Swapping the Nested Fixed Point Algorithm: A Class of Estimators for Discrete Markov Decision Models

Victor Aguirregabiria
University of Chicago

Pedro Mira
CEMFI

Working Paper No. 9904
February 1999

E-mail addresses: vaguire@midway.uchicago.edu and mira@cemfi.es.

CEMFI, Casado del Alisal 5, 28014 Madrid, Spain.
Tel: 34 91 4290551, fax: 34 91 4291056, www.cemfi.es.

Abstract

This paper proposes a procedure for the estimation of discrete Markov decision models and studies its statistical and computational properties. Our Nested Pseudo-Likelihood method (NPL) is similar to Rust's Nested Fixed Point algorithm (NFXP), but the order of the two nested algorithms is swapped. First, we prove that NPL produces the Maximum Likelihood Estimator under the same conditions as NFXP. Our procedure requires fewer policy iterations at the expense of more likelihood-climbing iterations. We focus on a class of infinite-horizon, partial likelihood problems for which NPL results in large computational gains. Second, based on this algorithm we define a class of consistent and asymptotically equivalent *Sequential Policy Iteration (PI)* estimators, which encompasses both Hotz-Miller's CCP estimator and the partial Maximum Likekihood estimator. This presents the researcher with a "menu" of sequential estimators reflecting a trade-off between finite-sample precision and computational cost. Using actual and simulated data we compare the relative performance of these estimators. In all our experiments the benefits in terms of precision of using a *2-stage PI* estimator instead of *1-stage* (i.e., Hotz-Miller) are very significant. More interestingly, the benefits of MLE relative to *2-stage PI* are small.

1 Introduction

The computational cost of estimating discrete choice dynamic programming models remains an important constraint which has limited their range of applications.[1] There are at least two features that make the estimation of these models substantially more costly than just solving the corresponding dynamic programming problem once. First, observable and unobservable time-invariant individual heterogeneity in preferences or technology imply that the model has to be solved for each type of individual in the sample (e.g., for each combination of sex, cohort, region, family background, etc.). And second, the nested solution-estimation algorithms that are used to obtain the maximum likelihood estimator require one to solve the dynamic decision problems of all types of individuals as many times as the number of iterations needed in the search for parameter estimates. This computational constraint has substantial implications for empirical work. The model specification has to be very parsimonious in terms of state variables and time-invariant explanatory variables. In some contexts, this can lead to important misspecifications.

Recent research has resulted in some important developments in the techniques for the solution of dynamic programming models. Rust (1997a, 1997b) shows that randomization and low discrepancy methods in the discretization of the state space can break the curse of dimensionality in the solution of these models. In other related work, Keane and Wolpin (1994, 1996) recently developed a method which combines interpolation and simulation techniques for the solution and estimation of discrete choice dynamic programming models. The emphasis of our paper is not in the algorithms for the solution of dynamic programs, but in the procedure that searches for the maximum likelihood estimator.

The contribution of our paper is twofold. First, we propose a new estimation procedure which we call Nested Pseudo-Likelihood algorithm (NPL). Our procedure is in the spirit of Rust's (1987, 1988) Nested Fixed Point algorithm (NFXP), but it is different in that *the nesting of the two algorithms (likelihood climbing and fixed point)*

[1] See Eckstein and Wolpin (1989), Rust (1994), and Miller (1997) for excellent surveys on the estimation of these models and their empirical applications.

is swapped. We show that this algorithm produces the maximum likelihood estimator (MLE) under the same conditions as NFXP. However, our nested algorithm does not require repeated solution of the dynamic programming problems. For a widely used class of problems, the computational cost of our procedure is of the same order of magnitude as solving the dynamic programming problems only once. Therefore, as the dimension of the state space increases our nested algorithm produces ML estimates at a much smaller computational cost than NFXP.

Second, we show that when non-parametric estimates of conditional choice probabilities are used as initial guesses of our NPL algorithm, this procedure yields a sequence of consistent and asymptotically equivalent estimators. In particular, in a partial likelihood context Hotz and Miller's estimator is obtained as a by-product of the first iteration. Based on this result we define a class of Policy Iteration (PI) estimators which encompasses both Hotz and Miller's and MLE. The advantage of our nested pseudo-likelihood algorithm is therefore not just a reduction of the computational cost of maximum likelihood estimation. It also presents the researcher with a 'menu' of sequential estimators reflecting a trade-off between computational cost and precision in finite samples.

Our paper builds on and extends previous results by Hotz and Miller (1993) and Manski (1991, 1993). These authors proposed and implemented estimation methods that do not require the explicit solution of the dynamic programming model. Hotz et al (1994) combined this method with a simulation-based approach to obtain an estimator which is computationally more efficient. However, a well known important limitation of these estimators is their finite sample properties. Because they do not fully exploit the structure of the model, relying instead on non-parametric estimates of conditional choice probabilities, they may be very imprecise even under the relatively large sample sizes which are common in micro applications.[2] We extend Hotz-Miller results by showing that their procedure can be applied recursively, and that in the limit it yields the ML estimator.

[2] This limitation was pointed out by Eckstein and Wolpin (1989) and Rust (1994a, 1994b), and it has been illustrated in the context of a Monte Carlo study by Hotz et al (1994).

In Rust's NFXP algorithm the outer algorithm is a "hill climbing" procedure that maximizes the likelihood function. At each iteration of this outer procedure there is an inner fixed point algorithm that solves the dynamic programming model given the current value of the parameters. In our nested method the outer algorithm solves the fixed point problem using a policy iteration method. Each policy iteration is performed with updated parameter values which are obtained from the inner procedure. This consists of a hill climbing algorithm which maximizes a *pseudo-likelihood function* where the discrete choice probabilities depend on a *pseudo-value function.* However, as the outer algorithm converges this pseudo-value function converges to the true value function and the pseudo-likelihood function to the true likelihood function. In some sense our algorithm exploits the following idea: why waste time obtaining an exact solution to the dynamic programming model at each iteration if the initial values of the structural parameters are far away from the final estimates? Why not obtain the full solution of the model gradually, "in parallel" with the computation of parameter estimates? The reason why our algorithm is faster than NFXP is thus quite intuitive as long as policy iterations are computationally more expensive than pseudo-likelihood climbing iterations. This is certainly the case for infinite horizon problems when we use partial likelihood methods as suggested by Rust (1987, 1994), even for relatively small dimensions of the state space and large sample sizes. Our NPL algorithm reduces the number of (outer) policy iterations dramatically at the expense of a larger number of hill-climbing iterations.

Regarding the family of *K-stage Policy Iteration* estimators, we analyze the trade-off between finite sample precision and computational cost with severalmMonte Carlo experiments based on Rust's bus replacement model. We find that the benefits of using the *2-stage PI estimator* instead of 1-stage (i.e., Hotz-Miller) are very significant. A more interesting result is that, in all our experiments, the benefits of using MLE instead of *2-stage PI* are very small. This is the case even when we generate artificially very imprecise initial guesses for the conditional choice probabilities.

The rest of the paper is organized as follows. Section 2 reviews the notation and solution methods for discrete Markov decision processes. In Section 3 we describe

Rust's NFXP algorithm, present our alternative nested procedure, and show that it yields the MLE. In Section 4 we introduce the class of Policy Iteration estimators and we obtain their asymptotic properties. In Section 5 we compare the performance of the NFXP and NPL algorithms using the bus engine dataset in Rust (1987), and present a Monte Carlo study which illustrates the precision in finite samples of the different policy iteration estimators. We conclude in Section 6 with a summary of our results and a discussion of several ideas for further research. Proofs of Lemmas and a detailed description of the algorithms are provided in the Appendices.

2 Discrete Markov Decision Processes

2.1 Definitions and notation

In this section we define a Discrete Markov Decision Process (DMDP) and we introduce the notation in this paper following Rust (1994). There are two types of variables in these models: the vector of state variables, s, and a control variable, d, that belongs to a finite set of mutually exclusive choice alternatives $D = \{1, 2, ..., J\}$. The space of state variables, S, is a subset of the $|S|$-dimensional Euclidean space, where $|S|$ is the number of state variables. Time is discrete and it is indexed by t. At each period t an agent observes s_t and decides d_t in order to maximize the expected sum of current and future discounted utilities. Future values of some state variables are uncertain for the agent. His beliefs about uncertain future states can be represented by a Markov transition probability $p(s_{t+1}|s_t, d_t)$, which may combine transition density functions for continuous state variables and transition probabilities for discrete state variables. The time horizon of the decision problem is infinite. Utility is time separable and $u(s_t, d_t)$ represents the one-period utility function. The parameter β represents the rate at which the agent discounts utility at future periods, and it belongs to the interval $(0, 1)$.

In this context, an agent can be represented by the set of primitives $\{u, p, \beta\}$. Under some regularity conditions about these primitives the optimal decision rule of the agent is a time-invariant function, $\delta(s_t)$, where the form of this function depends on the primitives of the problem (i.e., Blackwell's theorem). Therefore, the decision at period t is the same as the decision at period $t+j$ if $s_t = s_{t+j}$. For this reason we omit the subindex t for the rest of this section. We use s' to denote the vector of next period's state variables.

We assume that the researcher knows the utility and the transition probability functions up to the vectors of parameters θ_u and θ_p, respectively. We define $\theta \equiv (\theta_u, \theta_p)$. From an econometric point of view there are two types of state variables: those observable to the researcher, x, and the unobservables, ε, i.e., $s = (x, \varepsilon)$. Suppose we have a dataset with information about decisions and observable state variables for a random sample of individuals. Under the assumption that the sample was generated by a DMDP, our objective is to obtain an estimate of θ.

In order to guarantee the existence, consistency and asymptotic normality of the ML estimator of θ, some assumptions should be made about the primitives $\{u, p\}$. In particular, we impose certain restrictions that guarantee the continuity and twice differentiability of the log-likelihood function with respect to θ. A source of discontinuity of the log-likelihood function in discrete choice econometric models arises when the distribution of the unobservables is not *saturated*. In the context of our model, we say that the distribution of ε is *saturated* if, for any value of θ in a compact set Θ and for any combination of observables (d, x), the set $\{\varepsilon \mid \delta(x,\varepsilon) = d\}$ has positive probability over the distribution of the unobservables. Under this condition, the log-conditional choice probabilities $\ln \Pr(d|x;\theta)$ that appear in the log-likelihood function always take finite values.

There is not a unique set of assumptions on $\{u, p\}$ that guarantee that the distribution of ε is saturated. Here we consider the assumptions which have been most commonly used in the literature (see Rust, 1994). We discuss the role of these assumptions in our results at the end of section 2.3.

