Minimizing The Ackley Function

Developing a Basic Genetic Optimization Algorithm in C

Braeden Marchant

1 Introduction

In this project I will use a Genetic Algorithm to minimize the Ackley function. To do this, my goal is to minimize my fitness while using various functions and algorithms. The most important function is the crossover function, which emulates parents having children and creating new populations. I am also trying to optimize the time it takes to run this code and lower the CPU time.

2 Makefile

The Makefile compiles each all of the required files for the main function to run. The compiler used is set to gcc and the compilation flags include -Wall, -Wextra, and -lm. The -W flags are to enable warning messages, the -lm is to link the math library for the math constants in OF.c. There are several targets in my Makefile. The first is "all" which is what will run when I just run 'make' without specifying a target. It will build the executable 'GA' which depedns on three object files. GA.o, OF.o, and functions.o. These object files are responsible for compling the corresponding .c files. The clean target removes the GA executable and all the object files when running 'make clean'.

2.1 Results and Tables

I have ran my program using the parameters defined in the following 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	-2.629627	-4.586838	0.078207	0.000341
100	100	-4.601986	-4.860407	0.074511	0.000287
1000	100	4.548867	4.572635	0.069985	0.220027
10000	100	4.563190	4.593371	0.069943	20.719681
1000	1000	-4.602676	4.592629	0.069918	2.179273
1000	10000	-4.728659	-4.570522	0.070263	21.704112
1000	100000	-4.561901	-4.592675	0.069945	216.331819
1000	1000000	4.562983	5.480379	0.069926	1987.210923

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	-4.234242	-4.793568	0.074511	0.000287
100	100	-4.511490	-4.557875	0.070113	0.004479
1000	100	4.659980	4.600411	0.070002	0.354212
10000	100	4.598250	-4.601851	0.069917	33.816622
1000	1000	-4.614045	4.568012	0.069937	3.503184
1000	10000	4.618382	4.521774	0.070034	34.979671
1000	100000	-4.589894	4.587132	0.069921	259.063315
1000	1000000	-4.582750	-4.724086	0.069912	2460.183864