Su, Che-Lin; Judd, Kenneth L.

Working Paper

Constrained optimization approaches to estimation of structural models

Discussion Paper, Center for Mathematical Studies in Economics and Management Science, No. 1460

Provided in Cooperation with:
Kellogg School of Management - Center for Mathematical Studies in Economics and Management Science, Northwestern University

Suggested Citation: Su, Che-Lin; Judd, Kenneth L. (2011) : Constrained optimization approaches to estimation of structural models, Discussion Paper, Center for Mathematical Studies in Economics and Management Science, No. 1460, Northwestern Univ., Kellogg Graduate School of Management, Center for Mathematical Studies in Economics and Management Science, Evanston

This Version is available at:
http://hdl.handle.net/10419/59626

Constrained Optimization Approaches to Estimation of Structural Models*

Che-Lin Su†
The University of Chicago
Booth School of Business

Kenneth L. Judd‡
Hoover Institution
and NBER

December 20, 2011

Abstract

Estimating structural models is often viewed as computationally difficult, an impression partly due to a focus on the nested fixed-point (NFXP) approach. We propose a new constrained optimization approach for structural estimation. We show that our approach and the NFXP algorithm solve the same estimation problem, and yield the same estimates. Computationally, our approach can have speed advantages because we do not repeatedly solve the structural equation at each guess of structural parameters. Monte Carlo experiments on the canonical Zurcher bus-repair model demonstrate that the constrained optimization approach can be significantly faster.

Keywords: structural estimation, dynamic discrete choice models, constrained optimization

*We have greatly benefited from the comments and suggestions of a co-editor and four anonymous referees. We are grateful to Richard W. Cottle and Harry J. Paarsch for their careful reading of this paper and detailed suggestions. We thank Steven Berry, John R. Birge, Jean-Pierre Dubé, Jeremy T. Fox, A. Ronald Gallant, Philip A. Haile, Lars P. Hansen, James J. Heckman, Günter Hitsch, Panle Jia, Sven Leyffer, Todd Munson, Jorge Nocedal, Ariel Pakes, Peter E. Rossi, John Rust, and seminar participants at various universities and conferences for helpful discussions and comments. We acknowledge financial support from the NSF under Grant SES-0631622. Su is grateful for financial support from the IBM Corporation Faculty Research Fund at the University of Chicago Booth School of Business. The AMPL and MATLAB code used in this paper are available at: `http://faculty.chicagobooth.edu/che-lin.su/research/code.html`

†Corresponding author. `E-mail:che-lin.su@chicagobooth.edu`

‡`E-mail:kennethjudd@mac.com`

1 Introduction

Structural estimation of economic models is an important approach to analyzing economic data. The main advantage of the structural approach is that it allows researchers to conduct counterfactual policy analysis, which cannot be undertaken using the reduced-form approach. However, the computational burden of estimating structural models can be a problem. It is commonly believed that these computational demands make it difficult to implement the most powerful and efficient statistical methods. For example, Rust (1987) proposed a computational strategy for the maximum-likelihood estimation of single-agent dynamic discrete-choice models, an approach referred to as the *nested fixed-point* (NFXP) algorithm. The NFXP algorithm is computationally demanding because it repeatedly takes a guess for structural parameters and then solves for the corresponding endogenous economic variables with high accuracy at each guess of the parameters.

In the context of estimating games, the computational costs associated with the NFXP algorithm increase. Games differ from single-agent dynamic models in that multiple equilibria can exist. When applying the NFXP algorithm to maximum-likelihood estimation of games, a researcher must find *all* of the Nash equilibria for each vector of parameters considered in order to calculate the corresponding likelihood value. Finding all of the equilibria in a game is a daunting computational task.[1] Even if one can solve for all the equilibria for every guess of structural parameters, then the resulting likelihood function can be non-smooth, even discontinuous in the parameter space, which makes the optimization problem extremely difficult to solve.

Thus, the potential computational burden of implementing the NFXP algorithm has led to the development of computationally light estimators, such as the two-step estimator of Hotz and Miller (1993) for estimating singe-agent dynamic discrete-choice models as well as the two-step estimators of Bajari, Benkard, and Levin (2007); Pakes, Ostrovsky, and Berry (2007); and Pesendorfer and Schmidt-Dengler (2008) for estimating dynamic games. Many of these two-step estimators are not asymptotically efficient.[2] Also, in finite samples, two-step estimators can perform poorly if the first-step estimates are imprecise, or if researchers do not choose suitable criterion functions in the second step. For example, Pakes, Ostrovsky,

[1] See the discussions in Besanko, Doraszelski, Kryukov, and Satterthwaite (2010) as well as Judd, Renner, and Schmedders (2010).

[2] The two-step estimator where the second step is one Newton-Raphson step towards the maximum-likelihood estimator is efficient; see, e.g., Newey and McFadden (1994) and Aguirregabiria and Mira (2007).

and Berry (2007) have found that maximizing the pseudo logarithm of the likelihood function in a two-step estimator can lead to bias in finite samples. Recognizing these limitations, Aguirregabiria and Mira (2002, 2007) propose the nested pseudo-likelihood (NPL) estimator, a recursive two-step pseudo-likelihood estimator, to improve the asymptotic properties and finite-sample performance of two-step estimators for estimating dynamic discrete-choice models.

We propose a new constrained optimization strategy to estimate structural econometric models, referred to as the mathematical program with equilibrium constraints (MPEC) approach; see Luo, Pang, and Ralph (1996). The idea behind the MPEC approach is simple: choose the structural parameters and endogenous economic variables so as to maximize the likelihood (or minimize the moment conditions) of the data subject to the constraints that endogenous economic variables are consistent with an equilibrium for the structural parameters. By formulating the estimation problem as a constrained optimization problem, researchers can simply write down expressions that define the likelihood or the moments as the objective function and the equilibrium equations as constraints and use existing constrained optimization solvers to calculate the estimates.

Computationally, our approach has the following advantage: we do not repeatedly solve for an equilibrium at each guess of structural parameters. Most existing constrained optimization algorithms do not enforce constraints to be satisfied until the final iteration in the search process.[3] Consequently, the only equilibrium that needs to be solved exactly is the one associated with the final estimate of structural parameters. This feature reduces the computational burden of estimating structural models and makes it possible for our approach to be faster than the NFXP algorithm.

