CSC 680: Artificial Intelligence
Fall 2012
Assignment #5 (60 Points)
Due date: Monday 11/26/2012 (in class)

Objective: To gain programming experience using Evolutionary Algorithms

Machine Learning with Genetic Algorithms (GA)

In this assignment you will implement a complete GA algorithm to perform engineering design optimization for an I-beam as shown below:

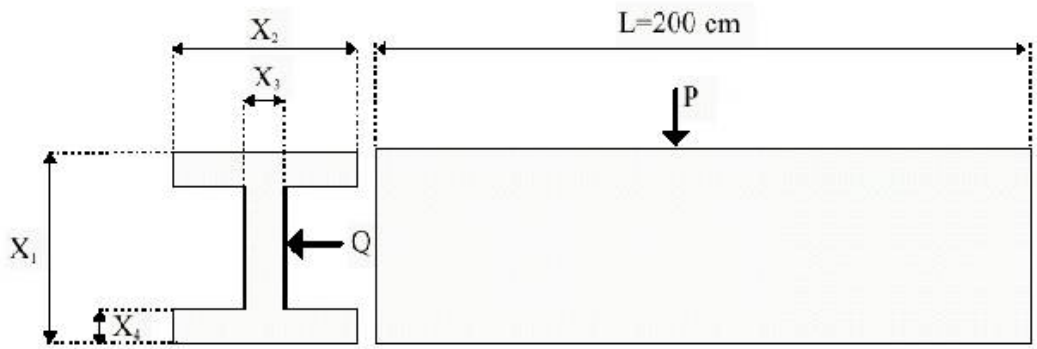


Figure 2. The frontal and side views of the I-beam

The general optimization goal for the I-Beam problem is to **minimize** $F=[f_1,f_2]^T$ where the cross section area (f_1) and the static deflection (f_2) of the I-beam, which are defined as:

$$f_1(X) = 2x_2x_4 + x_3(x_1 - 2x_4)$$

$$f_2(X) = \frac{60{,}000}{x_3(x_1 - 2x_4)^3 + 2x_2x_4\left[4x_4^2 + 3x_1(x_1 - 2x_4)\right]}$$

Subject to the bending stress constraint:

$$f_3(X) = \frac{180{,}000x_1}{x_3(x_1 - 2x_4)^3 + 2x_2x_4\left[4x_4^2 + 3x_1(x_1 - 2x_4)\right]} + \frac{15{,}000x_2}{(x_1 - 2x_4)x_3^3 + 2x_4x_2^3} \le 16$$

The geometric side constraints are:

$$10 \le x_1 \le 80, \quad 10 \le x_2 \le 50, \quad 0.9 \le x_3 \le 5.0, \quad 0.9 \le x_4 \le 5.0$$

Programming Guidelines

- Given the boundaries of the feasible design region as shown above, a linear search of the problem space can determine which extrema in the objectives space are simultaneously attainable.
- The table shown below depicts the computed ranges of responses for the two objective functions and the bending stress constraint. It is evident that f_1 is in conflict with f_2 and that the ideal solution $F^* = [25.38, 0.0059]^T$, where the two objectives are simultaneously minimized, can never be attained.

x_1	x_2	x_3	x_4	f_1	f_2	f_3
10	10	0.9	0.9	25.38_{min}	12.04_{max}	444.31_{max}
80	50	5	5	850.00_{max}	0.0059_{min}	2.01_{min}

- The fix for the above problem is to use the concept of Pareto optimality, but you are not expected to perform multi-objective optimization in this assignment. However, do test your GASolver on both f_1 and f_2 to make sure they are both optimized correctly in isolation from each other.
- Use the following population parameters:
    - numGenerations=100, populationSize=50, P_c=0.75, P_m=0.001

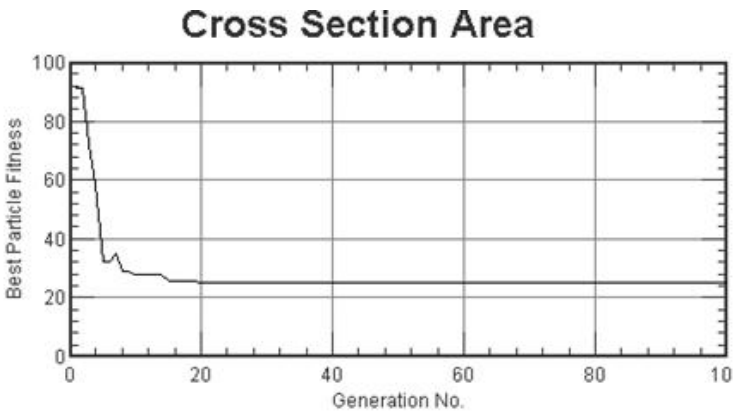
However, feel free to experiment with the above parameters and vary them to observe their effect on the algorithm's convergence.

