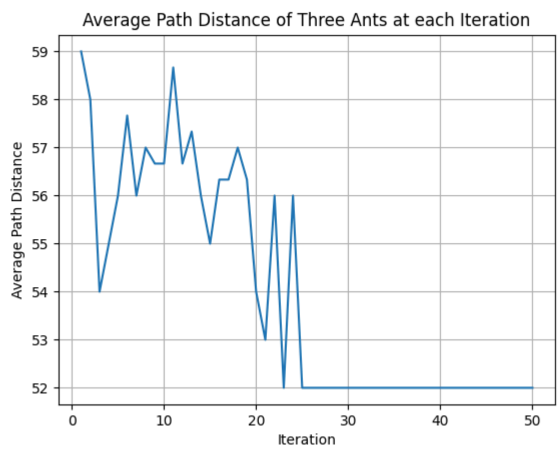
Noah Fabrick
AERO 470

Biomimetic Algorithms Assignment

Problem 1: Ant Colony Optimization
The file used is *ACO_Fabrick.py*
The optimal path found was some sequence in the order of 1 → 4 → 3 → 5 → 2 → 1 , with a value of 52.
Below is an example of how my ant colony optimization algorithm performs with each iteration.

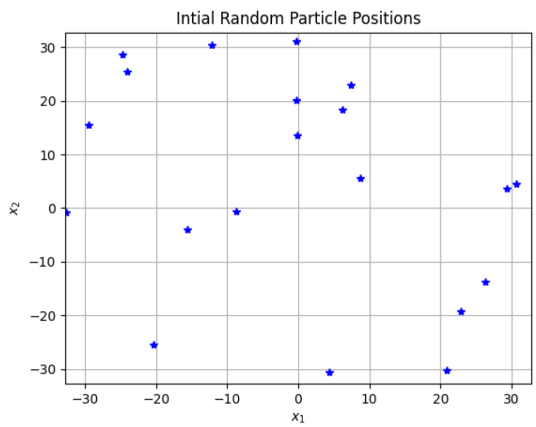


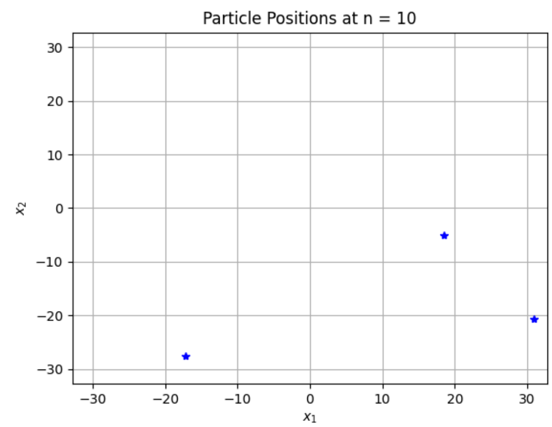
Problem 2: Particle Swarm Optimization
The files used are *PSO_Ackley.py* with module *PSO_Fabrick.py*
Given 20 particles and 50 iterations, my particle swarm optimization algorithm got the following results in ideal runs.

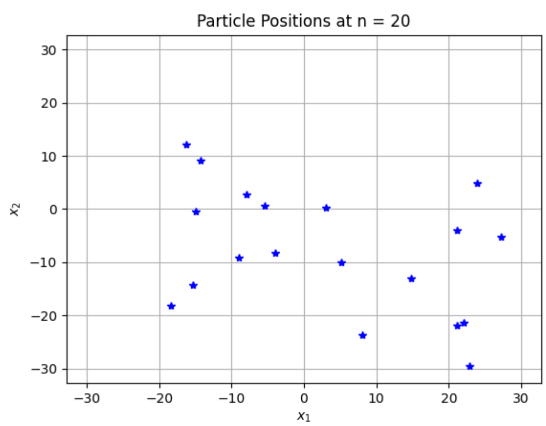
Number of Dimensions	Estimated Global Minimum of Ackley's Function
1	[0.04313609]
2	[-0.17135657, -0.30389297]
3	[0.42938549, 0.32163752, -0.01375126]
4	[0.38605053, -0.14263857, -0.05637054, 0.47237447]

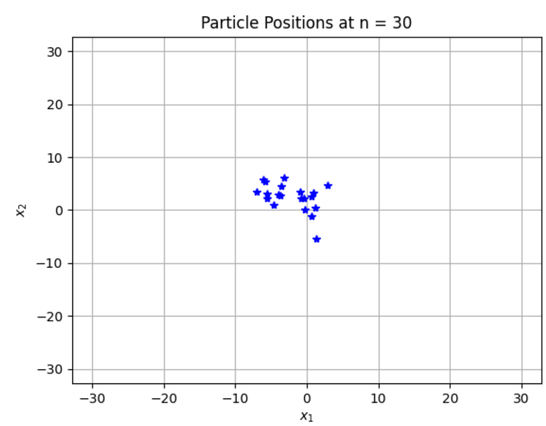
It's clear that my PSO can converge onto the true global minimum of [0,...,0] for Ackley's function. Given more particles and an ideal initial random distribution of particles, the confidence of convergence increases. If too many particles find a suitable local minimum, the swarm will converge onto that one, ignoring the global minimum. The complexity of the problem also increases with more dimensions, requiring more particles and iterations to converge onto [0,...,0] with the same confidence.

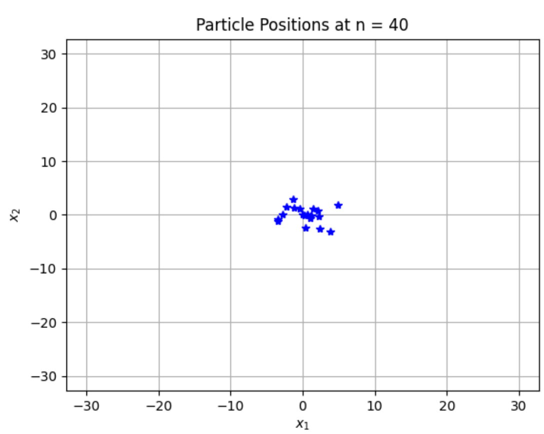
For 20 particles estimating the 2D Ackley function, their positions at specific iterations are shown below to show how the particles swarm.

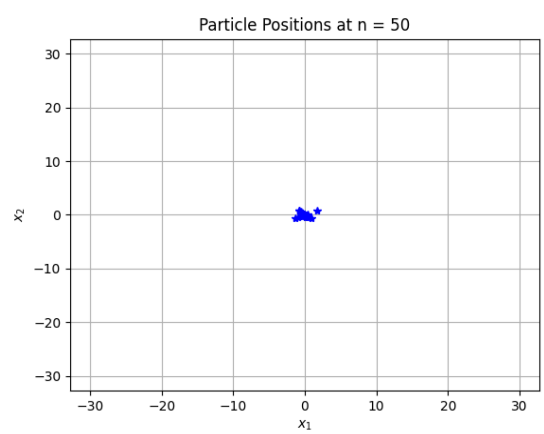












At iteration 10, a number of particles left the bounds of the problem, [-32.768, 32.768], and return to the swarm in later iterations. The fact that particles can leave the bounds may explain why my algorithm sometimes converges onto other local minima, typically outside of the bounds, and fixing this could increase the confidence of the algorithm.