The MPEC approach also builds on methods that have been developed in the statistics and econometrics literature. In particular, the examples that we shall examine below are constrained estimation problems of the kind described by Aitchison and Silvey (1958), Gallant and Holly (1980), Wolak (1987, 1989), as well as Gallant and Tauchen (1989).[4]

We first provide a general formulation of the MPEC approach to estimating structural models. We prove that MPEC and NFXP solve the same estimation problem and yield the same estimates. Thus, MPEC can be viewed as an alternative computational algorithm to

[3]Optimization solvers that incorporate this feature include FILTER (Fletcher and Leyffer (2002)), IPOPT (Wächter and Biegler (2006)), KNITRO (Byrd, Nocedal, and Waltz (2006)), SNOPT (Gill, Murray, and Saunders (2005)).

[4]We thank A. Ronald Gallant for pointing us to the constrained estimation literature.

NFXP for implementing the same statistical estimator.

We use the single-agent dynamic discrete-choice model studied by Rust (1987) as an illustrating example and derive the MPEC formulation for estimating this model. In the Monte Carlo experiments, we vary the discount factor in the dynamic programming model and investigate the corresponding performance of NFXP and MPEC. Monte Carlo results confirm that MPEC is significantly faster than NFXP, particularly when the discount factor is close to one. We also note that the computational time for MPEC did not change much across different discount factors, while the computational time for NFXP increased almost fivefold when the discounted factor increased from 0.975 to 0.995.

One natural concern is the computational feasibility of the MPEC approach; that is, the limitations concerning the size of problems that modern constrained optimization solvers can solve. If the constraint Jacobian and the Hessian of the Lagrangian are sparse, and first-order and second-order analytical derivatives and the corresponding sparsity patterns are supplied, then we believe that 100,000 variables and 100,000 constraints are reasonable size limitations for optimization problems to be solved using state-of-the-art constrained optimization solvers on a workstation. In fact, we have successfully solved a structural estimation problem with 100,052 variables and 100,042 constraints in an hour on a Macintosh workstation, MacPro, with 12 GB RAM. We expect that with the technological progress in computing hardware and software, the size of computationally tractable problems will grow accordingly, doubling perhaps every few years as has been the case for decades.

Dubé, Fox, and Su (2011) have illustrated the importance of supplying to optimization solvers the first-order and second-order analytical derivatives and sparsity patterns in the Jacobian and the Hessian. They applied MPEC to the GMM estimation of random-coefficients logit demand models. By deriving a constrained optimization formulation that allows optimization solvers to exploit the sparsity structure in the Jacobian and the Hessian, they demonstrated that MPEC can estimate random-coefficients demand models with thousands or tens of thousands of variables and constraints.

The remainder of the paper is organized as follows: in Section 2, we provide a general formulation of the MPEC approach to estimate structural models. We demonstrate the equivalence in solution between NFXP and MPEC. In Section 3, we consider the standard single-agent dynamic discrete-choice models in Rust (1987) and derive the corresponding formulation for MPEC. In Section 4, we present Monte Carlo experiments to investigate the computational performance of NFXP and MPEC. We summarize our results in Section 5.

2 MPEC Approach to Estimation

In this section, we provide a general formulation of the MPEC approach to estimate structural models. We prove the equivalence in formulation between NFXP and MPEC.

Suppose that an economic model is described by the parameter vector θ, a state vector x, and a vector of endogenous variables σ. The data are $X = \{x_i, d_i\}_{i=1}^{M}$, where x_i is the observed state, d_i is the observed equilibrium outcome of the underlying decision model and M is the number of data points. One example is a dynamic programming problem where θ contains the parameters for costs, profits, the laws of motion, and the stochastic shocks, while σ is the policy function of a decision maker in an environment described by the structural parameter vector θ. In general, σ will depend on θ through a set of equilibrium conditions such as the first-order conditions, Bellman equations, market balance conditions, and so forth. We express these conditions as a system of equations

$$h\left(\theta, \sigma\right) = 0. \tag{1}$$

For a given θ, let $\Sigma\left(\theta\right)$ denote the set of all σ such that $h\left(\theta, \sigma\right) = 0$; that is,

$$\Sigma(\theta) := \{\sigma : h(\theta, \sigma) = 0\}.$$

We let $\widehat{\sigma}(\theta)$ denote an element in $\Sigma(\theta)$. For some applications, such as single-agent dynamic discrete-choice models, $\widehat{\sigma}(\theta)$, which represents the expected value function, is uniquely determined, and the set $\Sigma(\theta)$ is a singleton for any given θ. In other applications, such as games, there can be multiple equilibria; hence, the set $\Sigma(\theta)$ can be multivalued for some θ.

We denote by $L\left(\theta, \widehat{\sigma}(\theta); X\right)$ the logarithm of the likelihood function of observing the data X, conditional on the parameter vector θ. If $\widehat{\sigma}(\theta)$ is unique for every θ, then the maximum-likelihood estimator is defined as

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmax}} \frac{1}{M} L\left(\theta, \widehat{\sigma}(\theta); X\right). \tag{2}$$

If there are multiple solutions in the set $\Sigma(\theta)$, then we assume that the data obtain from a single equilibrium, so that one can write down the likelihood function.[5] The maximum-

[5]Moment inequality methods do not require this assumption; see Ciliberto and Tamer (2009), and Pakes, Porter, Ho, and Ishii (2011).

likelihood estimator is then defined as

$$\widehat{\theta} = \underset{\theta}{\text{argmax}} \ \frac{1}{M} \left\{ \max_{\widehat{\sigma}(\theta) \in \Sigma(\theta)} L\left(\theta, \widehat{\sigma}(\theta); X\right) \right\}. \tag{3}$$

Notice that the consistency requirement between θ and σ described in condition (1) is implicitly taken into account by $\widehat{\sigma}(\theta)$ in the likelihood function.

