Estimation of Dynamic Discrete Choice Models in Continuous Time

Peter Arcidiacono          Patrick Bayer          Jason Blevins          Paul Ellickson
Duke University            Duke University          Duke University          University of Rochester

April 28, 2010

ERID Working Paper Number 50

This paper can be downloaded without charge from
The Social Science Research Network Electronic Paper Collection:
http://ssrn.com/abstract=1601350

Economic Research Initiatives at Duke
WORKING PAPERS SERIES
Estimation of Dynamic Discrete Choice Models in Continuous Time*

Peter Arcidiacono\textsuperscript{1}, Patrick Bayer\textsuperscript{1},
Jason R. Blevins\textsuperscript{2}, and Paul B. Ellickson\textsuperscript{3}

\textsuperscript{1}Duke University and NBER
\textsuperscript{2}Duke University
\textsuperscript{3}Simon Graduate School of Business, University of Rochester

April 28, 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 speed and scope of our framework using several large-scale Monte Carlo experiments.

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


\textsuperscript{*}We thank the attendees of the 2009 Cowles Foundation Conference on Structural Microeconomics and the 2010 cemmap Conference on Matching and Sorting as well as seminar participants at Chicago Booth, Duke, Ohio State, and Western Ontario for useful comments and suggestions.
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 methods have been applied to a wide range of economic problems including investment under uncertainty, savings and retirement, human capital accumulation, fertility decisions, labor market participation, resource extraction, and political decisions, among many others. Because dynamic decision problems are naturally high-dimensional, the empirical DDC literature has been accompanied from the outset by a parallel methodological literature aimed at reducing the computational burden of both estimating and computing these models.

The computational challenges raised by the dimensionality of these problems are even greater in the context of multi-agent strategic games, where the simultaneous actions of competing players introduces a further curse of dimensionality in computing expectations over rivals’ actions. Although a recent series of papers (Aguirregabiria and Mira, 2007; Bajari, Benkard, and Levin, 2007; Pesendorfer and Schmidt-Dengler, 2007; Pakes, Ostromsky, and Berry, 2007) have shown 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 dynamics, to more complex multi-agent settings, the computation of these models remains formidable, despite a growing number of methods for solving for equilibria (Pakes and McGuire, 1994, 2001; Doraszelski and Satterthwaite, 2010).  

A curse of dimensionality naturally arises in simultaneous move games because, in order to solve for their optimal policies, players must form expectations over all combinations of actions that each of their rivals can take. The burden of computing these expectations grows exponentially in the number of players and so, in many applications, the model that researchers can estimate (using two-step procedures) is far richer than what can be used for counterfactual policy simulations, leading some to suggest alternatives to the Markov perfect equilibrium concept in which firms condition on long run averages (regarding rivals’ states) instead of current information (Weintraub, Benkard, and Van Roy, 2008). The goal of this paper is to 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 challenges. Pakes and McGuire (2001) extend their seminal approach to solving dynamic games (Pakes and McGuire, 1994)

---

1Two-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.
by replacing explicit integration with simulation and utilizing an adaptive algorithm that targets only 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. Players in their model make continuous decisions that control the hazard rate of state changes (e.g., choose investment which results stochastically in a discrete productivity gain). Because state changes occur only one agent at a time, the dimension of expectations over rival actions 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.

Continuous time methods are frequently utilized in other areas of economics, including the macroeconomics, search, agency, and finance literatures. Continuous time modeling often simplifies the relevant mathematics, yielding closed form solutions for problems that would require computational methods under discrete time. In financial economics, it can even complete markets (Merton, 1990). The sequential “move” arrivals that arise naturally in continuous time have been used by macroeconomists to explain the stickiness of prices (Calvo, 1983) and in the search literature to explain price and wage dispersion (Burdett and Judd, 1983; Burdett and Mortensen, 1998). The analytical tractability of continuous time can also reduce the computational burden significantly. Doraszelski and Judd showed that this is especially true of theoretical IO models, due to the inherent combination of complex strategic interactions and high-dimensional state spaces. These issues become even more pronounced in empirical work, where the predictions of the model must be evaluated for a large number of parameter values.

In this paper, we seek to connect the computational advantages of continuous time with the empirical tractability of discrete choice models. To do so, we recast the dynamic decision problem as one in which competing Poisson processes stochastically control when players are able to move, with players then facing a standard discrete choice problem when given the opportunity to make a decision. This structure results in a simple, yet flexible mathematical structure that is computationally light enough to make even full solution (i.e., nested fixed point) estimation feasible for very large problems. The model also inherits many features of the standard discrete choice framework and, as a result, 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, further relaxing the computational burden of estimation. Having estimated the model, it is straightforward to 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.

Our framework easily accommodates a number of more complex sampling schemes, including some that are especially challenging in a discrete time setting. 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 of which are collected at regular intervals, rather than in real time). The mathematical structure of continuous time makes time aggregation simple. It is straightforward, for example, to calculate the likelihood of transitioning from any initial state to any final state over any discrete period of time in a manner that explicitly accounts for the complex combinations of possible actions and state changes that might have led to that final state. Time aggregation is much more difficult in discrete time and, as a result, researchers generally adopt the convention that players move at the same periodicity with which the data is observed, even when players clearly move far more frequently.

The potential advantages of modeling decisions using a continuous time framework extend beyond computation, highlighting aspects of strategic interaction that are muted by discrete time (e.g., first-mover advantage) and mitigating unnatural implications that can arise from simultaneity (e.g., ex post regret). In fact, a number of recent papers in the empirical games literature (e.g., Einav (2010); Schmidt-Dengler (2006)) have adopted a sequential structure for decision-making on the basis of the economic rather than computational implications of this modeling choice. The empirical relevance of these issues will depend on the institutional setting. In this way, a continuous time approach provides researchers with the option of modeling moves as sequential rather than simultaneous when this accords more closely with the actual economic setting, even if the data is only observed at regular intervals.

The 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 settings, including time aggregation, unobserved passive moves, and comparisons of computational times. 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 $\tau$ 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 $\tau_n$, with $n$ denoting 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

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

and the cumulative distribution function (cdf) is

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

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

where for $i \neq j$

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

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

\[
p_{ij} = \frac{q_{ij}}{\sum_{k \neq i} q_{ik}}.
\]

Finally, let $P_{ij}(t) = \Pr(X_{t+s} = j | X_s = i)$ 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

\[
P'(t) = P(t)Q,
\]

\[
P(0) = I.
\]

frequently referred to as the forward equations. It follows that

\[
P(t) = e^{tQ} = \sum_{k=0}^{\infty} \frac{(tQ)^k}{k!}.
\]

This quantity is the matrix exponential, the matrix analog of the scalar exponential $e^{x}$.

\[\text{2In practice, since we cannot calculate the infinite sum (2) directly, we compute } e^{tQ} \text{ using known algo-}\]
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 \( \lambda_i \) for \( i = 1, \ldots, n \), 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 properties are well known, but we present the following proposition for completeness.

**Proposition 2.1.** Suppose \( \tau_i \sim \text{Exponential}(\lambda_i) \), for \( i = 1, \ldots, n \), are independent and define \( \tau \equiv \min_i \tau_i \) and \( \iota \equiv \arg \min_i \tau_i \). Then

\[
\tau \sim \text{Exponential}(\lambda_1 + \cdots + \lambda_n)
\]

and

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

**Proof.** See Appendix A. \[\square\]

This proposition allows us to treat the \( n \) competing Poisson processes \( (\tau_1, \ldots, \tau_n) \) as a single joint process \((\tau, \iota)\) where the joint distribution is given above.

## 3 Single-Agent Dynamic Discrete Choice Models

In this section, we introduce a dynamic discrete choice model of single-agent decision-making in continuous time. The single-agent problem provides a simple setting in which to describe the main features of our continuous time framework; we show how these extend directly to multi-agent settings in the following section. We begin this section by laying out the notation and structure of the model in a general setting. We then introduce an example—the classic Rust bus engine replacement problem—to fix ideas. The example also serves as the basis for the Monte Carlo analysis presented later in the paper.

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 an element \( x \) 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 \( \lambda \) 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 \( \mathcal{A} = \{1, \ldots, J\} \), 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 \( \rho \). 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 \( \int_0^\tau e^{-\rho t} u_k \, dt \).

