluated during estimation and, if the value function is costly to compute, even if solving the model once might be feasible, doing so many times may not be. In the presence of multiple equilibria, they also require researchers to make an assumption on the equilibrium selection mechanism and solve for all the equilibria (cf. Bajari, Hong, and Ryan, 2007).[4] The Su and Judd (2008) MPEC approach provides one solution to both the computational bottleneck and the issue of multiplicity. CCP methods provide another attractive alternative, allowing the value function to be computed very quickly and the pseudo-likelihood function to condition upon the equilibrium that is played in the data.

Our model has the advantage of being estimable via either approach. It breaks one primary curse of dimensionality in that with probability one only a single player moves at any particular instant. Thus, full-solution methods are more likely to be computationally feasible with our model than with a standard discrete-time framework such as that of Ericson and Pakes (1995), as solved by Pakes and McGuire (1994, 2001). A further benefit

[4] When performing full-solution estimation in this paper, we assume that the equilibrium selection rule assigns probability one to the equilibrium obtained by our numerical fixed point routine. The computational advantages of continuous time, however, make it easier to explore more complex specifications with non-degenerate weightings.

of our model is that standard CCP methods may also be used for estimation. Finally, because it is feasible to both estimate *and* solve our model, it preserves a tight link between the estimated model and the model used for post-estimation exercises such as simulating counterfactuals.

This section is organized as follows. We begin by discussing estimation via full-solution methods with continuous time data in Section 5.1 before turning to cases where true continuous time data are not available. We consider the case when some moves may be unobserved in Section 5.2, and in Section 5.3 we consider the case where fully continuous-time observations are not available and the model is only observed at discrete intervals. Finally, we consider CCP-based estimation in Section 5.4.

5.1. Full-Solution Estimation

Consider a sample of size T consisting of a sequence of tuples $(\tau_t, i_t, a_t, x_t, x_t')$ each describing a jump or move event where, for each observation t, τ_t is the time interval since the previous event, i_t is the player index associated with this event ($i_t = 0$ indicates a move by nature), a_t is the action taken by player i_t (undefined for moves by nature), x_t denotes the state at the time of the event, and x_t' denotes the state immediately after the event.

Let $\ell_t(\theta)$ denote the likelihood of observation t given the parameters θ. Before stating the likelihood function explicitly, we must first introduce some additional notation. For some quantities, we use a slightly different notation in this section in order to make the dependence on θ explicit. Let $g(\tau;\lambda) = \lambda\,\mathrm{e}^{-\lambda\tau}$ and $G(\tau;\lambda) = 1 - \mathrm{e}^{-\lambda\tau}$ denote, respectively, the pdf and cdf of the exponential distribution with rate parameter λ. For two states x and x', let $q(x, x';\theta)$ denote the corresponding element of the intensity matrix $Q_0(\theta)$ and let $p(x, x';\theta)$ denote the corresponding transition probability, conditional on jumping, as defined in (1). Finally, let $\sigma(i_t, a_t, x_t;\theta)$ denote the conditional choice probability of player i_t taking action a_t in state x_t.

As we will see, $\ell_t(\theta)$ differs for moves and jumps. When a move is observed it provides information about the rate of move arrivals, through the density of the move arrival process $g(\tau_t;\lambda)$,[5] and the payoff parameters, through the CCPs $\sigma(i_t, a_t, x_t;\theta)$. In addition, it provides information about the rate at which jumps occur in state x_t since we observe that a jump *did not* occur over the interval of length τ_t, which happens with probability $1 - G\left(\tau_t; q(x_t, x_t;\theta)\right)$. Thus, in the case of a move, indicated by $i_t > 0$, the likelihood is

$$(9) \qquad \ell_t(\theta) = g(\tau_t;\lambda)\cdot\sigma(i_t, a_t, x_t;\theta)\cdot\left[1 - G\left(\tau_t; q(x_t, x_t;\theta)\right)\right].$$

When a jump is observed, it provides information first about the rate of jump arrivals in state x_t, through the density $g(\tau; q(x_t, x_t;\theta))$, and the state transitions themselves, through

[5] Or, for example, in a dynamic game with $n(x_t)$ active players in state x_t, the density would be $g(\tau_t; n(x_t)\lambda)$.

the conditional transition probability $p(x_t, x_t'; \theta)$. Similar to the case of moves considered above, a jump also provides information about the move arrival process with parameter λ since we know that over an interval of length τ_t we *did not* observe a move. Using the cdf, this happens with probability $1 - G(\tau; \lambda)$. Thus, the likelihood of a jump observation is

$$(10) \quad \ell_t(\theta) = g(\tau_t; q(x_t, x_t; \theta)) \cdot p(x_t, x_t'; \theta) \cdot [1 - G(\tau_t; \lambda)] .$$

Combining (9) and (10), we can write the log likelihood for the complete sample of size T as

$$\ln L_T(\theta) = \sum_{t=1}^{T} 1\{i_t > 0\} \Big[\ln g(\tau_t; \lambda) + \ln \sigma(i_t, a_t, x_t; \theta) + \ln \left[1 - G\left(\tau_t; q(x_t, x_t; \theta)\right)\right] \Big]$$
$$+ \sum_{t=1}^{T} 1\{i_t = 0\} \Big[\ln g(\tau_t; q(x_t, x_t; \theta)) + \ln p(x_t, x_t'; \theta) + \ln \left[1 - G(\tau_t; \lambda)\right] \Big].$$

5.2. Partially Observed Moves

We continue using the same notation is in the previous sections but now we suppose that the choice set is $\mathcal{A} = \{0, \dots, J-1\}$ and that only actions a_t for which $a_t > 0$ are observed by the econometrician. This complicates the estimation as now we only observe the truncated joint distribution of move arrival times and actions. Estimating λ using only the observed move times for observations with $a_t > 0$ would introduce a downward bias, corresponding to a longer average waiting time, because there could have been many unobserved moves in any interval between observed moves. Thus, in this setting τ_t is now the interval since the last *observed* event. For simplicity, we will consider only estimation of the single agent model of Section 3.

Over an interval where the state variable is constant at x_t, the choice probabilities for each action, $\sigma(a_t, x_t; \theta)$, are also constant. On this interval, conditional on receiving a move arrival, it will be observed with probability $1 - \sigma(a_t = 0, x_t; \theta)$.

For a given state x_t we can derive the likelihood of the waiting times between observed moves by starting with the underlying Poisson process generating the move arrivals. Let $N(t)$ denote the total cumulative number of move arrivals at time t and let $N_a(t)$ denote the number of move arrivals for which the agent chose action a. We will write $N_+(t)$ to denote $\sum_{a>0} N_a(t)$. We also define the waiting time before receiving a move arrival with corresponding action a, $W_a(t)$, defined as the smallest value of $\tau \geq 0$ such that $N_a(t+\tau) - N_a(t) \geq 1$. Let $W_+(t)$ and $W(t)$ be defined similarly.

By the properties of Poisson processes we know that $W(t)$, the waiting time until the next move arrival (both observed and unobserved) is independent of t and has an Exponential distribution with parameter λ. We have a similar result for $W_+(t)$. Because the probability of truncation (the probability of choosing $a = 0$) depends on x, so will the distribution of $W_+(t)$. We will derive the distribution for intervals where the state is constant which will be sufficient for the purposes of the likelihood function.

