Assistant Professor Rasul Bakhshi DASTJERDI
E-mail: rbakhshi@yazduni.ac.ir
Nahid Khaki NAJAFABADI, M.A. in Economics
Mohsen Dashti ARDAKANI, M.A in Mechanics
University of Yazd
Iran

SOLVING RAMSEY MODEL OF OPTIMAL GROWTH USING GENETIC ALGORITHM FOR UNDERGRADUATES

Abstract. *Ramsey Optimal Growth model is widely used in dynamic macroeconomics. Graduate and undergraduate students have problem in doing with Ramsey Model. In this paper we try to solve Ramsey model using Genetic Algorithm. Genetic Algorithm is a new optimization approach that is widely used in dynamic analysis. It seems that solving optimization problems with GA approach will produce better results than Excel solver.*
        Key words*: Ramsey Model, Genetic Algorithm, Excel, Optimal Control*

JEL Classification: A22; C61; D91; D92; Q00.

1. Introduction:

The Ramsey growth model is a neo-classical model of economic growth based primarily on the work of the economist and mathematician Frank P. Ramsey (1928). The Ramsey model differs from the Solow model in one crucial respect: it explicitly models the consumer side and endogenizes saving. As a result, unlike the Solow model, the saving rate in general is not constant and the convergence of the economy to its steady state is not uniform. Another implication of the endogenous saving rate is that the outcome in the Ramsey model is necessarily Pareto optimal or that the saving rate is at its golden rule level. Originally Ramsey set out the model as a central planner's problem of maximizing levels of consumption over successive generations. Only later was a model adopted by subsequent researchers as a description of a decentralized dynamic economy.

Ramsey growth model plays a central role in macroeconomics research. The Ramsey model adds optimization behavior to the Solow-Swan model of economic growth by modeling the consumption-saving decision of a forward looking consumer. Blanchard and Fischer (1989, ch. 2) present the model with continuous time, and Walsh (2003, ch. 2) uses discrete time (Weber, 2005, p. 15). We adopt discrete time because we are interested to compare our results with Weber. For this purpose we have also used Weber numerical example.

This paper has three parts. In the first part we review the Ramsey optimal growth in a discrete time version. In the second part we introduce the process of calibrating Ramsey model in GA. After all we compare our approach with Excel approach in the third part.

2. Ramsey Growth Model

According to Bakhshi & Dallali (2010) with Following Cass (1966) and Koopmans (1967), objective of the "representative agent" is to maximize utility over an infinite horizon. Let us suppose that the utility function at any time t is hedonistically defined as a positive function of consumption per capita at time t, U (c_t). Given an infinite horizon and discrete time, consumption will thus be infinite and discrete. The present value of future utility gains from individual consumption at any time period t is then:

$$\beta^t U(c_t)$$

Where, $\beta = \dfrac{1}{1+\theta}$ is discount factor (θ is rate of time preference) and $U(c_t)$ is utility function which we assume it has a CRRA (constant relative risk aversion) form as following:

$$U(c_t) = \frac{c_t^\gamma - 1}{1 - \gamma}$$

Where γ determines the curvature of the utility function. The objective function which needs to maximize, takes the following form:

$$MaxJ = \sum_0^T \beta^t U(c_t) \tag{1}$$

Where "J" is the total welfare and "now" is time 0. The objective of the representative agent is therefore to maximize this intertemporal stream of utility subject to economy-wide constraints. Therefore, function "J" is the present value of the continuous stream of utility-adding consumption from now to T (terminal date). This is our objective function that we try to maximize subject to a dynamic flow constraint.

The constraint is obvious enough: the more is consumed now, the less is saved and thus the less growth and consumption for future. Likewise, consuming less now means consuming more tomorrow. Therefore, higher time preference means to gives priorities to present consumption compared to future consumption. Thus, a

"balance" must be somehow struck between all the periods such that the total stream of utility is maximized - which means sacrificing some consumption in every period, but (because of time preference), not enough sacrifices are made. So we can write the constraint as follow:

$$S.t : k_t = (Ak_{t-1}^\alpha - c_t) + \frac{1-\delta}{1+n} k_{t-1} \tag{2}$$

Where, A and α are the technical parameters of Cobb-Douglas production function and k_t, δ and n are per capita capital stock, depreciation rate and population growth rate.