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 \( \psi_{jk} + \varepsilon_{jk} \) associated with making choice \( j \) in state \( k \), where \( \varepsilon_{jk} \) is a choice-specific payoff shock that is iid over time and across choices. Let \( \sigma_{jk} \) 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 \( v_{jk} = 0 \).

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 \( \tau \) denote the time until the next event: either an exogenous state change or a move opportunity for the agent. In state \( k \) we have

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

Here we have used Proposition 2.1 and the law of iterated expectations to evaluate the expectation over the joint distribution of \((\tau, \iota)\) by first conditioning on \( \tau \).

The value function, as expressed recursively in (3), is the expectation of two terms. The

\[\text{Footnote 3} \]
\[\text{Footnote 4} \]
\[\text{Footnote 5} \]

Alternatively, the search literature typically uses the instantaneous Bellman equation, which in our case can be written as \( \rho V_k = u_k + \sum_{l=1}^K q_{kl} V_l + \lambda \max_j \{ \psi_{jk} + \varepsilon_{jk} + v_{jk} \} \). This is roughly equivalent to the next-event representation in (3) in terms of computational complexity, since the integral for the flow utility has a known closed form. We use the next-event representation instead, primarily because it is more closely related to the next-move representation, which facilitates two-step estimation, as we show in Section 5.4. This is not the case with either the instantaneous or next-event representations.
first term represents the flow utility obtained in state \( k \) from the initial time until the next event (a move or jump), at time \( \tau \). 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 some state \( l \neq k \). The expectation is taken with respect to both \( \tau \) and \( \varepsilon \). It is important to note that although we have used the time of the next event, \( \tau \), as the point of reference in (3), another intuitive possibility is the time of the next move by the agent in question, \( \tau_i \) (here, \( \tau_i \)). We use the next event form in practice as it is computationally simpler.\(^6\)

A policy rule is a function \( \delta : \mathcal{X} \times \mathcal{E} \rightarrow \mathcal{A} \) 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, \( \delta \) assigns the action that maximizes the agent’s expected future discounted payoff. Thus, under the optimal policy rule, the conditional choice probabilities \( \sigma_{jk} \) satisfy

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

Note that the move arrival rate, \( \lambda \), and the choice probabilities of the agent, \( \sigma_{jk} \), also imply a Markov jump process on \( \mathcal{X} \) with intensity matrix \( Q_1 \), where \( Q_1 \) is a function of both \( \lambda \) and \( \sigma_{jk} \) for all \( j \) and \( k \). In particular, the hazard rate of action \( j \) in state \( k \) is simply \( \lambda \sigma_{jk} \), the product of the move arrival rate and the choice probability. The choice probability \( \sigma_{jk} \) is thus the proportion of moves in state \( k \), which occur at rate \( \lambda \), that result in action \( j \). Summing the intensity matrices \( Q_0 \) and \( Q_1 \) yields the intensity matrix of the combined (compound) 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 \( \mathcal{X} = \{1, \ldots, K\} \). The agent has a binary choice set \( \mathcal{A} = \{0, 1\} \), which represents the choice over whether or not to replace the engine, thereby resetting the mileage to its baseline level. The agent faces a cost minimization problem.

\(^6\)The next move form introduces matrix exponentials which must be calculated to evaluate the first term.
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 \( k = 1 \).

A representative sample path generated by this model is shown in Figure 2. Inter-arrival times are indicated by \( \tau_{in} \), where \( i \) denotes the identity of the player (with \( i = 0 \) denoting nature) and \( n \) denotes the event number. The agent’s decisions \( (a_{tn}) \) are indicated at each decision time. For example, at time \( t_1 \), the agent chooses to continue without replacement \( (a_{t_1} = 0) \), while at time \( t_4 \), the agent chooses to replace \( (a_{t_4} = 1) \), resetting the mileage.

The \( K \times K \) intensity matrix for the jump process on \( X \) is

\[
Q_0 = \begin{bmatrix}
-q_1 - q_2 & q_1 & q_2 & 0 & \ldots & 0 \\
0 & -q_1 - q_2 & q_1 & q_2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & -q_1 - q_2 & q_1 & q_2 \\
0 & 0 & \ldots & 0 & -q_1 - q_2 & q_1 + q_2 \\
0 & 0 & \ldots & 0 & 0 & 0
\end{bmatrix}.
\]

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 level or two (and only one at state \( K - 1 \)). Conditional on jumping, the probabilities of moving forward one level 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

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

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

If we assume that the \( \varepsilon_j \) 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 \( E_{\tau, \varepsilon} \) with \( E_{\tau} E_{\varepsilon|\tau} \)) to obtain:

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

and thus,

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

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, \( \tau \) 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 \( \tau \), and \( \varepsilon \). 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 to 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 \( \tau \) onward, after the next event occurs. At the arrival time \( \tau \), the state jumps to \( k + l, l \in \{1, 2\} \) with probability \( q_l / (\lambda + q_1 + q_2) \), while 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 \( j = 1 \) 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:

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

We return to this example in Section 6 below, where we conduct Monte Carlo experiments for various parameterizations of this single-agent renewal problem.

4 Multi-Agent Dynamic Discrete Games

Extending the single-agent model of Section 3 to the case of dynamic discrete games with many 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. As is standard in the literature, we focus on Markov strategies, eliminating the need to keep track of the full history of play. We begin this section by describing the general structure of the model followed by an example—the Ericson-Pakes quality ladder model—to fix ideas.

Suppose there are \( \bar{N} \) players indexed by \( i = 1, \ldots, \bar{N} \). The state space \( \mathcal{X} \) is now a set of vectors of length \( \bar{N} \), with the \( i \)-th component corresponding to the state of player \( i \). Player \( i \)’s discount rate is \( \rho_i \). We shall simplify the notation later by assuming 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, \( \sigma_{ijk} \), instantaneous payoffs, \( \psi_{ijk} \), and transition probabilities resulting from the action of a player, \( \phi_{ijkm} \).
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 $\tau_{in}$, 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.

Although it is still sufficient to have only a single state jump process on $X$ (with some intensity matrix $Q_0$) to capture moves by nature, there are now $N$ competing Poisson processes with rates $\lambda_i$ 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 the uncertainty about the moves of rival firms into the intensity matrix. This allows us to construct the value function for the multi-agent model in much the same way as in the single-agent case. Let $\tau$ 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 $\lambda_i$ for $i = 1, \ldots, N$.

A representative sample path from a two-player game is depicted in Figure 3. Moves by nature are indicated by $i = 0$. The moves and inter-arrival times at the $n$-th event are denoted by $a_{it}$ and $\tau_{in}$ 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

7As in the single player model, we assume here that the move arrival rate is constant across states. However, with a large enough dataset, the inclusion of state specific arrival rates would be straightforward.
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 $\tau$, no other events may take place since, by definition, $\tau$ 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 $\lambda_i$ 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 $\sigma_{ijk}$ 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 $\phi_{ijkl}$ 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

$$V_{ik} = E\left[\int_0^\tau e^{-\rho_i t} u_{ik} dt + e^{-\rho_i \tau} \frac{1}{\sum_{l=1}^{N} \lambda_l + q_{kk}} \left(\sum_{l \neq k} q_{kl} V_{il} + \sum_{l \neq i} \lambda_l \sum_{j=1}^{J} \sigma_{ljk} \sum_{m=1}^{K} \phi_{lkm} V_{im} + \lambda_i \max_j \{\psi_{ijk} + \varepsilon_{ij} + v_{ijk}\}\right)\right] + \varepsilon_i \delta_i(k, \varepsilon_i) = j \iff \psi_{ijk} + \varepsilon_{ij} + v_{ijk} \geq \psi_{ilk} + \varepsilon_{il} + v_{ilk} \quad \forall l \in A_i. \quad (7)$$

This expression is complicated only for the sake of generality. In many applications, it will be the case that the $\phi_{lkm}$ terms are degenerate, with deterministic state transitions following moves. Further simplifications are also possible under symmetry.