ASSUMPTION 1 (Additivity): The one period utility function is additively separable in the observable and unobservable components.

$$u(s, d; \theta_u) = u(x, d; \theta_u) + \varepsilon(d) \tag{1}$$

where $\varepsilon(d)$ is the d-th component of the $Jx1$ vector ε and $u(.)$ is continuous and twice differentiable with respect to θ_u.

ASSUMPTION 2 (Conditional independence): The transition probability of the state variables factors as:

$$p(x', \varepsilon' |\ x, d, \varepsilon; \theta_p) = g(\varepsilon' |\ x'; \theta_q) \quad f(x' |\ x, d; \theta_f) \tag{2}$$

where $g(.)$ is continuous and twice differentiable in ε' and θ_q, and $f(.)$ is continuous and twice differentiable with respect to θ_f.

Although the results below can be easily extended to the case of continuous observable state variables, the implementation of the different algorithms requires the discretization of these variables. For the sake of simplicity we consider the following assumption.

ASSUMPTION 3: Finite domain for the observable state variables.

$$x \in X = \{x^1, ..., x^M\} \tag{3}$$

Let $V(s; \theta)$ be the value function and let $v(s, d; \theta)$ be the value function conditional on the (hypothetical) choice of alternative d. Then $V(s; \theta)$ satisfies Bellman's equation:

$$v(s, d; \theta) = u(x, d; \theta_u) + \varepsilon(d) + \beta \sum_{m=1}^{M} f(x^m | x, d; \theta_f) \int V(x^m, \varepsilon'; \theta) \ g(d\varepsilon' | x^m; \theta_q) \tag{4}$$

and:

$$V(s; \theta) = \max_{d \in D} \{v(s, d; \theta)\} \tag{5}$$

The optimal decision rule is thus:

$$\delta(s; \theta) = \arg\max_{d \in D} \{v(s, d; \theta)\} \tag{6}$$

2.2 Smooth Bellman equation

We now define versions of the value function and the Bellman operator which take into account the existence of unobservables. These versions will prove more useful than (4)-(6) in the analysis of the estimation problem. McFadden (1973) introduced the *social surplus function*:

$$S(x;\theta) = \int \max_{d\in D} \{v(x,\varepsilon,d;\theta)\} \;\; g(d\varepsilon|x;\theta_q) \tag{7}$$

The social surplus is the expectation of the value function conditional on the information available to the econometrician about the state variables, i.e., conditional on x. Combining expressions (4), (5) and (7), we can obtain the following *smooth Bellman equation:*

$$S(x;\theta) \;\; = \int \max_{d\in D} \{u(x,d;\theta_u) + \varepsilon(d) + \beta \; F(x,d;\theta_f)' \; S(\theta)\} \;\; g(d\varepsilon|x;\theta_q) \tag{8}$$

where $F(x,d;\theta_f)$ is the vector of transition probabilities for x' conditional on x and d, and $S(\theta)$ is the vector of social surpluses for each value of x. We can write the system of M equations in (8) (one for each value of $x \in X$) using the following matrix form:

$$S(\theta) \;\; = \int \max_{d\in D} \{u(d;\theta_u) + \varepsilon(d) + \beta \; F(d;\theta_f) \; S(\theta)\} \;\; g(d\varepsilon;\theta_q) \tag{9}$$

where $u(d;\theta_u)$ is the Mx1 vector of utilities associated with alternative d; $F(d;\theta_f)$ is the MxM matrix of transition probabilities conditional on the choice of alternative d; and $g(d\varepsilon;\theta_q)$ stacks the M distribution functions $g(d\varepsilon|x;\theta_q)$ for all states. It is straightforward to show that the operator on the right hand side of (9) satisfies Blackwell's sufficient conditions for a contraction mapping (i.e., monotonicity and discounting).check? Therefore, $S(\theta)$ is the unique fixed point of this contraction mapping.

Associated with the solution to Bellman's equation there is a set of *conditional choice probabilities* or *smooth optimal decision rule.* Given the vector of state variables x, the probability that alternative d is the optimal choice is:

$$\begin{aligned} P(d,x;\theta) \;\; &= \int I\left(\delta(x,\varepsilon;\theta) = d\right) \;\; g(d\varepsilon|x;\theta_q) \\ &= \int I\left(d = \arg\max_{j\in D} \{u(x,j;\theta_u) + \varepsilon(j) + \beta \; F(x,j;\theta_f)' \; S(\theta)\}\right) \;\; g(d\varepsilon|x;\theta_q) \\ &\equiv \lambda(d,x,\theta,S[\theta]) \end{aligned} \tag{10}$$

where $I(.)$ is the indicator function. The last line emphasizes that $P(d, x; \theta)$ depends on the vector of social surpluses $S(\theta)$. Let $\Lambda(.)$ be the vector of $\lambda(.)$ functions for all x, d. Notice that $\Lambda(\theta, S)$ gives the *current* optimal response of a decision maker whose vector of social surpluses as of tomorrow is S. Finally, stack all conditional choice probabilities in an MJx1 vector $P(\theta)$,

$$P(\theta) \equiv \left[\begin{array}{c} (P(1, x^1; \theta), P(2, x^1; \theta), \ldots, P(J, x^1; \theta))', \ \ldots, \\ (P(1, x^m; \theta), P(2, x^m; \theta), \ldots, P(J, x^m; \theta))' \end{array} \right]'$$

Then the system of equations in (10) can be written as follows in matrix form:

$$P(\theta) = \Lambda(\theta; S[\theta]) \tag{11}$$

2.3 Smooth policy iteration

We now describe a smooth policy iteration algorithm to obtain $S(\theta)$ and $P(\theta)$ numerically for a given value of θ. We will need to define 'smooth' versions of policy valuation and policy iteration operators.

Using equation (8) and the definition of conditional choice probabilities, we can obtain the following alternative expression for the social surpluses:

$$S(x; \theta) = \sum_{d \in D} P(d, x; \theta) \left\{ u(x, d; \theta_u) + E[\varepsilon(d) | x, \delta(x, \varepsilon; \theta) = d] + \beta \ F(x, d; \theta_f)' \ S(\theta) \right\} \tag{12}$$

Hotz and Miller's Proposition 1 (1993) implies that, under Assumptions 1 and 2, the expectations $E[\varepsilon(d) | x, \delta(x, \varepsilon; \theta) = d]$ are functions of the vector of conditional choice probabilities $P(x; \theta) \equiv \{P[d, x; \theta] : d \in D\}$. We will denote these functions by $e(d, P[x; \theta]; \theta_q)$. For instance, if ε has an Extreme value Type I distribution, it is possible to show that $e(d, P[x; \theta]; \theta_q) = c - \ln P(d, x; \theta)$, where c is Euler's constant. In matrix form, expression (12) becomes:

$$S(\theta) = \sum_{d \in D} P(d; \theta) * \left\{ u(d; \theta_u) + e(d, P[\theta]; \theta_q) + \beta \ F(d; \theta_f) \ S(\theta) \right\} \tag{13}$$

where $*$ is the Hadamard product (or element-by-element product); and $e(d, P[\theta]; \theta_q)$ is the $Mx1$ vector of expectations $e(d, P[x; \theta]; \theta_q)$. Finally, we solve for $S(\theta)$ in the previous system of equations to obtain:

$$S(\theta) = \Big(I_M - \beta\ F^U(\theta_f, P[\theta])\Big)^{-1} \left\{ \sum_{d \in D} P(d;\theta) * [u(d;\theta_u) + e(d, P[\theta]; \theta_q)] \right\} \tag{14}$$

where $F^U(\theta_f, P[\theta])$ is the MxM matrix of unconditional transition probabilities induced by $F(d;\theta_f)$ and the conditional choice probabilities $P(\theta)$:

$$F^U(\theta_f, P[\theta]) = \sum_{d \in D} P(d;\theta) * F(d;\theta_f) \tag{15}$$

Notice that both the conditional expectation functions $e(d, \Pi; \theta_q)$ and the unconditional transition probabilities function $F^U(\theta_f, \Pi)$ are defined for *any* set of conditional choice probabilities Π, *optimal or not.* Therefore, the expression on the right hand side of (14) can be used to compute the vector of social surpluses induced by an arbitrary Π, or the *valuation* of Π. We define the *smooth policy valuation operator* as

$$\sigma(\theta, \Pi) \equiv \Big(I_M - \beta\ F^U(\theta_f, \Pi)\Big)^{-1} \left\{ \sum_{d \in D} \Pi(d) * [u(d;\theta_u) + e(d, \Pi; \theta_q)] \right\} \tag{16}$$

where $\Pi(d)$ is the vector of M conditional probabilities for alternative d, and $F^U(\theta_f, \Pi)$ is the matrix $\sum_{d \in D} \Pi(d) * F(d;\theta_f)$. We can now define a *smooth policy iteration operator* which combines the mappings $\sigma(\theta, \Pi)$ and $\Lambda(\theta; S)$:

$$\Psi(\theta, \Pi) \equiv \Lambda(\theta; \sigma[\theta, \Pi]) \tag{17}$$

$\Psi(., \Pi)$ should be interpreted as giving the current optimal policy under the assumption that policy Π will be used forever as of next period. Clearly, the optimal choice probabilities $P(\theta)$ are the unique fixed point of this mapping: $P(\theta) = \Psi(\theta, P(\theta))$.

The policy iteration algorithm consists on the repeated application of the smooth policy iteration operator. The algorithm proceeds as follows. Let $\Pi_0 \in [0,1]^{(J-1)M}$ be an initial guess for the set of conditional choice probabilities. At iteration k,

Step 1. Given Π_{k-1}, obtain $\sigma_{k-1}(\theta) = \sigma(\theta, \Pi_{k-1})$ using expression (16).

Step 2. Given $\sigma_{k-1}(\theta)$, obtain a new vector of conditional choice probabilities, Π_k, as $\Pi_k = \Lambda(\theta; \sigma_{k-1}[\theta])$.

Step 1 is a policy valuation; in step 2, the best policy is obtained under the assumption that policy Π_{k-1} will be used forever as of next period. Steps 1 and 2 can be written together as $\Pi_k = \Psi(\theta; \Pi_{k-1})$. The algorithm iterates until a convergence criterion with respect to Π is satisfied.[3]

We now state an important property of the smooth policy operators. As we show later, this property plays a crucial role in our result in Section 3 on the equivalence of the two nested algorithms.

