IEE/CSE 598: *Bio-Inspired AI and Optimization*

Mini-Project 1: Implementing a Simple GA with a Base-10 Genotype (30 points)

Description. *This first mini-project is a small challenge meant to help solidify your understanding of how to implement a simple GA; it also provides an example of the kind of deliverable that is expected in future mini-projects. This assignment follows from class discussions of Chapter 4 from Bozorg-Haddad et al. (2017) and Chapter 14 from Passino (2005), both of which are available for download on the Canvas site under "Unit 1" in "Modules."*

You may discuss approaches for solving this problem with other students in your class. However, ultimately, this activity is to be completed ***individually*** *without the aid of other students in the class.* ***Please upload your submission to Canvas by the due date.*** *Any submissions after the due date will be considered late and may be penalized accordingly.*

Submit **a *single* DOC, DOCX, or PDF file** with solutions to the following questions. **Include any code** you used **in an appendix** in your document, and **upload any source code** as additional files in your submission.

The **next *two* questions** refer to the following optimization problem. Suppose that you are given the function

$$f(x) = x\sin(10\pi x) + 1$$

and you are asked to:

$$\text{maximize } f(x) \quad \text{subject to} \quad x \in [-0.5, 1]$$

Question 1 (20 points): Implement a simple genetic algorithm that encodes the decision variable x as a string of a sign variable and 4 base-10 digits (e.g., -0.1234 is (-1,1,2,3,4)). You are free to choose whatever GA parameters would like (e.g., crossover operator, mutation operator, population size, number of parents, crossover probability, mutation probability), but your algorithm must have ***both*** crossover and mutation.

- Describe your GA-implementation choices (5 points)
- Generate two plots:
  - Best, worst, and average fitness for each successive generation of the GA (5 points)
  - Best individual for each successive generation generation of the GA (5 points)
- Plot the function f and evaluate the quality of the solution found by the GA (5 points)

Question 2 (10 points): Instead of encoding the decision variable as a string of base-10 digits, use a genotype with **a single gene** representing the **continuous** value of the decision variable. Because there is only one gene, there is **no crossover,** and parents selected for crossover are simply copied without any recombination. In this case, the GA is effectively **only doing mutation.**

- Implement this mutation-only GA, and **compare the efficiency** (i.e., number of generations needed to find a solution) of your mutation-only GA to the base-10 discrete GA from Question 1. **For the same number of generations, is one algorithm favorable over the other for some reason? Justify your response.**

Question 3 (BONUS) (4 points): Assume that a base-10 genotype (x_1, x_2, x_3, x_4) represents the number $0.x_1x_2x_3x_4$. While analyzing successive population generations from the operation of a GA, you observe that once all members of the population are within 0.001 of each other, the recombination operator **very rarely** produces offspring that differ from the parent generation, and all new variation in the population is driven by mutation only.

- Explain why this occurs (2 points).
- Why might it be **useful** for the recombination operator to act this way? ***HINT:*** *Think about about "global" and "local" search as well as the* ***exploration–exploitation tradeoff*** *when forming your answer.* (2 points)

References

Omid Bozorg-Haddad, Mohammad Solgi, and Hugo A. Loáiciga. *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*. John Wiley & Sons, 2017. ISBN 978-1119386995.

Kevin M. Passino. *Biomimicry for Optimization, Control, and Automation*. Springer-Verlag, London, 2005. doi:10.1007/b138169.