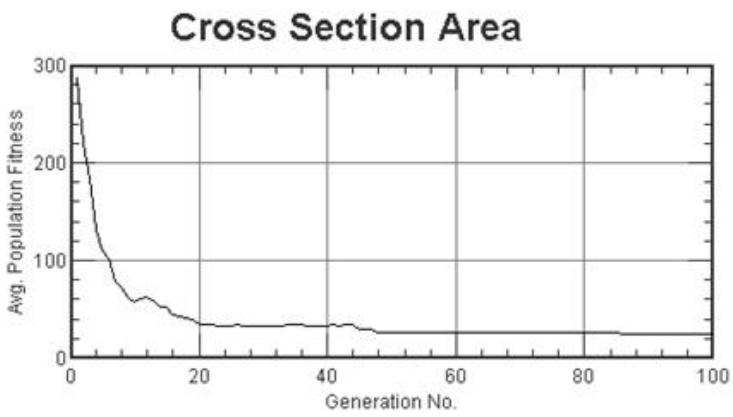
Input/Output Guidelines

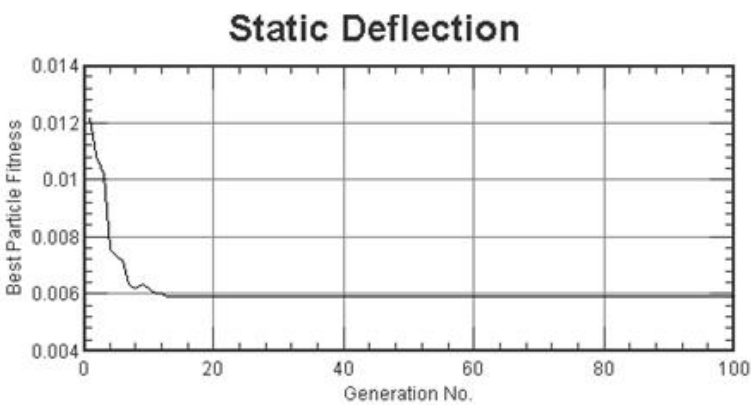
- Your program is to be called **GASolver.java** and compiled/run from the command line as follows:

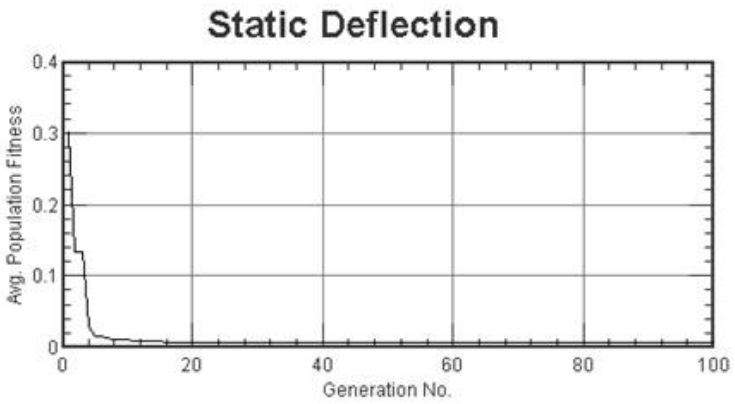
```
javac GASolver.java
java GASolver
```

- Your program must output two types of graph for each of the two objective functions – the best fitness of each particle in each generation and the average population fitness:

