OPTIMIZATION USING GENETIC ALGORITHMS: AN APPLICATION TO THE REAL BUISINESS CYCLE MODEL

Christian Andrew Johnson

DOCUMENTOS DE TRABAJO DEL BANCO CENTRAL

OPTIMIZATION USING GENETIC ALGORITHMS: AN APPLICATION TO THE REAL BUISINESS CYCLE MODEL

Christian Andrew Johnson
Head of the International Desk
Department of the
Central Bank of Chile

Resumen

Este artículo utiliza algoritmos genéticos (GAs) para encontrar los valores de los parámetros en la solución del modelo de Ciclos Reales. Para generar las funciones de política de los agentes, aproximamos las expectativas condicionales de la ecuación de Euler usando una función polinomial exponencial, basado en el método de Marcet (1991). La ambigüedad en la selección de los valores iniciales para el algoritmo propuesto permite la aplicación de GAs para mejorar las simulaciones macroeconómicas.

Abstract

This paper uses genetic algorithms (GAs) to find the optimal parameter values in the solution of the Real Business Cycle model. To generate the policy functions of the individual, we approximate the conditional expectation of the Euler equation using an exponential polynomial function, based on the method proposed by Marcet (1991). The ambiguity in the selection of the starting values for the proposed algorithm allows the application of the GAs methodology to improve the macroeconomic simulations.

The Article was prepared while the author was a full time professor at the Department of Economics, University of Santiago de Chile. It shall be presented at the Latin American Meeting of the Econometric Society (August 1996). Also presented at the International Symposium of Forecasting, and the International Symposium of Artificial Intelligence, both at Istanbul, Turkey (June 1996), and the IEEE International Symposium on Intelligent Control, Dearborn, Michigan USA (September 1996).

1. Introduction

This research is based on the implementation of genetic algorithms to find the optimal parameter structure which is going to be included in the solution of the agent-individual maximization problem, also known as Real Business Cycle model.

Since the pioneer work of Kydland and Prescott (1982) in which they calibrate and simulate a macroeconomic microfundated model, the introduction of computers in the solution of economic models became an obligation. Dynamic programming was included into macroeconomics and, as a result, the necessity to simulate these models with random generated variables was obvious. The outcome, a **policy function** which represents the reaction of the agent mapping endogenous and exogenous state variables into decision variables. So, based on a set of simulated random numbers (usually obtained from a normal probability density function), we can generate the path for all the control and state variables of the model.

However, the first order condition or Euler equation (the conditional expectation) needs to be approximated. Several methods have been used to solve nonlinear rational expectations models with endogenous state variables (Taylor and Uhlig (1990)). Among these, the method of parameterized expectations is particularly suited to be used with a simulation estimate because it produces simulated observations that satisfy the Euler conditions without having the necessity to solve the policy function. This method approximates the conditional expectation using an exponential polynomial, which considers as input a vector of parameters previously obtained from nonlinear least squared methods.

The paper will be organized in three sections. First, we will introduce the nonlinear rational expectation model to be solved using the parameterized expectations approach (Den Hann and Marcet (1990), (1994), and Marcet (1991)). In the next section, we will present the algorithm in order to solve and simulate the model. Finally, the last section will cover the main results and conclusions of the implementation.

2. *The Real Business Cycle Model*

This section presents the basic model, also known as benchmark economy. The structure consists of a representative agent, which is suppose to be rational, whose decisions are made in a world without distortions (i.e., taxes and money). The agent maximizes the present value of a stream of utility levels represented by an utility function, discounted by a deterministic and constant discount factor. Hence, the maximization problem can be represented by the following stochastic growth model:

$$\underset{\{c,k'\}}{\text{Max}}\ E_0 \sum_{t=0}^{\infty} \beta^t \frac{c^{1-\tau}}{1-\tau}$$

$$\text{s.a.} \quad c_t + k_{t+1} - (1-\delta)\cdot k_t = \theta_t k_t^{\alpha}$$

$$\log\theta_t = \rho\log\theta_{t-1} + \varepsilon_t$$

where consumption and capital stock are represented respectively by c and k. The utility function has the property of being concave with a parameter τ known as relative risk aversion parameter. This particular utility function is called Constant Relative Risk Averse (CRRA) utility function. The discount factor is β, a constant between 0 and 1. The first equation represents the present value of the utility streams, while the second equation represents the feasibility constraint: total output must be consumed or invested. The data generation process for the logarithm of the technology input (θ) evolves according to a first order autoregressive process (AR(1)) in which ρ is the autoregressive coefficient, and where the last term on the right hand side of the third equation represents the productivity shock, drawn from a normal density function, with zero mean and finite-homoscedastic variance.

