17-5

Particle Swarm Optimization Containing Multiple Swarms

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

1. Introduction

The particle swarm optimization (PSO) is a popular optimization technique for the solution of object function and is an evolutionary algorithm to simulate the movement of flocks birds. Due to the simple concept, such as easy implementation and quick convergence, PSO has gained much attentions and wide applications in different fields. However, if a particle is trapped into a local optimum, the bust-out is difficult.

This study proposes Multiple PSO (MPSO). The feature of MPSO is that the swarm of MPSO is not one but multiple. The swarms share an information of the best value in each group. Except the swarm including the best particle in whole swarms, all the particles are repositioned to escape the local optimum. We investigate behaviors of MPSO and confirm its efficiency.

2. Multiple PSO (MPSO) Algorithm

The most important feature of MPSO is that MPSO has K swarms, i.e. multiple particle groups; $Swarm_k$ ($k = 1, 2, \ldots, K$). Each swarm contains N/K particles whose position $\boldsymbol{X}_{k_i} = (x_{ki1}, x_{ki2}, \ldots, x_{kiD})$ ($i = 1, 2, \ldots, N/K$) are located randomly. The velocity vector $\boldsymbol{V}_{k_i} \in [\boldsymbol{V}_{\min}, \boldsymbol{V}_{\max}]$ for each particle k_i belonging to $Swarm_k$ is assigned randomly.

[MPSO1](Initialization) Let $t = 0$ and $t_R = 0$, i.e., t is the simulation step and t_R is the time step for the reposition. The quality of a particle position depends on a cost function $F(\boldsymbol{X}_{k_i})$. We find the personal best $pbest_{k_i}$, which is the best cost of the particle i in $Swarm_k$, and the global best $gbest_k$, which is the best value of $pbest_{k_i}$ in $Swarm_k$. The whole global best $gbest_w$, which is the best value in whole particles in all swarms, is also found;

$$gbest_k = \min_i(pbest_{k_i}), \quad gbest_w = \min_k(gbest_k). \quad (1)$$

The swarm including the particle with $gbest_w$ is called the best swarm. $\boldsymbol{P}_{k_i} = (p_{k_{i1}}, p_{k_{i2}}, \ldots, p_{k_{iD}})$, $\boldsymbol{P}_{k_g} = (p_{k_{g1}}, p_{k_{g2}}, \ldots, p_{k_{gD}})$ and $\boldsymbol{P}_w = (p_{w_1}, p_{w_2}, \ldots, p_{w_D})$ represent the position of particle k_i with $pbest_{k_i}$, the position of the particle with $gbest_k$, and the position of the particle with $gbest_w$, respectively.

[MPSO2] Compare the current cost $F(\boldsymbol{X}_{k_i}(t))$ of each particle with its best cost so far: if $F(\boldsymbol{X}_{k_i}(t)) < pbest_{k_i}$ then $pbest_{k_i} = F(\boldsymbol{X}_{k_i}(t))$ and $\boldsymbol{P}_{k_i} = \boldsymbol{X}_{k_i}(t)$. Similarly, $gbest_k$ and $gbest_w$ are evaluated according to Eq. (1). Therefore, $\boldsymbol{P}_{k_g}$ and $\boldsymbol{P}_w$ are also evaluated.

[MPSO3](Update) The velocity vector $\boldsymbol{V}_{k_i}$ for each particle and each particle position $\boldsymbol{X}_{k_i}$ are updated;

$$\begin{aligned} v_{k_{id}}(t+1) &= wv_{k_{id}}(t) + c_1 r_1 \left\{p_{k_{id}}(t) - x_{k_{id}}(t)\right\} \\ &\quad + c_2 r_2 \left\{p_{k_{gd}}(t) - x_{k_{id}}(t)\right\} + c_3 r_3 \left\{p_{w_d}(t) - x_{k_{id}}(t)\right\}, \\ x_{k_{id}}(t+1) &= x_{k_{id}}(t) + v_{k_{id}}(t+1), \end{aligned} \quad (2)$$

where $d = 1, 2, \ldots, D$, r_1, r_2 and r_3 are random variables distributed uniformly in $[0, 1]$, w is an inertia weight, and c_1, c_2, and c_3 are positive acceleration coefficients.

[MPSO4] If $t_R = T/K$, we perform [MPSO5], if not, we perform [MPSO6]. Thus, we perform [MPSO5] every time when T/K simulation steps are performed. T is the maximum number of the simulation.

[MPSO5](Reposition) We reposition the particle positions $\boldsymbol{X}_{k_i}$ of all the $Swarms_k$ except the best swarm at random and reassign their velocities $\boldsymbol{V}_{k_i}$ at random. We reset $t_R = 0$.

[MPSO6] Let $t = t + 1$ and $t_R = t_R + 1$. Go back to [MPSO2], and repeat until $t = T$.

3. Numerical Experiments

In order to confirm the performance of MPSO algorithm, we apply MPSO to the Rastrigin Function;

$$F(\boldsymbol{X}_{k_i}) = DA + \sum_{d=1}^{D}(x_{k_{id}}^2 - A\cos(2\pi x_{k_{id}})) \quad (3)$$

where $-5.12 \leq \boldsymbol{X}_i \leq 5.12$, $D = 30$, and $A = 5$. We set parameters as follows; $w = 0.5, c_1 = 2.0, c_2 = 1.7, c_3 = 0.3, T = 2000$. MPSO has $K = 4$ swarms, and each swarm contains 10 particles, i.e. $N = 40$.

We carry out the simulation 100 times and the results of MPSO, the standard PSO and PSO*, which is the standard PSO with the reposition process, are shown in Table 1. We can see that the result of MPSO is the best value in three algorithms. In MPSO, all the particles share the information of the best value in each swarm, and the particles except the best swarm are repositioned. As the result, if the particle is trapped into local optimum, the particle easily escapes from the local optimum. Therefore, we can confirm that MPSO algorithm is the most effective.

Table 1: Result of Optimization

	PSO	PSO*	MPSO
AVG	35.94	22.51	13.83
MIN	11.87	10.75	6.92
MAX	87.80	45.52	27.26

4. Conclusions

In this study, we have proposed the new PSO algorithm, Multiple PSO (MPSO). We have investigated its behaviors 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.