In this formulation, the dependence of L on σ is implicit. In many applications, σ is more directly related to the likelihood than to some components of θ. In a life-cycle problem, the likelihood of an observation depends on the consumption and labor decisions, not on, for example, the elasticity of labor supply, which could be a component of θ. Elasticities affect likelihood only indirectly through their impact on decision rules.

Applying the NFXP algorithm to compute the maximum-likelihood estimator (3) requires the following: in an outer loop, search over the structural parameter vector θ to maximize the likelihood function; in an inner loop, for a given θ, find all the solutions $\widehat{\sigma}(\theta) \in \Sigma(\theta)$ and choose the solution that returns the highest likelihood value.

By simply replacing the equilibrium solution $\widehat{\sigma}(\theta)$ in the likelihood function by any σ, we construct an *augmented likelihood function*, denoted by $L\left(\theta, \sigma; X\right)$.[6] This function explicitly expresses the dependence of the likelihood on σ and makes no requirement that θ and σ be consistent with the equilibrium conditions of the economic model. For example, in a life-cycle model, the policy function σ together with the components of θ related to exogenous shock processes defines a stochastic process for the observables. A stochastic process is well defined even when σ and θ are inconsistent with an equilibrium. An augmented likelihood function allows us to treat σ and θ independently when we compute the likelihood.

When we estimate the structural parameter vector θ, we need σ to be consistent with θ. We achieve this by imposing the equilibrium condition (1). Therefore, we can formulate the maximum-likelihood estimation problem as a constrained optimization problem

$$\begin{aligned} \max_{(\theta,\sigma)} \quad & \frac{1}{M} L\left(\theta, \sigma; X\right) \\ \text{subject to} \quad & h\left(\theta, \sigma\right) = 0. \end{aligned} \tag{4}$$

[6] We thank Peter Rossi for suggesting this term.

The following proposition states the equivalence of the two formulations (3) and (4).[7]

Proposition 1. Let $\widehat{\theta}$ be the maximum likelihood estimator defined in (3) and $(\bar{\theta}, \bar{\sigma})$ be a solution of the constrained optimization problem (4). Define $\widehat{\sigma}^*(\theta) = \underset{\widehat{\sigma}(\theta)}{\operatorname{argmax}} L(\theta, \widehat{\sigma}(\theta))$. Then $L(\widehat{\theta}, \widehat{\sigma}^*(\widehat{\theta})) = L(\bar{\theta}, \bar{\sigma})$. If the model is identified, then we have $\widehat{\theta} = \bar{\theta}$.

Proof. Since $(\bar{\theta}, \bar{\sigma})$ satisfies equation (1), we have $\bar{\sigma} \in \Sigma(\bar{\theta})$ and hence, $L(\widehat{\theta}, \widehat{\sigma}^*(\widehat{\theta})) \geq L(\bar{\theta}, \bar{\sigma})$. Conversely, since $\widehat{\sigma}^*(\widehat{\theta}) \in \Sigma(\widehat{\theta})$, the pair $(\widehat{\theta}, \widehat{\sigma}^*(\widehat{\theta}))$ is a feasible solution for (4). It follows that $L(\bar{\theta}, \bar{\sigma}) \geq L(\widehat{\theta}, \widehat{\sigma}^*(\widehat{\theta}))$.

If the model is identified, then the solution is unique, so $\widehat{\theta} = \bar{\theta}$. ∎

The MPEC approach described above is not limited to maximum-likelihood estimation. It can be applied to any loss (or gain) function, such as least squares, weighted GMM, or simulated maximum likelihood, as the objective; see, for example, Dubé, Fox, and Su (2011).

3 Single-Agent Dynamic Discrete Choice Models

We consider the single-agent dynamic discrete-choice model in Rust (1987). We chose the Rust model because it has become the standard framework for modeling dynamic discrete-choice problems. It is also often used to evaluate the performance of alternative estimators. We first describe the NFXP algorithm proposed by Rust. We then derive the MPEC formulation for estimating this model.

3.1 Model of Bus-Engine Replacement Problem

In the bus-engine replacement problem of Rust (1987), the dynamic decisions faced by the manager, Harold Zurcher, are considered. In each period, a bus arrives in state x, the accumulated mileage since the last engine replacement. Given the observed mileage state x and other unobserved state variables ε, Zurcher must decide whether to perform extensive repairs on a bus or to implement less costly activities. Formally, the discrete decision in time

[7] The equivalence of (2) and (4) when $\widehat{\sigma}(\theta)$ is unique for every θ is simply a special case of Proposition 1.

t is defined as follows:

$$d_t = \begin{cases} 1, & \text{replacing the engine;} \\ 0, & \text{performing regular maintenance.} \end{cases}$$

If Zurcher chooses to replace the engine, then the mileage state x after the replacement is reset to zero, and the bus then goes out to travel for another period.

Let $c(x;\theta_1)$ denote the expected operating costs per period at mileage x, where θ_1 is a vector of parameters to be estimated. Let RC denote the expected replacement cost to install a new engine, net of any scrap value of the old engine. Given the state (x, ε) and the action d, the utility per period is

$$u(x, d, \varepsilon; \theta_1, RC) = \nu(x, d; \theta_1, RC) + \varepsilon(d)$$

where

$$\nu(x, d; \theta_1, RC) = \begin{cases} -c(x;\theta_1), & \text{if } d = 0 \\ -RC - c(0;\theta_1), & \text{if } d = 1 \end{cases}$$

and $\varepsilon = [\varepsilon(0), \varepsilon(1)]$ represents utility shocks.

The state variables (x_t, ε_t) evolve after a decision d_t has been made, so their evolution is described by the transition probability $p(x_{t+1}, \varepsilon_{t+1}|x_t, \varepsilon_t, d_t; \theta_2, \theta_3)$, where θ_2 and θ_3 are parameter vectors to be estimated. Note that the transition probability p is *Markovian*: it depends only on current state (x_t, ε_t). Let $\beta \in (0, 1)$ denote the discount factor. Given the current state (x_t, ε_t), the agent chooses a sequence of decision to maximize the following total expected discounted utility over an infinite horizon:

$$\max_{\{d_t, d_{t+1}, d_{t+2}, \ldots\}} \mathbb{E}\left[\sum_{\tau=t}^{\infty} \beta^{\tau-t} u(x_\tau, d_\tau, \varepsilon_\tau; \theta_1, RC)\right] \tag{5}$$

where the expectation is taken over the stochastic evolution of future states.

