CS 776: Evolutionary Computing -Assignment 1 Fall 2018

Full Name: Batyr Charyyev
Email address: bcharyyev@nevada.unr.edu

Part-1.

For **eval.o** I used simple hillclimbing by just checking one digit per iteration and comparing result obtained with my old result. In lecture it was mentioned that it is just one hill, so by applying simple hill climbing I was able to get **100**. Strength of this algorithm is it is good for simple functions with one hill and it can find global optimum really quick. Weakness is, it is not good for complex functions with many local optimum because it may stuck on local optimum values.

How to run:

```
g++ -o solver main.cpp eval.o
./solver
```

For **eval1.o** first I tried to find pattern because in lecture it was mentioned to look for patterns. I observed that digits between **0-30** and **30-50** and **50-100** have similar behavior. Thus, I tried to split 100 digits into smaller partitions and consider each one of them as one dimension in my search space. I have two variables named **window** and **offset.** Window represents size of my partition and gap between two consecutive 1 digits in that partition, for example if window is 2 in that iteration first 20 digits will be considered as one partition and one 1 digit will be followed by 1 zero digit. Offset represents offset from which that partition starts.
Strength is it may find optimum values for functions following similar patterns. It may give good results for functions in which some block of digits show similar behaviors, however, it may give really bad results for other functions which doesnt follow this pattern.

How to run:

```
g++ -o solver main1.cpp eval1.o
./solver
```

Part-2.

You can use eval.cpp to generate another eval object which is harder to solve. Min and max values are 0 and 100 respectively.
It is hard because I generated random combination which does not follow any pattern thus, does not give any clue. Furthermore, I check 2 consecutive digits at once in each iteration. Thus, it is hard for my hillclimbers to solve it.

Command: *gcc -Wall -c eval.cpp.*