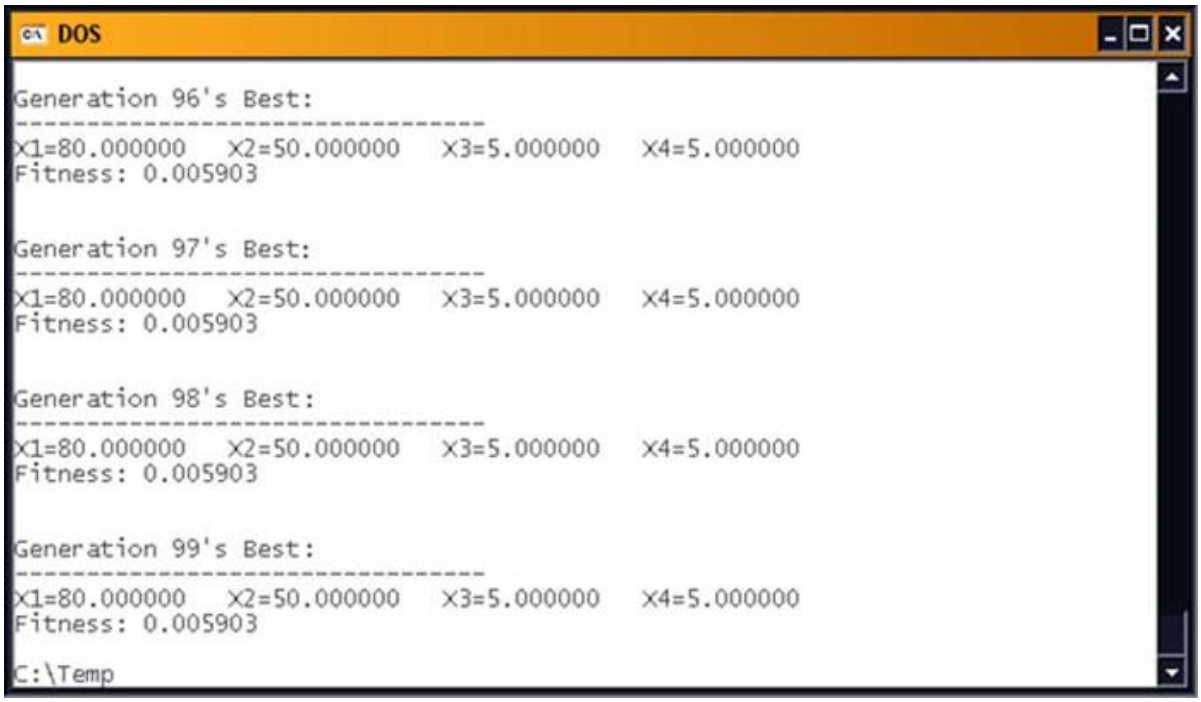








- In addition to the two graphs, have your program display in the console area the best obtained solution for each generation using the format:



- For plotting your results, use the following freely available PlotPackage API which can very easily be set up as shown below:

```
import jahuwaldt.plot.*;

import java.awt.*;

import javax.swing.*;


//***   Create a Simple 2D XY plot window.

double[] xArr = new double[numGenerations];

double[] yArr = new double[numGenerations];


for (int i=0; i<numGenerations; i++){
```

```
    xArr[i] = i+1;
    yArr[i] = avgPopulationFitness[i];   // your GASolver should fill up this array
}

Plot2D aPlot = new SimplePlotXY(xArr, yArr, "Static Deflection", "Generation No.", "Avg. Population Fitness", null, null, null);

//        Make the horizontal axis a log axis.
PlotAxis xAxis = aPlot.getHorizontalAxis();
xAxis.setScale(new LinearAxisScale());

PlotPanel panel = new PlotPanel(aPlot);
panel.setBackground( Color.white );

PlotWindow window = new PlotWindow("SimplePlotXY Plot Window", panel);
window.setSize(500, 300);
window.setLocation(250,250);   // location on screen
window.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
window.show();
```

Grading Guidelines

- I do not accept multiple submissions
- Read my policy on late submissions

Technical Correctness • **Program compiles and executes correctly** • **Proper error checking is performed on input file, if applicable, and during execution** • **Assignment objectives are met** • **Specified I/O guidelines are followed**	**90%**
Coding Style & Documentation • **Opening comments (author name, assignment number, date,**	**10%**

purpose, etc.) are included • General comments appear in code • Code is properly indented and spaced • Meaningful variable names are used • Code is properly modularized	

Submission Guidelines

1. Place your program files (**GASolver.java**) in a directory called **YourName-HW5** and archive it using Winzip or any other compression utility that you might have available on your PC. Include any special instructions to run your code in a **README** file.
2. Email me the zipped archive on or before the due date using the following subject line for your email message:
   Your Name – AI – HW5

   In addition, submit a **printout of all your program files** in class on the due date.