In steady state, saving must equal $\frac{n+\delta}{1+n} k^*$ to compensate for the negative effect of depreciation, δ, and population growth, n, on the amount of capital per worker. To see this, set $k_t k_{t-1} = k^*$ and solve for saving. Walsh (2003, ch. 2) also shows that the steady state amount of capital per worker is:

$$k^* = \left(\frac{A\alpha(1+n)}{(1+n)(1+\theta)-(1-\delta)} \right)^{\frac{1}{1-\alpha}} \tag{3}$$

The steady state output per worker, y^*, can be calculated by substituting k^* into the production function and the steady state consumption per worker, c^*, is output minus saving:

$$c^* = A(k^*)^\alpha - \frac{n+\delta}{1+n} k^* \tag{4}$$

Weber (2005) has solved this problem using Excel for a numerical example. In his example, all parameters have taken following magnitudes:

$$\theta = 0.04, \gamma = 1.5, A = 1, \alpha = 0.3, \delta = 0.05, n = 0.01$$

Solving the model with above parameters using Excel solver tool generates following results:

Weber' worksheet 6

	A	B	C	D	E
1		Initial	Shock		
2	θ =	0.04	**0.05**		
3	β =	0.9615	0.9524		
4	γ =	1.5	1.5		
5	A =	1	1		
6	α =	0.3	0.3		
7	δ =	0.05	0.05		
8	n =	0.01	0.01		
9	k*	4.8451	4.2250		
10					
11	Time	c(t)	k(t)	y(t)	PVU(t)
12	0		4.8451		
13	1	1.3770	4.7856	1.6054	0.2956
14	2	1.3697	4.7312	1.5995	0.2772
15	3	1.3626	4.6816	1.5940	0.2600
16	4	1.3561	4.6364	1.5890	0.2441
17	5	1.3502	4.5951	1.5844	0.2294
18	6	1.3448	4.5574	1.5801	0.2157
19	7	1.3399	4.5230	1.5762	0.2031
20	8	1.3355	4.4915	1.5726	0.1914
21	9	1.3315	4.4625	1.5693	0.1806
22	10	1.3279	4.4358	1.5663	0.1704
23	11	1.3247	4.4110	1.5635	0.1611
24	12	1.3219	4.3879	1.5608	0.1523
25	13	1.3195	4.3661	1.5584	0.1442
26	14	1.3174	4.3454	1.5561	0.1366
27	15	1.3157	4.3254	1.5538	0.1295
28	16	1.3143	4.3058	1.5517	0.1229
29	17	1.3133	4.2863	1.5496	0.1167
30	18	1.3126	4.2666	1.5475	0.1110
31	19	1.3122	4.2463	1.5453	0.1056
32	20	1.3121	4.2250	1.5431	0.1005
33					
34				PVLTU	3.5478

Worksheet 6 shows the optimal consumption path and optimal capital per capita accumulation. It also shows the maximum total utility attained equals 3.4578.

But we can solve above program using Genetic Algorithm. To see how to work with GA it is necessary to introduce GA step by step.

3. Genetic Algorithm

Basically, all random algorithms start to work based on sampling from responds space. Genetic algorithm as any other optimization algorithm begins by defining optimization variables and ends by testing convergence. This algorithm is a searching technique drawn from the nature applied to discovery in a search space. Although, setting this algorithm was in the field of computer science, nowadays it

is used in different fields such as economy, management, mechanic, electricity, industry and etc. The reason of usage is its simple calculation and search power to find the best respond.

Genetic algorithm is usable to any problem regardless of having any information about the problem, target function type, calculation procedure without any limitation on its type and the rate of variables and it has a fixed efficiency in finding a comprehensive optimal respond. In the other word, most of traditional optimization methods have got this main problem that when reaching the first local optimal point, they stopped and have no ability to exit this point and to move toward global optimal point (Kzarlis,1996, p.89). The continuous attempts for estimating the optimal answer was with more precision and less time resulted in finding creative methods. One of these methods is genetic algorithm (Syswerda, 1991, p. 135).