Define the value function V by

$$V(x_t, \varepsilon_t) = \max_{\{d_t, d_{t+1}, d_{t+2}, \ldots\}} \mathbb{E}\left[\sum_{\tau=t}^{\infty} \beta^{\tau-t} u(x_\tau, d_\tau, \varepsilon_\tau; \theta_1, RC)\right]. \tag{6}$$

Since the transition probability is Markovian, the optimal decision rule for the infinite-

horizon decision problem is time invariant, and we can drop the time index t. The infinite-horizon optimal decision problem (5) is then formulated as a solution to the Bellman equation

$$V(x,\varepsilon) = \max_d \left\{ \nu(x,d;\theta_1,RC) + \varepsilon(d) + \beta \int_{x'} \int_{\epsilon'} V(x',\varepsilon') p(x',\varepsilon'|x,\varepsilon,d;\theta_2,\theta_3) dx' d\varepsilon' \right\} \quad (7)$$

where (x',ε') denote the next-period state variables.

Rust (1987) discussed the computational difficulties in estimating the model specified above and introduced the *conditional independence* assumption on the transition probability p to simplify the estimation problem.

Assumption: Conditional Independence. The Markov transition probability of state variables (x,ε) can be decomposed as $p(x',\varepsilon'|x,\varepsilon,d;\theta_2,\theta_3) = p_2(\varepsilon'|x';\theta_2)p_3(x'|x,d;\theta_3)$.

Under the CI assumption, one can define the expected value function

$$EV(x) = \int_\varepsilon V(x,\varepsilon) p_2(\varepsilon|x;\theta_2) d\varepsilon$$

and the choice-specific value function

$$EV(x,d) = \nu(x,d;\theta_1,RC) + \varepsilon(d) + \beta \int_{x'} EV(x') p_3(x'|x,d;\theta_3) dx'.$$

Taking the expectation with respect to ε on both side of the Bellman equation (7) and using $EV(x)$ and $EV(x,d)$ defined above, one can obtain the integrated Bellman equation

$$EV(x) = \int_{\varepsilon'} \max_d \{EV(x,d)\} p_2(\varepsilon'|x;\theta_2) d\varepsilon'. \quad (8)$$

Let $\theta = (RC,\theta_1,\theta_3)$ denote the vector of parameters to be estimated. Assuming ε is a multivariate extreme-value distribution with θ_2 being the Euler's constant, Rust derived the following multinomial logit formula to characterize *conditional choice probability*:

$$P(d|x;\theta) = \frac{\exp\left[\nu(x,d;\theta_1,RC) + \beta EV(x,d)\right]}{\sum_{d' \in \{0,1\}} \exp\left[\nu(x,d';\theta_1,RC) + \beta EV(x,d')\right]} \quad (9)$$

where $P(d|x;\theta)$ is the probability of choosing decision d given the mileage state x and parameter vector θ. Furthermore, the choice-specific value function $EV(x,d)$ in equation (9)

is the unique fixed-point solution to the contraction mapping

$$EV(x,d) = \int_{x'=0}^{\infty} \log \left\{ \sum_{d' \in \{0,1\}} \exp\left[\nu(x',d';\theta_1,RC) + \beta EV(x',d')\right] \right\} p_3(dx'|x,d,\theta_3). \quad (10)$$

Since the mileage state space x is continuous, $EV(x,d)$ is a functional equation. Following Rust (1987), we discretize the mileage state space x into K grid points $\hat{\mathbf{x}} = \{\hat{x}_1, \ldots, \hat{x}_K\}$ with $\hat{x}_1 = 0$. We require the fixed-point equation (10) to be satisfied at those grid points. Given the current state $\hat{x}_k$, we assume that the mileage state in the next period x' can move up at most J grid points, and we specify the Markov transition probabilities as

$$p_3(x'|x_k,d,\theta_3) = \begin{cases} \Pr\{x' = \hat{x}_{k+j}|\theta_3\}, & \text{if } d=0 \\ \Pr\{x' = \hat{x}_{1+j}|\theta_3\}, & \text{if } d=1 \end{cases} \quad (11)$$

for $j = 0, 1, \ldots, J$. We can then rewrite the fixed-point equation (10) as

$$EV(\hat{x}_k,d) = \sum_{j=0}^{J} \log \left\{ \sum_{d' \in \{0,1\}} \exp\left[\nu(x',d';\theta_1,RC) + \beta EV(x',d')\right] \right\} p_3(x'|\hat{x}_k,d,\theta_3). \quad (12)$$

With a slight abuse of notation, we now denote

$$EV = [EV(\hat{x}_1,0), \ldots, EV(\hat{x}_K,0), EV(\hat{x}_1,1), \ldots, EV(\hat{x}_K,1)]$$

as a vector of expected value functions. We also denote the fixed-point equation (12) by

$$EV = T(EV,\theta). \quad (13)$$

The above characterization allows us to calculate the conditional choice probability $P(d|x;\theta)$ for a given vector of structural parameters θ by first solving the fixed-point equation (12) for $EV(x,d')$ for all d'. We refer readers to Rust (1987, 1994) for detailed derivation and discussions.[8]

[8]With the Markov transition probabilities $p_3(x'|x_k,d,\theta_3)$ defined in equation (11), we have $EV(\hat{x}_k,1) = EV(\hat{x}_1,0)$ for all $k = 1,\ldots,K$. Consequently, one can define $EV = [EV(\hat{x}_1,0),\ldots,EV(\hat{x}_K,0]$; see Rust (2000, p. 14).

3.2 Maximum-Likelihood Estimation

The econometrician observes the mileage state and the corresponding decision in each period and for each bus in the data. Let $X^i = (x_t^i, d_t^i)_{t=1}^T$ be the time-series data for bus $i = 1, \ldots, M$; x_t^i is the mileage state of bus i examined in period t and d_t^i is the replacement decision made for that bus. For simplicity, we assume x_t^i stays on the grid $\hat{\mathbf{x}} = \{\hat{x}_1, \ldots, \hat{x}_K\}$.[9] The full data set including all buses is denoted by $X = (X^i)_{i=1}^M$.

