Assignment 1

CS 776: Evolutionary Computing
Fall 2023
Max Score: 100

Black box function optimization

0.1 Hill climbing or gradient ascent

Write a simple hill climber which you may implement along the lines of the pseudocode from class.

Solve the following two black box problems using your hill climber. The black boxes are provided as precompiled .o files that implement an `eval` function. Both `eval` functions take a 100 length array of integers (ints) and return a double between 0 and 100 inclusive. Only the values 0 and 1 are allowed as values for members of the integer array. Solving the problem means setting each element of the array to either a 0 or a 1 such that the evaluation function returns 100.

1. First black box evaluation function.
2. Second black box evaluation function.

Here is a sample main.cpp that shows you how to call the `eval` function and what `eval` returns. I expect you to solve the first black box, but not the second.

0.2 Easy Function

Write a function `eval` (using the same arguments as the evals above) that is easy for your hill climber to optimize. Specify the optimum. Implement this function and make it publicly available. Run your hill climber algorithm with different starting points on this function and show that the algorithm finds the optimum every time.

0.3 Hard Function

Write a function `eval` (using the same arguments as the evals above) that is difficult or impossible for your hill climber to optimize. Specify the optimum. Implement this function and make it publicly available. Run your hill climber algorithm on this function with different starting points and show that your hillclimber never finds the optimum.

0.4 Questions

For each of the four problems above, answer the following questions. The three questions are related and may need to be answered for each problem differently.

1. What is time complexity of your hill climber in terms of the number of evaluations?
2. What is the reliability of your hill climber?
3. What is the quality of solutions found by your hill-climber? That is, how close does your hill climber get to optimum? What is your metric for measuring "closeness to optimum?"

Turning in your assignment

Turn in a document (pdf) with the following information (through canvas) before the due date.

1. Your full name and email address
2. Hill climber source, easy function source, optimum, hard function source, optimum
3. Transcripts of your program running on all four problems
4. Answers to the questions above

Talk to me if you have questions. I'll be glad to discuss ideas with you!