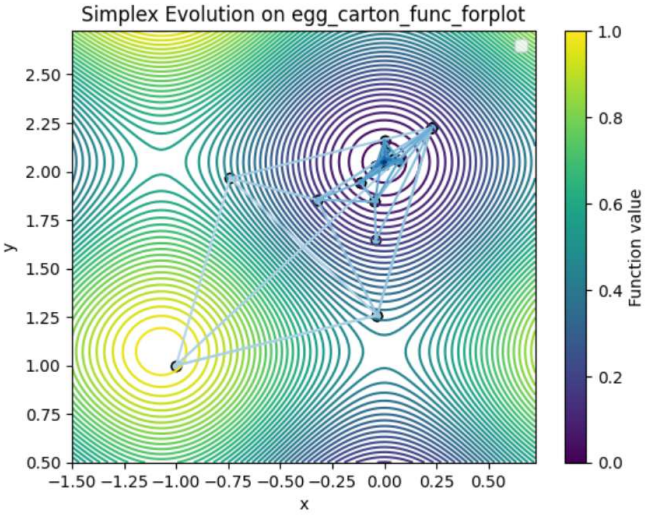
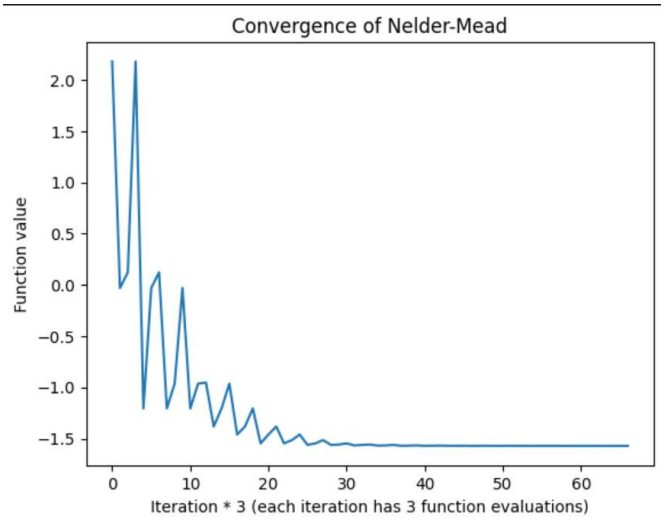
HW6_2

Thursday, March 20, 2025 1:19 AM

6.1 Nelder Mead





X0 was [-1,1]

Note that the convergence plot shows each point on the simplex, not just the best etc, thus the oscillations.

6.2 Me vs Scipy

X0 Point	Me (iters), f_val	Scipy (iters), f_val	Lower	Same location?
[1,1]	21, -1.14	48, -1.57	Scipy	No
[5,5]	23, 1.86	45, 1.43	Scipy	No
[-3,-3]	21, -1.14	47, 0.14	Mine	No
[0,0]	20, -2	12, -2	Tie	Yes
[-1,1]	22, -1.57	48, -1.57	Tie	No

Observations: At first glance it appears that my algorithm did better, but in fact it converged to different local minimums 3/5 times and was worse 2/5, tied 2/5 and was better 1/5. When it did tie it had more iterations than scipy. This highlights pros and cons of Nelder Mead: it can't avoid local minimums, and how the initial simplex is made likely effects things like direction etc which explains why we converged to different values, not always worse.

