FastPSO: Documentation and Technical Report

B Bharadwaj
B.Tech Computer Science, NIT Warangal

July 6, 2025

Contents

1 Introduction

FastPSO is a modified Particle Swarm Optimization algorithm developed with a focus on improving convergence speed while maintaining the quality of solutions. This document presents the motivation, algorithmic modifications, implementation details, and usage instructions.

2 Motivation

Traditional PSO algorithms focus on finding good solutions but may take more time to converge. FastPSO introduces four simple yet effective modifications aimed at accelerating convergence.

3 Algorithm Overview

3.1 Original PSO

Particle Swarm Optimization (PSO) is a population-based metaheuristic algorithm inspired by the social behavior of birds and fish. It optimizes a function by iteratively trying to improve a set of candidate solutions, called particles, based on a given fitness function.

Each particle in the swarm has:

- A **position** representing a possible solution in the search space.
- A **velocity** that determines the direction and speed of its movement.
- A **personal best position (pbest)** which stores the best position it has found so far.

The swarm as a whole keeps track of the **global best position (gbest)** — the best position any particle has achieved across all iterations.

The algorithm works by updating the velocity and position of each particle using the following formula:

$$v_i(t+1) = v_i(t) + c_1 r_1 (pbest_i - x_i) + c_2 r_2 (gbest - x_i) \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

Where:

- x_i is the current position of particle i
- v_i is the velocity of particle i
- $pbest_i$ is the best position found by particle i
- $gbest$ is the global best found so far
- r_1, r_2 are random values between 0 and 1

- c_1, c_2 are acceleration constants

The random factors and social influence guide the particles toward optimal regions of the search space, balancing exploration and exploitation.

3.2 FastPSO Modifications

1. **Uniform Initialization of Positions:** Particles are initialized across equally divided blocks of the search space.
2. **Non-Zero Initial Velocities:** Small random velocities are assigned at start.
3. **Sudden Movement to Gbest:** A random particle is moved to the global best.
4. **Selective Position Updates:** A position is only updated if it improves or maintains fitness.

4 Test Function

FastPSO was tested on a 1D function:

$$f(x) = -x^2 + 10x + 20 \quad \text{where } x \in [-10, 10] \tag{3}$$

5 Results and Observations

To evaluate the performance of FastPSO, it was tested on the benchmark function multiple times and compared against the standard PSO algorithm.

5.1 Convergence Results (FastPSO)

Attempt	Iterations	x	y
1	1	5.00000	45.0000
2	2	4.98398	44.9997
3	3	4.99617	45.0000
4	15	4.99406	45.0000
5	15	4.95687	44.9981
6	1	5.00000	45.0000
7	15	5.03181	44.9990
8	14	4.99055	44.9999
9	1	5.00000	45.0000
10	6	5.02383	44.9994

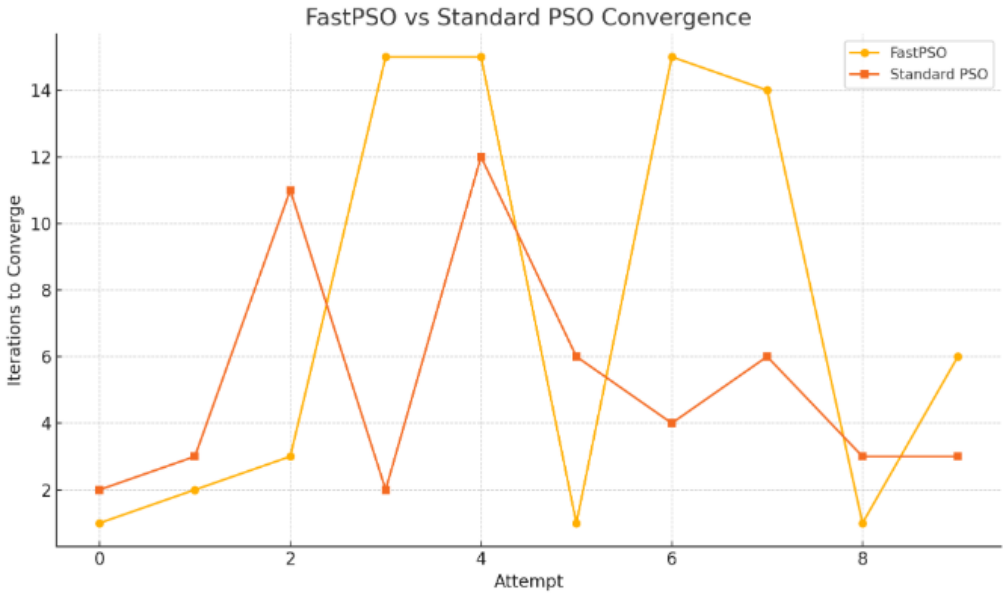
Best Case: 1 iteration **Average Case:** 5.2 iterations **Worst Case:** 15 iterations

5.2 Convergence Results (Standard PSO)

Attempt	Iterations	x	y
1	2	5.01834	44.9997
2	3	5.02813	44.9992
3	11	5.03322	44.9989
4	2	5.01316	44.9998
5	12	5.04163	44.9983
6	6	5.00898	44.9999
7	4	5.02634	44.9993
8	6	4.95814	44.9982
9	3	4.95502	44.9980
10	3	4.98970	44.9999

Best Case: 2 iterations **Average Case:** 5.2 iterations **Worst Case:** 12 iterations

5.3 Comparison Plot



As observed, FastPSO achieves comparable or better convergence speed with more consistency in certain cases, and converges as fast as in 1 iteration in some runs.

6 Future Work

- Extend to high-dimensional and benchmark test functions
- Compare against standard PSO and other variants
- Apply to real-world optimization problems

7 License

This project is licensed under the MIT License. See the LICENSE file for more details.