Rust (1987) proposed to estimate the structural parameter vector θ by the method of maximum likelihood. The likelihood of observing data X^i for bus i is

$$\ell_i(X^i; \theta) = \prod_{t=2}^{T} P(d_t^i | x_t^i; \theta) p_3(x_t^i | x_{t-1}^i, d_{t-1}^i; \theta_3),$$

where $P(d|x, \theta)$ is given by (9). The likelihood function of the full data set $X = (X^i)_{i=1}^M$ is

$$\ell(\theta) = \prod_{i=1}^{M} \ell_i(X^i; \theta) = \prod_{i=1}^{M} \prod_{t=2}^{T} P(d_t^i | x_t^i; \theta) p_3(x_t^i | x_{t-1}^i, d_{t-1}^i; \theta_3), \tag{14}$$

and the logarithm of the likelihood function is denoted by

$$L(\theta) = \log \ell(\theta) = \sum_{i=1}^{M} \sum_{t=2}^{T} \log \left[P(d_t^i | x_t^i; \theta) \right] + \sum_{i=1}^{M} \sum_{t=2}^{T} \log \left[p_3(x_t^i | x_{t-1}^i, d_{t-1}^i; \theta_3) \right]. \tag{15}$$

We calculate the estimates of unknown parameter vector $\theta = (RC, \theta_1, \theta_3)$ by maximizing the sample-averaged logarithm of the likelihood function

$$\max_{\theta} \frac{1}{M} L(\theta). \tag{16}$$

To evaluate the logarithm of the likelihood function $L(\theta)$ for a given θ, one needs to know the value of $P(d|x; \theta)$, which in turn requires the value of EV, a solution of the fixed-point equation (12). This fact motivates the NFXP algorithm proposed in Rust (1987). NFXP consists of an outer loop and an inner loop. In the outer loop, an optimization procedure is used to maximize $\frac{1}{M} L(\theta)$ by changing the structural parameters θ; in the inner loop, for a given θ, one solves the fixed-point equation (12) for EV to calculate the conditional choice probabilities $P(d_t^i | x_t^i; \theta)$, and to evaluate $L(\theta)$. Rust (1988) proved that $L(\theta)$ is a smooth

[9] If x_t^i is not on the grid $\hat{\mathbf{x}}$, then we round x_t^i up or down to the nearest grid point.

function of θ, an attractive property that makes Newton's method a valid computational approach for solving this maximization problem. Rust derived formulae for calculating the first-order derivatives of $L(\theta)$ using implicit function theorem. Supplying the analytical first-order derivatives in the implementation of Newton's method makes NFXP faster and more robust than when finite-difference derivatives are used.

NFXP requires the fixed-point equation (12) to be solved accurately for every vector of parameters θ considered in the maximization procedure.[10] Aguirregabiria and Mira (2002) have noted that repeatedly solving the fixed-point equation is not computationally efficient because it is unnecessary to enforce the fixed-point equation to be satisfied during the search process in the NFXP outer loop. They proposed a nested pseudo-likelihood estimator to avoid repeated solution of the fixed-point equation. With the same motivation in mind, we propose the MPEC approach for maximum-likelihood estimation of Rust's bus model.

For MPEC, the augmented log-likelihood function is the following function of both structural parameter vector θ and expected value functions EV:

$$\begin{aligned}
\mathcal{L}(\theta, EV) &= \sum_{i=1}^{M}\sum_{t=2}^{T} \log\left[P(d_t^i|x_t^i;\theta)\right] + \sum_{i=1}^{M}\sum_{t=2}^{T} \log\left[p_3(x_t^i|x_{t-1}^i, d_{t-1}^i;\theta_3)\right] \\
&= \sum_{i=1}^{M}\sum_{t=2}^{T} \log\left(\frac{\exp\left[\nu(x_t^i, d_t^i, \theta) + \beta EV(x_t^i, d_t^i)\right]}{\sum_{d'\in\{0,1\}} \exp\left[\nu(x_t^i, d', \theta) + \beta EV(x_t^i, d')\right]}\right) \\
&\quad + \sum_{i=1}^{M}\sum_{t=2}^{T} \log\left[p_3(x_t^i|x_{t-1}^i, d_{t-1}^i;\theta_3)\right].
\end{aligned} \tag{17}$$

Note that, in defining the augmented log-likelihood function (17), we have substituted the conditional-choice probability $P(d|x;\theta)$ into $L(\theta)$ in equation (15) by the multinomial logit formula given in equation (9). The function $\mathcal{L}(\theta, EV)$ is a well-defined smooth function of θ and EV, and its first-order and second-order derivatives with respect to θ and EV exist.

To ensure that EV and θ are consistent in the structural equation, we impose the integrated Bellman equations defined by equation (13) as constraints. Combining the sample-averaged augmented log-likelihood function as the objective function and the integrated

[10] Rust (1987) started with contraction mapping (at a tolerance level of 10^{-13}) and then switched to Newton's method to solve the fixed-point equation with very high accuracy; see also Rust (2000).

Bellman equations as constraints yields the following constrained optimization problem:

$$\begin{aligned} \max_{(\theta, EV)} \quad & \frac{1}{M} \mathcal{L}(\theta, EV) \\ \text{subject to} \quad & EV = T(EV, \theta). \end{aligned} \tag{18}$$

It follows from Proposition 1 that the maximum-likelihood estimation problem defined by (16) and the constrained optimization problem (18) are mathematically equivalent. Consequently, NFXP and MPEC yield the same parameter estimates. Dubé, Fox, and Su (2011) proved the equivalence in solutions to the first-order conditions of NFXP and MPEC in the context of random-coefficients demand estimation. Their results and proof can be applied directly to the single-agent dynamic models considered here as well.