3-1. Genetic algorithm performing stages for determining optimal consumption trend:

3-1-1. Producing chromosome: In a genetic algorithm at the first stage, a population of chromosomes is produced in definite numbers. Chromosomes are generated from putting the parameters coded figure beside each other and are as a web of bits. This initial population is usually generated randomly. Each chromosome is a reprehensive of a search place sample called "individual". The collection of these "individuals" is called population or present generation. Our aim is to obtain optimal consumption trend in an economy with the conditions just like as Weber exercise. We assume population grows at a constant rate of 1 percent per tear during 20 years. Each of per capita consumptions (c_i) in these 20 years is as a gene and this twenty- fold chain of consumption forms a chromosome. It is necessary to state that after writing a program on *MATLAB* labeled voluntary as *PCGA200* (appendix I), by writing *"gatool"* on *MATLAB* environment, the window related to *GA* will be appeared (appendix II). Here, program name, variable numbers and ranges in which the optimal consumption rates is estimated in it (based on the solved problem with excel, variable range of c in range is (1, 2) will be introduced to genetic algorithm.

In the next step, the population size needs to be determined. It should be mentioned that the population size (number of chromosomes involving genes that are the same optimal consumption trends), has a frequent effect on function and efficiency of genetic algorithms.

In this problem, the population size was identified 10000 based on the trail and error. It means that 10000 chromosome chains (involving 20 c_i gene) in introduced range will obtain randomly. Table 1 shows the results of the trial and error related to the population size. Regarding that the target function rates (maximum ideality) at the population size of 100000 varied 0.025 and on the other hand saving the time, *GA* population size in this problem is considered 10000.

_____

Table 1: Trial and error to determine GA population size

P.Z	100	1000	10000	100000
Time	0:7.40	0:9.73	0:22.21	7:34.82
F.F=(1/ u)	0.0610 25	0.0539 45	0.0537 85	0.05371 21
U	16.386 72	18.537 35	18.592 48	18.6177 7

Source: Calculations
P.Z: Population Size
F.F: Fitness Function

3-1-2. Parents Selection Mechanism: The rate of encoded chromosome in the domain of making- decision variable provides the probability of determining the function or the objective value of persons is the member of population. This function is done by the target function defining the function characteristic of individual chromosome in the domain of problem.

Hence, basic target function is to select the pairs combined each other during reproduction phase. During reproduction phase, objective value belongs to any individual based on determined solid function rate by the target function. This rate applied to direct the selection process toward the more suitable individuals. Individuals with high objective value in proportion to the total population have more chance being selected and mated. Conversely, individuals with low objective value have less chance being selected. This phase is called "reproduction" and Roulette wheel is used based on Goldberg approach (Mitsuo & Cheng, 2000). Therefore, the best chromosomes need to be selected to do mating process. In the noted problem with respect to the target function, the chromosome having the highest ideality is the best one. Thus, based on the ideality presented function (u) in *PCGA200* program, the ideality resulted from any c_i in any chromosome chain is calculated and then the sum of these idealities will obtain. The sum of these obtained idealities of any chain is arranged descending (downward) and an order belonged to any one based on Rank function. Then, objective value percent of each one is identified and put into the roulette wheel. Obviously, a chain with more ideality sum will get more objective value percent and attain more surface of roulette wheel and has more chance to be selected.

It should be mentioned that a chromosome chain with the most ideality put aside as a crossover operator. At least, the wheel begins to rotate and is stopped accidently. When wheel lover locates in front of part, that chromosome is selected as the first parent and in re- rotating the second parent also selected. This trend will be repeated 9999 times that forms the same initial population (10000 chains) together with the elite member. In this way, the parents are selected to do mating.

3-1-3. Generating the child: After determining the objective value rate of population members, they may be selected by an analogous probability with their relative objective value and combined to generate the next generation. We use cross-over and mutation operators to generate children. The crossover operator rate is usually selected 0.7 to 0.8. It means that 70 percent of children are generated by two- point, multi point or uniform crossover and the rest generated by mutation operator. However the sum of generated children by genetic operators with elite member produces 10000 chains totally.

