18-34

Particle Swarm Optimization Containing Plural Swarms with Influence

Masaki SUGIMOTO (Tokushima University) **Taku HARAGUCHI** (Tokushima University) **Haruna MATSUSHITA** (Tokushima University) **Yoshifumi NISHIO** (Tokushima University)

1. Introduction

Particle Swarm Optimization (PSO) [1] is a popular optimization technique for the solution of object function and is an algorithm to simulate the movement of flock of birds and the movement of a school of fish toward foods. In this study, we propose PSO containing plural swarms with influence (PPSO). The important feature of PPSO is that each particle of PPSO belongs to one of plural swarms, which have different characteristics and influence other swarms. We investigate the behavior of PPSO and confirm its efficiency.

2. PPSO

[PPSO1] (Initialization) Let a generation step $t = 0$ and $t_c = 0$, i.e., t_c is the time step for update the inertia weight w_k of each swarm S_k. Randomly initialize the particle position $\boldsymbol{X}_i$ and its velocity $\boldsymbol{V}_i$ for all particles i and initialize $\boldsymbol{P}_i = (p_{i1}, p_{i2}, ..., p_{iD})$ with a copy of $\boldsymbol{X}_i$. Attach each particle i to any swarm S_k at random.

[PPSO2] Evaluate the current cost $f(\boldsymbol{X}_i)$. Update the personal best position *pbest* $\boldsymbol{P}_i$ for each particle i and the global best position *gbest* $\boldsymbol{P}_g$ among the all particles in all the swarms so far.

[PPSO3] Let $\boldsymbol{Ps}_k$ represents the swarm best position with the best cost among the particles belonging to the swarm S_k so far (called *sbest*). Update $\boldsymbol{Ps}_k$ for each swarm S_k, if needed. **[PPSO4]** Updated $\boldsymbol{V}_i$ and $\boldsymbol{X}_i$ of each particle i depending on its *pbest*, its swarm best *sbest* and *gbest*;

$$\begin{aligned} v_{id}(t+1) &= w_k v_{id} + c_1 r_1 \{p_{id} - x_{id}(t)\} \\ &\quad + c_2 r_2 \{p_{s_k d} - x_{id}(t)\} + c_3 r_3 \{p_{gd} - x_{id}(t)\}, \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1), \end{aligned} \tag{1}$$

where r_1, r_2 and r_3 are three random variables distributed uniformly on $[0, 1]$, and c_1, c_2 and c_3 are positive acceleration coefficients. w_k is an inertia weight of each swarm S_k. In other words, the particles behave in a different way according to the inertia weight w_k of each swarm.

[PPSO5] If $t_c = T_c$, perform [PPSO6]. If not, perform [PPSO7]. Thus, perform [PPSO6] every T_c generation. T_c is a fixed parameter and is the number of generations to reconstitute the swarms.

[PPSO6] Update the inertia weight w_k of each swarm S_k as

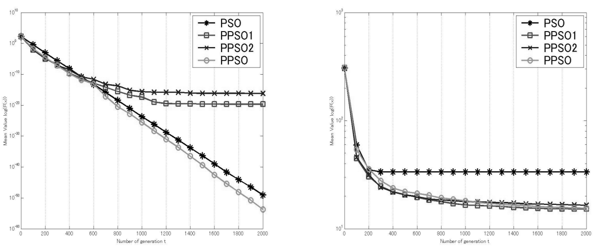
$$w_k(t+1) = w_k(t) - (w_k(t) - w_{\text{top}}) \times Q, \tag{2}$$

w_{top} is an inertia weight of the swarm containing *gbest*. Q is the rate of change of the inertia weight. And also, reset $t_c = 0$.

[PPSO7] Let $t = t + 1$ and $t_c = t_c + 1$. Go back to [PPSO2], and repeat until $t = T$.

3. Numerical Experiments

In order to evaluate the efficiency of PPSO and investigate the behavior of PPSO, we compare the four algorithms that PSO, PPSO1, PPSO2 and PPSO. PSO is the standard PSO and PPSO is the proposed algorithm. PPSO1 and PPSO2 algorithms are PPSO algorithm which does not perform [PPSO6]. The inertia weight w_k of PPSO1 is all the same as the inertia weight w of PSO. PPSO2 uses different inertia weight w_k depending on each swarm, however they are constant values unlike PPSO. We carry out the simulation 30 times for two optimization functions with 3000 generations. Figures 1(a) and (b) show the mean *gbest* values of every generation over 30 runs for Sphere function and Rastrigin function with 30-dimension. From these results, we can confirm that the mean values of PPSO are the best for two problems. Therefore, we can confirm that PPSO algorithm is the most effective.



(a) (b)

Figure 1: Mean *gbest* value of every generation for 30-dimensional two functions. (a) Sphere function. (b) Rastrigin function.

4. Conclusions

In this study, we have proposed PPSO. We have investigated its behavior with the simulation and have confirmed the efficiency.

Reference

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization", *Proc. of IEEE Int. Conf. Neural Networks*, vol. 4, pp. 1942–1948, 1995.