Department of Electrical and Computer Engineering

ECE 457A Adaptive and Cooperative Algorithms

Instructor: Dr. Otman A. Basir

Assignment 4
Due Date: December 1, 2024 11:59PM

Instructions

- This assignment can be approached in groups. Each group consists of upto three students.
- You are expected to do all problems of the assignment even if you are not doing it in a group.
- You should upload your answers as a PDF file on learn before 11:59pm of the deadline date
- Attach any codes used for the assignment separately as a compressed file to the same dropbox
- You can use any programming language (Matlab and Python are preferred)
- Submitted work with more than 30% similarity will not be marked
- Communicate any issues or concerns with the TAs.
- Deliverables: provided for each question
- The three problems are equally weighted

Problem 1

PSO and Application to Function Optimization
NetLogo has a model that implements a simple PSO in 2-D searching space [Models Library → Computer Science → Particle Swarm Optimization]. Refer to the work from Trelea 1 to change the model's parameters.

Deliverables

a (5 pts)Run experiments with different populations $(30, 80)$, speed limits (2 and 6), particle's inertia $(0.60, 0.729)$, personal-best $(1.7, 1.494)$ and the global-best factor the same as personal factor. Examine the PSO algorithm's characteristics (speed of converge and ability to find global optima), report your observations.

b (5 pts)What is the difference between the motion formulation of the given NetLogo implementation and the classical PSO? Explain the difference by referring to the code.

Problem 2

The six hump camelback problem is given as: $z = \left(4 - 2.1x^2 + \frac{x^4}{3}\right) x^2 + xy + \left(-4 + 4y^2\right) y^2$ where x and y lie between $+/-5$. 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.

[1]

Deliverables

a (20 pts): Code a simple PSO to solve the problem.

i To do this you need to encode the problem,

ii initialize a population, select a velocity update equation, and

iii select a stopping criterion.

iv Run your code and report: final solution, plot the progress of the average fitness and the best particle fitness.

b (15 pts):

i (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.

ii (5 pts)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.

iii (5 pts) 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.

c (10 pts): Write a report [max 2 pages, 700 words, 12pt font size] in which you describe:

i (3 pts) the choices you made to set the parameters of the populations,

ii (3 pts) Your insights and observations on the perforamce of each implementation,

iii (4 pts) Your observations on the comparative performance of the different implementations.

Problem 3

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 $a_1 = 1$, the address is

[1] [Trelea2003] I. C. Trelea. "The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection". Information Processing Letter, vol. 85, no. 6, pp. 317-325, 2003.

11 and the answer is the value of d_3. 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.

Deliverables

a (25/50 pts): Use genetic programming to discover the solution to the 6-multiplexer problem.

b (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.

c (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

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

Thus, as an example, 16-middle-3(1000100100111001) = 1 and 16-middle-3(111111111111110111) = 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.