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

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

For simplicity, we assume here that all players are active in all states. In this case, the rate at which moves occur is the sum $\sum \lambda_i$, which is independent of the state. In general, these move arrival rates may be state-specific, as illustrated in the quality ladder model of the following section, where inactive firms cannot move (i.e., $\lambda_{ik} = 0$ in states $k$ where firm $i$ is inactive).
That is, when given the opportunity to choose an action, \( \delta_i \) 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 \( i \), \( \sigma_{ijk} \), satisfy

\[
\sigma_{ijk} = \Pr[\delta_i(k, \varepsilon_i) = j \mid k].
\]

A Markov perfect equilibrium is a collection of policy rules \( (\delta_1, \ldots, \delta_N) \) and a set of beliefs \( \{\sigma_{ijk} : i = 1, \ldots, N; j = 1, \ldots, J; k = 1, \ldots, K\} \) 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 a discrete control version of the quality ladder model proposed by 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 (2010), Doraszelski and Pakes (2007), and several others. The model consists of at most \( N \) firms 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 \( \Omega = \{1, 2, \ldots, \bar{\omega}, \bar{\omega} + 1\} \) 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. In contrast to Pakes and McGuire (1994), all controls here are discrete: given a move arrival, firms choose whether or not to move up the quality ladder, not how much to spend to increase their chances of doing so.

We consider the particular example 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 \( \omega_i \) is the share of consumers who are aware of firm \( i \)'s product. Gowrisankaran (1999a) develops a model of endogenous horizontal mergers where \( \omega_i \) 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 \( \omega_i \).

### 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 move 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, \((\bar{\omega} + 1)\bar{N}\), to the number of possible distributions of \(\bar{N}\) 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 \(\bar{N}\):

\[
S = \{(s_1, \ldots, s_{\bar{\omega} + 1}) : \sum_j s_j = \bar{N}, s_j \in \mathbb{Z}^*\},
\]

where \(\mathbb{Z}^*\) is the set of nonnegative integers. In this notation, each vector \(\omega = (\omega_1, \ldots, \omega_{\bar{N}}) \in \Omega^{\bar{N}}\) maps to an element \(s = (s_1, \ldots, s_{\bar{\omega} + 1}) \in S\) with \(s_j = \sum_{i=1}^{\bar{N}} 1\{\omega_i = j\}\) for each \(j\).

In practice we map the multidimensional space \(S\) to an equivalent one-dimensional state space \(\mathcal{X} = \{1, \ldots, |S|\}\). Payoff relevant market configurations from the perspective of firm \(i\) are then uniquely described by two integers \((x, \omega_i)\), where \(x \in \mathcal{X}\) denotes the market structure and \(\omega_i\) is firm \(i\)’s own quality level.

### 4.1.2 Product Market Competition

Again, we follow Pakes and McGuire (1994) in assuming a continuum of consumers with measure \(M > 0\) and that consumer \(j\)’s utility from choosing the good produced by firm \(i\) is \(g(\omega_i) - p_i + \varepsilon_i\), where \(\varepsilon_i\) is iid across firms and consumers and follows a type I extreme value distribution. Pakes and McGuire (1994) use the \(g\) function to enforce an upper bound on profits. As in Pakes, Gowrisankaran, and McGuire (1993), for some constant \(\omega^*\) we specify 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 \(\omega\) and prices \(p\). From McFadden (1974), we know that the share of consumers purchasing good \(i\) is

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

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

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

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

---

9 In particular, we use the “probability density space” encoding algorithm described in Gowrisankaran (1999b), to map market structure tuples \(s \in S\) to integers \(x \in \mathcal{X}\).
Caplin and Nalebuff (1991) show that (in this single-product firm setting) there is a unique Bertrand-Nash equilibrium, which is 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

\[-(p_j - c)(1 - c_j) + 1 = 0.\]

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 \((\omega_i, \omega_{-i})\).

### 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 \(\kappa\) in order to increase the quality of their product from \(\omega_i\) to \(\omega'_i = \min\{\omega_i + 1, \bar{\omega}\}\), or they may exit the market and receive some scrap value \(\eta\). We denote these choices, respectively, by the choice set \(\mathcal{A}_i = \{0, 1, 2\}\). When an incumbent firm exits the market, \(\omega_i\) jumps deterministically to \(\bar{\omega} + 1\). Associated with each choice \(j\) is a private shock \(\varepsilon_{ijt}\). 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 \(\omega_k = (\omega_{1k}, \ldots, \omega_{Nk})\) denote its counterpart in \(\Omega^N\). In the general notation introduced above, the instantaneous payoff \(\psi_{ijk}\) 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 \(\omega_{k'i} = \min\{\omega_{ki} + 1, \bar{\omega}\}\) and \(\omega_{k'j} = \omega_{kj}\) for all
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

\[
V_{ik} = E \left[ \int_0^\tau e^{-\rho t} \pi_{ik} \, dt + e^{-\rho \tau} \left( \sum_{l \neq k} q_{kl} V_{il} + \sum_{l \neq i} \lambda \sum_{j=1}^J \sigma_{ljk} \sum_{m=1}^K \phi_{ljkm} V_{im} \right) \right]
\]

where \( \pi \) 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 \( \tau \) and \( \varepsilon_{ij} \) for all \( i \) and \( j \). Conditional upon moving while in state \( k \), incumbent firms face the following maximization problem:

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

The resulting choice probabilities are

\[
\sigma_{i0k} = \frac{\exp(V_{ik})}{\exp(V_{ik}) + \exp(-\kappa + V_{ik'}) + \exp(\eta)},
\]

\[
\sigma_{i1k} = \frac{\exp(-\kappa + V_{ik'})}{\exp(V_{ik}) + \exp(-\kappa + V_{ik'}) + \exp(\eta)},
\]

\[
\sigma_{i2k} = 1 - \sigma_{i0k} - \sigma_{i1k},
\]

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

### 4.1.4 Potential Entrants

Whenever the number of incumbent is smaller than \( \bar{N} \), a single potential entrant receives the opportunity to enter at rate \( \lambda \). Potential entrants are short-lived and do not consider the option value of delaying entry. The potential entrant is counted in \( N_k \), the total number of active firms in state \( k \), and thus the rate at which incumbents receive the opportunity to move is \( (N_k - 1)\lambda \) but the rate at which any type of move opportunity occurs is \( N_k \lambda \).

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 \( \eta^e \) and entering the market immediately in a predetermined entry state \( \omega^e \in \Omega \) or it can choose not to enter \( (a_i = 0) \) at no cost.

Associated with each choice \( j \) is a stochastic private payoff shock \( \varepsilon^e_{ijt} \). 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 \( X \) 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 \( \psi_{i0k} = V_{i0k} = 0 \). Thus, conditional upon moving in state \( k \), a potential entrant faces the problem

\[
\max \{ \varepsilon_{i0}^e, -\eta^e + V_{ik'} + \varepsilon_{i1}^e \}
\]

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 \( \gamma \), the quality of each firm \( i \) jumps from \( \omega_i \) to \( \omega'_i = \max\{\omega_i - 1, 1\} \). 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 depreciation happens with some probability \( \delta \), implying that the quality of all firms falls on average every \( 1/\delta \) periods. Our assumption of a rate \( \gamma \) 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 \( \gamma \).

### 5 Estimation

Methods that solve for the value function 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 inside of an optimization routine that 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.
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 (or similarly “plugged-in” to a corresponding GMM criterion function).

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. These methods can become quite computationally expensive 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). The Su and Judd (2008) MPEC approach provides one solution to 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. As in Doraszelski and Judd (2008), the use of continuous time breaks one primary curse of dimensionality in that only a single player moves at any particular instant. An attractive and novel feature of our framework is that it is easily estimable using standard CCP methods. This greatly reduces the computational costs of estimation relative to full solution methods. Having estimated a large problem with CCP methods, it is then straightforward to use the model for post-estimation exercises, since the computational burden of computing the equilibrium a few times for these purposes is not as great as nesting several such solutions into an estimation routine. In this way, our framework preserves a tight link between the estimated model and that used for post-estimation analysis, something which has proven infeasible for many empirical applications that have been modeled in discrete time.

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 with imperfectly sampled data. We consider the case when some moves may be unobserved in Section 5.2, and in Section 5.3 we consider the case where the data is only sampled at discrete intervals. We consider two-step CCP-based estimation in Section 5.4 and close with a discussion of extensions to models which admit unobserved heterogeneity in Section 5.5.

---

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.
5.1 Full-Solution Estimation

We begin our discussion of estimation by considering the application of a full solution method in a setting where continuous time data is available. In particular, suppose we observe all continuous-time events in a single market over the time interval \([0, T]\). Let \(N\) denote the total number of observed events during the observation window, that is, the number of actions (moves) plus the number of exogenous state changes (jumps). The \(n\)-th event is characterized by five pieces: the elapsed time since the last event, \(\tau_n\), the index of the player associated with the event, \(i_n\) (where \(i_n = 0\) indicates a move by nature), the observed choice, \(a_n\), and the states immediately before and after the event, \(x_n\) and \(x'_n\).