Proposition 3. *Let the state of the model be x and let $\sigma(a, x)$ denote the choice probability of action a in state x. Then $W_+(t)$ has an Exponential distribution with rate parameter $(1 - \sigma(0, x))\,\lambda$.*

Proof. See Appendix A. ■

As in (9) and (10) we have two cases for the likelihood of an observation $(\tau_t, i_t, a_t, x_t, x_t')$. For a move, when $i_t > 0$, we have

$$\ell_t(\theta) = g(\tau_t; (1 - \sigma(0, x_t))\lambda) \cdot \frac{\sigma(a_t, x_t; \theta)}{1 - \sigma(0, x_t; \theta)} \cdot [1 - G(\tau_t; q(x_t, x_t; \theta))] .$$

The only differences here are the first and second terms in which the parameter λ is now scaled to reflect the rate of *observed* move arrivals and the choice probabilities are scaled to condition on the fact that the move was actually observed. For a jump the likelihood becomes

$$\ell_t(\theta) = g(\tau_t; q(x_t, x_t; \theta)) \cdot p(x_t, x_t'; \theta) \cdot [1 - G(\tau; (1 - \sigma(0, x_t))\lambda)] .$$

Here, only one term is different: the probability of *not* having observed a move over the interval τ_t. Estimation can now proceed as usual by constructing and maximizing the log-likelihood function of the full sample.

5.3. Time Aggregation

Suppose we only observe the stochastic process X_t at n discrete points in time $\{t_1, t_2, \dots, t_n\}$. Let $\{x_1, x_2, \dots, x_n\}$ denote the corresponding states. Through the aggregate intensity matrix Q, where $Q = Q_0 + \sum_i Q_i$, these discrete-time observations provide information about the underlying state jump process as well as the rate of move arrivals and the conditional choice probabilities. We use these observations to estimate the parameters of Q_0 and the structural parameters implicit in the conditional choice probabilities σ_{ijk} which appear in Q_i for each $i = 1, \dots, N$. In this section, we describe a full-solution approach in which the value function is solved for each value of θ in order to obtain the implied CCPs to construct each Q_i.

Let $P(t)$ denote the transition probability function from (2) corresponding to the aggregate intensity matrix Q. These probabilities summarize the relevant information about a pair observations (t_{j-1}, x_{j-1}) and (t_j, x_j). That is, $P_{x_{j-1},x_j}(t_j - t_{j-1})$ is the probability of the process moving from x_{j-1} to x_j after an interval of length $t_j - t_{j-1}$. This includes cases where $x_j = x_{j-1}$ since the transition probabilities account for there having been no

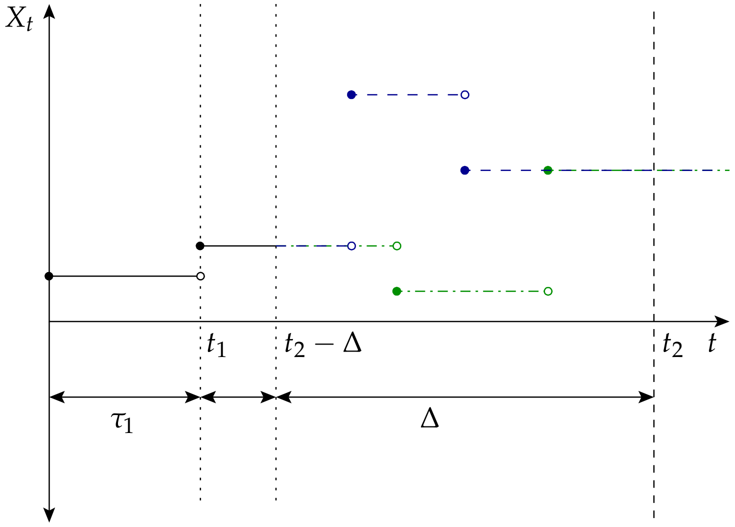

Figure 4. Time aggregation: two distinct paths which begin in the same state at $t_2 - \Delta$ and end in the same state at t_2 but differ over intermediate interval of length Δ.

jump or any of an infinite number of combinations of jumps to intermediate states before coming back to the initial state. The likelihood for a sample $\{(t_j, x_j)\}_{j=1}^n$ is thus

$$\ln L_n(\theta) = \sum_{j=1}^{n} \ln P_{x_{j-1}, x_j}(t_j - t_{j-1}).$$

To illustrate the issues involved, Figure 4 displays two distinct paths which coincide both before and after an interval of length Δ, but which take different intermediate steps. Consider the possible paths of the process between times $t_2 - \Delta$ and t_2. The dashed path first moves to a higher state before arriving at the resulting state x_{t_2}, while the dashed and dotted path first moves to a lower state and arrives in x_{t_2} at a later time (but before t_2). There are an infinite number of such paths since time is continuous, but the dynamics of the process over the interval are summarized by the transition matrix $P(\Delta)$.

To be more concrete, consider the single agent renewal model of Section 3.1 with $K = 5$ states. The intensity matrix Q_0 gives the rates at which the state changes due to nature. Suppose that the state increases one state at rate γ_1 and two states at rate γ_2. Then, Q_0 for this model is

$$Q_0 = \begin{bmatrix} -\gamma_1 - \gamma_2 & \gamma_1 & \gamma_2 & 0 & 0 \\ 0 & -\gamma_1 - \gamma_2 & \gamma_1 & \gamma_2 & 0 \\ 0 & 0 & -\gamma_1 - \gamma_2 & \gamma_1 & \gamma_2 \\ 0 & 0 & 0 & -\gamma_1 - \gamma_2 & \gamma_1 + \gamma_2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Let σ_k denote the conditional choice probability of choosing to renew—moving the state back to 1 deterministically—in state k. Note that σ_k is determined endogenously and depends on the parameters θ through the value function as in (5). Under our assumptions, σ_k will have a logit form. If λ is the rate at which moves arrive then Q_1 is

$$Q_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \lambda\sigma_2 & -\lambda\sigma_2 & 0 & 0 & 0 \\ \lambda\sigma_3 & 0 & -\lambda\sigma_3 & 0 & 0 \\ \lambda\sigma_4 & 0 & 0 & -\lambda\sigma_4 & 0 \\ \lambda\sigma_5 & 0 & 0 & 0 & -\lambda\sigma_5 \end{bmatrix}.$$

The first row contains only zeros because the model remains at state 1 regardless of which action is taken. The remaining diagonal elements are $-\lambda + \lambda(1-\sigma_k)$ where λ is the rate at which the model potentially *leaves* state k and $\lambda(1-\sigma_k)$ is the rate at which the state potentially remains unchanged yielding a net exit rate of $-\lambda\sigma_k$. The aggregate intensity matrix in this case is $Q = Q_0 + Q_1$, where the corresponding probability function $P(t)$ is used for estimation.

5.4. CCP-Based Estimation

We consider CCP estimation in terms of the single-agent model, but application to the multi-agent model follows directly and is discussed briefly in Section 5.4.1. CCP estimation relies on finding a mapping from CCPs σ_{jk} to the value function V_k. When separated at the time of the next *event*, the value function as expressed in (3) contains both terms involving V_k directly, as well as the familiar "social surplus" term which is typically used to obtain the inverse mapping. These extra terms preclude the use of the usual inverse CCP mapping. However, when the value function is separated instead at the time of the player's next *move*, the inverse mapping is straightforward.

The derivation is very similar to the next-event representation of Section 3, but we now need to consider that between any two moves by the agent, any number of other state jumps could have occurred. For example, if the model is initially in state k and no move arrival occurs on the interval $[0, \tau_1)$ while the state follows the dynamics of the underlying Markov jump process, we know that the probability of being in any state l at time $t \in [0, \tau_1)$ is $P_{kl}(t)$, where $P(t)$ are the jump probabilities associated with the intensity matrix Q_0. The total payoff obtained over $[0, \tau_1)$, discounted to the beginning of the interval, is therefore $\int_0^{\tau_1} \mathrm{e}^{-\rho t} \sum_{l=1}^{K} P_{kl}(t) u_l \, dt$.

