# A Genetic Algorithm for the Structural Estimation of Games: A First Report[1]

**Victor Aguirregabiria and Pedro Mira**
Boston University and CEMFI, Madrid

## Abstract

This paper proposes a hybrid algorithm to obtain maximum likelihood estimates of structural parameters in discrete games with multiple equilibria. The algorithm combines a pseudo maximum likelihood (PML) procedure with a genetic algorithm (GA). The GA searches efficiently over the huge space of possible combinations of equilibria in the data. The PML procedure avoids the repeated computation of equilibria for each trial value of the parameters of interest.

**Keywords:** Genetic algorithms; Maximum likelihood estimation; Multiple equilibria.

## 1. Introduction

Multiplicity of equilibria is a prevalent feature in empirical discrete games. Models with multiple equilibria do not have a unique reduced form and this indeterminacy poses practical problems in maximum likelihood estimation. Aguirregabiria and Mira (2003), Pakes, Ostrowsky and Berry (2003), and Pesendorfer and Smichdt-Dengler (2003) propose pseudo maximum likelihood (PML) estimators that deal with this problem. These PML methods proceed in two steps. The first step identifies nonparametrically the equilibrium (or equilibria) played in the data. The second step estimates structural parameters by maximizing a (pseudo) likelihood function based on best response functions evaluated at the equilibrium estimated in the first step. The main advantage of PML estimation is its computational simplicity. However, the method is not statistically efficient, and it can perform poorly in small samples. In this paper we propose an algorithm to obtain the efficient maximum likelihood estimator of structural parameters in games with multiple equilibria.

A problem associated with the ML estimation of models with multiple equilibria is that we should maximize the likelihood not only with respect to the structural parameters but also with respect to the *equilibrium types* that generate the observations in the data. There are two main reasons why optimization with respect to *equilibrium types* can be a very complicated task. First, computing all the equilibria associated with each trial value of the parameters can be computationally very demanding. And second, the number of possible combinations of equilibria in the data increases exponentially with the sample size and it is huge even for the simplest problems. In this paper we propose a relatively simple but computationally effective algorithm to obtain the ML estimator. The algorithm combines the PML procedure in Aguirregabiria and Mira (2003) with a Genetic Algorithm (GA). The GA searches efficiently over the huge space of possible combinations of equilibria in the data. The PML procedure avoids the repeated computation of equilibria for each trial value of the structural parameters.

GAs were first proposed by Holland (1975). Among their many applications, GA have been successfully used to search for global optima of discrete and step functions with very large search spaces (see chapter 4 in Mitchell, 1996, and Mitchell, Holland and Forrest, 1994). The problem of maximum likelihood estimation of models with multiple equilibria belongs to this class. Although GAs have been extensively used in experimental and evolutionary economics, the application of GAs in econometrics has been rare. Important exceptions are Dorsey and Mayer (1995) and Beenstock and Szpirob (2002).

## 2. Model

There are $N$ players which are indexed by $i \in I = \{1, 2, ..., N\}$. Each player should choose an action from a set of choice alternatives $A = \{0, 1, ..., J\}$. We represent the decision of player $i$ by the variable $a_i \in A$. The utility function of player $i$ is:

$$U_i = u_i(a_i, a_{-i}, x) + \varepsilon_i(a_i) \qquad (1)$$

where $a_{-i}$ is the vector with the decisions of players other than $i$; $x$ is a vector of players' exogenous characteristics which are common knowledge; and $\varepsilon_i \equiv (\varepsilon_i(0), \varepsilon_i(1), ..., \varepsilon_i(J))$ represents characteristics that are private information of player $i$.

*ASSUMPTION 1: For any $i \in I$ the vector $\varepsilon_i \in R^{J+1}$ is: (1) independent of common knowledge variables $x$; and (2) independently distributed across*

[1]This material is based upon work supported by the National Science Foundation under Grant No. 0241943.