We now discuss several computational aspects of implementing NFXP and MPEC. First, for each θ considered under NFXP, there are two steps: (i) solving the dynamic programming problem with high accuracy; and (ii) evaluating the likelihood function. Value function iteration requires evaluating the Bellman equations many times under NFXP each time a dynamic program is solved. In contrast, for each θ considered in MPEC, the Bellman equation is evaluated *once*, but not *solved*. This feature reduces the computational burden and makes it possible for MPEC to be much faster than NFXP.

Second, NFXP and MPEC require about the same amount of memory. The constraint Jacobian and the Hessian of the Lagrangian are highly sparse in the bus-engine replacement model. Let $dim(z)$ denote the dimension of a vector z. The total number of elements in the Jacobian and the Hessian are $(dim(\theta) + dim(EV)) \times dim(EV)$ and $(dim(\theta) + dim(EV))^2$. One can verify that the number of nonzero elements in the Jacobian is around $(dim(\theta) + dim(\theta_3) + 1) \times dim(EV)$. The Hessian is even more sparse than the Jacobian: the number of nonzero elements in the Hessian is around $(2 \times (1 + dim(\theta_1)) + dim(\theta_3)) \times (dim(\theta) + dim(EV))$. In most dynamic discrete-choice models, $\dim(\theta)$ is much smaller than $dim(EV)$. Thus, the memory required for the constrained optimization approach on Zurcher's model is on the order of $dim(\theta) + dim(EV)$, the number of variables in the MPEC approach.

4 Monte Carlo Experiments

We investigated a specific example of Rust's bus-engine replacement model, following the specifications used to report the estimates in Table X in Rust (1987, p. 1022). In particular, we chose the fixed-point dimension to be 175 (i.e., we discretized the mileage state space into 175 grid points) and the maintenance cost function to be $c(x, \theta_1) = 0.001\theta_{11}x$. The Markov transition probabilities in the mileage state were defined as $\theta_{3j} = \Pr(x_{t+1} = x_t + j | x_t, i_t = 0)$, for $j = 0, 1, \ldots, 4$. Hence, θ_{3j} is the probability that the mileage state will move up j grid points in the next period. The unknown structural parameters to be estimated are replacement cost RC, maintenance cost function parameter θ_{11}, and mileage transition probabilities $\theta_3 = (\theta_{3j})_{j=0}^4$. We did not estimate the discount factor β. Instead, we examined five cases with β ranging from 0.975 to 0.995 in an interval of 0.005.[11]

We used the parameter values reported for bus groups 1, 2, and 3 in Table X in Rust (1987) as the true values of the data generating process:

$$\begin{aligned} RC^0 &= 11.7257, \\ \theta_{11}^0 &= 2.4569, \\ \theta_3^0 &= (0.0937, 0.4475, 0.4459, 0.0127, 0.0002). \end{aligned}$$

We simulated time-series data for $T = 120$ time periods (10 years of monthly data) and $M = 50$ buses. To simulate data for each of the five cases and the corresponding β, we fixed the structural parameters at $\theta^0 = (RC^0, \theta_{11}^0, \theta_3^0)$ and solved the integrated Bellman equation (12) for EV^0. We then used θ^0 and EV^0 to compute the conditional choice probabilities $P(d|x, \theta)$ and to simulate mileage transitions and decisions for 250 data sets for each β in our Monte Carlo study.

We conducted three implementations of algorithms. In the first, we coded the MPEC approach in AMPL, an algebraic modeling language; see Fourer, Gay, and Kernighan (2003).[12] AMPL has two advantages: first, it will compute the exact first-order and second-order analytic derivatives using automatic differentiation (see Griewank and Walther (2008)); second, it will analyze the sparsity structure of the constraint Jacobian and the Hessian, and provide that information to optimization solvers.[13] We denote this implementation as

[11] Rust (1997) chose $\beta = 0.9999$.

[12] We cannot implement NFXP in AMPL because AMPL does not allow nested function calls.

[13] For this example, the number of nonzeros in Jacobian is 2255 ($\approx (7 + 5 + 1) \times 175$) and the number of nonzeros in Hessian is 1584 ($\approx (2 \times (1 + 1) + 5) \times 182$).

MPEC/AMPL.

We also coded both MPEC and NFXP in MATLAB. For MPEC, we provided the hand-coded first-order analytical derivatives of the objective function and constraints and the sparsity pattern of the constraint Jacobian. For NFXP, we provided the first-order analytical derivatives of objective function; we used only contraction mapping iterations to solve the integrated Bellman equation (12) with an inner-loop tolerance 10^{-10}.[14] We denote these two implementations as MPEC/MATLAB and NFXP/MATLAB, respectively.

Rust (1987) estimated the transition-probability parameters θ_3 from the mileage transition data in the first stage and then estimated the remaining parameters (RC, θ_{11}) in the second stage. For both MPEC and NFXP, we estimated the parameters $(RC, \theta_{11}, \theta_3)$ jointly by solving the full likelihood function (15). For all three implementations, we solved the estimation problem using the optimization solver, KNITRO; see Byrd, Nocedal, and Waltz (2006). For each of the 1,250 replications (250 replications for each of the five βs) , we used five different starting values to do a better job at finding a global solution. For a fair comparison, all three implementations used the same starting values in the structural parameter vector θ.

In Table 1, we report the Monte Carlo results. These three implementations produced almost identical means and standard deviations of estimates in each of the five scenarios when we varied the discount factor β. We believe the differences arise because different local maximizers were found in these three implementations. The estimation results demonstrate that the maximum-likelihood estimator works extremely well on this model, particularly in recovering the transition probabilities. For $\beta = 0.99$ and 0.995, all mean parameter estimates except for θ_{30} are within two standard errors of the true values.