4. After recombination and mutation, if necessary, the chains of population individuals (genotypes) are decoded, target function evaluated the rate of objective value of any individual determined, individuals selected according to objective value rate for mating and the process continues to generate the next generation. In this method, it is expected that the mean rate of individuals function increases in the population after each generation and besides good individuals are kept and combined with each other, improper individuals omitted. In other word, the processes determining the rate of new chains ideality, determining the rate of objective value percent and all of the aforementioned steps are repeated. Each reduplication of algorithm is called a generation that algorithm stopping condition also is introduced according to the desired problem. The stopping condition may be the number of identified duplication, achieving a definite rate of objective value or lack of changing the respond after several sequence generation or ending after a definite time in which the best chain of the final generation is considered as output. We considered the generation number of 500 suitable here based on trial, error and analyze. Therefore in GA, at 500 times, 10000 strains are generated. IN other word, five million strains each strain contains 20 genes (c_i) are generated and provides the probably of selecting the best strain while excel software is only able to estimate one optimal trend for consuming during 20 years. Finally, after ending GA, optimal consumption path and maximum utility will be generated. The results are introduced in table 2. According to this table, maximum utility is 18.59248. This maximum is more than the maximum utility generated using Excel solver (Weber' Worksheet 6)

3-2. Sensitivity Analysis
We can check the reaction of model to any change in the parameters. The rate of time preference in the model and technological parameters has an important role in determining the optimal path. Changing in time preference will change capital formation and optimal consumption path and so maximum utility will be change. We also check the response of system to change technological parameters of model. For example, by 1 percent increasing in time preference, optimal consumption will be changed and thereby maximum utility will be 16.83482 (Table 3)

Table 2. GA Optimal Results

Time	Optimal Consumption	k	y	Utility
0		4.8451		
1	1.72222	4.44048	1.605428	1.72222
2	1.13111	4.609551	1.563972	1.087606
3	1.10187	4.81545	1.581603	1.018741
4	1.3231	4.808758	1.602474	1.176231
5	1.13186	4.993035	1.601806	0.967519
6	1.37169	4.944708	1.619979	1.127429
7	1.03678	5.229442	1.615259	0.819382
8	1.401	5.1604	1.642618	1.064645
9	1.6136	4.876324	1.636082	1.179042
10	1.29821	4.896956	1.608525	0.912105
11	1.1074	5.109211	1.610564	0.74812
12	1.24502	5.19187	1.631196	0.808741
13	1.44013	5.08238	1.639069	0.899501
14	1.40292	5.006158	1.628621	0.842557
15	1.26963	5.060388	1.621255	0.73318
16	1.62287	4.763405	1.626504	0.901122
17	1.49504	4.582649	1.597259	0.798214
18	1.01412	4.875121	1.578829	0.520622
19	1.3633	4.830616	1.608406	0.672963
20	1.2484	4.899235	1.603987	0.592544
Total				18.59248

Table 3. GA Optimal Results after 1 percent increase in time preference

Time	Optimal Consumption	k	y	Utility
0		4.8451		
1	1.42307	4.73963	1.605428	1.42307
2	1.1651	4.887831	1.594863	1.109619
3	1.1237	5.083427	1.609663	1.019229
4	1.27153	5.138634	1.628722	1.098395
5	1.04651	5.420867	1.634008	0.860966
6	1.6179	5.141365	1.66043	1.267667
7	1.19972	5.270487	1.634269	0.89525
8	1.22576	5.378104	1.646475	0.871125
9	1.66243	5.052672	1.65649	1.125198
10	1.07013	5.308143	1.62576	0.689815
11	1.09691	5.545893	1.649996	0.673408

12	1.4109	5.477362	1.671827	0.824924
13	1.25178	5.565797	1.665603	0.697038
14	1.01506	5.89372	1.673625	0.538308
15	1.18043	6.065785	1.702617	0.596197
16	1.39912	6.0237	1.717379	0.673001
17	1.48325	5.896402	1.713796	0.679494
18	1.15018	6.09879	1.702849	0.50182
19	1.63797	5.818692	1.720177	0.68061
20	1.54064	5.628472	1.696085	0.609684
Total				*16.83482*

As table 3 shows, the maximum utility is achieved at 16.83482. This level is higher than the level attained by Weber (2005, p. 21) using excel solver.