Let \(\ell_n(\theta)\) denote the likelihood of the \(n\)-th event where \(\theta\) is the vector of parameters of interest which includes the move-arrival rates, the parameters of the process generating exogenous state changes, and the parameters of the payoff functions. The likelihood function depends on the aggregate and individual intensity matrices, which are in turn functions of \(\theta\). We let \(q(x, x'; \theta)\) denote the absolute value of the \((x, x')\) element of the aggregate intensity matrix \(Q = Q_0 + \sum_{i=1}^{N} Q_i\). Taking the absolute value serves to ensure that all rates are positive, since the diagonal elements of the aggregate intensity matrix, the rates of leaving a particular state, are negative. Define \(q_0(x, x'; \theta)\) and \(q_N(x, x'; \theta)\) similarly for the matrices \(Q_0\) and \(\sum_{i=1}^{N} Q_i\).

The likelihood of a single event consists of several parts, the first of which is the pdf of the inter-event interval, \(g(\tau_n; q(x_n, x_n; \theta))\), where \(g\) is the pdf of the exponential distribution. The rate parameter in this case is \(q(x_n, x_n; \theta)\), the negative of the diagonal element of the aggregate intensity matrix \(Q\), which is the overall rate at which the process leaves state \(x_n\). The remaining components of the likelihood for jumps (indicated by \(i_n = 0\)) are the probability that the event is indeed a jump, \(q_0(x_n, x_n; \theta)/q(x_n, x_n; \theta)\), and the state transition probability, \(p(x_n, x'_n; \theta) \equiv q_0(x_n, x'_n; \theta)/q_0(x_n, x_n; \theta)\). On the other hand, for moves (indicated by \(i_n > 0\)), the remaining components are the probability that the event is a move, \(q_N(x_n, x_n; \theta)/q(x_n, x_n; \theta)\), the choice probability, \(\sigma(i_n, a_n, x_n; \theta)\), and the probability of the resulting state transition, \(\phi(i_n, a_n, x_n, x'_n; \theta)\).\(^{11}\)

Combining these components, the likelihood of a single event is

\[
\ell_n(\theta) = g(\tau_n; q(x_n, x_n; \theta)) \left[ \frac{q_0(x_n, x_n; \theta)}{q(x_n, x_n; \theta)} \cdot p(x_n, x'_n; \theta) \right]^{1\{i_n=0\}} \times \left[ \frac{q_N(x_n, x_n; \theta)}{q(x_n, x_n; \theta)} \cdot \sigma(i_n, a_n, x_n; \theta) \cdot \phi(i_n, a_n, x_n, x'_n; \theta) \right]^{1\{i_n>0\}}.
\]

When the last observation occurs before the end of the observation window (i.e., when \(t_N < T\)), we must also account for the fact that no event was observed over the interval

\(^{11}\)In many cases, the state will change deterministically with the choice.
The likelihood of this is the probability of observing no events over an interval of length $T - t_N$ while the state was $x_N$, given by $1 - G(T - t_N; q(x_N, x_N; \theta))$, where $G$ is the cdf of the exponential distribution. Thus, after simplifying the observation likelihood, the log likelihood for the time period $[0, T]$ is

$$\ln L_T(\theta) = \sum_{n=1}^{N} \ln g(\tau_n; q(x_n, x_n; \theta)) + \sum_{n=1}^{N} 1\{i_n = 0\} \ln q_0(x_n, x'_n; \theta)$$

$$+ \sum_{n=1}^{N} 1\{i_n > 0\} \left[ \ln q(x_n, x_n; \theta) + \ln \sigma(i_n, a_n, x_n; \theta) + \ln \phi(i_n, a_n, x_n, x'_n; \theta) \right]$$

$$- \sum_{n=1}^{N} \ln q(x_n, x_n; \theta) + \ln \left[ 1 - G(T - t_N, q(x_N, x_N; \theta)) \right].$$

We have replaced the transition probabilities by the relevant elements of the intensity matrices, canceled the $q_0(x_n, x_n; \theta)$ terms for move observations, and collected the common $q(x_n, x_n; \theta)$ terms. Note that it is straightforward to generalize this to the case where multiple markets are observed over potentially market-specific time horizons.

### 5.2 Partially Observed Moves

We continue using the same notation as in the previous sections, but now we suppose that the choice set is $\mathcal{A} = \{0, \ldots, J - 1\}$ and that only actions $a_n$ for which $a_n > 0$ are observed by the econometrician, where, without loss of generality, $a_n = 0$ denotes the unobserved action.\footnote{We also assume that information about the unobserved action is not revealed through changes in the state, that is, the state remains constant following the choice $a_n = 0$. Stated formally, for all $x_n$ and $i_n > 0$, $\phi(i_n, 0, x_n, x_n; \theta) = 1$.} This assumption is likely to be the relevant one in most empirical settings with continuous-time data, as we generally expect the arrival of the right to move to be latent.\footnote{Estimation with time-aggregated data, discussed in Section 5.3, naturally accounts for this.} This complicates the estimation as now we only observe the truncated joint distribution of move arrival times and actions. Estimating $\lambda$ using only the observed move times for observations with $a_n > 0$ 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 $\tau_n$ 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 $x_n$, the choice probabilities for each action, $\sigma(i_n, a_n, x_n; \theta)$, are also constant. On this interval, conditional on receiving a move arrival, the move will be observed by the researcher with probability $1 - \sigma(i_n, 0, x_n; \theta)$.

For a given state $x_n$ 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 \( N_a(t) \) 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 \( \lambda \). 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 5.1.** Let the state of the model be \( x \) and let \( \sigma(i, a, x) \) denote the choice probability of player \( i \) for action \( a \) in state \( x \). Then \( W_+(t) \) has an exponential distribution with rate parameter \((1 - \sigma(i, 0, x)) \lambda\).

**Proof.** See Appendix A. ■

The primary difference here is that we will use \( \bar{x}_{xa} = [1 - \sigma(i_n, 0, x_n; \theta)] \lambda \), the rate of observed moves in state \( x_n \), to construct an intensity matrix for the observed processes, which will be used for estimation. We let \( \bar{q}(x, x'; \theta) \) denote the absolute value of the \((x, x')\) element of the resulting intensity matrix \( \bar{Q} \). Similarly, the choice probability \( \sigma(i_n, a_n, x_n; \theta) \) is replaced by the choice probability conditional on having observed the choice,

\[
\bar{\sigma}(i_n, a_n, x_n; \theta) = \frac{\sigma(i_n, a_n, x_n; \theta)}{1 - \sigma(i_n, 0, x_n; \theta)}.
\]

As before, we first write the likelihood for a single observation \((\tau_n, i_n, a_n, x_n, x'_n)\):

\[
\ell_n(\theta) = g(\tau_n; \bar{q}(x_n, x'_n; \theta)) \left[ \frac{\bar{q}(x_n, x'_n; \theta)}{\bar{q}(x_n, x_n; \theta)} \cdot p(x_n, x'_n; \theta) \right]^{1_{\{i_n=0\}}} \\
\times \left[ \frac{\bar{q}_N(x_n, x'_n; \theta)}{\bar{q}(x_n, x_n; \theta)} \cdot \bar{\sigma}(i_n, a_n, x_n; \theta) \cdot \phi(i_n, a_n, x_n, x'_n; \theta) \right]^{1_{\{i_n>0\}}}.
\]

Estimation can now proceed as before by constructing and maximizing the log-likelihood function of the full sample.

### 5.3 Time Aggregation

Having considered settings with partially observed moves, now suppose we only observe the process at \( N \) discrete points in time \( \{t_1, t_2, \ldots, t_N\} \). Let \( \{x_1, x_2, \ldots, x_N\} \) denote the
corresponding states. Through the aggregate intensity matrix $Q \equiv Q_0 + \sum Q_i$, 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 structural parameters $\theta$, which appear in $Q$ both directly and indirectly through the conditional choice probabilities $\sigma_{ijk}$. In this section, we describe a full-solution approach in which the value function is solved for each value of $\theta$ in order to obtain the implied CCPs which, in turn, are used to construct $Q$.