*players with distribution function $G_i(.)$ that is absolutely continuous with respect to the Lebesgue measure.*

Let $\sigma = \{\sigma_i(x, \varepsilon_i) : i \in I\}$ be a set of strategy functions where $\sigma_i : X \times R^{J+1} \to A$. Associated with a set of strategy functions we can define a set of *choice probabilities* $P^\sigma(x) = \{P_i^\sigma(a_i|x) : (a_i, i) \in A \times I\}$ such that:

$$P_i^\sigma(a_i|x) \equiv \int I\{\sigma_i(x, \varepsilon_i) = a_i\} \; dG_i(\varepsilon_i) \quad (2)$$

where $I\{.\}$ is the indicator function. These probabilities represent the expected behavior of player $i$ from the point of view of the other players (who do not know $\varepsilon_i$) when he follows his strategy in $\sigma$. Let $u_i^\sigma(a_i, x)$ be player $i$'s expected utility if he chooses alternative $a_i$ and the other players behave according to their respective strategies in $\sigma$. By the independence of private information in Assumption 1,

$$u_i^\sigma(a_i, x) = \sum_{a_{-i}} \left( \prod_{j \neq i} P_j^\sigma(a_j|x) \right) u(a_i, a_{-i}, x) \quad (3)$$

*DEFINITION: A Bayesian Nash equilibrium (BNE) in this game is a set of strategy functions $\sigma^*$ such that for any player $i$ and for any $(x, \varepsilon_i) \in X \times R^{J+1}$,*

$$\sigma_i^*(x, \varepsilon_i) = \arg\max_{a_i \in A} \left\{ \; u_i^{\sigma^*}(a_i, x) + \varepsilon_i(a_i) \; \right\} \quad (4)$$

We can represent this BNE in probability space. Let $\sigma^*$ be a set of BNE strategies, and let $P^*$ be the choice probabilities associated with these strategies. By definition, $P_i^*(a_i|x) = \int I\{a_i = \sigma_i^*(x, \varepsilon_i)\} \; dG_i(\varepsilon_i)$. Solving the equilibrium condition (4) in this expression we get that for any $(a_i, i) \in A \times I$:

$$P_i^*(a_i|x) = \int I\left( a_i = \arg\max_{a \in A} \{u_i^*(a, x) + \varepsilon_i(a)\} \right) dG_i(\varepsilon_i) \quad (5)$$

Notice that the function $u_i^\sigma$ depends on players' strategies only through the choice probabilities $P^\sigma$ associated with $\sigma$. To emphasize this point, we modify the notation and use the symbol $u_i^P$, instead of $u_i^\sigma$, to denote these expected utility functions. Therefore, the right hand side in equation (5) is a function that we define as $\Lambda_i(a_i|x; P)$. We call the functions $\Lambda_i$ *best response probability functions.* The vector of equilibrium probabilities $P^*(x) \equiv \{P_i^*(a_i|x) : (a_i, i) \in A \times I\}$ is a fixed point of the best response mapping $\Lambda(x, P) \equiv \{\Lambda_i(a_i|x, P) : (a_i, i) \in A \times I\}$. Given Assumption 1, best response probability functions are continuous in the compact set of players' choice probabilities. By Brower's theorem, there exists at least one equilibrium. In general, the equilibrium is not unique.

The primitives of the model $\{u_i, G_i : i \in I\}$ can be described in terms of a vector of parameters $\theta \in \Theta \subseteq R^K$. Primitives are continuously differentiable in $\theta$. We use $\Lambda(x, \theta, P)$ to denote the equilibrium mapping associated with $(x, \theta)$. And $P(x, \theta) = \{P_i(a|x, \theta) : (a, i) \in A \times I\}$ represents an equilibrium associated with $(x, \theta)$ such that $P(x, \theta) = \Lambda(x, \theta, P(x, \theta))$. For some values of $(x, \theta)$ the model has multiple equilibria. Let $\{P^\tau(x, \theta) : \tau = 1, 2, ...\}$ be the set of equilibria associated with $(x, \theta)$. The equilibria are indexed by the variable $\tau \in \{1, 2, ...\}$ that is called the *equilibrium type.*

# 3. Estimation

## 3.1 Data generating process

Suppose that the game is played at different moments in time or at different locations or markets. We have a random sample of $T$ realizations of the game where we observe players' actions and common knowledge state variables $\{a_t, x_t : t = 1, 2, ..., T\}$ with $a_t = (a_{1t}, a_{2t}, ..., a_{Nt})$. Let $\theta^0 \in \Theta$ be the true value of $\theta$ in the population under study. We are interested in the estimation of $\theta^0$.