The dynamic programming problem for the representative agent can be represented by the following value function, also known as the Stochastic Bellman Equation:

$$v(k,\theta) = \underset{[c,k']}{\text{Max}}\left\{ u(c) + \beta \int v(k',\theta')f(\theta',\theta)d\theta' \right\}$$
$$c + k' - (1-\delta)\cdot k = \theta k^{\alpha}$$
$$\log\theta' = \rho\log\theta + \varepsilon'$$
$$\varepsilon \xrightarrow[\text{approx}]{} \text{iid } N\left(0,\sigma_{\varepsilon}^{2}\right)$$

The Euler equation for the capital stock (once we included the market clearing condition in to the value function $v(k,\theta)$) represents the conditional expectation that needs to be approximated. The first order condition for k is:

$$c_t^{-\tau} = \beta E\left[c_{t+1}^{-\tau}\left(\alpha k_{t+1}^{\alpha-1}\theta_{t+1} + (1-\delta)\right)\middle| I_t\right]$$

As we can see, the conditional expectation of the right hand side of the Euler equation is a function of the endogenous and exogenous state variables (k,θ). The method proposed by Den Hann and Marcet (1990) and Marcet (1991) in order to solve this equation for the endogenous variable, consists in parameterize the conditional expectation by any class of dense functions[2], such as polynomials or neural nets. The particular form used to approximate the conditional expectation is:

$$E(\bullet) = \exp\left[\text{poly}(k,\theta)^{\lambda}\right]$$

[2]A function has a dense trajectory if given any point p on the dynamic system, any initial condition q, and any $\varepsilon > 0$, there is some time t finite such that the trajectory starting at q passes within a distance ε of p. It means that each trajectory comes arbitrarily close to any given point on the dynamic system. This is not to say that the trajectory passes *through* each point; it just comes arbitrarily close. (Strogatz (1994)).

where poly(k,θ) is a polynomial of order λ (usually first or second order), whose arguments are the state variables (k,θ). In practice, the first order polynomial we use consists of linear terms of the elements of the polynomial.

3. The Algorithm

The problem to be solved consists in generate an approximation of the Euler equation such as the mean square error of the following function be minimized:

$$c_{t+1}^{-\tau}\left(\alpha k_{t+1}^{\alpha-1}\theta_{t+1}+(1-\delta)\right)=\delta_1\cdot e^{\left[\delta_2\cdot\log(k_{t-1})+\delta_3\cdot\log(\theta_t)\right]}+\xi_{t+1} \qquad \textbf{(1)}$$

This equation indicates that the choice of the parameter structure is crucial to obtain good results in terms of the fitness function. Our problem is to determine the value of this parameter vector in such a way that we can obtain the best fitness value in our objective function.

The procedure proposed by Marcet consists in the estimation by successive approximations using an iterative updating of the parameter space, with a convergence speed near 1 and in which each vector of parameters in this scheme is calculated by nonlinear least squared methods.

The algorithm is presented in the following steps:

Step 1. Choose some fixed arbitrary vector of parameters ψ and simulate the time series c,k, using a fixed realization of productivity shocks.

Step 2. Estimate the parameters of the polynomial (1) by nonlinear least squared.

Step 3. Compare the estimates with the vector of parameters used to generate the series c,k. Based on a distance criteria, if tolerance is binding then go to step 4, else go to step 1 but now using the following learning procedure to make the updating of the parameter vector:

$$\psi_{j+1}=(1-\lambda)\cdot\psi_j+\lambda\cdot \mathrm{NLLS}(\psi_j)$$

where NLLS(ψ_j) represents the estimates previously obtained in step 2.