LEMMA 1:

Under Assumptions 1-3, and for any value θ, the Jacobian matrices of partial derivatives of $\sigma(\theta, \Pi)$ and $\Psi(\theta, \Pi)$ with respect to Π are zero at the fixed point $\Pi = P(\theta)$.

$$\frac{\partial \sigma(\theta, P[\theta])}{\partial \Pi'} = 0$$

Therefore,

$$\frac{\partial \Psi(\theta, P[\theta])}{\partial \Pi'} = \frac{\partial \Lambda(\theta, S[\theta])}{\partial S'} \frac{\partial \sigma(\theta, P[\theta])}{\partial \Pi'} = 0$$

Proof:

See Appendix 1.

The first part of Lemma 1 establishes that at the fixed point it is not possible to "improve" social surpluses by changing the conditional choice probabilities. That is, the optimal choice probabilities maximize the valuation operator locally. This result is consistent with the interpretation of the social surplus as a *smooth* value function, and the set of conditional choice probabilities as a *smooth* optimal decision rule. The interpretation of the second part of the lemma is that the optimal response today to a small deviation from optimal policies as of tomorrow is still not to deviate.

We end this section with a remark on the role of Assumptions 1-3. Assumption 3 allowed us to use compact matrix notation, but it is not crucial here. On the contrary, assumptions 1 (Additivity) and 2 (Conditional Independence) are required to prove Hotz & Miller's (1993) Proposition 1 (Invertibility), and this is needed in

[3] For a detailed description of the numerical and computational properties of policy iteration methods see Puterman (1994) and Rust (1996).

order to define the 'smooth' versions of policy rules, policy valuation and policy iteration operators. Assumption 1 can be relaxed to allow for multiplicative separability between observable and unobservable components. However, Assumption 2 is a necessary condition for future utility differences not to depend on current unobservable state variables, and this plays a crucial role in Hotz & Miller's Proposition 1.[4]

3 Maximum likelihood estimation and nested algorithms

Suppose our panel dataset consists of observations from a random sample of individuals $\{x_{it}, d_{it} : i = 1, ..., N; t = 1, ..., T_i\}$ where i is the subindex for individuals and t for time. Under Assumption 2, the log-likelihood function of the DMDP model can be decomposed into choice probability and transition probability terms as follows:

$$l(\theta) = l_1(\theta) + l_2(\theta_f) = \sum_{i=1}^{N}\sum_{t=1}^{T_i} \ln P(d_{it}, x_{it}; \theta) + \sum_{i=1}^{N}\sum_{t=1}^{T_i-1} \ln f(x_{it+1}|x_{it}, d_{it}; \theta_f) \tag{18}$$

Let z_{it}^{mj} be the indicator for the choice of alternative j observed in state m, i.e., $z_{it}^{mj} = I(d_{it} = j)\ I(x_{it} = x^m)$, and let y_{it}^{mrj} be the indicator for the transition from state m to state r conditional on the choice of alternative j, i.e., $y_{it}^{mrj} = I(x_{it+1} = x^r)I(x_{it} = x^m)I(d_{it} = j)$. Finally, let z_{it} and y_{it} be the vectors that contain the set of indicator variables $\{z_{it}^{jm}\}$ and $\{y_{it}^{mrj}\}$, respectively, for observation (i, t). Using these definitions:

$$l_1(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T_i}\sum_{m=1}^{M}\sum_{j=1}^{J} z_{it}^{jm} \ln P(j, x^m; \theta) = n\ \ln P(\theta)'\ \bar{z} \tag{19}$$

where $P(\theta)$ is the vector of all MJ conditional choice probabilities, n is total sample size and $\bar{z}$ is the sample mean of z_{it}, i.e., the vector with the sample frequencies of

[4] Hotz and Miller (1993) defined a mapping $Q()$ from the $J-1$ (future) utility differences to the choice probabilities, conditional on observable state variables. Their Proposition 1 establishes that this mapping is invertible. Thus, policy rules can be described either in the space of utility differences, or as 'smooth' policies in the simplex. A corollary of their proposition is that the expectations of choice-specific unobservables are functions of the choice probabilities (the $e()$ functions we use here).

every possible state-choice cell. And:

$$l_2(\theta) = \sum_{i=1}^{N} \sum_{t=1}^{T_i-1} \sum_{m=1}^{M} \sum_{r=1}^{M} \sum_{j=1}^{J} y_{it}^{rjm} \ln f(x^r | \ x^m, j; \theta_f) = (n - N) \ \ln F(\theta_f)' \ \bar{y} \tag{20}$$

where $F(\theta_f)$ is the vectorization of all the matrices of conditional choice transition probabilities, $F(d; \theta_f)$, and $\bar{y}$ is the sample mean of y_{it}.

The Maximum Likelihood Estimator (MLE) of θ can be implicitly defined in terms of two equations: the first order conditions,

$$n \ \frac{\partial \ln P(\hat{\theta}_{ML})'}{\partial \theta} \ \bar{z} + (n - N) \ \frac{\partial \ln F(\hat{\theta}_{ML})'}{\partial \theta} \ \bar{y} = 0 \tag{21}$$

and the equation defining $P(\theta)$ implicitly as the fixed point of the smooth policy iteration mapping:

$$P(\widehat{\theta}_{ML}) = \Psi(\widehat{\theta}_{ML}, P(\widehat{\theta}_{ML})) \tag{22}$$

ASSUMPTION 4: Identification assumption.[5]

There is a unique vector θ^* such that:

$$E\left(\frac{\partial \ln P(\theta^*)'}{\partial \theta} \sum_{t=1}^{T} \ z_{it} + \frac{\partial \ln F(\theta^*)'}{\partial \theta} \sum_{t=1}^{T} \ y_{it} \right) = 0 \tag{23}$$

We now define two algorithms to obtain the MLE of θ^*. The first is Rust's *Nested Fixed Point Algorithm* (NFXP) (see Rust 1987, 1988).

Nested Fixed Point Algorithm (NFXP).[6]

Start with an initial guess of θ, θ_0. At iteration $k \geq 1$, apply the following two steps.

[5] Since both our procedure and NFXP exploit gradient algorithms, we consider a strong identification assumption instead of the weaker assumption that θ^* is the unique global maximum for the expected value of the log-likelihood function.

[6] Rust (1987, 1988) studied the properties of this algorithm. In order to guarantee convergence of the 'inner' algorithm in Step 1, he proposed a *hybrid fixed point algorithm* which begins with successive approximation iterations in the surplus function and then switches to policy iterations. The nested algorithm always converges, and under Assumption 4 the NFXP algorithm converges to the MLE of θ^*. If Assumption 4 is relaxed (i.e., there may be several local maxima), whether the value of θ reached by this algorithm is a local or a global maximum will depend on the initial guess θ_0. We describe a pure policy iteration version of NFXP here in order to emphasize its relationship with our nested algorithm.

Step 1: Given θ_{k-1}, obtain the optimal choice probabilities $P(\theta_{k-1})$. That is, iterate in the smooth policy iteration mapping until convergence to compute the fixed point:

$$P(\theta_{k-1}) = \Psi(\theta_{k-1}, P(\theta_{k-1})) \tag{24}$$

Also, compute $\partial P(\theta_{k-1})'/\partial\theta$.

Step 2: Apply one step of the gradient method for the maximization of the likelihood function. That is, obtain a new value of θ, θ_k, according to

$$\theta_k = \theta_{k-1} - H(\theta_{k-1}) \left(n \frac{\partial \ln P(\theta_{k-1})'}{\partial\theta} \, \bar{z} + \frac{\partial l_2(\theta_{k-1})}{\partial\theta} \right) \tag{25}$$

where the form of $H(.)$ will depend on the particular hill climbing algorithm used.

Iterate in k until convergence with respect to θ is reached.

In the second nested procedure we swap the order of the two algorithms. That is, the outer algorithm solves the fixed point problem, and the inner algorithm maximizes a pseudo-likelihood function in which the conditional choice probabilities are $\Psi(\theta, \Pi)$ instead of $P(\theta)$, where Π is fixed at the value obtained from the previous outer iteration. However, we will show that, as the outer policy iterations converge, the pseudo-social surplus converges to the true social surplus and the pseudo-likelihood function to the true likelihood function. In other words, the task of obtaining the true likelihood function by solving the dynamic programming model is performed "in parallel" with the computation of the ML estimate of θ.

Nested Pseudo Likelihood Algorithm (NPL).

Start with an initial guess for the conditional choice probabilities, $\Pi_0 \in [0,1]^{MJ}$. At iteration $k \geq 1$, apply the following three steps:

Step 1: Given Π_{k-1}, construct the smooth policy iteration mapping $\Psi(\theta, \Pi_{k-1})$ described in section 2.3.

Step 2: Obtain a new value of θ, θ_k, as the value that maximizes a pseudo-likelihood function defined in terms of the conditional choice probabilities $\Psi(\theta, \Pi_{k-1})$.

$$\theta_k = \arg\max_{\{\theta\}} \quad \ln \Psi(\theta, \Pi_{k-1})'\bar{z} + l_2(\theta_{k-1}) \tag{26}$$

Step 3: Update Π using the 'max ' from step 2, i.e.

$$\Pi_k = \Psi(\theta_k, \Pi_{k-1}). \tag{27}$$

Iterate in k until convergence with respect to Π (and θ) is reached.

The following result establishes that the NPL algorithm does indeed provide an alternative way of computing maximum likelihood estimates:

LEMMA 2 Equivalence of NFXP and NPL:

If the Nested Pseudo Likelihood algorithm (NPL) converges, it does so to a value of θ that solves equations (21) and (22). Under Assumption 4, this is the Maximum Likelihood estimator of θ^*

Proof:

See Appendix 1.

The proof of Lemma 2 shows that the main idea behind this result is relatively simple. Our econometric model is such that in order to obtain the probabilities that enter the likelihood function (given the parameters of interest) we need to solve a fixed point problem. If the mapping that defines this fixed point (i.e., $\Psi(\theta, \Pi)$ in our case) is such that, at the fixed point, the partial derivatives of the mapping with respect to the probabilities are zero, then the NPL algorithm provides the Maximum Likelihood estimates of the parameters of interest. Therefore, the result in Lemma 1 is crucial to obtain the convergence of the NPL algorithm to the MLE.

Under Assumption 2, consistent estimates of the conditional transition probability parameters θ_f can be obtained from transition data without having to solve the Markov decision model. In the rest of the paper we focus on the estimation of preference parameters (θ_u) and the parameters describing the distribution of unobservables (θ_q). That is, we consider a partial Maximum Likelihood estimator of (θ_u, θ_q) based on the likelihood $l_1(\theta)$, given consistent estimates of θ_f obtained from likelihood $l_2(\theta_f)$.[7] This two-stage estimation strategy, which reduces the computational burden of estimation, was used in Rust (1987) and Hotz and Miller (1993).[8] In this partial likelihood analysis the identification assumption is now:

ASSUMPTION 4': Identification assumption (Partial Maximum Likelihood).

There is a unique vector $\theta^* = (\theta^*_{uq}, \theta^*_f)$ such that:

$$E\left(\frac{\partial \ln P(\theta^*)'}{\partial \theta_{uq}} \sum_{t=1}^{T} z_{it}\right) = 0, \text{ and } E\left(\frac{\partial \ln F(\theta^*)'}{\partial \theta_f} \sum_{t=1}^{T} y_{it}\right) = 0$$

It is straightforward to see that Lemma 2 (equivalence) still applies to the partial likelihood versions of NFXP and NPL.[9] In this context, using the NPL algorithm rather than NFXP results in large additional computational gains. We address this question in Section 5 using Rust's bus replacement problem as an example. Furthermore, the NPL algorithm defines a recursive extension of Hotz and Miller's Conditional Choice Probability (CCP) estimator. We turn to this issue in the next section.

4 Sequential policy iteration estimators

In the context of partial likelihood estimation, let $\widehat{\theta}_f$ denote the (consistent) estimates of conditional transition probability parameters obtained in the first stage from the likelihood $l_2(\theta_f)$. Furthermore, let $\widehat{\Pi}_0$ be consistent, nonparametric estimates of the conditional choice probabilities $P(\theta^*)$. Consider using $\widehat{\Pi}_0$ as an initial guess in our NPL algorithm. Performing one, two and in general K iterations of the NPL algorithm yields a sequence $\{\widehat{\theta}_{uq1}, \widehat{\theta}_{uq2}, ..., \widehat{\theta}_{uqK}\}$ of statistics which can be used as estimators of θ^*_{uq}. We call them *sequential Policy Iteration (PI) estimators.* Let us

[7] The discount factor β is assumed to be known.

[8] See Rust (1987, 1994) for a 3-stage extension which is asymptotically equivalent to full maximum likelihood.

[9] Notice that the only changes are: (a) $l_2(\theta_f)$ drops from both the likelihood and pseudo-likelihood functions; and (b) in the choice probabilities $P()$ and in the pseudo-choice probabilities $\Psi()$ the vector of parameters θ_f is fixed.

drop the ${}_{uq}$ subscript for notational simplicity. Thus, $\widehat{\theta}_K$ denotes the $K - stage\ PI$ estimator of θ^*_{uq} which is obtained after K iterations of the Nested Pseudo Likelihood algorithm when: 1) θ_f is fixed at $\widehat{\theta}_f$; and 2) Consistent nonparametric estimates of $P(\theta^*)$ are used as initial values Π_0. In this section we study the statistical properties of this sequence of estimators. It is clear from Lemma 2 that it encompasses the partial Maximum Likelihood estimator: for any sample size, as the NPL algorithm converges in Π_k the corresponding $K - stage\ PI$ estimator converges to the partial ML estimator. In Lemma 3 we show that, for any value of K, the PI estimators are asymptotically equivalent to the MLE. Lemma 4 shows that the PI estimator for $K = 1$ is Hotz-Miller's CCP estimator. Therefore, for intermediate values of K we get asymptotically equivalent estimators which are significantly cheaper to obtain than the MLE yet significantly more precise in finite samples than the Hotz-Miller estimator.

The $K - stage\ PI$ estimator is defined as:

$$\hat{\theta}_K = \arg\max_{\theta} \sum_{i=1}^{N} \widetilde{l}_{1i}(\theta, \widehat{\theta}_f, \hat{\Pi}_{K-1}) \equiv \arg\max_{\theta} \sum_{i=1}^{N}\sum_{t=1}^{T_i} \ln \Psi(d_{it}, x_{it}; \theta, \widehat{\theta}_f, \hat{\Pi}_{K-1}) \quad (28)$$

where $\widetilde{l}_{1i}(\theta, \widehat{\theta}_f, \hat{\Pi}_{K-1})$ is individual i's contribution to the partial pseudo log-likelihood, $\Psi(d_{it}, x_{it}; \cdot)$ denotes a pseudo-choice probability for the specific choice-state combination in observation it and $\hat{\Pi}_{K-1}$ is the vector of pseudo-choice probabilities obtained from the previous stage, i.e., $\hat{\Pi}_{K-1} = \Psi(\hat{\theta}_{K-1}, \widehat{\theta}_f, \hat{\Pi}_{K-2})$. This is a particular case of a $Quasi - Generalized\ M - estimator$ as defined in Gourieroux and Monfort (1995), with $(\widehat{\theta}_f, \hat{\Pi}_{K-1})$ as the vector of nuisance parameters. Let the * superscript denote a function evaluated at the true parameter values; if the vector

$$\left[\frac{1}{\sqrt{n}} \sum_i \frac{\partial \widetilde{l}^*_{1i}}{\partial \theta'}, \sqrt{n}(\widehat{\theta}_f - \theta^*_f)', \sqrt{n}(\hat{\Pi}_{K-1} - \Pi^*)'\right]'$$

is asymptotically normally distributed with mean zero and variance - , the Quasi-Generalized M-estimator is asymptotically normal and its asymptotic variance is given by the following general expression:

$$V(\hat{\theta}_K) = J^{-1} \left((I, H_f, H_\Pi) \begin{pmatrix} -_{00} & -_{0f} & -_{0\Pi} \\ -_{f0} & -_{ff} & -_{f\Pi} \\ -_{\Pi 0} & -_{\Pi f} & -_{\Pi 0} \end{pmatrix} \begin{pmatrix} I \\ H'_f \\ H'_\Pi \end{pmatrix} \right) J^{-1} \quad (29)$$

where $J = E\left(-\dfrac{\partial^2 \widetilde{l}_{1i}(\theta^*, \theta_f^*, \Pi^*)}{\partial\theta\partial\theta'}\right)$, $H_f = E\left(\dfrac{\partial^2 \widetilde{l}_{1i}(\theta^*, \theta_f^*, \Pi^*)}{\partial\theta\partial\theta_f'}\right)$ and

$H_\Pi = E\left(\dfrac{\partial^2 \widetilde{l}_{1i}(\theta^*, \theta_f^*, \Pi^*)}{\partial\theta\partial\Pi'}\right)$.

Likewise, the partial MLE is:

$$\hat{\theta}_{PMLE} = \arg\max_{\theta} \sum_{i=1}^{N} l_{1i}(\theta, \widehat{\theta}_f) \equiv \arg\max_{\theta} \sum_{i=1}^{N}\sum_{t=1}^{T_i} \ln P(d_{it}, x_{it}; \theta, \widehat{\theta}_f) \qquad (30)$$

which can also be seen as a Quasi-Generalized M-estimator. Notice that the score replaces the pseudo-score and that the only nuisance parameters are now $\widehat{\theta}_f$. Using our result in Lemma 1 it is possible to show that H_Π above is zero and that scores and pseudo-scores are equivalent, and this implies that $V(\hat{\theta}_K)$ is in fact the asymptotic variance of the partial MLE.

LEMMA 3:

The estimators in the sequence $\left\{\hat{\theta}_K : K \geq 1\right\}$ are consistent, asymptotically normal and asymptotically equivalent to the partial likelihood estimator. The asymptotic covariance matrix of all K-stage PI estimators is:

$$V(\hat{\theta}_K) = -_{00}^{-1} + -_{00}^{-1}\left\{H_f -_{f0} + -_{0f}\, H_f' + H_f\; V(\hat{\theta}_f)\; H_f'\right\} -_{00}^{-1} \qquad (31)$$

where

$-_{00} = E\left(\dfrac{\partial\widetilde{l}_{1i}^*}{\partial\theta}\dfrac{\partial\widetilde{l}_{1i}^*}{\partial\theta'}\right)$; $H_f = E\left(\dfrac{\partial\widetilde{l}_{1i}^*}{\partial\theta}\dfrac{\partial\widetilde{l}_{1i}^*}{\partial\theta_f'}\right)$;

$-_{f0} = \left[E\left(-\dfrac{\partial^2 l_{2i}^*}{\partial\theta_f\partial\theta_f'}\right)\right]^{-1} E\left(\dfrac{\partial l_{2i}^*}{\partial\theta_f}\dfrac{\partial\widetilde{l}_{1i}^*}{\partial\theta'}\right)$; $-_{0f} = -_{f0}'$;

$V(\hat{\theta}_f) = \left[E\left(\dfrac{\partial l_{2i}^*}{\partial\theta_f}\dfrac{\partial l_{2i}^*}{\partial\theta_f'}\right)\right]^{-1}$ with $E\left(-\dfrac{\partial^2 l_{2i}^*}{\partial\theta_f\partial\theta_f'}\right) = E\left(\dfrac{\partial l_{2i}^*}{\partial\theta_f}\dfrac{\partial l_{2i}^*}{\partial\theta_f'}\right)$ and

$\dfrac{\partial\widetilde{l}_{1i}^*}{\partial\theta} = \sum_t \dfrac{\partial \ln P_{it}(\theta^*, \theta_f^*)}{\partial\theta}$; $\dfrac{\partial\widetilde{l}_{1i}^*}{\partial\theta_f'} = \sum_t \dfrac{\partial \ln P_{it}(\theta^*, \theta_f^*)}{\partial\theta_f'}$;

$\dfrac{\partial l_{2i}^*}{\partial\theta_f} = \sum_t \dfrac{\partial \ln f(x_{i,t+1}, x_{it}; \theta_f^*)}{\partial\theta_f}$

Proof: See Appendix 1.

LEMMA 4:

The estimator obtained after one iteration of the NPL algorithm is a Hotz-Miller estimator of θ^*_{uq}, which is consistent and asymptotically normal.

Proof: See Appendix 1.

We have thus established that the Hotz-Miller estimator, the $K-stage$ estimators and the partial maximum likelihood estimator are asymptotically equivalent.[10] An important limitation of Hotz-Miller's estimator, however, is that initial nonparametric estimates of the conditional choice probabilities can be very imprecise, and this lack of accuracy is transmitted to the estimates of the structural parameters (see Eckstein and Wolpin, 1989, Rust, 1994, and Hotz et al., 1994). Our $K - stage$ estimator overcomes this problem by iterating K times in the smooth policy iteration operator. The computational cost of these $(K - 1)$ additional iterations is equal to the cost of $(K - 1)$ policy iterations and $(K - 1)$ pseudo-maximum likelihood estimations. An important issue is how fast the variance of the estimator decreases with K in finite samples. Is a 2-stage or 3-stage estimator enough to obtain significant gains in precision with respect to Hotz-Miller? We address this issue with a Monte Carlo experiment in Section 5.2.

5 The performance of the NPL algorithm and PI estimators

5.1 Relative speed of NPL and NFXP

In order to illustrate the performance of our NPL algorithm in maximum likelihood estimation, we use Rust's well known bus replacement model and dataset.[11] We obtain *partial* ML estimates using the NPL and the NFXP algorithms under several scenarios about the dimension of the state space, the number of parameters, and the quality of initial guesses of parameters and conditional choice probabilities. For a detailed technical description of the two algorithms used in this section see Appendix

[10] Asymptotic equivalence of K-stage and maximum likelihood estimators can also be established in a full likelihood context.

[11] See Rust (1987). Rust's model has been used in other studies to evaluate the performance of alternative algorithms and estimators, e.g. Hotz et al. (1994) and Rust (1997b).

2. [12]

We show that for large problems the cost of policy iterations is the main component of the computational burden of estimation, and that the reduction in the number of policy iterations achieved by use of the NPL algorithm in infinite horizon problems would lead to significant computational gains. An important feature of this example In the context of partial likelihood estimation, policy iterations are much more expensive than pseudo-likelihood climbing iterations. Notice that both types of iteration use the policy iteration operator defined in equation (17). However, pseudo-likelihood climbing iterations at each stage do not require repeated matrix inversions in order to compute the policy valuation operator in (16) because this inverse depends on parameters (θ_f, β) which are not being estimated.

Figures 1A and 1B plot the fraction of total estimation time which is spent in policy iterations against the size of the problem as measured by the number of points in the state space, i.e., M. Figure 1A refers to the NFXP algorithm. We see that, even for relatively small number of cells in the state space (e.g., $M = 200$), policy iterations represent almost 100% of the CPU time in estimation. Figure 1B shows that for the NPL algorithm this ratio tends to increase more slowly, but it is practically equal to 1 for $M \geq 500$.[13] Therefore, it is quite relevant to assess to what extent the use of NPL instead of NFXP reduces the number of policy iterations in the estimation procedure.

Before we address this question, it is important to notice that the size of the state space does not affect the number of policy iterations. Figures 2A and 2B illustrate this point by presenting the number of policy iterations to obtain the MLE under the NFXP and the NPL, respectively, for different sizes of the state space. Therefore, taking into account Figures 1 and 2, it should be clear that, for M large enough:

[12] It is important to underline that the NFXP algorithm that we use has two features which contribute very significantly to improve its computational efficiency (both features have been considered by Rust, 1987, 1988). First, we use a closed-form expression for the gradient of the likelihood function. Second, at each outer iteration, we use "smart guesses" for the vector of choice probabilities that initialize the policy iteration algorithm. For further details see Appendix 2.

[13] The reason why under the NPL this ratio increases more slowly with M is that, as we will show below, the number of policy iterations under NPL is smaller than under NFXP.

(a) the ratio between the CPU times of the two algorithms will be very close to the ratio between their respective number of policy iterations; and *(b)* this ratio does not depend on the size of the state space. However, the number of policy iterations *is* affected by the accuracy of initial guesses of parameter values (for NFXP) and initial guesses of conditional choice probabilities (for NPL). In principle, it might depend also on the number of parameters to be estimated. For this reason, in Table 1 we report results for only one size of the state space, $M = 500$, but we consider different cases for the other factors affecting the relative speed of the two algorithms.[14]

Table 1 presents absolute and relative values for CPU time and number of policy iterations for specifications with 2 and 4 structural parameters. [15] For the NFXP algorithm, the initial guesses θ_0 have been chosen in two different ways. Since $\theta_0 = 0$ is a commonly used initial guess, we present results for it. Other guesses for θ_0 have been obtained so that they lie in the frontier of a confidence region with a predetermined significance level. In particular, we have considered a confidence region obtained from the actual ML estimates with significance level equal to 10^{-6}.[16] We believe this is a reasonable metric for the quality of a researcher's guesses. Our choice of the significance level is necessarily arbitrary. However, notice that, both with 2 and 4 structural parameters, $\theta_0 = 0$ lies outside this frontier, i.e., it is a worse guess. Furthermore, we obtain similar results with significance levels between 10^{-4} and 10^{-7}. As for the NPL algorithm we consider 2 cases for the initial guesses of the choice probabilities, i.e., kernel estimates and random draws from a $U(0, 1)$. Figures 3 and 4 present the true choice probabilities and the 2 guesses for Π_0.[17] The estimations with random initial choice probabilities try to represent a *worst case scenario* for NPL.

[14] All the computations in this paper have been made using a Pentium MMX 233 MHz. The programs have been written in GAUSS.

[15] As one might expect from Lemma 2, the values of the ML estimates using NFXP and NPL are numerically equivalent (i.e., up to the twelfth significant digit). In fact, this paper was originally motivated by this finding.

[16] We generate these values for θ_0 as follows. We fixed all the parameters except one at their maximum likelihood estimates. Using the asymptotic Normal distribution of the vector of ML estimates, and its estimated covariance matrix, the value of the remaining parameter is determined so that θ_0 lies in the frontier of the confidence region with a predetermined significance level.

[17] Kernel estimates have been obtained using a Nadaraya-Watson estimator with a Gaussian kernel function and Silverman's rule of thumb for the choice of the bandwidth parameter.

But notice that, in this application, the kernel estimates are quite inaccurate too.

Table 1 shows that, even when using random draws for the initial choice probabilities, the NPL algorithm is between 7 and 13 times faster than NFXP. When using kernel estimates as initial probabilities, this ratio is between 9 and 19. It is clear that for models with large state spaces this range of speed ratios implies huge savings in CPU time.

The number of structural parameters in the model does not affect the relative performance of the two estimators.

5.2 The precision of PI estimators: A Monte Carlo exercise

In section 4 we established that all K-stage PI are asymptotically equivalent. We now look into their behavior in finite samples. The Hotz-Miller estimator is known to be quite sensitive to the quality of the initial non-parametric estimates of choice probabilities. This suggests that there is a trade-off between increasing computational cost and increasing precision as we move along the sequence of estimators. In order to illustrate and analyze this trade-off in finite samples, we carried out the following Monte Carlo experiment. We used Rust's bus engine model with parameters equal to ML estimates as the DGP. For sample sizes 2500, 5000 and 10000 we generated 500 samples and for each of them we obtained the sequence of PI estimators and its limit, the ML estimator.[18]

Tables 2 to 5 present summary statistics for several experiments. For the 1, 2 and 3 -stage PI estimators and the ML estimator, we report in these Tables the mean and median of the absolute estimation error, and the empirical standard deviations. We also report the average number of stages to obtain the MLE. The size of the state space grid in the DGP is 200. Recall from Figure 2 that grid size does not affect the number of policy iterations to convergence of the NPL. Thus, if the precision

[18] Each sample has been obtained in the following way. First, we obtain the steady-state distribution of the observable state variable (cumulative mileage) using transition probabilities and true choice probabilities. Second, from this distribution we obtain n random draws of the state variable, where n is the sample size. Finally, using the optimal decision rule and simulations of the unobservable epsilons, we obtain the optimal decisions associated with each of the n simulations of the state variables.

of the initial nonparametric estimates of Π_0 is the same for different state spaces, the efficiency of the $K - stage$ estimators relative to MLE does not depend on the dimension of the state space. Of course, in an actual application where the sample size is given, the larger the dimension of the state space the lower the precision of the initial nonparametric estimates. We take into account this issue by considering different sample sizes, and fixing the grid size.

In experiments 1 to 3, we find very significant benefits of doing more than one policy iteration, even for relatively large sample sizes. In general, the 1-PI estimator performs poorly relative to the other estimators. More interestingly, the relative performance of the 2-PI is excellent. Even for very small sample sizes (and, consequently, very imprecise kernel estimates of the choice probabilities) the 2-PI estimator is very similar to MLE. Most of the benefits of additional policy iterations are obtained when one goes from 1 to 2 iterations.

However, this result might be misleading. Since this model is binary and has only one observable state variable, it is not possible to have very large discrepancies between the precision of the ML estimates of the choice probabilities and the precision of the kernel estimates. In other words, the only way to get very imprecise kernel estimates is by considering very small sample sizes, but this implies also imprecise ML estimates. In order to overcome this limitation, we consider Experiment 4. Instead of kernel estimates of the initial choice probabilities we use random draws from a $U(0,1)$. This results in artificially poor initial values for the choice probabilities. The results of this experiment (in Table 5) actually reinforce the ones from Experiments 1 to 3. Poor initial guesses have a very serious effect on the 1-PI estimator, but a negligible effect on the 2-PI, which is still very close to the MLE.

Therefore, in the context of this parametric model the fixed point structure has very important identifying power. When we construct expected value functions using choice probabilities which are closer to the fixed point our estimates become much more precise. However, it seems that 2 policy iterations are enough to get choice probabilities which are close enough to the fixed point.

In order to interpret this finding, note that there exists an important qualitative difference between going from 1 to 2 policy iterations and going from 2 to more iterations. The second policy iteration is the first one in which we incorporate the structure of the model to obtain the choice probabilities used to compute the pseudo-social surplus $\sigma(\theta, \Pi)$. These probabilities incorporate for the first time the following information which is not contained in the initial nonparametric estimates: (1) a distributional assumption about the unobservable state variables; (2) a parametric assumption for the one-period utility function; (3) the assumption of additivity between observables and unobservables in the one-period utility function; and (4) the assumption of additivity between current and future utilities in the intertemporal utility function. All subsequent policy iterations impose (recursively) that (5) the choice probabilities should be a fixed point of the policy iteration mapping, but no further assumptions about the functional form of the primitives are incorporated.

In our experiment the incorporation of (1) to (4) leads to very important improvements in the estimates of choice probabilities and structural parameters. The results also show that once we incorporate this structure the choice probabilities are very close to the ones that solve the fixed point problem for the MLE. An interesting issue would be to see whether this result still holds when we relax the parametric assumptions (1) or (2). To answer this question we would have to incorporate a semi-parametric estimation method like the one in Klein and Spady (1993). We consider this a topic for further research. .

6 Concluding remarks

We have proposed a new algorithm to obtain Maximum Likelihood estimates of dynamic programming discrete choice models. We have shown that this algorithm produces the MLE using significantly fewer policy iterations than the usual Nested Fixed Point Algorithm, resulting in large computational gains for a class of infinite-horizon, partial likelihood problems. Based on our algorithm, we have defined a family of sequential *Policy Iteration estimators* which encompass the Hotz-Miller estimator and the ML estimator as extreme cases. All K-stage PI estimators are consistent and asymptotically equivalent. Our Monte Carlo experiments show that the performance of the $2-stage\ PI$ estimator is excellent. Its finite sample variance and bias are very similar to those of the MLE. However, the $2-stage\ PI$ overcomes the well-known limitations of Hotz and Miller's estimator in terms of finite sample properties. Since this estimator is computationally much cheaper to obtain than MLE, our ability to obtain precise estimates of interesting economic models with more than two or three state variables is significantly enhanced. We would like to see these encouraging results confirmed by the use of 2-stage and k-stage PI estimators in actual applications.

It has been pointed out that the Conditional Independence assumption on which the Hotz & Miller estimator relies is perhaps too strong in many interesting applications. This drawback is shared by our K-stage PI estimators. However, one can show that our NPL algorithm can accommodate a form of serial correlation in the unobservables which has proved useful in many applications, the case in which the unobserved state variables include a permanent component with a non-parametric discrete distribution as in Heckman-Singer (1984). In this case the Hotz-Miller and k-stage PI estimators are not feasible because it is not possible to obtain non-parametric consistent estimates of conditional choice probabilities. But our NPL algorithm initiated with (inconsistent) guesses of choice probabilities will still converge to the Maximum Likelihood estimator, and in many applications it will do so much faster than the alternative NFXP algorithm.

As mentioned in the Introduction, the emphasis of this paper is not in algorithms for the solution of the dynamic programming models, but on the procedures used to search for parameter estimates in those models. In principle, our approach could be combined with the discretization techniques in Rust (1997b). Furthermore, the idea behind our algorithm is simple and it might be applied to the estimation of other type of models that incorporate the solution of a fixed point problem, i.e., game theoretic models, general equilibrium models. The main question is whether Lemma 1 can be extended to this type of models or not. We believe these are interesting topics for further research.

APPENDIX 1. Proof of Lemmas.

Proof of Lemma 1.

For the sake of notational simplicity, we omit θ as an argument of all the functions below. First of all, notice that the matrix Π contains only $M(J-1)$ conditional choice probabilities. That is, for each of the M possible values of the conditioning variables, there are only $J-1$ free probabilities. Without loss of generality, the probabilities that we exclude from Π are the ones associated with choice alternative 1.

[**1**] By definition,

$$\sigma(\Pi) = \sum_{j=1}^{J} \Pi(j) * \{u(j) + e(j,\Pi) + \beta\ F(j)\ \sigma(\Pi)\}\ \ ,$$

Let π_m be the column vector containing the $(J-1)$ choice probabilities in Π associated with $x = x^m$, and let $\pi_m(j)$ be the $j-th$ component of π_m. Differentiating in both sides with respect to π_m and collecting terms, one can show that

$$\frac{\partial\sigma(\Pi)}{\partial\pi_m} = [I_M - \beta F_\Pi]^{-1} \left[\left(\sum_{k=1}^{J} \Pi(k) * \frac{\partial e(k,\Pi)}{\partial\pi'_m} \right) + i_m\ \{\Delta e'_m + \Delta v'_m\} \right], \tag{A.1}$$

where i_m is an Mx1 vector with 1 in the $j-th$ position and zeros elsewhere; Δe_m is the $(J-1)$x1 vector $\{e_m(j,\Pi) - e_m(1,\Pi) : j = 2,...,J\}$; and $\Delta v_m(j)$ is the $(J-1)$x1 vector $\{v_m(j) - v_m(1) : j = 2,...,J\}$, where $v_m(j) \equiv u_m(j) + \beta\ F_m(j)\ \sigma(\Pi)$.

¿From Proposition 1 in Hotz and Miller (1993), $e_m(k,\Pi) = e_m(k,\pi_m)$. That is, the expectations $E(\varepsilon(j) \mid x^m, \delta = j)$ do not depend on choice probabilities associated with values of x different to x^m. Therefore, using expression (A.1), it is straightforward to see that $\dfrac{\partial\sigma_n}{\partial\pi'_m} = 0$, for any $m \neq n$. It remains to prove that $\dfrac{\partial\sigma_m}{\partial\pi'_m} = 0$. According to expression (A.1), it will suffice to show that

$$\left(\sum_{k=1}^{J} \pi_m(k) \frac{\partial e_m(k,\pi_m)}{\partial\pi'_m} \right) + \Delta e'_m + \Delta v'_m = 0. \tag{A.2}$$

For notational simplicity we omit the subindex m in our expressions below. However, it should be understood that all vectors, matrices and functions below refer to a particular value of x.

[2] To prove (A.2) we should introduce some additional notation. First, remember that $\pi(j)$ can be expressed as a function of Δv.

$$\pi(j) = Q(j, \Delta v) \equiv \int_{-\infty}^{+\infty} G_j(\xi + \Delta v(j), ..., \xi, ..., \xi + \Delta v(j) - \Delta v(J))\ d\xi,$$

where $G(.)$ is the *cdf* of ε, and $G_j(.)$ is its partial derivative with respect to $\varepsilon(j)$. Also, $e(j)$ is a function of Δv. For $j > 1$,

$$e(j) = W(j, \Delta v) \equiv \frac{H(j, \Delta v)}{Q(j, \Delta v)},$$

where:

$$H(j, \Delta v) \equiv \int_{-\infty}^{+\infty} \xi\ G_j(\xi + \Delta v(j), ..., \xi, ..., \xi + \Delta v(j) - \Delta v(J))\ d\xi.$$

And for $j = 1$,

$$e(1) = W(1, \Delta v) \equiv \frac{H(1, \Delta v)}{1 - \sum\limits_{j \neq 1} Q(j, \Delta v)},$$

We also introduce the following definitions. For any pair (j, k), we define:

$$q_k(j) = \int_{-\infty}^{+\infty} G_{jk}(\xi + \Delta v(j), ..., \xi, ..., \xi + \Delta v(j) - \Delta v(J))\ d\xi,$$

and,

$$h_k(j) = \int_{-\infty}^{+\infty} \xi\ G_{jk}(\xi + \Delta v(j), ..., \xi, ..., \xi + \Delta v(j) - \Delta v(J))\ d\xi$$

where $G_{jk}(.)$ is the partial derivative of $G(.)$ with respect to $\varepsilon(j)$ and $\varepsilon(k)$. The following properties of these functions will be used below.

Property 1: $q_k(j) = q_j(k)$
Property 2: $h_k(j) = h_j(k) + [\Delta v(j) - \Delta v(k)]\ q_k(j)$

According to these definitions, the $(J-1)$x$(J-1)$ Jacobian matrix $\dfrac{\partial Q(\Delta v)}{\partial \Delta v'}$ has the following expression. For $j > 1$ and $k > 1$:

$$DQ_{jk} \equiv \frac{\partial Q(j, \Delta v)}{\partial \Delta v(k)} = \begin{cases} -q_k(j) & if\ k \neq j \\ \sum\limits_{i \neq j} q_i(j) & if\ k = j \end{cases}$$

We can also obtain the expression for the Jx$(J-1)$ Jacobian matrix $\dfrac{\partial H(\Delta v)}{\partial \Delta v'}$. For $k > 1$:

$$DH_{jk} \equiv \frac{\partial H(j, \Delta v)}{\partial \Delta v(k)} = \begin{cases} -h_k(j) & if\ k \neq j \\ \sum\limits_{i \neq j} h_i(j) & if\ k = j \end{cases}$$

[3] We now proceed with the proof of (A.2). First, we obtain $\dfrac{\partial e(\pi)}{\partial \pi'}$. Since $Q(\Delta v)$ is an invertible system of equations (see Proposition 1 in Hotz and Miller), $\Delta v = Q^{-1}(\pi)$. Therefore, applying the chain rule and the inverse function theorem, we can obtain:

$$\frac{\partial e(k,\pi)}{\partial \pi'} = \frac{\partial W(k, \Delta v)}{\partial \Delta v'} \frac{\partial Q^{-1}(\pi)}{\partial \pi'} = \frac{\partial W(k, \Delta v)}{\partial \Delta v'} \left(\frac{\partial Q(\Delta v)}{\partial \Delta v'} \right)^{-1}, \tag{A.3}$$

Using the definition of $W(k, \Delta v)$ and the definitions in **[2]**, it is simple to show that for $k > 1$,

$$\frac{\partial W(k, \Delta v)}{\partial \Delta v'} = \frac{1}{\pi(k)} \left[DH_{k\bullet} - e(k)\ DQ_{k\bullet} \right],$$

where $DH_{k\bullet}$ and $DQ_{k\bullet}$ are the $k-th$ rows of DH and DQ, respectively. For $k = 1$,

$$\frac{\partial W(1, \Delta v)}{\partial \Delta v'} = \frac{1}{\pi(1)} \left[DH_{1\bullet} - e(1)\ (i'DQ) \right],$$

where i is a $(J-1)$x1 vector of ones.

Now, solving the previous equations in (A.3) we get,

$$\begin{aligned} \frac{\partial e(k,\pi)}{\partial \pi'} &= \frac{1}{\pi(k)} \left[DH_{k\bullet}\ (DQ)^{-1} - e(k)\ i_k' \right] \\ \frac{\partial e(1,\pi)}{\partial \pi'} &= \frac{1}{\pi(1)} \left[DH_{1\bullet}\ (DQ)^{-1} + e(1)\ i' \right] \end{aligned}$$

where i_k is a $(J-1)$x1 vector with 1 in the $k-th$ position and zeros elsewhere. Thus,

$$\sum_{k=1}^{J} \pi(k) \frac{\partial e(k,\pi)}{\partial \pi'} = \left(\sum_{k=1}^{J} DH_{k\bullet} \right)\ (DQ)^{-1} - \Delta e' \tag{A.4}$$

[4] Solving (A.4) into (A.2) it becomes clear that (A.2) is true if:

$$\sum_{k=1}^{J} DH_{k\bullet} = -(\Delta v')DQ$$

To prove this we rely on the expressions for DH and DQ in [**2**], as well as on *Properties 1 and 2.* The $j-th$ element in the left-hand side of the previous equation is:

$$\sum_{k\neq j} DH_{kj} + DH_{jj} = \sum_{k\neq j}[h_k(j) - h_j(k)] = \sum_{k\neq j}[\Delta v(j) - \Delta v(k)]\ q_k(j).$$

And the j-th element in the right-hand side is:

$$-\sum_{k=2;k\neq j}^{J} \Delta v(k)\ DQ_{kj} - \Delta v(j)\ DQ_{jj} = \sum_{k\neq j}[\Delta v(j) - \Delta v(k)]\ q_k(j).$$

what completes the proof.