The next-move representation of the value function in state k, is

$$\text{(11)} \quad V_k = \mathrm{E}\left[\int_0^{\tau_1} \mathrm{e}^{-\rho t} \sum_{l=1}^{K} P_{kl}(t) u_l \, dt + \mathrm{e}^{-\rho\tau_1} \sum_{l=1}^{K} P_{kl}(\tau_1) \sum_{j=1}^{J} \sigma_{jl} \left(\psi_{jl} + \varepsilon_j + v_{jl}\right)\right].$$

Note that this is simply an alternate representation of the value function in (3), expressed in terms of the next *move* time instead of the next *event* time. Both representations are equivalent.

The first term above represents the flow utility obtained from the initial time until the first move arrival at time τ_1. The second term represents the expected instantaneous and future utility obtained from making a choice at time τ_1. The resulting state l at time time τ_1 is stochastic, as is the optimal choice j and, possibly even the continuation value v_{jl}. The expectation operator is needed because τ_1 itself is random and is not known a priori.

If $\varepsilon_j \sim \text{TIEV}(0, 1)$, then the CCPs admit the following closed form:

$$(12) \quad \sigma_{jk} = \frac{\exp(\psi_{jk} + v_{jk})}{\sum_{m=1}^{J} \exp(\psi_{mk} + v_{mk})}.$$

Suppose we wish to express this probability with respect to another state, say state 1, then we can write

$$(13) \quad \sigma_{jk} = \frac{\exp(\psi_{jk} + v_{jk} - \psi_{j1} - v_{j1})}{\sum_{m=1}^{J} \exp(\psi_{mk} + v_{mk} - \psi_{m1} - v_{m1})}.$$

The ψ_{jk}'s typically have closed forms in terms of the parameters. Thus, if we know differences in the continuation values $v_{jk} - v_{j1}$, we effectively know the CCPs and can estimate the model. In what follows, we show how to obtain these differences using first stage estimates of the CCPs and a closed form inverse relationship with the value function.

First, note that from (12) we can write

$$(14) \quad \ln \left[\sum_{m=1}^{J} \exp(\psi_{mk} + v_{mk}) \right] = -\ln \sigma_{ijk} + \psi_{jk} + v_{jk}.$$

The left side of this expression is precisely the closed form for the ex-ante future value term in the value function.

In many model specifications we can then obtain an expression for the differences in (13) by choosing an appropriate normalizing state.[6] We use the example model of Section 3.1 to illustrate this point. In terms of this model, we can write (14), for $j = 1$ as

$$(15) \quad \ln \left[\exp(V_k) + \exp(V_1 - c) \right] = -\ln \sigma_{1k} + V_1 - c.$$

Note that the left-hand side of the above equation is exactly the expression in the value function as expressed in (4). Substituting (15) into (4) gives the following expression for

[6] See Arcidiacono and Miller (2008) for a general discussion.

the value function for each state k:

$$\begin{aligned} V_k &= \mathrm{E}\left[\int_0^{\tau_1} \mathrm{e}^{-\rho t} \sum_{l=1}^{K} P_{kl}(t) u_l \, dt + \mathrm{e}^{-\rho \tau_1} \sum_{l=1}^{K} P_{kl}(\tau_1) \left(-\ln \sigma_{1l} + V_1 - c\right)\right] \\ &= \mathrm{E}\left[\int_0^{\tau_1} \mathrm{e}^{-\rho t} \sum_{l=1}^{K} P_{kl}(t) u_l \, dt - \mathrm{e}^{-\rho \tau_1} \sum_{l=1}^{K} P_{kl}(\tau_1) \ln \sigma_{1l} + \mathrm{e}^{-\rho \tau_1} (V_1 - c)\right] \end{aligned}$$

where in the second equality we have used the fact that $V_1 - c$ does not depend on l and that the probabilities $P_{kl}(t)$ must sum to one over $l = 1, \ldots, K$. Evaluating the above expression at $k = 1$ and differencing gives

$$V_k - V_1 = \mathrm{E}\left[\int_0^{\tau_1} \mathrm{e}^{-\rho t} \sum_{l=1}^{K} \left[P_{kl}(t) - P_{1l}(t)\right] u_l \, dt - \mathrm{e}^{-\rho \tau_1} \sum_{l=1}^{K} \left[P_{kl}(\tau_1) - P_{1l}(\tau_1)\right] \ln \sigma_{1l}\right].$$

This expression gives differences in the value function in terms of the conditional choice probability σ_{1l}. With first-stage estimates of σ_{1l} for each l we can use this expression to "invert" the estimated CCPs to obtain an approximation of $V_k - V_1$ which can then be used, along with (5), to approximate $\sigma(a_t, x_t; \theta)$ in the likelihood. The result is a pseudo-likelihood function which can be maximized to obtain an estimate of θ.

5.4.1. Multi-Agent Models

In dynamic games, in the interval between an arbitrary time $t < \tau_i$ and τ_i, any combination of state jumps and moves by other players may take place. Q_0 describes the dynamics of state jumps, and we can construct similar intensity matrices Q_i that describe the dynamics of events caused by the actions of rival players. In any state k, player i moves at a rate λ_i which is constant across k.

Thus, the rate at which the model leave state k due to player i is λ_i. The rate at which the model enters another state $l \neq k$, the (k, l) element of Q_i, is given by the sum

$$\lambda_i \sum_{j=1}^{J} \sigma_{ijk} \phi_{ijkl},$$

which accounts for uncertainty both over the choice and the resulting state. Intuitively, this is the probability of moving to state l expressed as a proportion of λ_i, the rate at which the model leaves state k. Note that we must also allow for the state to remain at k, in which case the diagonal (k, k) element of Q_i is

$$-\lambda_i + \lambda_i \sum_{j=1}^{J} \sigma_{ijk} \phi_{ijkk}.$$

From the perspective of player i, the dynamics of the model follow an intensity matrix $Q_{-i} \equiv Q_0 + \sum_{j \neq i} Q_j$ which captures all events caused by nature and player i's rivals. With this intensity matrix in hand, the flow utility portion of the value function can be expressed exactly as before with $P^{-i}(t)$ being constructed using the intensity matrix Q_{-i}: $\int_0^{\tau_i} \mathrm{e}^{-\rho t} \sum_{l=1}^{K} P_{kl}^{-i}(t) u_{il} \, dt$. The value function for player i is then

$$(16) \quad V_{ik} = \mathrm{E}\left[\int_0^{\tau_i} \mathrm{e}^{-\rho_i t} \sum_{l=1}^{K} P_{kl}^{-i}(t) u_{il} \, dt + \mathrm{e}^{-\rho \tau} \sum_{l=1}^{K} P_{kl}^{-i}(\tau_i) \sum_{j=1}^{J} \sigma_{ijl} \left(\psi_{ijl} + \varepsilon_{ij} + v_{ijl} \right) \right].$$

CCP estimation can now proceed as in the single agent case by recognizing that exiting is a terminal state. Hence, at the time of the next move, the continuation value can be expressed simply as the negative of the log probability of exiting.

5.4.2. Computational Issues

There are several practical computational issues to consider when attempting to evaluate the next-move based value function both in the single- and multi-agent cases. In practice, for CCP estimation, we are actually interested in approximating the difference $V_k - V_1$. For simplicity, we will discuss methods for approximating V_k. Approximating the difference is straightforward using the same procedures.