4. Conclusion

Ramsey optimal growth theory is widely used in theoretical and applied economics. Based on its importance and also its hardship to learn for undergraduates, economists try to facilitate learning it, using some techniques. In this paper we try to solve Ramsey problem using Genetic Algorithm. In comparison to excel solver technique, results show that genetic algorithm can reach the representative agent to higher maximum.

REFERENCES

[1]**Bakhshi Dastjerdi. R & Dallali Esfahani, R. (2011),** *Equity and Economic Growth, A Theoretical & Empirical Study: MENA Zone*. *Journal of Economic Modelling*, In Press and forthcoming, Vol. 28, pp.694-700

[2]**Blanchard, O.J. and S. Fischer (1989),** *Lectures on Macroeconomics*. Cambridge, MA: MIT Press;

[3]**Conrad, J.M. (1999),** *Resource Economics*. Cambridge: Cambridge University Press;

[4]**G. Syswerda, (1991),** *Schedule Optimization Using Algorithms*. In: Handbook of GA's L. Davis, Ed. New York: van Nostrand Reinhold, pp. 129-144;

[5]**Goldberg, D.E. (1989b),** *Sizing Population for Serial and Parallel Genetic Algorithms;* Proceedings of the Third International Conference on Genetic Algorithms. San Mateo, CA: Morgan Kaufman. PP.70-79;

[6]**Goldborg, D. E. (1989),** *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison Wesley;

[7] **Holland, J. (1975),** *Adaptation in Natural and Artificial Systems*; Ann Arbor, MI, university of Michigan Press;

[8]**Kazarlis,S.A., A.G. Bakirtzis and V. Petridis (1996),** *A Genetic Algorithm Solution to the Unit Commitment Problem;* IEEE, Trans. Power syst. Vol.1, pp. 83-92;

[9]**Nævdal, E. (2003),** *Solving Continuous-Time Optimal Control Problems with a Spreadsheet.* Journal of Economic Education 34, 99-122;

[10]**Ramsey, F.P. (1928),** *A Mathematical Theory of Saving*. Economic Journal, 38, 543-559.

11]**Syswerda, G. (1991),** *A Study of Reproduction in Generational and Steady- state Genetic Algorithms*. G. Rawlins, ed. Morgan- Kuufmuun. pp. 4- 101;

[12]**Walsh, C.E. (2003),** *Monetary Theory and Policy*. Cambridge, MA: MIT Press;

[13]**Weber, E. J. (2005),** *Optimal Control Theory for Undergraduates*. *Computers in Higher Education Economics Review*, Economics Network, University of Bristol, vol. 19(1), pages 4-15.

Appendix I:

```
Function y = PCGA200 (X)
Ct = X;

teta = 0.04;
beta = 1/(1+teta);
gama = 1.5;
A = 1;
alfa = 0.3;
delta = 0.05;
n = 0.01;
t = 0:1:20;


ks = ((A*alfa*(1+n))/((1+n)*(1+teta)-(1-delta)))^(1/(1-alfa));

for i=1:20
   u(i) = beta^t(i)*(((ct(i)^(1-gama))-1)/(1-gama));
   if i==1
      kt(i) = (A*ks^alfa - ct(i))+((1-delta)/(1+n))*ks;
```

```
   else
      kt(i) = (A*kt(i-1)^alfa - ct(i))+((1-delta)/(1+n))*kt(i-1);
   end
end

tag = 1;
if kt(20) < ks
   tag = 0;
end
for i = 1:20
   if kt(i) < 0
      tag = 0;
   end
end
for i = 1:20
   if ct(i) < 0
      tag = 0;
   end
end

s = sum(u);

if tag == 0
   y = inf;
else
   y = 1/s;
end
```

Appendix II: