PAKISTAN INSTITUTE OF ENGINEERING AND APPLIED SCIENCES

Department of Computer and Information Sciences (DCIS)

Dated: 03-08-2017

Assignment: Artificial Intelligence

Name: Muhammad Dawood

Registration Number: 30251

Assignment Title: Genetic Algorithm Applications

Identify at least 3 novel (meaning -- new, not existing on the net!) real-world (meaning, practical!) problems and how to solve them using Genetic Algorithms. Specifically, for each of the problems -- do the following:

- ***Describe the problem***
- ***Describe its objective function / fitness function***
- ***Formulate the objective or fitness function mathematically or algorithmically***
- ***Describe your chromosome representation and your reasoning behind choosing it***
- ***Describe your cross-over and mutation operators and the reasoning behind them***
- ***Describe your selection strategy and the reasoning behind it***
- ***What Python or Matlab packages can you use to implement this?***

Application # 1: Eigenvalue Calculation for Checking Building Strength

Problem Description:

Eigenvalue tell you about some of the important characteristic of a system. Consider system in which a vehicle is travelling on bridge. The movement vehicle start traveling, two thing happen one is the bridge start with some oscillation and second is the decay time of the oscillation. Both of these is represented by eigenvalue of the system.

Figure 1 show another example of the eigenvalue. Consider a ruler that is standing up. At start the ruler is capable to accept certain value of load and it stay steady. When the load increases after a certain limit the ruler start bending, this load is called critical load and this load can be calculated using eigenvalue.

Google use eigenvalue to rank a page based on backlinks.

Let discuss an example of second order system. The equation for this system is:

$$k + \Lambda D + \mathrm{M}\Lambda^2$$

In the system above we have three matrices.

K matrix represent the stiffness of the structure

D matrix represent the damping

M matrix represent the mass of that specific structure.

Since we have three matrices we have to determine specific value for each matrix so that the resultant system is stable.

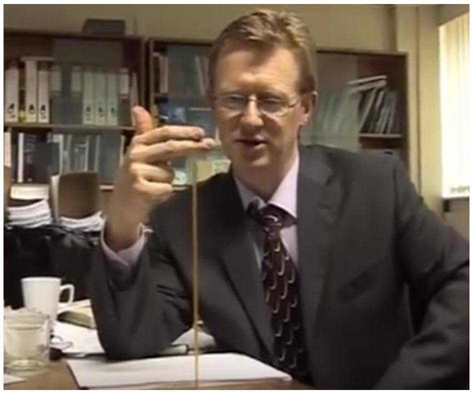

Figure 1

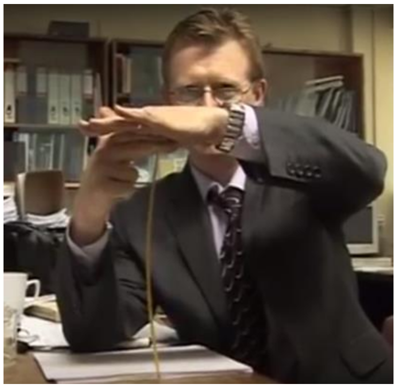

Figure 2

Fitness Function:

We have to select a specific value for each of these matrices and then calculate the eigenvalue. After calculating eigenvalue we have to check wither the resultant system is stable or not.

If the real part of each eigenvalue is strictly negative, the system is asymptotically stable. If some eigenvalues have negative real part but one or more of them has zero real part, the system is marginally stable but not asymptotically stable. If any eigenvalue has positive real part, the system is unstable.

Mathematical formulation of fitness function:

If all eigenvalues $\Lambda 1, \Lambda 2 \ldots \ldots .. \Lambda n$ are negative then system is stable.

If $\Lambda 1, \Lambda 2 \ldots . \Lambda x/2$ are zero and $\Lambda x+1, \Lambda x+2 \ldots \ldots \Lambda n$ are negative x<=n. then system is marginally stable and we have to improve the stability further.

If all eigenvalues $\Lambda 1, \Lambda 2 \ldots \ldots .. \Lambda n$ are positive then system is unstable.

Chromosome representation:

The encoding technique I have to use is binary encoding, every chromosome is representing some value. Let suppose the range of stiffness(S) is 0 <=S<=240 and that of Damping (D) is 0 <= D <=240 and Mass (M) range from (0, 240). We are using eight bit representation for each value as show below.

Stiffness	Damping	Mass
01010101	11011011	11110011

Cross-over and mutation operators

Since we have tree parameter, we need three point crossover as shown below.

01111101 01011011 00111010 (parent A)

01111001 00111001 00111001 (parent B)

01111001 01011001 00111001   (child A)

01111101 00111011 00111010 (Child B)

We need this type of crossover so that the result population is diverse and since each 8bit represent different entity so we have to crossover only inside that entity.

In mutation we are crossing over a random bit in child strings to increment or decrement the value of a parameters.