Consider the single-agent version in (11). In expectation $\sum_{j=1}^{J} \sigma_{jl} \left(\psi_{jl} + \varepsilon_j + v_{jl} \right)$ is the ex-ante expected future value of choosing optimally in state l. We can isolate this term using the law of iterated expectations, replacing $\mathrm{E}_{\tau_1, \varepsilon}$ with $\mathrm{E}_{\tau_1} \mathrm{E}_{\varepsilon | \tau_1}$. If we then make the standard assumption that the ε_j are iid and distributed according to the type I extreme value distribution, we can simplify this expression using the known closed form for the maximum of J values $\{\delta_1 + \varepsilon_1, \dots, \delta_J + \varepsilon_J\}$:

$$\mathrm{E}\left[\max\{\delta_1 + \varepsilon_1, \dots, \delta_J + \varepsilon_J\}\right] = \ln\left[\exp(\delta_1) + \dots + \exp(\delta_J)\right].$$

See, for example, McFadden (1984) for details.

Now, we must still choose how to evaluate both the flow utility term as well as the expectation over τ_1. We describe two Monte Carlo integration methods for doing so. The first involves simulating from the distribution of τ_1 and using a closed form for the flow utility term. The second involves averaging the flow utility and discounted future value over many simulated paths of the combined jump process, starting from the current time and ending at next move by the player in question. The first method involves a lower-dimensional integral but requires many matrix exponential calculations. The second approach involves approximating a more complex integral, but avoids potentially costly matrix calculations.

In the first approach, we simply approximate the expectation over τ_1 using Monte Carlo integration by drawing R values of τ_1, $\{\tau_1^s\}_{s=1}^R$, and forming the following approximation:

$$V_k \approx \frac{1}{R} \sum_{s=1}^{R} \left[\int_0^{\tau_1^s} \mathrm{e}^{-\rho t} \sum_{l=1}^{K} P_{kl}(t) u_l \, dt + \mathrm{e}^{-\rho \tau_1^s} \sum_{l=1}^{K} P_{kl}(\tau_1^s) \sum_{j=1}^{J} \sigma_{jk} \left(\psi_{jk} + \varepsilon_j + v_{jk} \right) \right]. \tag{17}$$

In matrix notation, the flow utility term above actually has a relatively simple closed form which allows (17) to be calculated directly. To see this, let $b_i(\tau_1) = \int_0^{\tau_1} \mathrm{e}^{-\rho s} \sum_j P_{ij}(s) u(x_j) \, ds$, $B(\tau_1) = (b_1(\tau_1), b_2(\tau_1), \dots, b_K(\tau_1))^\top$ and $U = (u(x_1), \dots, u(x_n))^\top$. Define $C \equiv -(\rho I - Q)$ for simplicity. Then we can write the first term inside the expectation in matrix notation as

$$\begin{aligned} B(\tau_1) &= \int_0^{\tau_1} \mathrm{e}^{-\rho s I} \, \mathrm{e}^{sQ} \, U \, ds = \left[\int_0^{\tau_1} \mathrm{e}^{-s(\rho I - Q)} \, ds \right] U \\ &= \left[\int_0^{\tau_1} C^{-1} C \, \mathrm{e}^{sC} \, ds \right] U = C^{-1} \left[\int_0^{\tau_1} C \, \mathrm{e}^{sC} \, ds \right] U = C^{-1} \left[\mathrm{e}^{\tau_1 C} - I \right] U. \end{aligned}$$

Finally, substituting for C we have

$$B(\tau_1) = -(\rho I - Q)^{-1} \left[\mathrm{e}^{-\tau_1 (\rho I - Q)} - I \right] U.$$

The alternate approach is able to avoid the computation of matrix exponentials altogether. We can approximate V_k using a forward simulation procedure where we simulate R paths of the joint jump process governed by the aggregate intensity matrix $Q = Q_0 + \sum_i Q_i$. Each path begins at the current time, in state k, and ends when the player in question moves next (a simulated realization of τ_1). The flow utility obtained over each path is accumulated and the discounted future value term at the final state is calculated (via CCP inversion when working in differences). Averaging both the flow utility and future value terms over the R simulated paths and discounting appropriately provides an approximation to V_k.

In either case, it is important to note that the value function only needs to be approximated at states that are relevant for estimation. We can focus only on those states that are actually observed in the sample and any related states which are used in the choice probability calculations that appear in the log likelihood function. That is, we only need to know the value function at each observed state k and each additional state that might arise as a result of some action at state k (e.g., exit or investment). As a result, even when the state space is very large the number of components of the value function that need to be calculated is simply a function of the observations in the sample. This can result in considerable computational savings.

5.5. Unobserved Heterogeneity

Incorporating permanent unobserved heterogeneity into the models above follows the same method commonly used in the dynamic discrete choice literature. Namely, we can use finite mixture distributions to allow for permanent unobserved characteristics.[7]

Consider, for example, the bus engine problem where now certain buses have higher replacement costs or mileage transitions. The type-specific likelihood for a particular bus is composed of the type-specific probabilities of the mileage and engine transitions over the course of the sample period. The log likelihood for a particular bus with the unobserved state integrated out is then the log of the sum of the type-specific likelihoods weighted by the population probabilities of being each of the different types. For the nested fixed point algorithm, estimation is then straightforward. With CCP estimation, the techniques developed by Arcidiacono and Miller (2008) also apply to the continuous time setting.

6. Monte Carlo Experiments

6.1. Single Agent Dynamic Discrete Choice

Here, we generate data according to the simple single player binary choice model of Section 3.1. The primitives of the model are the payoff (mileage cost) parameter β, the intensity matrix (mileage transition) parameters q_1 and q_2, the reset (engine replacement) cost c, the discount rate ρ, and the move arrival rate λ. We fix $\rho = 0.05$ and focus on estimating $\theta = (\lambda, q_1, q_2, \beta, c)$.

In the first set of experiments, we use a full solution approach to estimate the model. Namely, the value function is obtained through value function iteration for each value of θ while the log likelihood function is maximized in an outer loop. We estimate the model under several different scenarios including full continuous-time data, continuous-time data when the decision not to replace the engine is not observed, and discrete time data of varying resolution. Additional details regarding data generation and estimation can be found in the appendix. The means and standard deviations of the parameter estimates are reported in Table 1. All are centered around their true values and quite precisely estimated. The loss in precision from moving away from continuous time data is initially greatest for the move arrival rate, λ, yet all estimates of this parameter are still precise. The replacement cost, c, also loses precision with more coarsely sampled data, but the increases are not large until we move to seeing only one in four events on average in the sampling period.

Results for two-step estimation using conditional choice probabilities are displayed in Table 2. We use a simple bin estimator to obtain the CCPs in a first stage. Details about how these first-stage estimates were obtained in the time aggregation and partial move cases can be found in the appendix. Using CCPs increases the standard deviations slightly, reflecting noise from the first stage. However, the estimates are still very good, particularly

[7] See Keane and Wolpin (1997), Eckstein and Wolpin (1999), and Arcidiacono (2005) among many others.

when the average number of state changes per sampling interval is small.

Finally, we also estimated the model with continuous-time data while allowing for buses to be of two distinct types, where the type is not observed by the econometrician. In this specification, the type affected both the mileage transition probabilities and payoff parameters. In particular, with probability π, the bus is of the first type and with probability $1 - \pi$, the bus is of the second type. For buses of type $m = 1, 2$, the mileage jumps forward one unit at rate q_1 and two units at rate q_{2m}, the cost of mileage is β, and the cost of replacement is c_m. Again, estimation proceeded quickly with little difficulty in separating the unobserved heterogeneity from the other model parameters. The results are reported in Table 3.