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 of observations $(t_{n-1}, x_{n-1})$ and $(t_n, x_n)$. That is, $P_{x_{n-1},x_n}(t_n - t_{n-1})$ is the probability of the process moving from $x_{n-1}$ to $x_n$ after an interval of length $t_n - t_{n-1}$. This includes cases where $x_n = x_{n-1}$ since the transition probabilities account for there having been no 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 $\{(t_n, x_n)\}_{n=1}^N$ is thus

$$\ln L_N(\theta) = \sum_{n=1}^N \ln P_{x_{n-1},x_n}(t_n - t_{n-1}).$$

In this way, dealing with time aggregation is remarkably simple in continuous time. Because transition probabilities for an interval of any length can easily be calculated from the underlying intensity matrix, it is straightforward to characterize the likelihood of transitioning from any observed initial state to any observed final state over any given interval. This approach naturally accounts for all of the myriad ways that such a transition could have occurred, something which would be much more computationally demanding in discrete time.\(^{14}\)

To illustrate the issues involved, Figure 4 displays two distinct paths which coincide both before and after an interval of length $\Delta$, 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 $P(\Delta)$.

For example, 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 level at rate $q_1$ and two levels at rate $q_2$. Then, $Q_0$ for this

\(^{14}\)In a discrete-time, sequential-move model with a prespecified number of sub-periods, time aggregation would require enumerating and integrating over all possible paths. Conveniently, in continuous time this information is summarized by the matrix exponential from Section 2.
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 \( \Delta \).

The model is

\[
Q_0 = \begin{bmatrix}
-q_1 - q_2 & q_1 & q_2 & 0 & 0 \\
0 & -q_1 - q_2 & q_1 & q_2 & 0 \\
0 & 0 & -q_1 - q_2 & q_1 & q_2 \\
0 & 0 & 0 & -q_1 - q_2 & q_1 + q_2 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Let \( \sigma_k \) denote the conditional choice probability of choosing to renew—moving the state back to 1 deterministically—in state \( k \). Note that \( \sigma_k \) is determined endogenously and depends on the parameters \( \theta \) through the value function as in (5). Under our assumptions, \( \sigma_k \) will have a logit form. If \( \lambda \) 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 \( \lambda \) 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.

5.3.1 Embeddability and Identification

There are two fundamental issues regarding estimation using time aggregated data, which arise in both the continuous-time and discrete-time settings. The first is the embeddability problem: could the observed discrete-time transition matrix $P(\Delta)$, associated with a time interval of length $\Delta$, have in fact been generated by the proposed data generating process (some continuous-time Markov structure with intensity matrix $Q$ or some discrete-time chain over fixed time periods of length $\delta$). If so, then we can address the second question regarding identification: whether there is a unique parametrization of the underlying model which, when sampled at intervals of length $\Delta$, gives rise to the transition matrix $P(\Delta)$.

In the continuous-time framework, let $Q$ be the set of all valid intensity matrices and $P_0 \equiv \{\exp(Q) : Q \in Q\}$ be the set of all possible generated transition matrices. The question of embeddability asks whether $P(\Delta) \in P_0$. Singer and Spilerman (1976) provide several necessary conditions for embeddability involving testable conditions on the determinant and eigenvalues of $P(\Delta)$. Essentially, this is simply a model specification issue: was the data actually generated by a continuous time process? In this paper, we assume that the model is correctly specified and therefore, such a $Q$ matrix exists.

The problem of identification is about whether there is a unique matrix $Q \in Q$ such that $P(\Delta) = \exp(Q)$. Singer and Spilerman (1976) provide several sufficient conditions that must be verified to guarantee that $Q$ is unique, for example if the eigenvalues of $P(\Delta)$ are distinct, real, and positive, if $\min_{i} \{P_{ii}(\Delta)\} > 1/2$, or if $\det P(\Delta) > e^{-\pi}$.

Although these two issues are typically discussed in the context of continuous-time models, both apply to discrete-time models as well. Consider a discrete-time model with a fixed move interval of length $\delta$ which may be different from the fixed interval $\Delta$ at which observations are sampled. In practice, researchers typically assume (implicitly) that $\delta = \Delta$, where $\Delta = 1$ is normalized to be some specific unit of time (e.g., one quarter). This assumption is convenient but effectively assumes away the embeddability and identification problems.

In this case, the issue of embeddability is then whether the observed discrete time transition matrix $P(\Delta)$ could have in fact been generated by (another) discrete time transition matrix over intervals of length $\delta$. Stated somewhat differently, embeddability is satisfied if

\footnotesize
\begin{itemize}
  \item This problem was first proposed by Elfving (1937). Kingman (1962) derived the set of embeddable processes with $K = 2$ and Johansen (1974) gave an explicit description of the set for $K = 3$.
  \item Our continuous-time model is more flexible than the discrete-time model in the sense that we estimate the rate of move arrivals, rather than fixing it at unity. The closest analog to discrete time is found by setting $\lambda \equiv 1$.
\end{itemize}

\normalsize
there exists a matrix $P_0$ such that $P_0^{\Delta / \delta} = P(\Delta)$. In general, the root $P_0$ need not exist. The identification question is then, if the root $P_0$ exists, is it unique? In general there may be multiple such matrices (Singer and Spilerman, 1976, p. 49). These issues become trivial under the aforementioned assumption that $\delta = \Delta$.

### 5.4 CCP-Based Estimation

We introduce CCP estimation in terms of the single-agent model for simplicity. Application to the multi-agent model follows directly and is discussed at the end of this section. CCP estimation relies on finding a mapping from the CCPs $\sigma_{jk}$ 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, application of 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 $\int_0^{\tau_1} e^{-\rho t} \sum_{l=1}^K P_{kl}(t) u_l dt$.

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

$$V_k = E \left[ \int_0^{\tau_1} e^{-\rho t} \sum_{l=1}^K P_{kl}(t) u_l dt + e^{-\rho \tau_1} \sum_{l=1}^K P_{kl}(\tau_1) \max_j \{ \psi_{jl} + \varepsilon_j + v_{jl} \} \right].$$

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. These representations are equivalent.\(^{17}\)

The first term above represents the flow utility obtained from the initial time until the first move arrival at time $\tau_1$. The second term represents the expected instantaneous and future utility obtained from making a choice at time $\tau_1$. The resulting state $l$ at time $\tau_1$ is stochastic, as is the optimal choice $j$ and, possibly even the continuation value $v_{jl}$. The

\(^{17}\)In Section 2 we noted that the next-move representation introduces matrix exponential calculations, increasing the computational burden of evaluating the functional mapping. However, in the context of CCP estimation this can be completed in a preliminary step and only needs to be carried out for the states actually observed in the data and the neighboring states required for calculating expectations.
expectation operator is needed because $\tau_1$ is also random and unknown a priori.

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

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

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

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

The $\psi_{jk}$’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 (10) we can write

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

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

### 5.4.1 Single-Agent Example

In many model specifications we can then obtain an expression for the differences in (11) by choosing an appropriate normalizing state.\(^{18}\) We use the example model of Section 3.1 to illustrate this point. In terms of this model, we can write (12), for $j = 1$ as

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

Note that the left-hand side of the above equation is exactly the expression in the value function as expressed in (4). Substituting (13) into (4) gives the following expression for the value function for each state $k$:

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

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

\(^{18}\)See Arcidiacono and Miller (2008) for a general discussion.
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

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

This expression gives differences in the value function in terms of the conditional choice probability \( \sigma_{1l} \). With first-stage estimates of \( \sigma_{1l} \) 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 \( \theta \).

### 5.4.2 Multi-Agent Models

In dynamic games, in the interval between an arbitrary time \( t < \tau_i \) and \( \tau_i \), 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 \( \lambda_i \) which is constant across \( k \).

Thus, the rate at which the model leaves state \( k \) due to player \( i \) is \( \lambda_i \). 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 \( \lambda_i \), 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 \( Q_{-i} \):
\[ \int_0^{\tau_i} e^{-\rho t} \sum_{l=1}^K P_{kl}^{-i}(t) u_{il} dt. \] The value function for player \( i \) is then

\[
V_{ik} = E \left[ \int_0^{\tau_i} e^{-\rho t} \sum_{l=1}^K P_{kl}^{-i}(t) u_{il} dt + e^{-\rho \tau_i} \sum_{l=1}^K P_{kl}^{-i}(\tau_i) \max_j \{ \psi_{ijl} + \varepsilon_{ij} + v_{ijl} \} \right].
\] (14)

CCP estimation of the quality ladder model, for example, 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.