Let $\tau_t$ be the equilibrium type of observation $t$, that is unobservable to the researcher. And let $P_t^0(x_t) \equiv \{\Pr(a_{it} = a|x_t) : (a, i) \in A \times I\}$ be the distribution of $a_t$ conditional on $x_t$ in the population that generates observation $t$. Since $a_t$ comes from an equilibrium of the game, we have that $P_t^0(x_t) = P^{\tau_t}(x_t, \theta^0)$. The following assumption establishes some conditions on the data generating process that guarantee the identification of $\theta^0$.

*ASSUMPTION 2: (A) For every observation $t$ the equilibrium type $\tau_t$ is determined by a function $\tau^0(.) \in \Gamma$ of the common knowledge state variables, i.e., $\tau_t = \tau^0(x_t)$. And (B) there is a unique pair $(\theta^0, \tau^0) \in \Theta \times \Gamma$ such that $P^0(x) = P^{\tau^0(x)}(x, \theta^0)$ for every $x \in X$.*

Under Assumption 2(A), two games with the same exogenous factors should have the same equilibrium probabilities. The function $\tau^0$ is called the *equilibrium selection mechanism.* Under this assumption we can identify nonparametrically the equilibrium probabilities in the population under study. Assumption 2(B) establishes the joint identification of the structural parameters and the equilibrium selection mechanism.

### 3.2 Maximum likelihood estimation

The maximum likelihood estimator (MLE) of $(\theta^0, \tau^0)$ maximizes the likelihood with respect to $\theta$ and with respect to the equilibrium types in the sample. Thus, the MLE is:

$$\hat{\theta} = \arg\max_{\theta\in\Theta} \left\{ \sup_{\tau_1,...,\tau_T} \sum_{t=1}^{T} \sum_{i=1}^{N} \log P_i^{\tau_t}(a_{it}|x_t;\theta) \right\} \quad (6)$$

subject to the restriction that $\tau_t = \tau_{t'}$ when $x_t = x_{t'}$. This estimator is consistent and asymptotically efficient. However, its implementation can be computationally very costly. The problem is in the maximization with respect to the equilibrium types. First, we need to know all the equilibrium types that the model has for every trial value of $\theta$ and for each sample value of $x$. This is impractical in many applications. And second, the number of of possible values of $\{\tau_1, ..., \tau_T\}$ is typically huge. For instance, if the number of equilibrium types is 3 and the number of observations is 200, we have $3^{200} \simeq 10^{95}$ possible values for the vector $\{\tau_1, ..., \tau_T\}$. It is clear that optimization with respect to the equilibrium types can be extremely costly. This problem has motivated the development of alternative methods, like pseudo maximum likelihood (PML) estimation, that avoid the search for the MLE of $\tau^0$.

### 3.3 Pseudo likelihood estimation

The MLE in (6) can be also described as:

$$\hat{\theta} = \arg\max_{\theta\in\Theta} \left\{ \begin{array}{ll} \sup\limits_{P_1,...,P_T} & Q(\theta, P_1, ..., P_T) \\ s.t: & P_t = \Lambda(x_t, \theta, P_t) \text{ for each } t \\ & P_t = P_{t'} \text{ for } x_t = x_{t'} \end{array} \right\} \quad (7)$$

where $Q(\theta, P_1, ..., P_T)$ is the *pseudo likelihood function* $\sum_{t=1}^{T}\sum_{i=1}^{N} \log \Lambda_i(a_{it}|x_t, \theta, P_t)$, that is defined for arbitrary choice probabilities, not necessarily in equilibrium. The two-step PML estimator is the value of $\theta$ that maximizes the pseudo likelihood function $Q\left(\theta, \hat{P}_1, ..., \hat{P}_T\right)$, where $\{\hat{P}_1, ..., \hat{P}_T\}$ are nonparametric estimates of players' choice probabilities conditional on $x$. Notice that the nonparametric estimates of choice probabilities can be interpreted as estimates of the equilibrium selection mechanism $\tau^0$. Therefore, this estimator avoids the search for the MLE of $\tau^0$ by estimating it nonparametrically.