6.2. A Dynamic Discrete Game

Our second set of Monte Carlo experiments corresponds to the quality ladder model described in Section 4.1. We estimate models ranging from 10 to 20 firms with 7 possible quality levels. The size of the state space for our largest problem is over four and a half million. Details about the sample construction can again be found in the appendix.

Table 4 summarizes the results for full-solution estimation, where we obtain the value function using value function iteration for each trial value of θ. Table 5 presents the analogous results obtained using CCP estimation, where we assume the true CCPs are available. In all cases, both full-solution methods and CCP estimation perform extremely well and there is virtually no change in the standard deviations across the different state space sizes.

We then compare the computational time required for both full-solution and CCP estimation in Table 6. We first report the number of players N, the market size M, the total number of states K. For each model, computational times are reported for only one replication. Since we consider many models, the overall trends are clear despite the fact that we do not report averages.[8]

The first timing column reports the time required to obtain the value function V for each model specification. This step is necessary to either generate a dataset and to simulate the model (e.g. to perform counterfactuals). In particular, we use datasets consisting of $M = 200$ markets with $T = 100$ continuous time events observed in each. Next we report the time required to estimate the first stage parameters λ and γ. This step is independent of the method used to obtain the value function. Next, we report the total time required to estimate the second stage parameters κ, η, and η^e via full solution estimation. For each

[8] All reported times are for estimation on a desktop PC with a quad-core 64-bit AMD Phenom II X4 920 processor. Our programs are written in Fortran and take advantage of parallel processing in obviously parallel segments of code. Again, we use L-BFGS-B to maximize the log-likelihood function in each case.

new trial value of θ, we use the value function at the previous θ as the starting value for the value function iteration. Finally, we report the setup time required to perform the initial forward simulation procedure described in Section 5.4.2 (with $R = 250$), the time required to estimate the second-stage parameters, and the sum of these two times (the total time).[9]

Even with over four and a half million states, full solution estimation took under five hours. Conditional on already having the CCPs from a first stage, two-step estimation times were incredibly fast, with the longest taking less than two minutes. To put these numbers in perspective, Doraszelski and Judd (2008) note that it would take about a year to just solve for the equilibrium of a 14 player game (with 7 levels of quality) using the basic Pakes-McGuire algorithm.[10] Our continuous time approach takes about 30 minutes to solve the game and under two hours to estimate the parameters using a full solution (NFXP) approach. CCP estimation requires less than a minute. These computational times suggest that very large classes of problems can be easily estimated in a continuous time framework. Furthermore, the computational time required to calculate the fixed point once in continuous time is small even for very large problems. This implies that simulating counterfactuals from large-scale models will not be an issue.

7. Conclusion

While recently developed two-step estimation approaches make it possible to estimate large-scale dynamic games, performing counterfactuals (and simulating data) can be severely limited by the curse of dimensionality that arises from simultaneous moves. To break this curse, we recast the standard simultaneous move game in discrete time as a sequential move game in continuous time. Games that would take a year to solve under discrete time can be estimated and solved in less than two hours, greatly expanding the breadth and applicability of these structural methods. We also show how to extend the framework to accommodate incomplete sampling schemes, including missing actions and time-aggregated data. Both are likely to be relevant for real-world datasets.

Our framework suggests a number of areas for future research. First, we currently do not allow players to influence the arrival rate of move opportunities. It is reasonably straightforward to endogenize these rates (over a finite set) using the methods described in Puterman (2005). This would, however, increase the data requirements substantially. Second, we have focused exclusively on discrete decision problems on discrete states.

[9] This table does not address the time required to estimate the CCPs which can vary significantly depending on which method is used. Parametric methods can be very fast while fully nonparametric methods can be computationally intensive.

[10] Similar computational times are also reported in Doraszelski and Pakes (2007).

Future work is needed to extend these models to both continuous controls and continuous states. These extensions will be key to allowing for asymmetric information in continuous time games. If players only observe their rivals' actions with a lag, then the time since player i's last move, which is continuous, becomes a relevant state variable for player i's value function. We believe this is a particularly promising area for future research.

A. Proofs

Proof of Proposition 1. The result follows directly from the joint distribution function:

$$\begin{aligned}\Pr\left(\tau \leq t\right) = \Pr\left(\min_i \tau_i \leq t\right) &= 1 - \Pr(\tau_1 > t, \dots, \tau_n > t) \\ &= 1 - \prod_{i=1}^{n} \Pr(\tau_i > t) = 1 - \prod_{i=1}^{n} \mathrm{e}^{-\lambda_i t} = 1 - \mathrm{e}^{-(\sum_{i=1}^{n} \lambda_i)t}.\end{aligned}$$

Therefore, τ has an exponential distribution with rate parameter $\sum_{i=1}^{n} \lambda_i$. ■

Proof of Proposition 2.

$$\begin{aligned}\Pr(\tau_i \leq \tau_j \; \forall j) &= \mathrm{E}_{\tau_i}\left[\Pr(\tau_j \geq \tau_i \; \forall j \neq i) \bigm| \tau_i\right] \\ &= \int_0^\infty \left[\mathrm{e}^{-\sum_{j\neq i} \lambda_j}\right] \lambda_i \, \mathrm{e}^{-\lambda_i \tau_i} \; d\tau_i \\ &= \int_0^\infty \lambda_i \, \mathrm{e}^{-(\sum_{j=1}^{n} \lambda_j)\tau_i} \; d\tau_i \\ &= -\frac{\lambda_i}{\sum_{j=1}^{n}} \left[\mathrm{e}^{-(\sum_{j=1}^{n} \lambda_j)\tau_i}\right]_{\tau_i=0}^{\infty} \\ &= \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}.\end{aligned}$$

■

Proof of Proposition 3. We have

$$\begin{aligned}
\Pr(W_+(t) \geq \tau) &= \Pr\left[N_+(t+\tau) - N_+(t) = 0\right] \\
&= \sum_{k=0}^{\infty} \Pr\left[N(t+\tau) - N(t) = k, N_0(t+\tau) - N_0(t) = k\right] \\
&= \sum_{k=0}^{\infty} \Pr\left[N(t+\tau) - N(t) = k\right] \sigma(0,x)^k \\
&= \sum_{k=0}^{\infty} \frac{\mathrm{e}^{-\lambda\tau}(\lambda\tau)^k}{k!} \sigma(0,x)^k \\
&= \mathrm{e}^{-\lambda\tau} \sum_{k=0}^{\infty} \frac{(\sigma(0,x)\lambda\tau)^k}{k!} \\
&= \mathrm{e}^{-\lambda\tau}\, \mathrm{e}^{\sigma(0,x)\lambda\tau} \\
&= \mathrm{e}^{-(1-\sigma(0,x))\lambda\tau},
\end{aligned}$$

and therefore the cdf of $W_+(t)$ is

$$\Pr(W_+(t) \leq \tau) = 1 - \mathrm{e}^{-(1-\sigma(0,x))\lambda\tau}.$$

For a given x, this is precisely the cdf of the exponential distribution with parameter $(1 - \sigma(0,x))\lambda$. ■

B. Details of the Monte Carlo Experiments

B.1. Single Agent Model

To generate data for the single agent model we first choose values for θ and then use numerical fixed point methods to determine the value function over the state space $\mathcal{X}$ to within a tolerance of $\epsilon = 10^{-6}$ in the relative sup norm. To evaluate the expectation over τ in (4), we use Monte Carlo integration as described in Section 5.4.2, drawing R arrival intervals according to the appropriate exponential distribution and approximating the integral using the sample average. We set R to 250. We then use the resulting value function to generate data for various values of T.

