Particle Swarm Optimization - Simulation

Ronit Kumar Kataria
Department of Computer Science
Habib University
Karachi, Pakistan
rk06451@st.habib.edu.pk

Faraz Ali
Department of Computer Science
Habib University
Karachi, Pakistan
fa06396@st.habib.edu.pk

***Abstract*—Particle Swarm Optimization (PSO) is a metaheuristic algorithm that mimics the social behavior of birds flocking or fish schooling. In this report, we implemented the PSO algorithm in Python and created a visual simulation to demonstrate how the algorithm works. We first introduced the basic PSO algorithm and its key components, including the particle representation, the fitness function, the velocity update, and the position update. We then described how we implemented the PSO algorithm in Python and explained the design choices we made. Finally, we presented the results of our simulation, which showed how the PSO algorithm converges to the global optimum in a multimodal function optimization problem. Our visual simulation allowed us to see the behavior of the particles as they searched for the optimal solution and provided an intuitive understanding of how the PSO algorithm works.**

***Index Terms*—Particle Swarm Optimization, PSO, Swarm Algorithm, Simulation**

I. Introduction

Particle Swarm Optimization, or PSO, is inspired by the actions of a flock of birds or fish schooling. For our simulation, we made use of how birds in a flock behave. The intuition behind the algorithm is that there is a Food Source, or rather an optimum point, that is to be reached by the maximum number of birds in the flock. For PSO, it is assumed that there is a sense of communication amongst the particles or in this case, the birds. Initially, the system of birds are randomly allocated some start points and they fly around finding the source of food. In each generation, their velocity and position gets updated. Apart from this, we keep track of a pBest, i.e. Personal Best of each bird. Similarly, a Global Best, gBest, is also maintained which eventually leads to convergence of the flock. How the calculations are performed are explained in later sections.

II. Algorithm Description

The main aspect of the entire PSO algorithm is how it calculated the velocity and position of each bird at each generation. As highlighted earlier, the birds are initially assigned random solution which also include assigning them random velocities. The three main things that influence the calculation of the velocity in next iteration are as follows:

1) Current Velocity
2) Personal Best: Its fittest Position yet encountered
3) Global Best: The Global best and most fittest position visited by any bird in the flock

At each iteration the birds, or particles, are displaced from their current position by applying a velocity (gradient) vector to them. The magnitude and direction of velocity for each particle at each iteration is influenced by their velocity in the previous iteration of the algorithm, simulated momentum, and the location of the a particle relative to the location of its pbest and the gbest.
The Velocity and Position of each particle is updated with the following equations:

1) $V_i = \omega V_i + c_1 * \textit{rand} * (\textit{pBest} - P_i) + c_2 * \textit{rand} * (\textit{gBest} - P_i)$
2) $P_i = P_i + V_i$

Here P_i is the current position of the i^{th} particle and V_i is the velocity with same definition. ω is the inertia weight.

III. Methodology

This section discusses the problem formulation, concept, and methods of creating the simulation.

A. Problem Formulation

Processing for Python was used to create the entire simulation. As discussed in the *Algorithm Description*, we started with initially randomly populating birds. The number of birds in the flock was kept to 30 initially. The user, once plays the simulation, has the option to add more birds. This will add a complete flock of birds, containing 30 birds. Initially, the birds move randomly and there is no source of food. The update their *gBest* and *pBest* based on this randomness. Some convergence might be simulated even with lack of food. This happens because birds in the flock follow each other. Once the food is added, which user gets to add, the birds start to calculate their *pBest* and *gBest* and the updating of their velocities and position happens as discussed earlier. The simulation provides user the option to play around with several fun coefficients. This include:

1) Momentum
2) C_1 or Accel Const 1
3) C_2 or Accel Const 2
4) Size of Food
5) Colour of Birds

the last two doesn't impact the working of the algorithm but is added to create more interaction.

B. Concept

The concept, or motivation behind this simulation was to visually simulate how PSO works. Learning via reading is different from actually getting to simulate and play around with variables to see how they impact the overall working of an algorithm. Our initial idea was to implement a particle system and simulate the double slit experiment, but we went with the implementation and simulation of PSO instead.

IV. RESULT

The result of the simulation can be found at the following Youtube Link:
https://youtu.be/uifXwpcD7fk

REFERENCES

[1] OpenAI. *ChatGPT*. [Online]. Available: https://openai.com/chat. Last accessed: 19 March 2023.

[2] Osama Usuf. *GitHub*. [Online]. Available: https://github.com/osama-usuf/Interactive-Ant-Colony-Optimization-Simulation

[3] Dr. Saleha Raza. *CS 451 Notes*. [Online]. Available: https://hulms.instructure.com/courses/2501