### 5.4.3 Computational Issues

There are several computational issues to consider when evaluating the next-move based value function in both 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 (9). The expectation of the \( \max \{ \cdot \} \) term 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 \( E_{\tau_1, \varepsilon} \) with \( E_{\tau_1} E_{\varepsilon | \tau_1} \). If we then make the standard assumption that the \( \varepsilon_j \) 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 \( \{ \delta_1 + \varepsilon_1, \ldots, \delta_J + \varepsilon_J \} \):

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

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 \( \tau_1 \). We describe two Monte Carlo integration methods for doing so. The first involves simulating from the distribution of \( \tau_1 \) 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 the 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 \( \tau_1 \) using Monte Carlo
integration by drawing \( R \) values of \( \tau_1, \{ \tau_1^s \}_{s=1}^R \), and forming the following approximation:

\[
V_k \approx \frac{1}{R} \sum_{s=1}^R \left[ \int_0^{\tau_1^s} e^{-\rho t} \sum_{l=1}^K P_{kl}(t)u_l \, dt + e^{-\rho \tau_1^s} \sum_{l=1}^K P_{kl}(\tau_1^s) \text{E}_{max} j \{ \psi_{jk} + \varepsilon_j + v_{jk} \} \right].
\] (15)

In matrix notation, the flow utility term has a relatively simple closed form which allows (15) to be calculated directly. To see this, let \( b_i(\tau_1) = \int_0^{\tau_1} e^{-\rho s} \sum_j P_{ij}(s)u(x_j) \, ds \), \( B(\tau_1) = (b_1(\tau_1), b_2(\tau_1), \ldots, b_K(\tau_1))^\top \) and \( U = (u(x_1), \ldots, u(x_n))^\top \). Define \( C \equiv -(\rho I - Q) \) for simplicity. Then we can write the first term inside the expectation in matrix notation as

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

Finally, substituting for \( C \) we have

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

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 \( \tau_1 \)). 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.19

Consider, for example, the bus engine problem but where certain buses now 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) apply to the continuous time setting as well.

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 $\beta$, the intensity matrix (mileage transition) parameters $q_1$ and $q_2$, the reset (engine replacement) cost $c$, the discount rate $\rho$, and the move arrival rate $\lambda$. 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 $\theta$ 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. In each experiment we fixed the discount rate, $\rho = 0.05$, the number of states, $K = 10$, and the number of draws used for Monte Carlo integration, $R = 250$. 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 estimated quite precisely. The loss in precision from moving away from continuous time data is initially greatest for the move arrival rate, $\lambda$, yet all estimates of this parameter are still very 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.

---

19See Keane and Wolpin (1997), Eckstein and Wolpin (1999), Arcidiacono (2005) and several others.
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 Appendix B. Using CCPs increases the standard deviations slightly, reflecting noise from the first stage. However, the estimates are still very good, particularly 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 $\pi$, 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 $\beta$, 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. In all experiments, as before, we fixed $\rho = 0.05$ and used $R = 250$ draws for Monte Carlo integration. Further details can again be found in Appendix B.

Table 4 summarizes the results for full-solution estimation, where we obtain the value function using value function iteration for each trial value of $\theta$. 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 $\tilde{N}$, the market size $M$, and 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.\footnote{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.}

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 or 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 $\lambda$ and $\gamma$. 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 $\kappa$, $\eta$, and $\eta^p$ via full solution estimation. For each new trial value of $\theta$, we use the value function at the previous $\theta$ 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 5.4.3 (with $R = 250$), the time required to estimate the second-stage parameters, and the sum of these two times (the total time).\footnote{This table does not address the time required to estimate the first-stage CCPs, which can vary significantly depending on which method is used. Parametric methods can clearly be quite fast while fully nonparametric methods can be computationally intensive.}

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 9 levels of quality) using the basic Pakes-McGuire algorithm.\footnote{Similar computational times are also reported in Doraszelski and Pakes (2007).} Our continuous-time approach takes about 20 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 methods have made it possible to estimate large-scale dynamic games, performing simulations for counterfactual work or generating data remains severely limited by the curse of dimensionality that arises from simultaneous moves. We recast the standard discrete-time, simultaneous-move game as a sequential-move game in continuous time. This breaks the curse of dimensionality, greatly expanding the breadth and applicability of these structural methods and making even full-solution estimation feasible for very large games.

Furthermore, by building on an underlying discrete-choice random utility model, our model preserves many of the desirable features of discrete-time models. In particular, we show that the insights from two-step estimation methods can be applied directly in our
framework, resulting in another order of magnitude computational gain during estimation. We also show how to extend the model 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 models with discrete state spaces. Future work is needed to extend these models to continuous state spaces, which 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 2.1. The result follows directly from the joint distribution function:

\[
\Pr(\tau \leq t) = \Pr\left( \min_i \tau_i \leq t \right) = 1 - \Pr(\tau_1 > t, \ldots, \tau_n > t) \\
= 1 - \prod_{i=1}^{n} \Pr(\tau_i > t) = 1 - \prod_{i=1}^{n} e^{-\lambda_i t} = 1 - e^{-\left(\sum_{i=1}^{n} \lambda_i\right)t}.
\]

Therefore, \( \tau \) has an exponential distribution with rate parameter \( \sum_{i=1}^{n} \lambda_i \).

Furthermore,

\[
\Pr(\tau_i \leq \tau_j \forall j) = \mathbb{E}_{\tau_i} \left[ \Pr(\tau_j \geq \tau_i \forall j \neq i) \mid \tau_i \right] \\
= \int_{0}^{\infty} \left[ e^{-\sum_{j \neq i} \lambda_j} \lambda_i e^{-\lambda_i \tau_i} \right] d\tau_i \\
= \int_{0}^{\infty} \lambda_i e^{-\left(\sum_{j=1}^{n} \lambda_j\right)\tau_i} d\tau_i \\
= -\frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \left[ e^{-\left(\sum_{j=1}^{n} \lambda_j\right)\tau_i} \right]_{\tau_i=0}^{\infty} \\
= \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}.
\]
Proof of Proposition 5.1. We have

\[
\Pr(W_+(t) \geq \tau) = \Pr[N_+(t + \tau) - N_+(t) = 0] \\
= \sum_{k=0}^{\infty} \Pr[N(t + \tau) - N(t) = k, N_0(t + \tau) - N_0(t) = k] \\
= \sum_{k=0}^{\infty} \Pr[N(t + \tau) - N(t) = k] \sigma(0, x)^k \\
= \sum_{k=0}^{\infty} \frac{e^{-\lambda\tau} (\lambda\tau)^k}{k!} \sigma(0, x)^k \\
= e^{-\lambda\tau} \sum_{k=0}^{\infty} \frac{(\sigma(0, x)\lambda\tau)^k}{k!} \\
= e^{-\lambda\tau} e^{\sigma(0, x)\lambda\tau} \\
= e^{-(1 - \sigma(0, x))\lambda\tau},
\]

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

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

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

\[\blacksquare\]

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 \(\theta\) and then use numerical fixed point methods to determine the value function over the state space \(X\) to within a tolerance of \(\epsilon = 10^{-6}\) in the relative sup norm. To evaluate the expectation over \(\tau\) in (4), we use Monte Carlo integration as described in Section 5.4.3, 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 \(\theta\) while maximizing the likelihood function using the L-BFGS-B algorithm (Byrd, Lu, and Nocedal, 2000).
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 $\Delta \in \{0.625, 1.25, 2.5, 5.0, 10.0\}$.\footnote{While there are more efficient methods to evaluate the expectation over $\tau$ and taking analytic derivatives would clearly speed up estimation, the computational times are so fast that these steps were not needed.}

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 ($\lambda, q_1,$ and $q_2$) and the parameters of a logistic regression model for the probability of renewal with parameters $\alpha$. 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 $\theta = (\lambda, \gamma, \kappa, \eta, \eta')$ for each of 25 simulated datasets and report the means and standard deviations (in parenthesis). In all experiments, we hold $\bar{\omega}$ fixed at $\bar{\omega} = 7$, set $\omega^e = \lfloor \frac{\bar{\omega}}{2} \rfloor$, and vary the maximum number of players, $\bar{N}$, and the market size, $M$.\footnote{One could view 63'ss as the number of months in the data with $\{0.625, 1.25, 2.5, 5.0, 10.0\}$ indicating the number of months (or fraction of months) between samples. While 25,000 implies having over 2,000 years of data, this is following one time series. An almost equivalent structure would follow 1000 decision-makers over two years.}

We also increase the market size $M$ so that the average number of active players ($n_{\text{avg}}$) grows with the total number of possible players ($\bar{N}$). The average quality level of active firms is denoted $\omega_{\text{avg}}$. We also report $K$, the number of states from the perspective of player $i$—the number of distinct $(\omega, \omega_i)$ 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.