Step 4. Take the final estimates and generate the series c,k to draw inferences concerning actual data.

However, in practice, this method is very sensible to the hillclimbing technique used to find the optimal parameter vector. Starting points are decisive in these solutions. The method presented is efficient in finding the best parameter vector over the space that minimize the proposed mean square error (MSE) objective function. The idea is to generate a population of individuals where each one has its own economy and in which each one can estimate its own MSE. The strings will represent the value of the parameters, and the algorithm will search for the best vector, using the updating of the generation based on genetic algorithms.

We know that hillclimbing methods use the iterative improvement technique, i.e., we apply the method to a single point in the search space and, as a result, during a single iteration, a new point is selected from the neighborhood. The problem with this method is that usually provides local optimum values which are starting point dependent.

Genetic algorithm performs a multi-directional search by maintaining a population of potential solutions. "Good" solutions will tend to reproduce while the "Bad" ones will eventually die. To distinguish between good and bad solutions we use an objective or fitness function which plays the role of an environment. Among many others, we can mention the different studies carried out by Michalewicz (1992), Koza (1991), Koza (1993), Forrest (1993), Holland (1975), Goldberg (1989), and, in the field of Economics and Finance, we can name to Bauer (1994), and White (1991).

The method I propose is based on GA search and the algorithm follows the next steps:

Step 1. Define the parameter space to be used by the search, $\psi \in \Psi$, and the deep parameter values, such as the CRRA parameter, the depreciation rate, the discount factor, and the parameters in the normal PDF used to simulate the random productivity shocks.

Step 2. Define the parameters of the GA: size of the population, the size of the time series to be simulated, the number of generations, the probability of crossover, and the probability of mutation.

Step 3. Generate randomly the strings for each individual of the population in generation 1.

Step 4. Generate the control, and endogenous and exogenous state variables from the model. Evaluate the fitness function.

Step 5. Execute (if corresponds) the crossover and mutation.

Step 6. Repeat steps 4 and 5 until the population converges or until we have a number of generations equal to the max number of generation permitted.

