ASSIGNMENT 2

Developing a Basic Genetic Optimization Algorithm in C

NATHAN SINHA

1 MAKEFILE

```
1     CC = gcc
2     CFLAGS = -Wall -Wextra -std=c99
3     LIBS = -lm
4     SOURCES = GA.c OF.c functions.c
5     OBJECTS = $(SOURCES:.c=.o)
6     EXECUTION = GA
7
8     all: $(EXECUTION)
9
10    $(EXECUTION): $(OBJECTS)
11      $(CC) $(OBJECTS) -o $(EXECUTION) $(LIBS)
12
13    %.o: %.c
14      $(CC) $(CFLAGS) -c $< -o $@
15
16    clean:
17      rm -f $(OBJECTS) $(EXECUTION)
```

Line 1 sets the compiler to 'gcc'.
Line 2 picks the compiler flags. This will mark errors in the code.
Line 3 accounts for the math library.
Line 4 creates a list of all the files to compile with.
Line 5 makes .o files from the .c files.
Line 6 gives a name to the executable file.
Line 11 sets a template to compile the code with, similarly to how the code would be compiled without the Makefile.
Line 14 will compile the .c files to the respective .o files.
Line 17 will remove objects made via the compiler.

To run the makefile use 'make' in the terminal, to use the clean function use 'make clean'.

2 TABLES

Table 1: Results with Crossover Rate = 0.5 and Mutation Rate = 0.05

Pop Size	Max Gen	Best Solution			CPU time (Sec)
		x_1	x_2	Fitness	
10	100	-0.022005	-0.2722371	1.945565	0.000409
100	100	-0.045144	0.038342	0.258794	0.000804
1000	100	0.058413	-0.005530	0.255090	0.018999
10000	100	-0.035636	-0.019404	0.158102	0.096210
1000	1000	-0.007779	0.007002	0.032516	0.094375
1000	10000	0.001359	-0.003545	0.011121	0.877879
1000	100000	0.001559	-0.000773	0.005004	7.780251
1000	1000000	0.000489	0.000491	0.001973	83.814786

Table 2: Results with Crossover Rate = 0.5 and Mutation Rate = 0.2

Pop Size	Max Gen	Best Solution			CPU time (Sec)
		x_1	x_2	Fitness	
10	100	-0.209963	-0.013136	1.440489	0.000482
100	100	-0.021445	0.117224	0.675457	0.004225
1000	100	-0.003721	0.032372	0.120194	0.017529
10000	100	0.007886	0.004132	0.027291	0.113018
1000	1000	-0.006082	-0.000118	0.124004	0.124004
1000	10000	0.000284	0.002386	0.006951	0.907742
1000	100000	0.000154	-0.000192	0.000697	9.123633
1000	1000000	-0.000134	-0.000248	0.000800	100.425645

3 HOW TO COMPILE

To compile the code ensure functions.c, OF.c, GA.c, functions.h, and Makefile are in the same directory.

To compile the files, use the terminal and type 'make' this will compile the code.

To run the code after compiling use the following parameters:

```
./GA population_size max_generations crossover_rate mutate_rate stop_criteria
```

An example of a proper input is: ./GA 1000 10000 0.5 0.1 1e-16

NOTE: While compiling you might receive an error regarding M_PI and M_E. This is because they are undefined. To fix this set M_PI and M_E to the values of pi and e respectively.