Parallel Particle Swarm Optimization (PSO) Proposal

Siyi Hong, sh4325

**Topic Introduction**
The Particle Swarm Optimization (PSO) algorithm is a robust stochastic optimization technique based on the movement and intelligence of swarms. PSO applies the concept of social interaction to problem-solving as it simulates the social behavior of birds and fish. However, as the complexity of the problems increases, the computational burden also grows. Parallelizing PSO can significantly reduce computational time, making it suitable for complex and large-scale optimization problems.

**Why It Can Be Paralleled**
PSO consists of a swarm of particles moving through the search space, where each particle represents a potential solution. The parallelism in PSO is inherent as in each iteration, each particle’s position and velocity can be updated independently of the others. Moreover, the evaluation of the objective function, which is often the most computationally expensive task, can be done in parallel for different particles. This intrinsic independence makes PSO a perfect candidate for parallel computation.
How To Parallel It

By decomposing the algorithm into distinct segments that can be executed concurrently, the overall execution time can be significantly reduced. The flowchart above clearly indicates each process and check if it can be paralleled:

**Initial Swarm:** Each particle’s initial position and velocity can be determined independently, allowing for simultaneous initialization across the swarm.

**Calculate Self Score:** The fitness score of each particle is based on its position in the search space and can be computed in parallel, as it does not depend on the state of other particles.

**Update Self Best Score:** As each particle’s best-known position is updated based on its individual history, this step is inherently parallelizable.

**Update Swarm Best Score:** This step involves assessing all particles’ best scores to determine the global best. While each particle’s score can be calculated in parallel, the process of finding the global best is a sequential reduction operation.

**Update Position, Velocity for Each Particle:** The position and velocity updates for each particle are based on its current state and the global/local best positions, which can be computed in parallel for each particle.

**Development Plan**

The development of the parallel PSO algorithm will be a two-phase process to ensure efficacy and performance optimization. Initially, I will focus on implementing a serial version of the PSO to guarantee its functionality and use it as a benchmark. Once I establish a working serial model, I will incrementally parallel the algorithm following each process, which will allow us to refine and validate the parallelism at each step, ensuring that the final parallel algorithm is both accurate and efficient.