Proof of Lemma 2.

[**1**] Suppose the sequence $\{\Pi_k\}$ from the NPL algorithm converges to a vector Π_{NPL}. Then, the smoothness of the pseudo-maximum likelihood estimator mapping defined in (26) guarantees that the sequence $\{\theta_k\}$ converges also to a vector $\hat{\theta}_{NPL}$. Therefore, by construction, Π_{NPL} should be the fixed point of the policy iteration mapping for $\theta = \hat{\theta}_{NPL}$.

$$\Pi_{NPL} = \Psi(\hat{\theta}_{NPL}, \Pi_{NPL}) \tag{A.5}$$

Furthermore, $\hat{\theta}_{NPL}$ maximizes the pseudo log-likelihood function $\{\ln\Psi(\theta,\Pi_{NPL})'\ \bar{z} + l_2(\theta_f)\}$. Therefore, the following marginal conditions of optimality should hold:

$$\frac{\partial\Psi(\hat{\theta}_{NPL},\Pi_{NPL})'}{\partial\theta}\ diag\left\{1/\Psi_{jm}(\hat{\theta}_{NPL},\Pi_{NPL})\right\}\ \bar{z} + \frac{\partial l_2(\widehat{\theta}_{NPL})}{\partial\theta} = 0 \tag{A.6}$$

or, combining (A.5) and (A.6):

$$\frac{\partial\Psi(\hat{\theta}_{NPL},P[\hat{\theta}_{NPL}])'}{\partial\theta}\ diag\left\{1/P_{jm}[\hat{\theta}_{NPL}]\right\}\ \bar{z} + \frac{\partial l_2(\widehat{\theta}_{NPL})}{\partial\theta} = 0 \tag{A.7}$$

Expression (A.7) defines implicitly $\hat{\theta}_{NPL}$. Our next step is to show that this expression is equivalent to the one that defines the ML estimator.

[**2**] Under Assumption 4, the ML estimator $\hat{\theta}_{ML}$ is the unique value of θ such that:

$$\frac{\partial\ln P(\hat{\theta}_{ML})'}{\partial\theta}\ \bar{z} + \frac{\partial l_2(\widehat{\theta}_{ML})}{\partial\theta} = 0$$

where:

$$P(\hat{\theta}_{ML}) = \Psi(\hat{\theta}_{ML}, P[\hat{\theta}_{ML}])$$

Now, using the definition of $P(\theta)$ as the (unique) fixed point of the mapping $\Psi(\theta, \Pi)$, it is possible to express the matrix of partial derivatives $\dfrac{\partial P(\theta)}{\partial \theta'}$ in terms of the partial derivatives of $\Psi()$ with respect to θ and Π. In particular, applying the implicit function theorem we obtain the following expression:

$$\frac{\partial P(\theta)}{\partial \theta'} = \left[I_{JM} - \frac{\partial \Psi(\theta, P(\theta))}{\partial \Pi'} \right]^{-1} \frac{\partial \Psi(\theta, P(\theta))}{\partial \theta'} \tag{A.8}$$

Solving (A.8) in the marginal conditions of optimality for the MLE, we get:

$$\frac{\partial \Psi(\hat{\theta}_{ML}, P[\hat{\theta}_{ML}])'}{\partial \theta} \left(I_{JM} - \frac{\partial \Psi(\hat{\theta}_{ML}, P[\hat{\theta}_{ML}])'}{\partial \Pi} \right)^{-1} diag\left\{ 1/P_{jm}(\hat{\theta}_{ML}) \right\} \; \bar{z} + \frac{\partial l_2(\widehat{\theta}_{ML})}{\partial \theta} = 0 \tag{A.9}$$

Equations (A.7) and (A.9) are not equivalent for values of Π which are not the fixed point. However, by Lemma 1, at any fixed point $\dfrac{\partial \Psi(\theta, P[\theta])'}{\partial \Pi} = 0$. Therefore, equation (A.9) can be simplified to:

$$\frac{\partial \Psi(\hat{\theta}_{ML}, P[\hat{\theta}_{ML}])'}{\partial \theta} \; diag\left\{ 1/P_{jm}[\hat{\theta}_{ML}] \right\} \; \bar{z} + \frac{\partial l_2(\widehat{\theta}_{ML})}{\partial \theta} = 0$$

which completes the proof.

Proof of Lemma 3.

Consistency and asymptotic normality follow from the results in Gourieroux and Monfort (1995, pp. 214-216). We first note two properties which we use repeatedly: (a) At the true parameter values, the pseudo choice probabilities $\Psi(\theta^*, \Pi^*)$ are the true choice probabilities $P(\theta^*)$. It follows from this that the partial pseudo-likelihood $\widetilde{l}_{1i}(\theta^*, \theta_f^*, \Pi^*)$ satisfies the "information equivalence", i.e., $E\left(-\dfrac{\partial^2 \widetilde{l}_{1i}^*}{\partial \theta \partial \theta'} \right) = E\left(\dfrac{\partial \widetilde{l}_{1i}^*}{\partial \theta} \dfrac{\partial \widetilde{l}_{1i}^*}{\partial \theta'} \right)$ and so on. (b) Furthermore, as seen in (A.8) in Appendix 1, $\dfrac{\partial P(\theta)}{\partial \theta'} = \left[I_{JM} - \dfrac{\partial \Psi(\theta, P(\theta))}{\partial \Pi'} \right]^{-1} \dfrac{\partial \Psi(\theta, P(\theta))}{\partial \theta'}$. Lemma 1 then implies the equivalence of scores and pseudo-scores at the true parameter values.

Asymptotic normality of the vector $\left[\frac{1}{\sqrt{n}}\sum_i \frac{\partial \widetilde{l}^*_{1i}}{\partial \theta'}, \sqrt{n}(\widehat{\theta}_f - \theta^*_f)', \sqrt{n}(\hat{\Pi}_0 - \Pi^*)'\right]'$ and the form of its asymptotic variance - can be established using a Central Limit Theorem applied to score functions as well as Taylor series expansions of the first order conditions defining the nuisance parameters $(\widehat{\theta}_f, \widehat{\Pi}_0)$. Notice that since $\widehat{\Pi}_K = \Psi(\widehat{\theta}_K, \widehat{\theta}_f, \widehat{\Pi}_{K-1})$ with Ψ a smooth function, consitency and asymptotic normality of $\widehat{\Pi}_K$ follow.

We now prove that H_Π is zero:

$$H_\Pi = E\left(\frac{\partial^2 \widetilde{l}^*_{1i}}{\partial\theta\partial\Pi'}\right) = E\left(\frac{\partial \widetilde{l}^*_{1i}}{\partial\theta}\frac{\partial \widetilde{l}^*_{1i}}{\partial\Pi'}\right) = E\left(\sum_t \frac{\partial \ln \Psi^*_{it}}{\partial\theta}\sum_t \frac{\partial \ln \Psi^*_{it}}{\partial\Pi'}\right)$$

But by Lemma 1 $\frac{\partial \Psi^*_{it}}{\partial \Pi'}$ is zero for any (d_{it}, x_{it}). Therefore, $H_\Pi = 0$ and the expression for $V(\hat{\theta}_K)$ becomes:

$$V(\hat{\theta}_K) = J^{-1}\left(-_{00} + H_f -_{f0} + -_{0f}\ H'_f + H_f -_{ff}\ H'_f\right) J^{-1}$$

But $J = E\left(-\frac{\partial^2 \widetilde{l}^*_{1i}}{\partial\theta\partial\theta'}\right) = E\left(\frac{\partial \widetilde{l}^*_{1i}}{\partial\theta}\frac{\partial \widetilde{l}^*_{1i}}{\partial\theta'}\right) = -_{00}$ and $-_{ff} = V(\hat{\theta}_f)$ which establishes (31).

Proof of Lemma 4.

The $1 - \textit{stage PI}$ estimator referred to in Lemma 4 is characterized in terms of the following sample moment conditions:

$$\begin{array}{rcl} \frac{1}{N}\sum_i \frac{\partial l_{2i}(\widehat{\theta}_f; y_{i1}, \ldots, y_{iT})}{\partial \theta_f} & = & 0 \\ \text{diag}\left\{\widehat{\Pi}_{0mj}\right\} \cdot (\bar{z} - \bar{X}\ \hat{\Pi}_0) & = & 0 \\ \Delta(\hat{\theta}_1, \widehat{\theta}_f, \widehat{\Pi}_0)\{\bar{z} - \bar{X}\ \Psi(\hat{\theta}_1, \widehat{\theta}_f, \widehat{\Pi}_0)\} & = & 0 \end{array}$$

where $\bar{z}$ and $\bar{X}$ are the sample means of z_{it} and X_{it}, respectively. But, by definition, this is the Hotz-Miller estimator of θ^*_{uq} when optimal instruments are used in the second step.

APPENDIX 2:

NFXP algorithm

[1] Inner (fixed point) algorithm:

- Smooth policy iteration algorithm.

- Convergence criterion: $\max_{\{j,m\}}\{|\Pi^{k}_{jm} - \Pi^{k-1}_{jm}|\} < 10^{-6}$

- At outer iteration K, the initial guess for the vector of probabilities is $P(\theta^{k-1})$. These "smart guesses" reduce very significantly the number of policy iterations when we advance in the outer iterations.

[2] Outer (hill climbing) algorithm:

- BHHH method.

- Convergence criterion: $\max_{\{j\}}\{|\theta^{k}_{j} - \theta^{k-1}_{j}|\} < 10^{-6}$

- Analytical expression for the gradient of the likelihood function. This implies that we only have to solve one fixed point problem for each outer iteration.

- β is fixed at 0.9999.

NPL algorithm

[1] Inner (pseudo-ML) algorithm:

- It is a simple Multinomial logit estimation where we restrict the parameter β to be equal to 0.9999.

- Convergence criterion: $\max_{\{j\}}\{|\theta^{k}_{j} - \theta^{k-1}_{j}|\} < 10^{-6}$

[2] Outer algorithm:

- "Pseudo" Policy iteration procedure, because at each iteration we use a new value of θ to obtain the new vector of probabilities.

- The convergence criterion is: $\max_{\{j,m\}}\{|\Pi^{k}_{jm} - \Pi^{k-1}_{jm}|\} < 10^{-6}$.