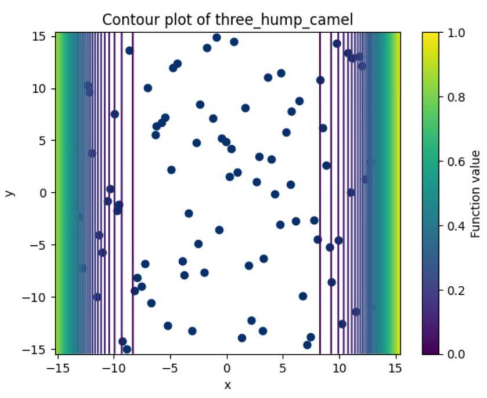
6.3: Particle Swarm Varying Parameters

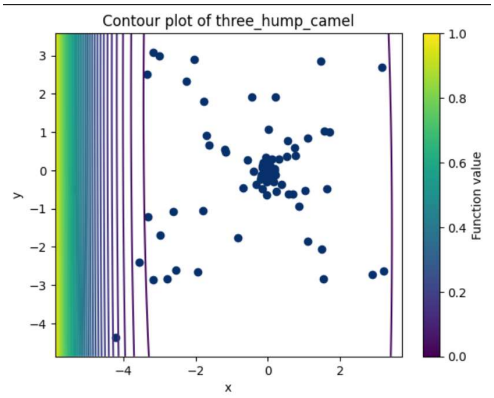
	Population	**Alpha (inertia)**	**Beta (cognitive/self-focused)**	**Gamma (social focused)**	**Iterations** (Avg of 3 runs), max_iter = 400	**Convergence Criteria**
Standard (Median Convg)	100	Rand	Rand	Rand	78.3	Median(Delta_x) < 1e-5
Standard (Mean Convg)	100	Rand	Rand	Rand	400	Mean(Delta_x) < 1e-5
Small	50	Rand	Rand	Rand	88	Median(Delta

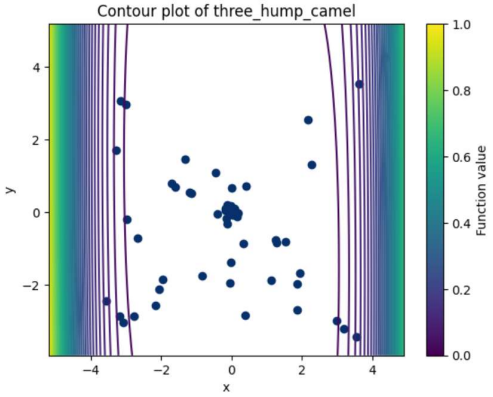
Population						_x) < 1e-5
Large Population	200	Rand	Rand	Rand	69	Median(Delta _x) < 1e-5
Inertia Focused	100	1.0	0.1	0.1	400	Median(Delta _x) < 1e-5
Cognitive Focused	100	0.1	1.0	0.1	31	Median(Delta _x) < 1e-5
Social Focused	100	0.1	0.1	1.0	12.3	Median(Delta _x) < 1e-5

Observations:
Comparing the mean and median convergence criteria shows that the median converges much faster, and the mean hit max_iter. This makes sense as some of the points will likely get stuck in local minimums or have too much inertia and not settle, so the mean will stay high for much longer. Population was as expected, the larger population performed better, but not by as much as I was expecting, and the small population worse. Inertia focused always hit max_iter, as they were likely jumping around too much, cognitive did very well, and social focused did extremely well. Social likely did the best because due to the bounds (-4,4) I probably had an overly large population size, so a very good point was likely found right at the beginning.

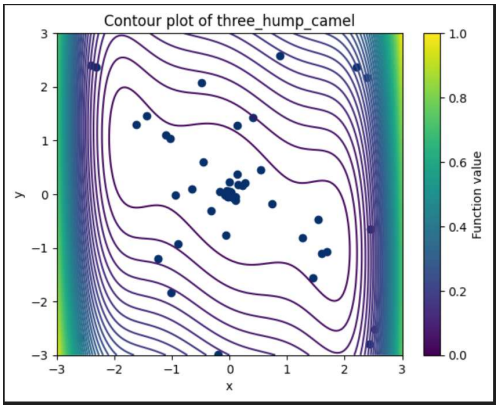
6.4 Three Hump Camel Test function:







Observations:
This is the Three Hump camel test function, it has a global minimum at (0,0) with a value of 0, the optimizer found: (2.30003314e-07, -1.08622550e-06) with a value of 1.4508532723308657e-13, from the best particle.
The spread is pretty large right at the beginning (the first plot), midway through the 117 iterations (the middle plot) the particles are all a lot closer to the middle, although the center spread is still semi large. The final plot shows some straggling points, but the large majority of the points have made it to the minimum. If I had tailored the coefficient weights it probably could have done better.



More zoomed in plot, a few iterations before convergence