COMP6776 **Winter 2018**

Assignment 4

Published: 2018-03-01 Due: 2018-03-16

Implementation of a tree-based genetic programming algorithm.

1. Input: use the A4_traingSamples.txt as your input file.

2. Representation: binary trees.

3. Population initialization: ramped half-and-half with the maximum depth (tree depth is defined as the maximum number of edges from a tree's root to its leaves) being 5.

4. Fitness function is the root-mean-squared error.

5. Terminal set: {x, y, 1, 2, 3, 4, 5}; Function set: {+, -, *, ÷ (protected)}.

5. Parent selection: tournament selection with replacement

6. Crossover: sub-tree swap with the rate 1

7. Mutation: random sub-tree replacement with the rate 0.2 (mutant sub-trees are also subject to the initial tree depth limit).

8. Population management: Steady-state

For each generation, the two winners of the parent tournament selection will go through crossover with rate 1 and then mutation with rate 0.2. Then the two offspring will replace the worst two individuals in the tournament.

9. Termination: when the maximum number of generations is reached.

10. Command line execution: There are three input arguments, population size N, parent tournament selection size S, and the maximum number of generations G.

11. Output: To the standard output, for every 100 generations, the best, average, and the worst fitness, and the tree presentation of the best individual in the final generation.

12. Report: Use the parameter setting: N= 100, S=6, G=10000, run your algorithm 10 times and report the final best fitness of each run. For the best run, draw a fitness progression figure (best, average, worst fitness as a function of every 100 generations) and a figure of the average tree depth of the population as a function of every 100 generations. Show the tree representation of the final best individual. Figure out what phenotype it represents.