References

Econometrica 75, 1–53. [2, 19]


with unobserved heterogeneity. Working paper, Duke University. [20, 28, 32]

competition. Econometrica 75, 1331–1370. [2, 19]

complete information. Working paper, University of Minnesota. [20]

[3]

Burdett, K. and D. T. Mortensen (1998). Wage differentials, employer size, and unemploy-

strained optimization. SIAM Journal on Scientific and Statistical Computing 16, 1190–
1208. [36]

etary Economics 12, 383–398. [3]


Doraszelski, U. and K. L. Judd (2008). Avoiding the curse of dimensionality in dynamic
stochastic games. Working paper, Harvard University. [3, 20, 34, 37]


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 \( \Delta \) 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>
<thead>
<tr>
<th>Sampling</th>
<th>( n )</th>
<th>( q_1 )</th>
<th>( q_2 )</th>
<th>( \lambda )</th>
<th>( \beta )</th>
<th>( c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>( \infty )</td>
<td>0.150</td>
<td>0.050</td>
<td>0.200</td>
<td>1.000</td>
<td>1.250</td>
</tr>
<tr>
<td>Continuous Time</td>
<td>10,000</td>
<td>0.150</td>
<td>0.050</td>
<td>0.200</td>
<td>1.009</td>
<td>1.254</td>
</tr>
<tr>
<td></td>
<td>(0.002)</td>
<td>(0.001)</td>
<td>(0.003)</td>
<td>(0.068)</td>
<td>(0.054)</td>
<td></td>
</tr>
<tr>
<td>Passive Moves</td>
<td>7,176</td>
<td>0.150</td>
<td>0.050</td>
<td>0.204</td>
<td>1.010</td>
<td>1.271</td>
</tr>
<tr>
<td>( \Delta = 0.625 )</td>
<td>40,000</td>
<td>0.137</td>
<td>0.053</td>
<td>0.189</td>
<td>1.107</td>
<td>1.305</td>
</tr>
<tr>
<td></td>
<td>(0.003)</td>
<td>(0.002)</td>
<td>(0.019)</td>
<td>(0.213)</td>
<td>(0.238)</td>
<td></td>
</tr>
<tr>
<td>( \Delta = 1.25 )</td>
<td>20,000</td>
<td>0.145</td>
<td>0.051</td>
<td>0.191</td>
<td>1.074</td>
<td>1.191</td>
</tr>
<tr>
<td></td>
<td>(0.003)</td>
<td>(0.002)</td>
<td>(0.024)</td>
<td>(0.210)</td>
<td>(0.297)</td>
<td></td>
</tr>
<tr>
<td>( \Delta = 2.5 )</td>
<td>10,000</td>
<td>0.147</td>
<td>0.051</td>
<td>0.198</td>
<td>1.014</td>
<td>1.167</td>
</tr>
<tr>
<td></td>
<td>(0.004)</td>
<td>(0.002)</td>
<td>(0.027)</td>
<td>(0.334)</td>
<td>(0.408)</td>
<td></td>
</tr>
<tr>
<td>( \Delta = 5.0 )</td>
<td>5,000</td>
<td>0.151</td>
<td>0.050</td>
<td>0.195</td>
<td>1.088</td>
<td>1.233</td>
</tr>
<tr>
<td></td>
<td>(0.007)</td>
<td>(0.003)</td>
<td>(0.019)</td>
<td>(0.249)</td>
<td>(0.402)</td>
<td></td>
</tr>
<tr>
<td>( \Delta = 10.0 )</td>
<td>2,500</td>
<td>0.158</td>
<td>0.048</td>
<td>0.200</td>
<td>1.010</td>
<td>1.108</td>
</tr>
<tr>
<td></td>
<td>(0.019)</td>
<td>(0.007)</td>
<td>(0.022)</td>
<td>(0.397)</td>
<td>(0.618)</td>
<td></td>
</tr>
</tbody>
</table>
Table 2: Single player Monte Carlo results: CCP estimation ($T = 25,000$).

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

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 $\Delta$ 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, $R = 250$. 

43
Table 3: Single player Monte Carlo results with unobserved heterogeneity.

<table>
<thead>
<tr>
<th>$M$</th>
<th>$n$</th>
<th>$q_1$</th>
<th>$q_{21}$</th>
<th>$q_{22}$</th>
<th>$\pi$</th>
<th>$\lambda$</th>
<th>$\beta$</th>
<th>$c_1$</th>
<th>$c_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>100</td>
<td>0.150</td>
<td>0.050</td>
<td>0.030</td>
<td>0.700</td>
<td>0.200</td>
<td>1.000</td>
<td>1.000</td>
<td>2.000</td>
</tr>
<tr>
<td>25</td>
<td>(0.006)</td>
<td>(0.004)</td>
<td>(0.005)</td>
<td>(0.115)</td>
<td>(0.005)</td>
<td>(0.303)</td>
<td>(0.111)</td>
<td>(0.255)</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>(0.004)</td>
<td>(0.003)</td>
<td>(0.004)</td>
<td>(0.070)</td>
<td>(0.004)</td>
<td>(0.188)</td>
<td>(0.067)</td>
<td>(0.141)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>(0.003)</td>
<td>(0.002)</td>
<td>(0.002)</td>
<td>(0.058)</td>
<td>(0.003)</td>
<td>(0.137)</td>
<td>(0.049)</td>
<td>(0.107)</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>(0.003)</td>
<td>(0.003)</td>
<td>(0.003)</td>
<td>(0.092)</td>
<td>(0.004)</td>
<td>(0.176)</td>
<td>(0.061)</td>
<td>(0.118)</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>(0.003)</td>
<td>(0.002)</td>
<td>(0.002)</td>
<td>(0.073)</td>
<td>(0.003)</td>
<td>(0.136)</td>
<td>(0.041)</td>
<td>(0.102)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>(0.002)</td>
<td>(0.001)</td>
<td>(0.002)</td>
<td>(0.047)</td>
<td>(0.002)</td>
<td>(0.096)</td>
<td>(0.029)</td>
<td>(0.062)</td>
<td></td>
</tr>
</tbody>
</table>

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, $R = 250$. 