REFERENCES:

Eckstein, Z., and K. Wolpin (1989): "The specification and estimation of dynamic stochastic discrete choice models", *Journal of Human Resources* 24, pp. 562-598.

Gourieroux, C., and A. Monfort (1995): "Statistics and Econometric Models: Volume I", Cambridge University Press.

Hansen, L. P. (1982): "Large Sample Properties of Generalized Method of Moments", *Econometrica*, 50, pp. 1029-1054

Heckman, J., and B. Singer (1984): "A Method for Minimizing the Impact of Distributional Assumptions in Econometric Models for Duration Data", *Econometrica*, 52, pp. 271-320.

Hotz, J., and R.A. Miller (1993): "Conditional choice probabilities and the estimation of dynamic models", *Review of Economic Studies* 60, pp. 497-529.

Hotz, J., R.A. Miller, S. Sanders, and J. Smith (1994): "A simulation estimator for dynamic models of discrete choice", *Review of Economic Studies* 61, pp. 265-89.

Keane, M. P., and K. I. Wolpin (1994): "The solution and estimation of discrete choice dynamic programming models by simulation and interpolation: Monte Carlo evidence", *The Review of Economics and Statistics* 76, pp. 648-672.

Keane, M. P., and K. I. Wolpin (1997): "The Career Decisions of Young Men", *Journal of Political Economy*, 105, pp. 473-522.

Klein, R. W., and R. H. Spady (1993): "An Efficient Semiparametric Estimator for Binary Response Models", *Econometrica,* 61, pp. 387-421.

Manski, C. (1991): "Nonparametric estimation of expectations in the analysis of discrete choice under uncertainty", in *Nonparametric and Semiparametric Methods in Econometrics and Statistics*, W. Barnett, J. Powell, and G. Tauchen (eds.). Cambridge University Press.

Manski, C. (1993): "Dynamic choice in social settings: Learning from the experience of others", *Journal of Econometrics* 58, pp. 121-36.

McFadden, D. (1973): "Conditional logit analysis of qualitative choice behavior", in P. Zarembka (ed.) *Frontiers of Econometrics*, New York, Academic Press, pp. 105-42.

Miller, R. A. (1997): "Estimating Models of Dynamic Optimization with Microeconomic Data", in H. Pesaran and P. Smidth (eds.) *Handbook of Applied Econometrics: Microeconomics (Vol 2).* Blackwell.

Newey, W. K. (1984): "A Method of Moments Interpretation of Sequential Estimators", *Economic Letters*, 15, pp. 201-206.

Pakes, A, and P. McGuire (1996): "Stochastic Algorithms for Dynamic Models: Markov Perfect Equilibrium ad the 'Curse of Dimensionality"', manuscript, Department of Economics, Yale University.

Puterman, M. (1994): "Markov Decision Processes", Wiley, New York.

Rust, J. (1987): "Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher", *Econometrica* 55, pp. 999-1033.

Rust, J. (1988): "Maximum Likelihood Estimation of Discrete Control Processes", *SIAM Journal of Control and Optimization*, pp. 1006-1024

Rust, J. (1994): "Structural estimation of Markov decision processes", in R. E. Engle and McFadden (eds.) *Handbook of Econometrics Volume 4*, North-Holland.

Rust, J. (1996): "Numerical Dynamic Programming in Economics", in H Amman, D. Kendrick, and J. Rust (eds.). *Handbook of Computational Economics*, Elsevier-North Holland, Chapter 14.

Rust, J. (1997a): "Using Randomization to Break the Curse of Dimensionality", *Econometrica*, 65, pp. 487-516.

Rust, J. (1997b): "A Comparison of Policy Iteration Methods for Solving Continuous-State, Infinite-Horizon Markovian Decision Problems Using Random, Quasi-Random, and Deterministic Discretizations", manuscript, Department of Economics, Yale University.

Table 1

CPU time and number of policy iterations for NFXP and NPL algorithms

Rust's bus replacement model ; Size of the state space = 500

Initial Π_0 in NPL	Initial θ_0 in NFXP	Policy iter. NFXP (1)	Policy iter. NPL (2)	Ratio (1)/(2) (3)	CPU seconds NFXP (4)	CPU seconds NPL (5)	Ratio (4)/(5) (6)
	Number of structural parameters = 2						
Kernel							
	$\theta_0 = 0$	59	3	19.7	955	51	18.7
	$\theta_0 \in$ Frontier with sig. level 10^{-6}	34.5	3	11.5	553	51	10.8
Random							
	$\theta_0 = 0$	59	5	11.8	955	80	11.8
	$\theta_0 \in$ Frontier with sig. level 10^{-6}	34.5	5	6.9	553	80	6.9
	Number of structural parameters = 4						
Kernel							
	$\theta_0 = 0$	65	3	21.7	1058	59	17.9
	$\theta_0 \in$ Frontier with sig. level 10^{-6}	33.8	3	11.3	550	59	9.3
Random							
	$\theta_0 = 0$	65	5	13.0	1058	81	13.0
	$\theta_0 \in$ Frontier with sig. level 10^{-6}	33.8	5	6.8	550	81	6.8

Note: For the NFXP algorithm, the MLE has been obtained using 4 different values for the initial θ_0. Here we report the average numbers for these four computations. The initial values for θ_0 are such that they lie in the frontier of a confidence region with significance level equal to 10^{-6}.

Table 2
Monte Carlo Experiment 1
Sample size = 10,000
Initial Probabilities = Kernel estimates

Parameter	Statistics	Estimators			
		1-PI	2-PI	3-PI	MLE
θ_0	Mean absolute error	0.835	0.631	0.629	0.628
	(% over θ_0)	(7.98)	(6.02)	(6.01)	(6.00)
	Median absolute error	0.653	0.530	0.524	0.524
	(% over θ_0)	(6.24)	(5.06)	(5.01)	(5.01)
	Std. dev. estimator	1.056	0.787	0.784	0.784
	(% over θ_0)	(10.08)	(7.52)	(7.49)	(7.49)
θ_1	Mean absolute error	0.0782	0.0521	0.0517	0.0515
	(% over θ_1)	(13.48)	(8.99)	(8.91)	(8.89)
	Median absolute error	0.0558	0.0448	0.0439	0.0437
	(% over θ_1)	(9.61)	(7.72)	(7.57)	(7.54)
	Std. dev. of estimator	0.1024	0.0660	0.0654	0.0652
	(% over θ_1)	(17.66)	(11.38)	(11.27)	(11.25)

Average number of Policy Iterations to obtain MLE = 5.37

True parameters: $\theta_0 = 10.47$; $\theta_1 = 0.58$; $\beta = 0.9999$
State space = 200 ; Number of replications = 500

Table 3
Monte Carlo Experiment 2
Sample size = 5,000
Initial Probabilities = Kernel estimates

Parameter	Statistics	Estimators			
		1-PI	2-PI	3-PI	MLE
θ_0	Mean absolute error	1.032	0.789	0.786	0.784
	(% over θ_0)	(9.86)	(7.54)	(7.51)	(7.49)
	Median absolute error	0.780	0.652	0.649	0.645
	(% over θ_0)	(7.45)	(6.23)	(6.20)	(6.16)
	Std. dev. of estimator	1.277	0.997	0.993	0.992
	(% over θ_0)	(12.19)	(9.52)	(9.49)	(9.47)
θ_1	Mean absolute error	0.0991	0.0667	0.0664	0.0661
	(% over θ_1)	(17.08)	(11.5)	(11.5)	(11.4)
	Median absolute error	0.0753	0.0573	0.0575	0.0575
	(% over θ_1)	(13.00)	(9.88)	(9.92)	(9.91)
	Std. dev. of estimator	0.1249	0.0838	0.0833	0.0831
	(% over θ_1)	(21.54)	(14.44)	(14.35)	(14.32)

Average number of Policy Iterations to obtain MLE = 5.48

True parameters: $\theta_0 = 10.47$; $\theta_1 = 0.58$; $\beta = 0.9999$
State space = 200 ; Number of replications = 500

Table 4
Monte Carlo Experiment 3
Sample size = 2,500
Initial Probabilities = Kernel estimates

Parameter	Statistics	Estimators			
		1-PI	2-PI	3-PI	MLE
θ_0	Mean absolute error	1.476	1.309	1.300	1.296
	(% over θ_0)	(14.09)	(12.50)	(12.41)	(12.38)
	Median absolute error	1.198	1.013	1.002	1.009
	(% over θ_0)	(11.44)	(9.67)	(9.58)	(9.64)
	Std. dev. of estimator	1.832	1.707	1.698	1.695
	(% over θ_0)	(17.50)	(16.31)	(16.22)	(16.19)
θ_1	Mean absolute error	0.1388	0.1075	0.1067	0.1064
	(% over θ_1)	(23.93)	(18.54)	(18.40)	(18.34)
	Median absolute error	0.1096	0.0837	0.0839	0.0836
	(% over θ_1)	(18.90)	(14.42)	(14.47)	(14.42)
	Std. dev. of estimator	0.1684	0.1381	0.1369	0.1367
	(% over θ_1)	(29.04)	(23.81)	(23.60)	(23.56)

Average number of Policy Iterations to obtain MLE = 5.86

True parameters: $\theta_0 = 10.47$; $\theta_1 = 0.58$; $\beta = 0.9999$
State space = 200 ; Number of replications = 500

Table 5
Monte Carlo Experiment 4
Sample size = 5,000
Initial Probabilities = Random draws

Parameter	Statistics	Estimators			
		1-PI	2-PI	3-PI	MLE
θ_0	Mean absolute error	3.114	0.791	0.780	0.784
	(% over θ_0)	(29.75)	(7.56)	(7.45)	(7.49)
	Median absolute error	3.185	0.711	0.656	0.645
	(% over θ_0)	(30.42)	(6.79)	(6.27)	(6.16)
	Std. dev. of estimator	1.620	0.961	0.986	0.992
	(% over θ_0)	(15.47)	(9.18)	(9.42)	(9.47)
θ_1	Mean absolute error	4.1986	0.0758	0.0665	0.0661
	(% over θ_1)	(723.8)	(13.07)	(11.47)	(11.40)
	Median absolute error	4.1662	0.0682	0.0569	0.0575
	(% over θ_1)	(718.3)	(11.75)	(9.80)	(9.91)
	Std. dev. of estimator	0.5650	0.0853	0.0831	0.0831
	(% over θ_1)	(97.42)	(14.71)	(14.32)	(14.32)

Average number of Policy Iterations to obtain MLE = 7.06

True parameters: $\theta_0 = 10.47$; $\theta_1 = 0.58$; $\beta = 0.9999$
State space = 200 ; Number of replications = 500