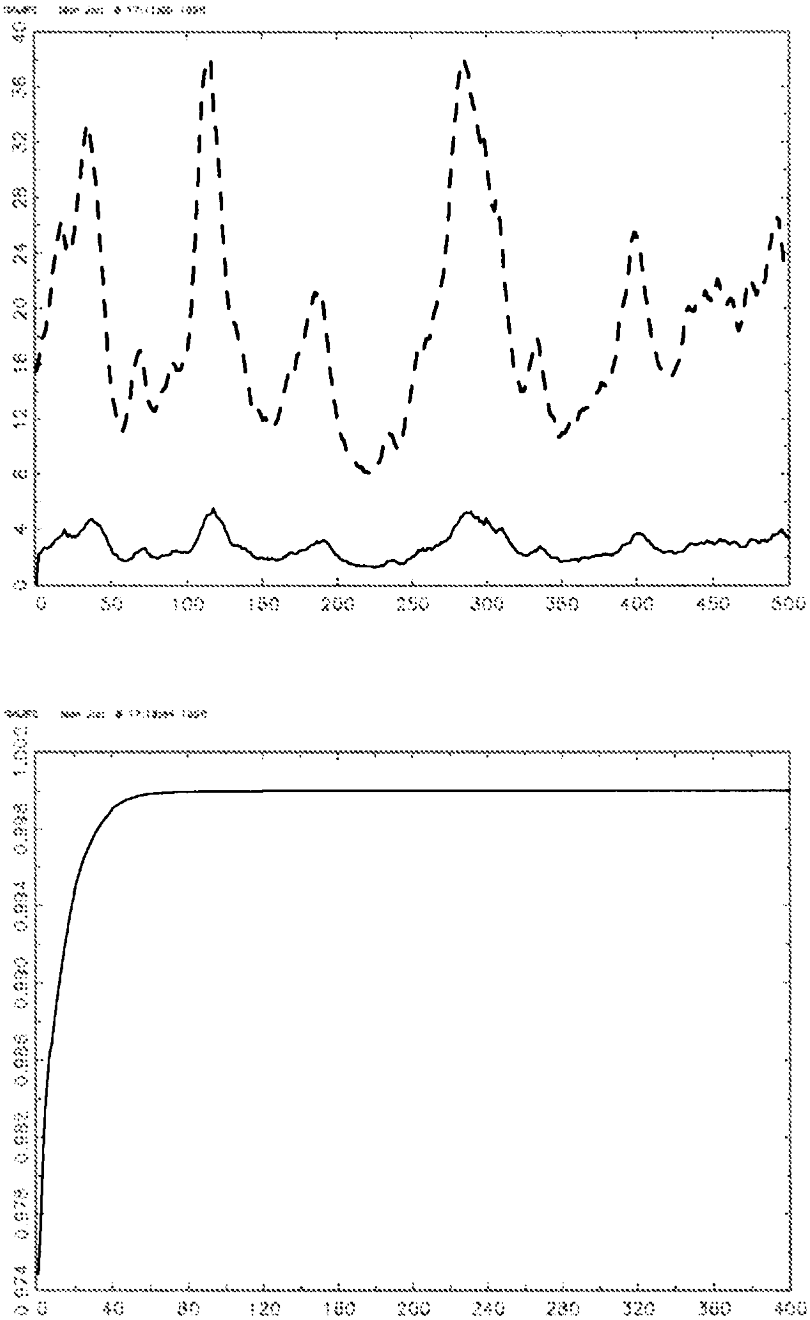
4. Results of the Simulations

The implementation was executed in a PC Pentium-90, with 8 RAM, and the parameters used are presented in the following table. The results gave us very positive estimates (we use a four digits accuracy level), whose values differ from those obtained using the nonlinear least square mechanism.

Summary of the GAs Simulations for the Real Business Cycle Model							
Pm=0.0033	Pc=0.5	Generation Size	Population Size	Time Series Simulated	Parameter 1	Parameter 2	Parameter 3
		400	100	500	2.7109	-0.5000	0.1132
		200	100	100	2.6679	-0.5000	-0.0619
		100	50	100	1.9903	-0.4054	-0.1235
NLLS (BFGS-SC)				1000	2.0469	-0.4091	-0.1161

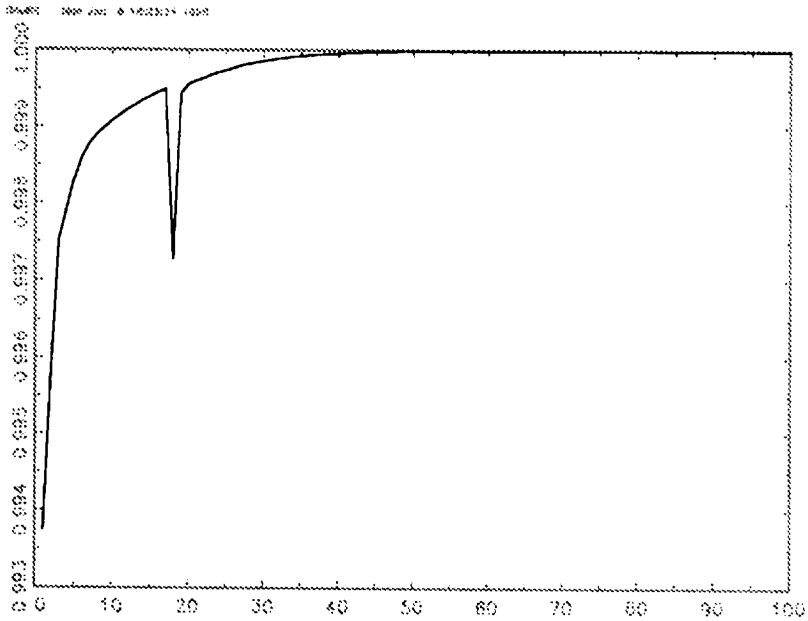
The table indicates part of the results in applying the GA multi-directional search. As we can see, the value of the objective function (fitness) is increasing when we move from search in small spaces to wide spaces. Also, the two simulated variables, population and time series, are the two most important determinants in the search procedure. With a generation and population size of 400 and 100, respectively, the convergence is reached around the 200th generation while in the case of the small example with 50 individuals the convergence is reached very quickly but at sub optimal parameter values. This last vector of parameters is very similar to the one obtained when using the nonlinear least squared method (with the BFGS-scaled procedure) proposed by Marcet (1991), which indicates a sub optimal solution to the Euler equation.

