

Modified Particle Swarm Optimization for Avoiding Mexican Hat Problem in One-Dimensional Search Space

IL-Ho Yang, Min-Seok Kim, and Ha-Jin Yu[†]

School of Computer Science, University of Seoul, Seoul 130-743, Korea

Phone: +82-2-2210-5613, Fax: +82-2-2210-5275, [†]hju@uos.ac.kr

Particle Swarm Optimization (PSO) is an optimization algorithm which uses the interactions between the particles. Each particle in a population is accelerated and moves to a better position from the current position. One of the problems is that the process can be stuck in a local optimum. In this research, we propose an algorithm that avoids convergence in local optimum and searches for all global optima in the search space by modifying the original function. The recent approaches on function transformation include deflection, stretching, and repulsion. The main idea of these approaches is to form a hill-shaped penalty area around the minimum point found to avoid local convergence. However, such approaches have a problem that the penalty area can make other minima which is called Mexican hat at each side of it. In this research, we use straight line segments for the penalty area. From the convergence points we select the closest local maximum points at each direction, and we modify the function by drawing a straight line segment from one maximum point to another maximum point. The filled function $F(x)$ formed by connecting the two maximum points $(x_1, f(x_1))$, $(x_2, f(x_2))$ can be derived using the following equations. ($x_1 < x_2$)

$$g(x) = \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) + f(x_1), \quad \left\{ \begin{array}{l} F(x) = \max(f(x), g(x)), \text{ if } (x_1 < x < x_2) \\ F(x) = f(x), \text{ othercase} \end{array} \right\}$$

Figure 1 through 3 show a simulation of the proposed method. Figure 4 and 5 show the comparison of the proposed method with the conventional PSO algorithm.

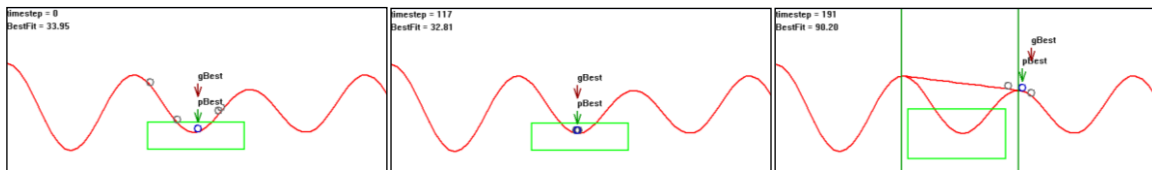


Fig. 1. PSO initialization.

Fig. 2. After all particles are converged to a local optimum.

Fig. 3. Making a penalty area.

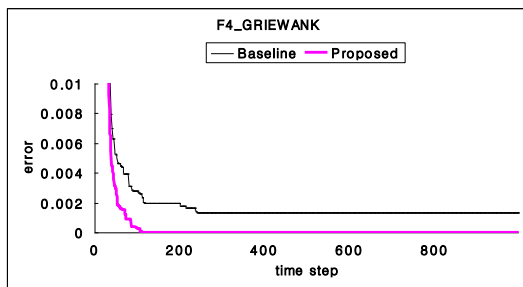


Fig. 4. experimental result (griewank function)

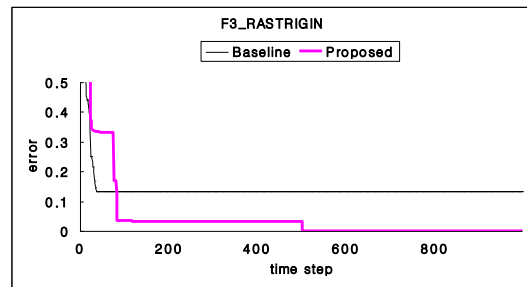


Fig. 5. experimental result (rastrigin function)