In the first set of experiments, we estimate the model using full solution methods. The value functions are obtained through value function iteration for each value of θ while maximizing the likelihood function using the L-BFGS-B algorithm (Byrd, Lu, and Nocedal, 1995; Zhu, Byrd, Lu, and Nocedal, 1997). We generate 100 data sets over the interval $[0, T]$ with $T = 25,000$ for an average of $10,000$ events and then estimate the model under several sampling regimes: true continuous time data, continuous time data when passive actions ($a = 0$, the choice not to renew) are unobserved, and discrete time data observed at intervals $\Delta \in \{0.625, 1.25, 2.5, 5.0, 10.0\}$.

We also carry out the same experiments using CCP-based estimation in the single agent model. Again, for $T = 25,000$, we estimate the model with full continuous-time data, a continuous-time dataset with missing passive actions, and several discrete-time datasets of varying granularity. For the full continuous-time dataset, we can nonparametrically estimate the CCPs using a simple bin estimator. When accounting for passive moves, we approximate the CCPs by dividing the number of times each particular observed choice was made in each state by the implied expected number of move arrivals in that state. Finally, when estimating the model with discrete-time data, we first jointly estimate the first-stage parameters (λ, q_1, and q_2) and the parameters of a logistic regression model for the probability of renewal with parameters α. The regressors in our logit model are a constant, the state x, and $\ln x$. Then, we invert the predicted CCPs obtained using the estimated parameters $\hat{\alpha}$ to obtain the value function which we use to estimate the remaining second stage parameters.

B.2. Quality Ladder Model

For the multi-agent quality ladder model, we obtain estimates of $\theta = (\lambda, \gamma, \kappa, \eta, \eta^e)$ for each of 25 simulated datasets and report the means and standard deviations (in parenthesis). In all experiments, we hold $\bar{\omega}$ fixed at $\bar{\omega} = 7$ and vary the maximum number of players, N, and the market size, M.[11]

We also increase the market size M so that the average number of active players (n_{avg}) grows with the total number of possible players (N). The average quality level of active firms is denoted ω_{avg}. We also report K, the number of states from the perspective of player i—the number of distinct (ω, ω_i) combinations. In these experiments, we used samples containing $T = 100$ continuous time events in each of $M = 1000$ markets. We fixed $\rho = 0.05$ and use $R = 250$ draws for Monte Carlo integration.

For the CCP estimation, we use the true CCPs. In practice, the CCPs must be estimated somehow in a preliminary step. However, because there are many possible methods for doing so, and because they tend to be application and data specific, we simply present the results for the second-stage parameters as if the true CCPs were known. We have estimated the CCPs nonparametrically using locally weighted averages with little change in the results.

[11] The size of the state space of our model with $\bar{\omega} = 7$ is roughly comparable to that of Doraszelski and Judd (2008) with $\bar{\omega} = 9$ since their model does not include entry and exit.

References

Aguirregabiria, V. and P. Mira (2007). Sequential estimation of dynamic discrete games. *Econometrica 75*, 1–53. [2, 18]

Arcidiacono, P. (2005). Affirmative action in higher education: How do admission and financial aid rules affect future earnings? *Econometrica 73*, 1477–1524. [28]

Arcidiacono, P. and R. A. Miller (2008). CCP estimation of dynamic discrete choice models with unobserved heterogeneity. Working paper, Duke University. [18, 24, 28]

Bajari, P., C. L. Benkard, and J. Levin (2007). Estimating dynamic models of imperfect competition. *Econometrica 75*, 1331–1370. [2, 18]

Bajari, P., H. Hong, and S. Ryan (2007). Identification and estimation of discrete games of complete information. Working paper, University of Minnesota. [18]

Byrd, R. H., P. Lu, and J. Nocedal (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing 16*, 1190–1208. [32]

Caplin, A. and B. Nalebuff (1991). Aggregation and imperfect competition: On the existence of equilibrium. *Econometrica 59*, 25–59. [15]

Doraszelski, U. and K. L. Judd (2008). Avoiding the curse of dimensionality in dynamic stochastic games. Working paper, Harvard University. [2, 3, 14, 30, 33]

Doraszelski, U. and S. Markovich (2007). Advertising dynamics and competitive advantage. *The RAND Journal of Economics 38*, 557–592. [14]

Doraszelski, U. and A. Pakes (2007). A framework for applied dynamic analysis in IO. In M. Armstrong and R. H. Porter (Eds.), *Handbook of Industrial Organization*, Volume 3, Chapter 30, pp. 1887–1966. North Holland. [13, 30]

Doraszelski, U. and M. Satterthwaite (2007). Computable Markov-perfect industry dynamics: Existence, purifcation, and multiplicity. Working paper, Harvard University. [2, 13]

Eckstein, Z. and K. I. Wolpin (1999). Why youth drop out of high school: The impact of preferences, opportunities, and abilities. *Econometrica 67*, 1295–1339. [28]

Ericson, R. and A. Pakes (1995). Markov-perfect industry dynamics: A framework for empirical work. *Review of Economics and Statistics 62*, 53–82. [13, 18]

Gotz, G. A. and J. J. McCall (1980). Estimation in sequential decision-making models: A methodological note. *Economic Letters 6*, 131–136. [1]

Gowrisankaran, G. (1999a). A dynamic model of endogenous horizontal mergers. *The RAND Journal of Economics 30*, 56–83. [14]

Gowrisankaran, G. (1999b). Efficient representation of state spaces for some dynamic models. *Journal of Economic Dynamics and Control 23*, 1077–1098. [14]

Harsanyi, J. (1973). Games with randomly disturbed payoffs: A new rationale for mixed-strategy equilibrium points. *International Journal of Game Theory 2*, 1–23. [11]

Hotz, V. J. and R. A. Miller (1993). Conditional choice probabilities and the estimation of dynamic models. *Review of Economic Studies 60*, 497–529. [2, 18]

Hotz, V. J., R. A. Miller, S. Sanders, and J. Smith (1994). A simulation estimator for dynamic models of discrete choice. *Review of Economic Studies 61*, 265–289. [2, 18]

Karlin, S. and H. M. Taylor (1975). *A First Course in Stochastic Processes* (Second ed.). Academic Press. [5]

Keane, M. and K. I. Wolpin (1997). The career decisions of young men. *Journal of Political Economy 105*, 473–522. [28]

McFadden, D. L. (1974). Conditional logit analysis of qualitative choice analysis. In P. Zarembka (Ed.), *Frontiers in Econometrics*, pp. 105–142. Academic Press. [15]

McFadden, D. L. (1984). Econometric analysis of qualitative response models. In Z. Griliches and M. Intrilligator (Eds.), *Handbook of Econometrics*, Volume 2, Chapter 24, pp. 1396–1456. Amsterdam: Elsevier. [10, 26]

Miller, R. A. (1984). Job matching and occupational choice. *Journal of Political Economy 92*(6), 1086–1120. [1]

Pakes, A. (1986). Patents as options: Some estimates of the value of holding european patent stocks. *Econometrica 54*, 755–784. [1]

Pakes, A., G. Gowrisankaran, and P. McGuire (1993). Implementing the Pakes-McGuire algorithm for computing Markov perfect equilibria in Gauss. Unpublished manuscript, Harvard University. [14, 15]