In the following pictures we report the results from these simulations, plotting the average maximum fitness as a percentage of the optimal value versus the generation number, and the time series for capital and consumption generated from an individual in the final generation.



0
4
8
12
16
20
24
28
32
36
40
0
50
100
150
200
250
300
350
400
450
500
0.974
0.978
0.982
0.986
0.990
0.994
0.998
1.000
0
40
80
120
160
200
240
280
320
360
400

Figure 1.



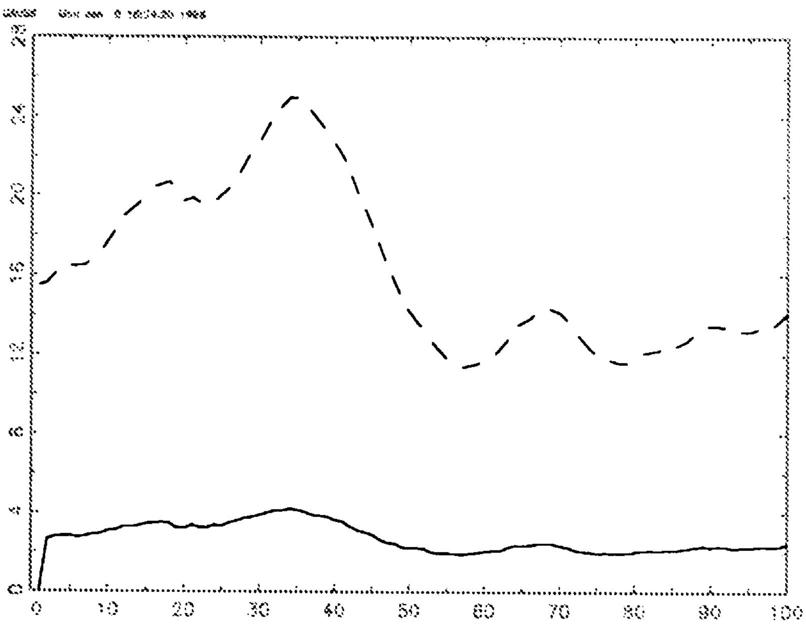


Figure 2.

