

**Assignment 4: PSO and EP**  
**(Due Date: August 4, 2020- midnight)**

**Question 3 PSO and Application to Function Optimization:** The six hump camelback problem is given as:

$$z = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2$$

where  $x$  and  $y$  lie between  $\pm 5$ . The objective is to minimize  $z$ . The global minimum lies at  $(-0.089840, 0.712659)$  or  $(0.089840, -0.712659)$  where  $z = -1.0316285$ . The global optimum to the problem is given for reference purpose. Do not use it in your solution methodology.

Part I [25 pts] Code a simple PSO to solve the problem. To do this you need to encode the problem, initialize a population, select a velocity update equation, and select a stopping criterion. Run your code and report: final solution, plot the progress of the average fitness and the best particle fitness.

Part II [15 pts]

- [5 pts] Use the Inertia Weight version of velocity update equation with Global best. Run your code and report: final solution, and plot the progress of the average fitness of the population and the global best particle fitness.
- Use Constriction Factor version of velocity update equation with Global Best. Run your code and report: final solution, and plot the progress of the average fitness of the population and the global best particle fitness.
- Use the Guaranteed Convergence PSO (GVPSO). Run your code and report: final solution, and plot the progress of the average fitness of the population and the global best particle fitness.

Part III [10 pts]

Write a report [max 2 pages, 700 words, 12pt font size] in which you describe: (1) the choices you made to set the parameters of the populations, (2) Your insights and observations on the performance of each implementation, (3) Your observations on the comparative performance of the different implementations.

**Question 2: Genetic Programming:** Implement a genetic programming algorithm and use it to solve the “6-multiplexer” problem. In this problem there are six Boolean-valued terminals,  $\{a_0, a_1, d_0, d_1, d_2, d_3\}$ , and four functions, AND, OR, NOT, IF. The first three functions are the usual logical operators, taking two, two, and one argument respectively, and the IF function takes three arguments. (IF X Y Z) evaluates its first argument X. If X is true, the second argument Y is evaluated; otherwise the third argument Z is evaluated. The problem is to find a program that will return the value of the  $d$  terminal that is addressed by the two  $a$  terminals. E.g., if  $a_0=0$  and  $a_1=1$ , the address is 01 and the answer is the value of  $d_1$ . Likewise, if  $a_0=1$  and

a1=1, the address is 11 and the answer is the value of d3. The fitness of a program should be the fraction of correct answers over all 64 possible fitness cases (i.e., values of the six terminals).

When “optimizing” you may wish to use only partial “test” sets however, when reporting performance you should report the performance on the full test set.

Part I [30/50 pts]: Use genetic programming to discover the solution to the 6-multiplexer problem.

Part II [15/50 pts]: Use genetic programming to discover a solution to the 11-multiplexer (3 address, 8 data lines) problem. You may find it computationally expensive to use the full test set on this problem (2048 cases), thus you may find it necessary to develop a technique which only use a fraction (changing) of the test cases each generation.

Part III [5/50 pts]: Use genetic programming to discover a solution to the 16-middle-3 problem. The correct logic for 16-middle-3 given input  $x \in \{0, 1\}^{16}$  then

$$\text{16-middle-3}(x) := \begin{cases} 0 & \text{not}(7 \leq \sum_{i=1}^{16} x_i \leq 9) \\ 1 & 7 \leq \sum_{i=1}^{16} x_i \leq 9 \end{cases}$$

Thus e.g.,  $\text{16-middle-3}(1000100100111001) = 1$  and  $\text{16-middle-3}(11111111111110111) = 0$ . Speed issues as above but now are exacerbated with 65536 test cases.

For each part: report your parameter choices, and a plot of the progress of the best program fitness in each iteration, and the fitness of the finalist program. Report the tree of the finalist program in each part.