The nested pseudo likelihood (NPL) estimator is a recursive extension of the two-step PML. Given the initial nonparametric estimates of choice probabilities, the NPL generates a sequence of estimators $\{\hat{\theta}^K : K \geq 1\}$ where the $K-$*stage* estimator is defined as:

$$\hat{\theta}^K = \arg\max_{\theta\in\Theta} Q\left(\theta, \hat{P}_1^K, ..., \hat{P}_T^K\right) \quad (8)$$

and the probabilities $\{\hat{P}_1^K, ..., \hat{P}_T^K : K \geq 2\}$ are obtained recursively as: $\hat{P}_t^{K+1} = \Lambda(x_t, \hat{\theta}^K, \hat{P}_t^K)$. The algorithm iterates until convergence. For any given sample, Brower's fixed-point theorem guarantees the existence of at least one NPL fixed-point. Aguirregabiria and Mira (2003) show that NPL is consistent and more efficient than the two-step PML. Monte Carlo experiments show that NPL has much better finite sample properties than the two-step estimator. However, NPL is not the MLE and therefore it is not efficient. Notice that the MLE should be a NPL fixed-point, but not every NPL is a MLE.

## 4. Genetic algorithm

A possible procedure to obtain the MLE is a *parallel NPL* method. That is, we can obtain $M$ fixed points of the NPL procedure by applying this method with $M$ different initial values for the choice probabilities. These $M$ initial values can be obtained by drawing $M$ bootstrap samples from the original sample and then calculating a nonparametric estimator of the probabilities for each bootstrap sample. Given the $M$ fixed-points of the NPL, the estimator of $\theta^0$ will be the fixed point with the highest value of the pseudo likelihood $Q(.)$. A limitation of this approach is that for some application we may need a very large number of initial values, $M$, to guarantee that this estimator is the MLE. To deal with this problem we combine the *parallel NPL* method with a GA method. In other words, at each step of the NPL, we perform three operations on the $M$ vectors of probabilities: crossover, mutation and selection. These operations, which are characteristic of GAs, make the NPL searches not independent and allow for a much more global search over the space of $(\theta, P_1, ..., P_T)$. Here we describe this algorithm.

*(0) Initial population.* The initial "population" of probability vectors is $\Pi^1 = \{\hat{P}_{mt}^1 : t = 1, .., T; m = 1, ..., M\}$, where $M$ is the size of the population. This initial population may be arbitrarily chosen, or it may come from a nonparametric estimates. For instance, the $M$ probabilities could be obtained as $M$ bootstrap nonparametric estimates of players' choice probabilities.

The GA generates a sequence of populations of size $M$ that we denote by $\{\Pi^K : K \geq 1\}$. Associated with this sequence of probabilities the GA also generates a sequence of sets of parameter estimates

$\{\hat{\theta}_m^K : m = 1, ..., M;\ K \geq 1\}$. An iteration of the algorithm consists in the creation of a new generation with the offsprings of the existing generation. An iteration can be described in terms of four processes or steps that are followed sequentially: (1) mating or selection of parents; (2) crossover and mutation; (3) NPL iteration; (4) selection of offsprings.

*(1) Selection of parents.* We draw, with replacement, $O > M$ pairs of probability vectors from the population $\Pi^K$. The probability that a vector is chosen depends on its relative fitness. Fitness is a term from evolution theory. In our problem, the fitness of a probability vector is the Lagrangian function:

$$\begin{aligned} l(\hat{P}_1, ..., \hat{P}_T) &= Q\left(\hat{\theta}, \hat{P}_1, ..., \hat{P}_T\right) \\ &- \lambda \sum_{t=1}^{T} \left\| \hat{P}_t - \Lambda(x_t, \hat{\theta}, \hat{P}_t) \right\| \end{aligned} \tag{9}$$

where $\hat{\theta}$ is the PML estimate associated with $\left(\hat{P}_1, ..., \hat{P}_T\right)$, and $\lambda$ is a small and positive constant. Given the measures of fitness of the $M$ elements of $\Pi^K$, the probability that the $m - th$ element is selected is:

$$S_m^K = \frac{\exp\left\{\sigma^{-1}\ l(\hat{P}_{m1}^K, ..., \hat{P}_{mT}^K)\right\}}{\sum_{j=1}^{M} \exp\left\{\sigma^{-1}\ l(\hat{P}_{j1}^K, ..., \hat{P}_{jT}^K)\right\}} \tag{10}$$

where $\sigma > 0$ is a parameter that measures the strength of the dependence of selection on fitness. If $\sigma = 0$, only the fittest individual is selected in the $O$ "random" draws. If $\sigma = \infty$, every individual has the same probability of being selected.