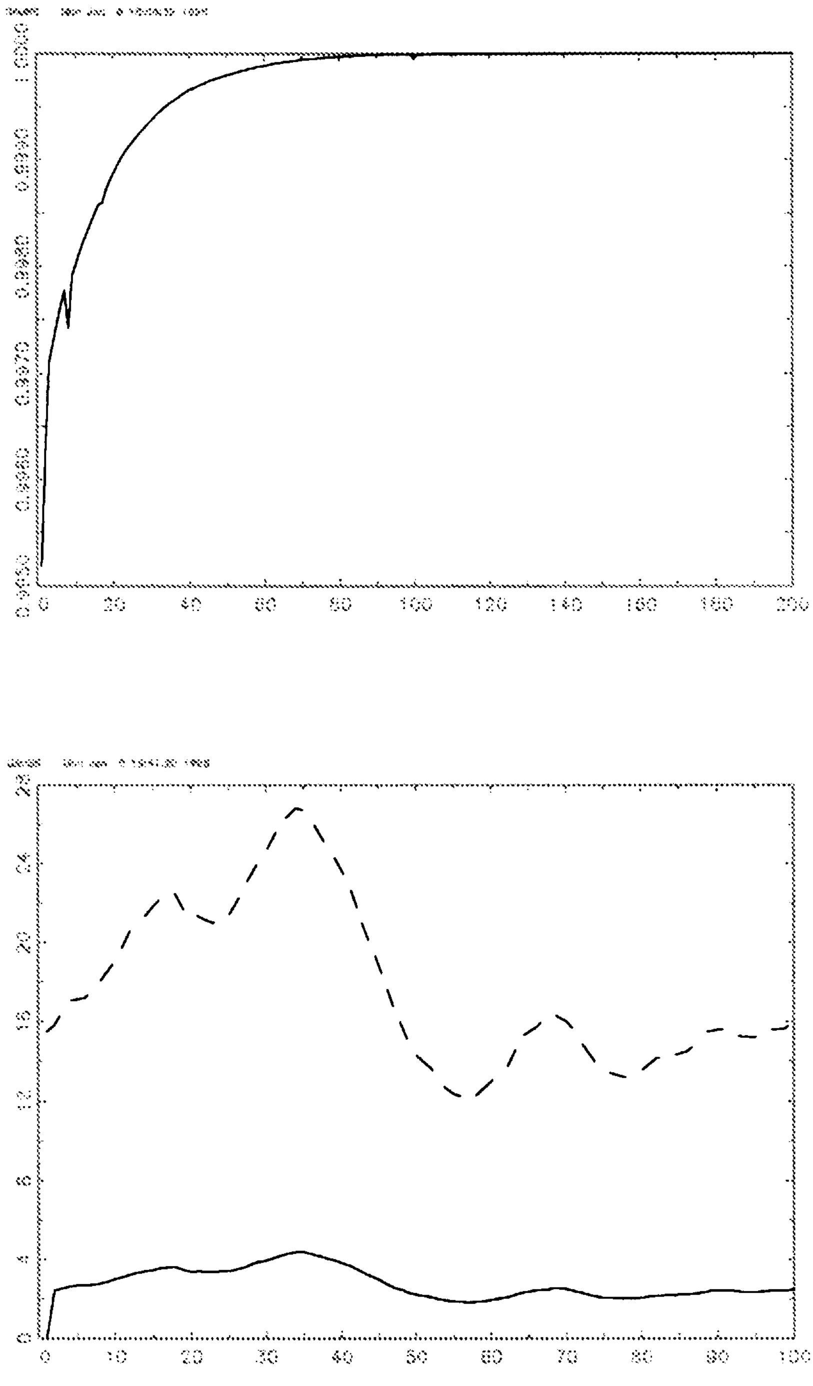


Figure 3.

5. Conclusions

The implementation of genetic algorithms in the solution of the parameterized expectation provides to be an excellent alternative to the nonlinear least squared method. It gave us better estimates of the parameter vector which enable us to obtain simulated accurate time series for the capital stock and consumption. This accuracy allows the evaluation of the economic model with respect to the actual data. Because, it is not common to obtain very different estimates depending on the starting point for polynomials of order greater than 1, this method has been proved to be very powerful in helping me to find the optimal vector of parameters. Now, the author is evaluating a GA for exponential polynomials of order two (six parameters) and three (ten parameters), models that have been problematic for other authors to estimate consistently.

REFERENCES

Bauer, Richard (1994). *Genetic Algorithms and Investment Strategies*, John Wiley & Sons.

Den Hann, W. and A. Marcet (1990). "Solving the Stochastic Growth Model by Parameterizing Expectations", *Journal of Business and Economic Statistics* 8 (31-34).

________________________ (1994). "Accuracy in Simulations", *Review of Economic Studies,* 61 (3-17).

Forrest, Stephanie (1993). *Genetic Algorithms: Proceedings of the Fifth International Conference,* editor.

Goldberd, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*, U. Of Michigan Press, Ann Harbor.

Koza, J.R. (1991). *Genetic Programming*, MIT Press.

Koza, J.R. (1993). *Genetic Programming II*, MIT Press.

Marcet, Albert (1991). "Solving Non-Linear Stochastic Models by Parameterizing Expectations: An Application to Asset Pricing with Production", mimeo, GSIA, Carnegie-Mellon University.

Michalewicz, Zbigniew (1992). *Genetic Algorithms + Data Structures = Evolution Program*, Springer-Verlag.

Strogatz, Steven H. (1994). *Nonlinear Dynamic and Chaos*, Adison-Wesley Publishing Company.

Taylor, J.B. and H. Uhlig (1990). "Solving Nonlinear Stochastic Growth Models: A Comparison of Alternative Solution Methods", *Journal of Business and Economic Statistics*, 8 (1-17).