Pakes, A. and P. McGuire (1994). Computing Markov-perfect nash equilibria: Numerical implications of a dynamic differentiated product model. *The RAND Journal of Economics 25*, 555–589. [2, 13, 14, 15, 17, 18]

Pakes, A. and P. McGuire (2001). Stochastic algorithms, symmetric Markov perfect equilibrium, and the 'curse' of dimensionality. *Econometrica 69*, 1261–1281. [2, 13, 18]

Pakes, A., M. Ostrovsky, and S. Berry (2007). Simple estimators for the parameters of discrete dynamic games (with entry/exit examples). *The RAND Journal of Economics 38*, 373–399. [2, 18]

Pesendorfer, M. and P. Schmidt-Dengler (2007). Asymptotic least squares estimators for dynamic games. *Review of Economic Studies 75*, 901–928. [2, 18]

Puterman, M. L. (2005). *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (Second ed.). Wiley-Interscience. [30]

Rust, J. (1987). Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica 55*, 999–1013. [1, 9, 18]

Rust, J. (1994). Estimation of dynamic structural models, problems and prospects: Discrete decision processes. In C. Sims (Ed.), *Advances in Econometrics: Sixth World Congress*, Volume 2, Chapter 4, pp. 119–170. Cambridge University Press. [2]

Sidje, R. B. (1998). Expokit: A software package for computing matrix exponentials. *ACM Trans. Mathematical Software 24*, 130–156. [6]

Su, C.-L. and K. L. Judd (2008). Constrained optimization approaches to estimation of structural models. Working paper, University of Chicago. [18]

Weintraub, G. Y., C. L. Benkard, and B. Van Roy (2008). Markov perfect industry dynamics with many firms. *Econometrica 76*, 1375–1411. [2]

Wolpin, K. I. (1984). An estimable dynamic stochastic model of fertility and child mortality. *Journal of Political Economy 92*, 852–874. [1]

Zhu, C., R. H. Byrd, P. Lu, and J. Nocedal (1997). Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software 23*, 550–560. [32]

Table 1. Single player Monte Carlo results: NFXP estimation ($T = 25,000$).

Sampling	n	q_1	q_2	λ	β	c
Population	∞	0.150	0.050	0.200	1.000	1.250
Continuous Time	10,000	0.150	0.050	0.200	1.009	1.254
		(0.002)	(0.001)	(0.003)	(0.068)	(0.054)
Passive Moves	7,176	0.150	0.050	0.204	1.010	1.271
		(0.002)	(0.001)	(0.020)	(0.127)	(0.126)
$\Delta = 0.625$	40,000	0.137	0.053	0.189	1.107	1.305
		(0.003)	(0.002)	(0.019)	(0.213)	(0.238)
$\Delta = 1.25$	20,000	0.145	0.051	0.191	1.074	1.191
		(0.003)	(0.002)	(0.024)	(0.210)	(0.297)
$\Delta = 2.5$	10,000	0.147	0.051	0.198	1.014	1.167
		(0.004)	(0.002)	(0.027)	(0.334)	(0.408)
$\Delta = 5.0$	5,000	0.151	0.050	0.195	1.088	1.233
		(0.007)	(0.003)	(0.019)	(0.249)	(0.402)
$\Delta = 10.0$	2,500	0.158	0.048	0.200	1.010	1.108
		(0.019)	(0.007)	(0.022)	(0.397)	(0.618)

The mean and standard deviation (in parenthesis) of the parameter estimates for 100 different simulated datasets are shown for various sampling regimes. Passive moves refers to datasets for which the choice $a = 0$ is not observed while Δ denotes the observation interval for discrete-time data. n denotes the average number of observations (continuous-time events or discrete-time intervals) when observing the model on the interval $[0, T]$. We fixed the discount rate, $\rho = 0.05$, the number of states, $K = 10$, and the number of draws used for Monte Carlo integration, $R = 250$.

Table 2. Single player Monte Carlo results: CCP estimation ($T = 25,000$).

Sampling	n	q_1	q_2	λ	β	c
Population	∞	0.150	0.050	0.200	1.000	1.250
Continuous Time	10,000	0.150	0.050	0.200	1.015	1.256
		(0.002)	(0.001)	(0.003)	(0.064)	(0.053)
Passive Moves	7,176	0.150	0.050	0.187	0.830	1.157
		(0.002)	(0.001)	(0.012)	(0.148)	(0.094)
$\Delta = 0.625$	40,000	0.137	0.053	0.196	1.114	1.367
		(0.003)	(0.002)	(0.041)	(0.267)	(0.272)
$\Delta = 1.25$	20,000	0.145	0.051	0.211	1.066	1.370
		(0.003)	(0.002)	(0.053)	(0.301)	(0.325)
$\Delta = 2.5$	10,000	0.147	0.051	0.219	1.094	1.377
		(0.004)	(0.002)	(0.103)	(0.333)	(0.421)
$\Delta = 5.0$	5,000	0.151	0.050	0.222	1.092	1.350
		(0.007)	(0.003)	(0.089)	(0.373)	(0.499)
$\Delta = 10.0$	2,500	0.154	0.049	0.241	1.159	1.356
		(0.018)	(0.008)	(0.157)	(0.516)	(0.733)

The mean and standard deviation (in parenthesis) of the parameter estimates for 100 different simulated datasets are shown for various sampling regimes. Passive moves refers to datasets for which the choice $a = 0$ is not observed while Δ denotes the observation interval for discrete-time data. The CCPs were estimated in a first step using a bin estimator for continuous-time data and via logistic regression on x and $\ln x$ for estimation with time aggregation. n denotes the average number of observations (continuous-time events or discrete-time intervals) when observing the model on the interval $[0, T]$. We fixed the discount rate, $\rho = 0.05$, the number of states, $K = 10$, and the number of draws used for Monte Carlo integration, $R = 250$.

Table 3. Single player Monte Carlo results with unobserved heterogeneity.

M	n	q_1	q_{21}	q_{22}	π	λ	β	c_1	c_2
Population		0.150	0.050	0.030	0.700	0.200	1.000	1.000	2.000
25	100	0.150	0.051	0.031	0.677	0.201	1.040	0.986	2.003
		(0.006)	(0.004)	(0.005)	(0.115)	(0.005)	(0.303)	(0.111)	(0.255)
50	100	0.151	0.050	0.030	0.693	0.201	1.045	0.995	2.001
		(0.004)	(0.003)	(0.004)	(0.070)	(0.004)	(0.188)	(0.067)	(0.141)
100	100	0.151	0.051	0.030	0.689	0.201	1.023	0.994	1.994
		(0.003)	(0.002)	(0.002)	(0.058)	(0.003)	(0.137)	(0.049)	(0.107)
25	200	0.150	0.050	0.030	0.685	0.200	1.025	1.004	2.002
		(0.003)	(0.003)	(0.003)	(0.092)	(0.004)	(0.176)	(0.061)	(0.118)
50	200	0.151	0.050	0.030	0.694	0.201	1.033	1.009	2.008
		(0.003)	(0.002)	(0.002)	(0.073)	(0.003)	(0.136)	(0.041)	(0.102)
100	200	0.151	0.050	0.030	0.701	0.201	1.014	1.002	1.995
		(0.002)	(0.001)	(0.002)	(0.047)	(0.002)	(0.096)	(0.029)	(0.062)

The mean and standard deviation (in parenthesis) of the parameter estimates for 100 different simulated datasets are shown for various sampling regimes. M denotes the number of markets used in the sample, each with n observed continuous-time events. We fixed the discount rate, $\rho = 0.05$, the number of states, $K = 10$, and the number of draws used for Monte Carlo integration, $R = 250$.