Selection strategy:

The selection strategy I am using is roulette wheel selection. Since we have to select one optimal value as a result the algorithm will converge fast, also there are less chances of too much difference in chromosome so diversity will be maintained.

What Python or Matlab packages can you use to implement this?

The python package I prefer is geneticalgs, because it has built in support of roulette selection and also support is available for multipoint crossover which is the demand of this problem to solve using current approach.

Application # 2: organic solar cell of polyaniline and zinc oxide:

Problem Description:

Monomolecular-layer polyaniline (PANI) was dispersed on the surface of zinc oxide (ZnO) and formed the hybrid effect between ZnO and PANI. Among conducting plastics polyaniline stands out due to its outstanding properties. It is one of the so-called doped polymers, in which conductivity results from a process of partial oxidation or reduction. Polyaniline compounds can be designed to achieve the required conductivity for a given application. To convert polyaniline into proper sheet zinc oxide is added to convert this into crystal form. We need specific ratio of polyaniline and ZnO to achieve high conductivity. Increasing ZnO quantity too much the Polyaniline sheet melting point is increased too much but the conductivity is reduced. We use genetic algorithm to find specific quantity of both these to achieve high conductivity as well as high melting point.

Fitness Function:

Check the individual if it has high melting point as well as high conductivity give it highest probability or score and select the individual.

Mathematical formulation of fitness function:

If conductivity > specific threshold which is required (depends on problem) && melting point > specific threshold value (depend on environment where to install the system)

Chromosome representation:

The encoding technique I have to use is binary encoding, every chromosome is representing some value. Let suppose the range of Conductivity(C) is 0 <=C<=24 and that of melting point (M) is 25 (degree centigrade) <= M <=100 (degree centigrade). We are using 6 bit representation for each value as show below.

Melting Point	Conductivity
010101	110110

Since we have two parameter, we need three point crossover as shown below.

011101 010011 (parent A)

011001 001001 (parent B)

011001 010001 (child A)

011101 001011 (Child B)

We need this type of crossover so that the result population is diverse and since each 6bit represent different entity so we have to crossover only inside that entity.

In mutation we are crossing over a random bit in child strings so that we can increase or decrease value of both ZnO and polyaniline to reach to an optimal value.

Selection strategy:

The selection strategy I am using is roulette wheel selection. Since we have to select one optimal value as a result the algorithm will converge fast, also we only have two parameters so diversity will be maintained easily.

What Python or Matlab packages can you use to implement this?

The python package I prefer is geneticalgs, because it has built in support of roulette selection and also support is available for multipoint crossover which is the demand of this problem to solve using current approach.

Application # 3: Design microprocessor in such a way so that the processor heading factor is minimum

Problem Description:

Increasingly, heat looms as the single largest obstacle to computing's continued advancement[1]. The problem is fundamental: the smaller and more densely packed circuits become, the hotter they get. To achieve that ever more difficult goal, engineers are exploring new ways of cooling by pumping liquid coolants directly on to chips, for example, rather than circulating air around them. In a more radical vein, researchers are also seeking to reduce heat flux by exploring ways to package the circuitry. Instead of being confined to two-dimensional (2D) slabs, for example, circuits might be arrayed in 3D grids and networks inspired by the architecture of the brain, which manages to carry out massive computations without any special cooling gear. In this section we are trying to evaluate different design choices so that the heat factor is reduced sufficiently using genetic algorithm.

Fitness Function:

Evaluate each design choice if the heating factor is small enough and occupy small space, also it is easy to design using current technology then it is best Fit individual.

Chromosome representation:

The encoding technique which I think to be suitable, is binary encoding. Every chromosome is representing a design choice. In processor design there may be a lot of factor to consider for simplicity we consider only three, Heat produced (range (0<=H<=40 degree centigrade), Space required to install processor (On chip area 1cm2<=Ac <= 10 cm^2), Energy consumption (10 joule <=E<= 120 joule). I am using 6 bit binary representation for each of these. There is specific relation between Heat produced, on chip area and Energy Consumption, each of them is defined in term of others. Table below show the representation of these three parameters.

Energy Consumption	On chip area	Heat produced
010101	011011	111011

Cross-over and mutation operators

Since we have tree parameter, but they are all dependent quantity changing one value the other will be effected, so in this case the random mutation will result in more efficient result as compare to Cross over, if we are using cross-over at a time we only do single point cross-over i.e only in one parameter.

Consider we have two individual A, B which are parent they generated two child using cross-over as shown below.

011101 010011 001110 (parent A)

011001 001001 001001 (parent B)

011001 010011 001110 (child A)

011101 001001 001001 (Child B)

In mutation a random bit is flipped in any of the three parameter but not in all only one parameter can be change at a time.

Selection strategy:

The selection strategy I am using is rank based selection because we want to get diverse design choice and not give too much priority to best one.

What Python or Matlab packages can you use to implement this?

The python package I prefer is geneticalgs, because it has built in support of roulette selection and also support is available for single point crossover & mutation, which is the demand of this problem to solve using current approach.