We report the computational performance of these three implementations in Table 2. One can see that the MPEC/MATLAB implementation is the most robust in terms of number of runs converged. In terms of timing, MPEC/AMPL is the fastest, requiring only around 0.15 seconds for each run and considerably fewer iterations and function evaluations than either MPEC/MATLAB or NFXP/MATLAB. The reason is that AMPL supplies both first-order and second-order analytical derivatives to the solver KNITRO, whereas for the other

[14] NFXP will need more computational time if researchers use tighter inner-loop tolerance level such as 10^{-13} or follow Rust's suggestion to switch to Newton's method after the contraction mapping iteration converges; see footnote 9. Since we provided hand-coded first-order derivatives in our implementation, we found that NFXP using contraction mapping iteration only with an inner loop tolerance 10^{-10} is quite robust.

two implementations we supplied only first-order analytical derivatives. This observation confirms that an optimization solver runs most efficiently and needs the fewest iterations and function evaluations when both first-order and second-order derivatives are supplied. When we remove the effect of supplying second-order derivatives and compare the performance of MPEC/MATLAB and NFXP/MATLAB, MPEC still dominates NFXP in all categories. As one can see, the number of major iterations and function evaluations for NFXP and MPEC do not increase when we increase the discount factor from 0.975 to 0.995; however, the computing time for NFXP increases steadily, while the computing time for MPEC/MATLAB (and MPEC/AMPL) requires the same. The reason for the increase in computing time for NFXP is that when the discount factor β increases, the number of contraction mapping iterations needed in solving the inner loop Bellman equations also increases.

Instead of calculating the asymptotic standard errors, one can use the bootstrap to construct confidence interval or test hypothesis. The feasibility of conducting bootstrap inference relies on having a computationally efficient method to perform parameter estimation of the underlying structural model since the bootstrap requires estimating parameters for each of the simulated samples. As we illustrated above, MPEC provides a computationally efficient method to estimate dynamic discrete-choice models and makes bootstrap inference possible. Indeed, the data sampling procedure that we describe above is a parametric bootstrap procedure to generate simulated samples and our Monte Carlo study demonstrates the uses of parametric bootstrap to compute standard errors on structural parameters.

5 Conclusion

In this paper, we have proposed a new constrained optimization approach, MPEC, for estimating structural econometrics models. We have illustrated that the MPEC approach can be applied directly to maximum-likelihood estimation of single-agent dynamic discrete-choice models. Our approach can be easily implemented using existing standard constrained optimization software. Monte Carlo results confirmed that MPEC is significantly faster than NFXP, particularly when the discount factor in the dynamic-programming model is close to one.

As shown by Dubé, Fox, and Su (2011), MPEC can also be applied to estimate random-coefficients logit demand models. We believe that our approach will be useful for estimating structural models in various contexts and applications. For future research, we plan to

investigate the applicability of the MPEC approach to estimate dynamic discrete-choice games studied in Aguirregabiria and Mira (2007); Bajari, Benkard, and Levin (2007); Pakes, Ostrovsky, and Berry (2007); Pesendorfer and Schmidt-Dengler (2008); and Arcidiacono and Miller (2011).

References

[1] Arcidiacono, P., and R. A. Miller (2011): "CCP Estimation of Dynamic Discrete Choice Models with Unobserved Heterogeneity," *Econometrica*, 79, 1823–1868.

[2] Aguirregabiria, V., and P. Mira (2002): "Swapping the Nested Fixed Point Algorithm: A Class of Estimators for Discrete Markov Decision Models," *Econometrica*, 70, 1519–1543.

[3] Aguirregabiria, V., and P. Mira (2007): "Sequential Estimation of Dynamic Discrete Games," *Econometrica*, 75, 1–53.

[4] Aitchison, J., and S. D. Silvey (1958): "Maximum Likelihood Estimation of Parameters subject to Restraints," *Annals of Mathematical Statistics*, 29, 813–828.

[5] Bajari, P., C. L. Benkard, and J. Levin (2007): "Estimating Dynamic Games of Imperfect Competition," *Econometrica*, 75, 1331–1370.

[6] Besanko, D., U. Doraszelski, Y. Kryukov, and M. Satterthwaite (2010): "Learning-by-Doing, Organizational Forgetting, and Industry Dynamics," *Econometrica*, 78, 453–508.

[7] Byrd, R. H., J. Nocedal, and R. A. Waltz (2006): "KNITRO: An Integrated Package for Nonlinear Optimization," in *Large-Scale Nonlinear Optimization* ed. by G. di Pillo and M. Roma. Springer Verlag, 35–59.

[8] Ciliberto, F., and E. Tamer (2009): "Market Structure and Multiple Equilibria in Airline Markets," *Econometrica*, 77, 1791–1828.

[9] Dubé, J.-P., J. T. Fox, and C.-L. Su (2011): "Improving the Numerical Performance of BLP Static and Dynamic Discrete Choice Random Coefficients Demand Estimation," Working paper, Booth School of Business, University of Chicago.

[10] Fletcher, R., and S. Leyffer (2002): "Nonlinear Programming without a Penalty Function," *Mathematical Programming*, 91, 239–270.

[11] Fourer, R., D. M. Gay, and B. W. Kernighan (2003): *AMPL: A Modeling Language for Mathematical Programming*, Brooks/Cole – Thomson Learning, Pacific Grove, California, second edition.

[12] Griewank, A., and A. Walther (2008): *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, SIAM, Philadelphia, second edition.

[13] Gallant, A. R., and A. Holly (1980): "Statistical Inference in an Implicit, Nonlinear, Simultaneous Equation Model in the Context of Maximum Likelihood Estimation," *Econometrica*, 48, 697–420.

[14] Gallant, A. R., and G. E. Tauchen (1989): "Seminonparametric Estimation of Conditionally Constrained Heterogeneous Processes: Asset Pricing Applications," *Econometrica*, 57, 1091–1120.

[15] Gill, P. E., W. Murray and M. A. Saunders (2005): SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, 47, 99–131.

[16] Hotz, J., and R. A. Miller (1993): "Conditional Choice Probabilities and the Estimation of Dynamic Models," *Review of Economic Studies*, 60, 497–529.

[17] Judd, K. L., P. Renner, and K. Schmedders (2010): "Finding All Pure Strategy Equilibria in Dynamic and Static Games with Continuous Strategies," Working paper, University of Zurich.

[18] Luo, Z.-Q., J.-S. Pang, and D. Ralph (1996): *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, UK.