44
The mean and standard deviation (in parenthesis) of the parameter estimates for 25 different samples are shown for choices of $\tilde{N}$, the total number of players, and $M$, the market size, with $\bar{\omega}$ fixed at 7. $K$ denotes the total number of distinct states, $n_{\text{avg}}$ denotes the average number of active players, and $\omega_{\text{avg}}$ 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>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>$M$</th>
<th>$n_{\text{avg}}$</th>
<th>$\omega_{\text{avg}}$</th>
<th>$\lambda$</th>
<th>$\gamma$</th>
<th>$\kappa$</th>
<th>$\eta$</th>
<th>$\eta^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.800</td>
<td>0.200</td>
<td>0.800</td>
<td>4.000</td>
<td>5.000</td>
</tr>
<tr>
<td>10</td>
<td>80,080</td>
<td>5.0</td>
<td>6.62</td>
<td>3.79</td>
<td>1.820</td>
<td>0.201</td>
<td>0.798</td>
<td>3.986</td>
<td>4.967</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.005)</td>
<td>(0.001)</td>
<td>(0.026)</td>
<td>(0.204)</td>
<td>(0.171)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>136,136</td>
<td>7.0</td>
<td>7.79</td>
<td>3.62</td>
<td>1.819</td>
<td>0.201</td>
<td>0.791</td>
<td>3.990</td>
<td>4.952</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.005)</td>
<td>(0.002)</td>
<td>(0.031)</td>
<td>(0.198)</td>
<td>(0.174)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>222,768</td>
<td>8.0</td>
<td>8.29</td>
<td>3.47</td>
<td>1.821</td>
<td>0.201</td>
<td>0.798</td>
<td>4.010</td>
<td>5.007</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.006)</td>
<td>(0.001)</td>
<td>(0.024)</td>
<td>(0.192)</td>
<td>(0.163)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>352,716</td>
<td>9.0</td>
<td>8.81</td>
<td>3.35</td>
<td>1.821</td>
<td>0.200</td>
<td>0.801</td>
<td>4.043</td>
<td>5.044</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.006)</td>
<td>(0.001)</td>
<td>(0.031)</td>
<td>(0.184)</td>
<td>(0.157)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>542,640</td>
<td>10.0</td>
<td>9.32</td>
<td>3.22</td>
<td>1.821</td>
<td>0.200</td>
<td>0.801</td>
<td>4.043</td>
<td>5.044</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.006)</td>
<td>(0.001)</td>
<td>(0.031)</td>
<td>(0.184)</td>
<td>(0.157)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>813,960</td>
<td>11.0</td>
<td>9.86</td>
<td>3.13</td>
<td>1.822</td>
<td>0.201</td>
<td>0.811</td>
<td>4.012</td>
<td>5.047</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.005)</td>
<td>(0.002)</td>
<td>(0.073)</td>
<td>(0.211)</td>
<td>(0.257)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1,193,808</td>
<td>13.0</td>
<td>10.89</td>
<td>3.00</td>
<td>1.822</td>
<td>0.200</td>
<td>0.837</td>
<td>3.967</td>
<td>5.152</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.005)</td>
<td>(0.001)</td>
<td>(0.103)</td>
<td>(0.223)</td>
<td>(0.402)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1,716,099</td>
<td>15.0</td>
<td>11.88</td>
<td>2.90</td>
<td>1.820</td>
<td>0.201</td>
<td>0.836</td>
<td>3.984</td>
<td>5.181</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.006)</td>
<td>(0.002)</td>
<td>(0.107)</td>
<td>(0.201)</td>
<td>(0.363)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>2,422,728</td>
<td>17.0</td>
<td>12.90</td>
<td>2.81</td>
<td>1.821</td>
<td>0.200</td>
<td>0.808</td>
<td>3.999</td>
<td>5.030</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.006)</td>
<td>(0.002)</td>
<td>(0.060)</td>
<td>(0.195)</td>
<td>(0.268)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>3,364,900</td>
<td>19.0</td>
<td>13.91</td>
<td>2.72</td>
<td>1.820</td>
<td>0.201</td>
<td>0.809</td>
<td>3.987</td>
<td>5.139</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.006)</td>
<td>(0.002)</td>
<td>(0.078)</td>
<td>(0.190)</td>
<td>(0.272)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>4,604,600</td>
<td>21.0</td>
<td>14.92</td>
<td>2.64</td>
<td>1.820</td>
<td>0.200</td>
<td>0.801</td>
<td>4.009</td>
<td>5.129</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(0.006)</td>
<td>(0.0020)</td>
<td>(0.084)</td>
<td>(0.194)</td>
<td>(0.308)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5: Quality ladder Monte Carlo results: CCP estimation.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>$M$</th>
<th>$n_{\text{avg}}$</th>
<th>$\omega_{\text{avg}}$</th>
<th>$\lambda$</th>
<th>$\gamma$</th>
<th>$\kappa$</th>
<th>$\eta$</th>
<th>$\eta^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>80,080</td>
<td>5.0</td>
<td>6.62 3.79</td>
<td>1.822 0.202 0.800 4.000</td>
<td>5.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>136,136</td>
<td>7.0</td>
<td>7.79 3.62</td>
<td>1.821 0.202 0.774 4.099</td>
<td>5.080</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>222,768</td>
<td>8.0</td>
<td>8.29 3.47</td>
<td>1.823 0.202 0.775 4.088</td>
<td>5.086</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>352,716</td>
<td>9.0</td>
<td>8.81 3.35</td>
<td>1.823 0.202 0.779 4.076</td>
<td>5.071</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>542,640</td>
<td>10.0</td>
<td>9.32 3.22</td>
<td>1.823 0.202 0.780 4.076</td>
<td>5.069</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>813,960</td>
<td>11.0</td>
<td>9.86 3.13</td>
<td>1.824 0.202 0.782 4.068</td>
<td>5.060</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1,193,808</td>
<td>13.0</td>
<td>10.89 3.00</td>
<td>1.824 0.202 0.787 4.067</td>
<td>5.060</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1,716,099</td>
<td>15.0</td>
<td>11.88 2.90</td>
<td>1.822 0.202 0.782 4.066</td>
<td>5.064</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>2,422,728</td>
<td>17.0</td>
<td>12.90 2.81</td>
<td>1.823 0.202 0.784 4.070</td>
<td>5.064</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>3,364,900</td>
<td>19.0</td>
<td>13.91 2.72</td>
<td>1.822 0.202 0.782 4.055</td>
<td>5.047</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>4,604,600</td>
<td>21.0</td>
<td>14.92 2.64</td>
<td>1.822 0.201 0.783 4.061</td>
<td>5.050</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 $\omega$ fixed at 7. $K$ denotes the total number of distinct states, $n_{\text{avg}}$ denotes the average number of active players, and $\omega_{\text{avg}}$ 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.

<table>
<thead>
<tr>
<th>$\bar{N}$</th>
<th>$M$</th>
<th>$K$</th>
<th>Solve $V$</th>
<th>First Stage</th>
<th>NFXP</th>
<th>CCP</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Estimation</td>
<td>Setup</td>
<td>Estimation</td>
<td>Total</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>2310</td>
<td>9.05</td>
<td>0.42</td>
<td>51.61</td>
<td>0.51</td>
<td>0.30</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>5544</td>
<td>15.59</td>
<td>0.30</td>
<td>107.41</td>
<td>1.54</td>
<td>0.32</td>
<td>1.86</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2.0</td>
<td>12012</td>
<td>29.26</td>
<td>0.28</td>
<td>172.40</td>
<td>2.17</td>
<td>0.47</td>
<td>2.64</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3.0</td>
<td>24024</td>
<td>58.27</td>
<td>0.30</td>
<td>256.19</td>
<td>4.02</td>
<td>0.62</td>
<td>4.64</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4.0</td>
<td>45045</td>
<td>107.40</td>
<td>0.26</td>
<td>375.26</td>
<td>5.58</td>
<td>1.01</td>
<td>6.59</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5.0</td>
<td>80080</td>
<td>185.78</td>
<td>0.35</td>
<td>535.83</td>
<td>7.13</td>
<td>1.42</td>
<td>8.55</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>7.0</td>
<td>136136</td>
<td>325.28</td>
<td>0.34</td>
<td>639.98</td>
<td>11.20</td>
<td>1.80</td>
<td>13.00</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>8.0</td>
<td>222768</td>
<td>518.57</td>
<td>0.33</td>
<td>1069.52</td>
<td>13.47</td>
<td>3.21</td>
<td>16.69</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>9.0</td>
<td>352716</td>
<td>821.83</td>
<td>0.34</td>
<td>1411.32</td>
<td>14.96</td>
<td>3.63</td>
<td>18.59</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>10.0</td>
<td>542640</td>
<td>1228.98</td>
<td>0.39</td>
<td>2436.61</td>
<td>17.21</td>
<td>4.10</td>
<td>21.31</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>11.0</td>
<td>813960</td>
<td>1719.72</td>
<td>0.38</td>
<td>3413.42</td>
<td>19.67</td>
<td>7.15</td>
<td>26.82</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>13.0</td>
<td>1193808</td>
<td>2499.98</td>
<td>0.44</td>
<td>4765.67</td>
<td>23.85</td>
<td>7.06</td>
<td>30.91</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>15.0</td>
<td>1716099</td>
<td>3642.02</td>
<td>0.43</td>
<td>13513.96</td>
<td>27.28</td>
<td>8.38</td>
<td>35.66</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>17.0</td>
<td>2422728</td>
<td>5109.30</td>
<td>0.41</td>
<td>10807.54</td>
<td>30.93</td>
<td>10.54</td>
<td>41.47</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>19.0</td>
<td>3364900</td>
<td>6929.01</td>
<td>0.43</td>
<td>13737.87</td>
<td>35.67</td>
<td>15.37</td>
<td>51.04</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>21.0</td>
<td>4604600</td>
<td>9377.57</td>
<td>0.41</td>
<td>16069.89</td>
<td>37.26</td>
<td>15.60</td>
<td>52.86</td>
<td></td>
</tr>
</tbody>
</table>

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. $\bar{N}$ 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.