*(2) Crossover and mutation.* Each couple generates one offspring. An offspring inherits "chromosomes" from its parents, but there maybe mutation as well. We represent this with two sets of binary variables: $\{z_t, d_t : t = 1, 2, ..., T\}$. $z_t$ is the indicator of a mutation for chromosome $t$ and it is i.i.d. over $t$ with $\Pr(z_t = 1) = \gamma$, i.e., the mutation probability. $d_t$ is the indicator for the identity of the parent who transmits the $t - th$ chromosome and it is i.i.d. over $t$ with $\Pr(d_t = 1) = 1/2$. Let $\{\hat{P}_{m1}, ..., \hat{P}_{mT}\}$ and $\{\hat{P}_{m'1}, ..., \hat{P}_{m'T}\}$ be a couple. Then, the offspring from this couple is $\{\hat{P}'_1, ..., \hat{P}'_T\}$ where for any $t$:

$$\begin{aligned} \hat{P}'_t &= d_t \left\{\hat{P}_{mt} + z_t\ \delta\ (\hat{P}_{mt} - U_t)\right\} \\ &+ (1 - d_t) \left\{\hat{P}_{m't} + z_t\ \delta\ (\hat{P}_{m't} - U_t)\right\} \end{aligned} \tag{11}$$

where $U_t$ is a vector of $N$ independent random draws from a $U(0, 1)$; and $\delta$ is a parameter that represents the magnitude of the mutation.

*(3) NPL iteration.* For each offspring we obtain its associated PML estimator of $\theta^0$, i.e., the value of $\theta$ that maximizes the pseudo likelihood $Q\left(\theta, \hat{P}'_1, ..., \hat{P}'_T\right)$. Then, for each offspring $\{\hat{P}'_1, ..., \hat{P}'_T\}$ and its PML estimator $\hat{\theta}$, we obtain a new offspring $\{\hat{P}''_1, ..., \hat{P}''_T\}$ such that $\hat{P}''_t = \Lambda(x_t, \hat{\theta}, \hat{P}'_t)$.

*(4) Selection of offsprings.* We calculate the fitness of each new $O$ offsprings and select the $M$ ones with highest fitness. This is the new population $\Pi^{K+1}$.

The algorithm iterates until convergence of the sequence of populations $\{\Pi^K\}$.

# REFERENCES


[1] Aguirregabiria, V., and P. Mira (2003): "Sequential Estimation of Dynamic Discrete Games," manuscript, Department of Economics, Boston University.

[2] Beenstock M. and G. Szpirob (2002): "Specification Search in Nonlinear Time-Series Models Using the Genetic Algorithm," *Journal of Economic Dynamics and Control,* 26, 811-835

[3] Dorsey, R. E. and W. J. Mayer (1995): "Genetic Algorithms for Estimation Problems with Multiple Optima, Nondifferentiability, and Other Irregular Features," *Journal of Business and Economic Statistics,* 13, 53-66.

[4] Holland, J. H. (1975): "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor.

[5] Mitchell, M. (1996): "An Introduction to Genetic Algorithms," MIT Press. Cambridge.

[6] Mitchell, M., J. H. Holland and S. Forrest (1994): "When Will a Genetic Algorithm Outperform Hill Climbing?" In J. D. Dowan, G. Tesauro, and J. Alspector, eds., *Advances in Neural Information Processing Systems 6*, Morgan Kauffman.

[7] Pakes, A., M. Ostrowsky, and S. Berry (2003): "Simple Estimators for the Parameters of Discrete Dynamic Games (with Entry/Exit Examples)," manuscript, Harvard University.

[8] Pesendorfer, M. and Schmidt-Dengler (2003): "Identification and Estimation of Dynamic Games," London School of Economics Working Paper.