[19] Newey, W. K. and D. L. McFadden (1994): "Large Sample Estimation and Hypothesis Testing," in *Handbook of Econometrics*, Vol. 4, ed. by R. F. Engle and D. L. McFadden, Elsevier, Amsterdam, North-Holland, 2113–2245.

[20] Pakes, A., J. Porter, K. Ho, and J. Ishii (2011): "Moment Inequalities and Their Applications," Working paper, Harvard University.

[21] Pesendorfer, M., and P. Schmidt-Dengler (2008): "Asymptotic Least Squares Estimators for Dynamic Games," *Review of Economic Studies*, 75, 901–928.

[22] Rust, J. (1987): "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher," *Econometrica*, 55, 999–1033.

[23] Rust, J. (1988): "Maximum Likelihood Estimation of Discrete Control Processes," *SIAM Journal on Control and Optimization*, 26, 1006–1024.

[24] Rust, J. (1994): "Structural Estimation of Markov Decision Processes," in *Handbook of Econometrics*, Vol. 4, ed. by R. F. Engle and D. L. McFadden, Elsevier, Amsterdam, North-Holland, 3082–3139.

[25] Rust, J. (2000): *Nested Fixed Point Algorithm Documentation Manual: Version 6*, Department of Economics, Yale University.

[26] Wächter, A., and L. T. Biegler (2006): "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, 106, 25–57.

[27] Wolak, F. A. (1987): "An Exact Test for Multiple Inequality and Equality Constraints in the Linear Regression Model," *Journal of American Statistical Association*, 82, 782–793.

[28] Wolak, F. A. (1989): "Testing Inequality Constraints in Linear Econometric Models," *Journal of Econometrics*, 41, 205–235.

Table 1: Monte Carlo Results for Various Discount Factors β

β	Implementation		Parameters						MSE
			RC	θ_{11}	θ_{30}	θ_{31}	θ_{32}	θ_{33}	
		True values	**11.726**	**2.457**	**0.0937**	**0.4475**	**0.4459**	**0.0127**	
0.975	MPEC/AMPL	Mean	12.212	2.607	0.0943	0.4473	0.4454	0.0127	3.111
		Std. dev.	(1.613)	(0.500)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–
	MPEC/MATLAB	Mean	12.212	2.607	0.0943	0.4473	0.4454	0.0127	3.111
		Std. dev.	(1.613)	(0.500)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–
	NFXP/MATLAB	Mean	12.213	2.606	0.0943	0.4473	0.4445	0.0127	3.123
		Std. dev.	(1.617)	(0.500)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–
0.980	MPEC/AMPL	Mean	12.134	2.578	0.0943	0.4473	0.4455	0.0127	2.857
		Std. dev.	(1.570)	(0.458)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	–
	MPEC/MATLAB	Mean	12.134	2.578	0.0943	0.4473	0.4455	0.0127	2.857
		Std. dev.	(1.570)	(0.458)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	–
	NFXP/MATLAB	Mean	12.139	2.579	0.0943	0.4473	0.4455	0.0127	2.866
		Std. dev.	(1.571)	(0.459)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	–
0.985	MPEC/AMPL	Mean	12.013	2.541	0.0943	0.4473	0.4455	0.0127	2.140
		Std. dev.	(1.371)	(0.413)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	–
	MPEC/MATLAB	Mean	12.013	2.541	0.0943	0.4473	0.4455	0.0127	2.140
		Std. dev.	(1.371)	(0.413)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	–
	NFXP/MATLAB	Mean	12.021	2.544	0.0943	0.4473	0.4455	0.0127	2.136
		Std. dev.	(1.368)	(0.411)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	–
0.990	MPEC/AMPL	Mean	11.830	2.486	0.0943	0.4473	0.4455	0.0127	1.880
		Std. dev.	(1.305)	(0.407)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–
	MPEC/MATLAB	Mean	11.830	2.486	0.0943	0.4473	0.4455	0.0127	1.880
		Std. dev.	(1.305)	(0.407)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–
	NFXP/MATLAB	Mean	11.830	2.486	0.0943	0.4473	0.4455	0.0127	1.880
		Std. dev.	(1.305)	(0.407)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–
0.995	MPEC/AMPL	Mean	11.819	2.492	0.0942	0.4473	0.4455	0.0127	1.892
		Std. dev.	(1.308)	(0.414)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–
	MPEC/MATLAB	Mean	11.819	2.492	0.0942	0.4473	0.4455	0.0127	1.892
		Std. dev.	(1.308)	(0.414)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–
	NFXP/MATLAB	Mean	11.819	2.492	0.0942	0.4473	0.4455	0.0127	1.892
		Std. dev.	(1.308)	(0.414)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	–

For each β, there are 250 replications. Standard deviations are reported in parentheses. MSE is calculated by summing over all structural parameters.

Table 2: Numerical Performance of NFXP and MPEC in the Monte Carlo Experiments

β	Implementation	Runs Converged (out of 1250 runs)	CPU Time (in sec.)	# of Major Iter.	# of Func. Eval.	# of Contraction Mapping Iter.
0.975	MPEC/AMPL	1240	0.13	12.8	17.6	–
	MPEC/MATLAB	1247	7.90	53.0	62.0	–
	NFXP	998	24.60	55.9	189.4	134,748
0.980	MPEC/AMPL	1236	0.15	14.5	21.8	–
	MPEC/MATLAB	1241	8.10	57.4	70.6	–
	NFXP	1000	27.90	55.0	183.8	162,505
0.985	MPEC/AMPL	1235	0.13	13.2	19.7	–
	MPEC/MATLAB	1250	7.50	55.0	62.3	–
	NFXP	952	43.20	61.7	227.3	265,827
0.990	MPEC/AMPL	1161	0.19	18.3	42.2	–
	MPEC/MATLAB	1248	7.50	56.5	65.8	–
	NFXP	935	70.10	66.9	253.8	452,347
0.995	MPEC/AMPL	965	0.14	13.4	21.3	–
	MPEC/MATLAB	1246	7.90	59.6	70.7	–
	NFXP	950	111.60	58.8	214.7	748,487

For each β, we use five starting points for each of the 250 replications. CPU time, number of major iterations, number of function evaluations and number of contraction mapping iterations are the averages for each run.