* results/analysis\_std\_pso.txt (300)

In all 10 runs, the initial movement towards the optimum was the fastest. The fitness decrease slowed down, before finally converging completely at about the 200th iteration. This is evident not only from the global best value, which stagnates at this point but also the standard deviation, which approaches 0, indicating that all particles are at similar positions. The mean velocity also decreases fast, with the particles’ mean speed being almost 0 at the 300th iteration.

All 10 runs performed similarly, some converged early (such as: light blue, brown), while the orange run showed a slight movement towards the optimum at the 800th iteration, but this is an outlier. Overall, in 8/10 runs the mean particle position was <3 units away from the optimum by the end of the run, with 1/10 being <1 unit away.

Overall, the PSO performed well, considering the number of dimensions. The limiting factors seemed to be the fast convergence to a very uniform swarm, which reduces the velocity and means that particles have a hard time escaping local optima (which are plentiful in the Ackley function). Once all particles are in the same general area, it is unlikely that any further meaningful movement towards the global optimum can be made, as they lose their momentum.

* results/analysis\_weight\_adjust.txt (300)

The main difference in this approach is the dynamic velocity adjustment, as it decreases with increased iterations. This seemed to decrease the likelihood of a premature swarm-wide convergence, as the fitness value only seemed to stop decreasing at around the 500th iteration. Interestingly, while the global best was either worse or the same over different runs compared to the standard PSO that uses a constriction method, the mean position distance was much better, being zero for 9/10 runs. While overall this method seemed to improve the performance, it also vastly increased the likelihood of particles suddenly leaving the bounds and straying far away from the optimum, before returning to an optimum. This is apparent from the blue run, where particles suddenly “jumped” away from the mean, increasing the mean distance from the optimum and the standard deviation, but this resolved itself again within 70 iterations.

The speed seemed to decrease much slower than in the standard PSO, which meant that particles were able to escape local optima, with the speed even surging again for some runs as particles moved further apart. It is likely that this dynamic speed adjustment was both responsible for causing the sudden leaps out of bounds, as well as correcting the particle’s courses again.

The value we used for wmin was \_ and for wmax was \_. In our testing these performed the best, but it is possible that there is a combination of these values that might improve performance even further.

* results/analysis\_swarm\_size.txt (300)
* results/analysis\_topologies.txt (300)
* results/analysis\_combinations.txt (300)