Table 4. Quality ladder Monte Carlo results: NFXP estimation.

N	K	M	n_{avg}	ω_{avg}	λ	γ	κ	η	η^e
Population					1.800	0.200	0.800	4.000	5.000
10	80,080	5.0	6.62	3.79	1.820	0.201	0.798	3.986	4.967
					(0.005)	(0.001)	(0.026)	(0.204)	(0.171)
11	136,136	7.0	7.79	3.62	1.819	0.201	0.791	3.990	4.952
					(0.005)	(0.002)	(0.031)	(0.198)	(0.174)
12	222,768	8.0	8.29	3.47	1.821	0.201	0.798	4.010	5.007
					(0.006)	(0.001)	(0.024)	(0.192)	(0.163)
13	352,716	9.0	8.81	3.35	1.821	0.200	0.801	4.043	5.044
					(0.006)	(0.001)	(0.031)	(0.184)	(0.157)
14	542,640	10.0	9.32	3.22	1.821	0.200	0.801	4.043	5.044
					(0.006)	(0.001)	(0.031)	(0.184)	(0.157)
15	813,960	11.0	9.86	3.13	1.822	0.201	0.811	4.012	5.047
					(0.005)	(0.002)	(0.073)	(0.211)	(0.257)
16	1,193,808	13.0	10.89	3.00	1.822	0.200	0.837	3.967	5.152
					(0.005)	(0.001)	(0.103)	(0.223)	(0.402)
17	1,716,099	15.0	11.88	2.90	1.820	0.201	0.836	3.984	5.181
					(0.006)	(0.002)	(0.107)	(0.201)	(0.363)
18	2,422,728	17.0	12.90	2.81	1.821	0.200	0.808	3.999	5.030
					(0.006)	(0.002)	(0.060)	(0.195)	(0.268)
19	3,364,900	19.0	13.91	2.72	1.820	0.201	0.809	3.987	5.139
					(0.006)	(0.002)	(0.078)	(0.190)	(0.272)
20	4,604,600	21.0	14.92	2.64	1.820	0.200	0.801	4.009	5.129
					(0.006)	(0.0020	(0.084)	(0.194)	(0.308)

The mean and standard deviation (in parenthesis) of the parameter estimates for 25 different samples are shown for choices of N, the total number of players, and M, the market size, with $\bar{\omega}$ fixed at 7. K denotes the total number of distinct states, n_{avg} denotes the average number of active players, and ω_{avg} denotes the average quality level. Samples consisted of 1000 markets each with 100 observed events. We fixed $\rho = 0.05$ and used $R = 250$ draws for Monte Carlo integration.

Table 5. Quality ladder Monte Carlo results: CCP estimation.

N	K	M	n_{avg}	ω_{avg}	λ	γ	κ	η	η^e
	Population				1.800	0.200	0.800	4.000	5.000
10	80,080	5.0	6.62	3.79	1.822	0.202	0.777	4.075	5.072
					(0.005)	(0.001)	(0.012)	(0.236)	(0.235)
11	136,136	7.0	7.79	3.62	1.821	0.202	0.774	4.099	5.080
					(0.005)	(0.002)	(0.014)	(0.253)	(0.247)
12	222,768	8.0	8.29	3.47	1.823	0.202	0.775	4.088	5.086
					(0.006)	(0.001)	(0.013)	(0.236)	(0.228)
13	352,716	9.0	8.81	3.35	1.823	0.202	0.779	4.076	5.071
					(0.006)	(0.001)	(0.010)	(0.226)	(0.227)
14	542,640	10.0	9.32	3.22	1.823	0.202	0.780	4.076	5.069
					(0.005)	(0.001)	(0.015)	(0.238)	(0.228)
15	813,960	11.0	9.86	3.13	1.824	0.202	0.782	4.068	5.060
					(0.005)	(0.002)	(0.014)	(0.218)	(0.210)
16	1,193,808	13.0	10.89	3.00	1.824	0.202	0.787	4.067	5.060
					(0.005)	(0.001)	(0.016)	(0.212)	(0.199)
17	1,716,099	15.0	11.88	2.90	1.822	0.202	0.782	4.066	5.064
					(0.006)	(0.002)	(0.013)	(0.209)	(0.201)
18	2,422,728	17.0	12.90	2.81	1.823	0.202	0.784	4.070	5.064
					(0.006)	(0.002)	(0.014)	(0.210)	(0.189)
19	3,364,900	19.0	13.91	2.72	1.822	0.202	0.782	4.055	5.047
					(0.006)	(0.002)	(0.016)	(0.210)	(0.185)
20	4,604,600	21.0	14.92	2.64	1.822	0.201	0.783	4.061	5.050
					(0.006)	(0.002)	(0.017)	(0.204)	(0.189)

The mean and standard deviation (in parenthesis) of the parameter estimates for 25 different samples are shown for choices of N, the total number of players, and M, the market size, with $\bar{\omega}$ fixed at 7. K denotes the total number of distinct states, n_{avg} denotes the average number of active players, and ω_{avg} denotes the average quality level. Samples consisted of 1000 markets each with 100 observed events. We fixed $\rho = 0.05$ and used $R = 250$ draws for Monte Carlo integration. The true CCPs were used in estimation.

TABLE 6. Computational times (in seconds): NFXP vs CCP.

					NFXP	CCP		
N	M	K	Solve v	First Stage	Estimation	Setup	Estimation	Total
5	0.5	2,310	9.05	0.42	51.61	0.51	0.30	0.81
6	1.0	5,544	15.59	0.30	107.41	1.54	0.32	1.86
7	2.0	12,012	29.26	0.28	172.40	2.17	0.47	2.64
8	3.0	24,024	58.27	0.30	256.19	4.02	0.62	4.64
9	4.0	45,045	107.40	0.26	375.26	5.58	1.01	6.59
10	5.0	80,080	185.78	0.35	535.83	7.13	1.42	8.55
11	7.0	136,136	325.28	0.34	639.98	11.20	1.80	13.00
12	8.0	222,768	518.57	0.33	1069.52	13.47	3.21	16.69
13	9.0	352,716	821.83	0.34	1411.32	14.96	3.63	18.59
14	10.0	542,640	1228.98	0.39	2436.61	17.21	4.10	21.31
15	11.0	813,960	1719.72	0.38	3413.42	19.67	7.15	26.82
16	13.0	1,193,808	2499.98	0.44	4765.67	23.85	7.06	30.91
17	15.0	1,716,099	3642.02	0.43	13513.96	27.28	8.38	35.66
18	17.0	2,422,728	5109.30	0.41	10807.54	30.93	10.54	41.47
19	19.0	3,364,900	6929.01	0.43	13737.87	35.67	15.37	51.04
20	21.0	4,604,600	9377.57	0.41	16069.89	37.26	15.60	52.86

Computational times are reported in seconds for estimation on a desktop PC with a quad-core 64-bit AMD Phenom II X4 920 processor using code written in Fortran using OpenMP for parallel processing in obviously parallel segments of code. Times are reported for only one replication of each specification. N denotes the total possible number of players, M denotes the market size, and K denotes the total number of distinct states. We have fixed the number of possible quality levels at $\bar{\omega} = 7$, the discount rate at $\rho = 0.05$, and used $R = 250$ draws for Monte Carlo integration. Obtaining the value function v is used for generating data. Obtaining the first stage estimates is a common step for both NFXP and CCP estimation.