IEICE Proceeding Series

A Fission-and-Recombination Particle Swarm Optimizer

Takumi Sato, Kazuki Maruyama, Toshimichi Saito

Vol. 2 pp. 193-196
Publication Date: 2014/03/18
Online ISSN: 2188-5079

A Fission-and-Recombination Particle Swarm Optimizer

Takumi Sato†, Kazuki Maruyama† and Toshimichi Saito†

†EE Department, Hosei University, Koganei, Tokyo 184-8584, Japan, Email: tsaito@hosei.ac.jp

Abstract—This paper studies a fission-and-recombination particle swarm optimizer and its application. At this algorithm, the particles have lifetime. Lifetime enables self-organization and flexible search. We examine performance of FRPSO by comparing it with standard PSO.

1. Introduction

The particle swarm optimizer (PSO) is a population-based optimization method inspired by flocking behavior of living beings [1]. The particles correspond to potential solutions and search some objective solution based on inter-particle communication in a swarm of the particles. The PSO is a simple in concept, fast and has been applied to various practical/potential engineering applications such as design of circuits and neural networks [2]-[4]. However, in standard PSO, particles tend to be trapped into local-minima depending on the problems. In order to escape the trapping, diversity of particle swarm and variation of particle movement are required [5]-[10].

This paper presents a fission-and-recombination particle swarm optimizer (FRPSO). The FRPSO has self-organizing function that can control the number of particles: each particle has the lifetime that controls the number of particles. If particle movement is stagnated, then the lifetime decreases. When the lifetime is expired, the particle disappears and a new sub-swarm is generated: the number of particles increases and the particle topology changes. This growing structure of the sub-swarms aims at escape from local minima. The plural sub-swarms operate in parallel. In the sub-swarm, if the lifetime of some particle is expired, the particle disappears: the number of particles decreases in total.

We apply this algorithm to several benchmark functions. Performing basic numerical experiments, we have confirmed that, depending on the problems, the number and topology of the particles can vary flexibly. As compared with standard PSOs, our algorithm can realize higher success rate and lower number of iteration.

2. Algorithm

First, we introduce a standard ring type PSO (RPSO). The RPSO is one of the standard PSO. In the RPSO, the particle exchanges information for only the neighborhood particles. There are the following two kinds of information. The Pbest is own best position in the self-search process. The Lbest is best position in the neighborhood particles and oneself. Let $\boldsymbol{x}^P$ and $\boldsymbol{x}^L$ be the posotion of the Pbest and Lbest respectively. The positions of Pbest and Lbest are decided by fitness. The evaluate of fitness is done by substituting the position $\boldsymbol{x}$ for function $f(\boldsymbol{x})$. The characteristic of the ring type uniting is to be able to oppose the localized solution though it is simple.

Step 1 (Initialization): Let search step $t = 0$. The number of particle is N. The particle velocity $\boldsymbol{v}_i(0)$ is initialized to zeros and position $\boldsymbol{x}_i(0)$ is randomly initialized in the search space $(i = 1 \cdots N)$. After that evaluate the fitness and set $\boldsymbol{x}_i^P(0), \boldsymbol{x}_i^L(0)$, $t \leftarrow t + 1$.
Step 2 (State Update): Update $\boldsymbol{v}_i$ and $\boldsymbol{x}_i$ at all particles.

$$\begin{aligned}\boldsymbol{x}_i(t) &\leftarrow \boldsymbol{x}_i(t-1) + \boldsymbol{v}_i(t-1)\\ \boldsymbol{v}_i(t) &\leftarrow \omega\boldsymbol{v}_i(t-1)\\ &\quad +\rho_1\left\{\boldsymbol{x}_i^P(t-1) - \boldsymbol{x}_i(t-1)\right\}\\ &\quad +\rho_2\left\{\boldsymbol{x}_i^L(t-1) - \boldsymbol{x}_i(t-1)\right\}\end{aligned} \tag{1}$$

where ω is inertia weight. In general, $\omega = 0.7$. ρ_1 and ρ_2 are random number in the range of $[0, 1.4]$.
Step 3 (Information Update): Calculate evaluation of the fitness at each particle position and update of the Pbest and Lbest.

$$\begin{aligned}&\boldsymbol{x}_i^P(t) \leftarrow \boldsymbol{x}_i(t) \text{ if}\\ &\quad f(\boldsymbol{x}_i(t)) < f(\boldsymbol{x}_i^P(t-1))\\ &\boldsymbol{x}_i^L(t) \leftarrow \boldsymbol{x}_{i-1}^P i(t), \boldsymbol{x}_i^P(t), \boldsymbol{x}_{i+1}^P(t) \text{ if}\\ &\quad f(\boldsymbol{x}_{i-1}^P(t)) < f(\boldsymbol{x}_i^L(t-1))\\ &\quad f(\boldsymbol{x}_i^P(t)) < f(\boldsymbol{x}_i^L(t-1))\\ &\quad f(\boldsymbol{x}_{i+1}^P(t)) < f(\boldsymbol{x}_i^L(t-1))\end{aligned} \tag{2}$$

If the condition is not met, Pbest and Lbest keep it intact.
Step 4 (Fitness Judgment): When best fitness evaluation of Lbest became less than the criteria C, finish the search.
Step 5 (Step Judgment): Let $t \leftarrow t+1$, return Step 2 and repeat until $t = T_{max}$.

Next, we define the FRPSO.
Step 1 (Initialization): Let generation step $t = 0$. The particle velocity $\boldsymbol{v}_{i,0}(0)$ is initialized to zeros and

position $\boldsymbol{x}_{i,0}(0)$ is randomly initialized in the search space ($i = 1 \cdots N$). Lifetime $L_{i,0}$ is set to L. After that evaluate the fitness and set $\boldsymbol{x}_{i,0}^P(0), \boldsymbol{x}_{i,0}^L(0)$, $t \leftarrow t+1$. There are no sub particles at this step. **Step 2 (State Update):** Update $\boldsymbol{v}_{i,j}$ and $\boldsymbol{x}_{i,j}$ at each living particles. Living means that L is not equal to zero.

$$\begin{aligned} \boldsymbol{x}_{i,j}(t) &\leftarrow \boldsymbol{x}_{i,j}(t-1) + \boldsymbol{v}_{i,j}(t-1) \\ \boldsymbol{v}_{i,j}(t) &\leftarrow \omega \boldsymbol{v}_{i,j}(t-1) \\ &\quad + \rho_1 \left\{ \boldsymbol{x}_{i,j}^P(t-1) - \boldsymbol{x}_{i,j}(t-1) \right\} \\ &\quad + \rho_2 \left\{ \boldsymbol{x}_{i,j}^L(t-1) - \boldsymbol{x}_{i,j}(t-1) \right\} \end{aligned} \tag{3}$$

Step 3 (Information Update): Evaluate the each particle fitness and update Pbest and Lbest based on $\boldsymbol{x}_{i,j}$. If Pbest is not update, decrease lifetime -1. Lifetime $L_{i,0} = 0$, the particle is dead and cannot be updated. When lifetime of main particle, $L_{i,0}$, is equal to 0, go to Step 4. Otherwise, go to Step 5. When sub particle dies, stop update. If the number of sub particles is 1, replace the particle as a main particle. Then, the information succeeds a sub particle. However, initialize only lifetime.
Step 4 (Addition of the particle): Scatter sub particles with a random number around the main particle which died at Step3 of this search step. The number of new sub particles is 3 to J. Dispersion width E is determined by next equation.

$$E = E_{max} \times \frac{T_{max} - t}{T_{max}} \tag{4}$$

Next, construct each ring unitting with only sub particles. This unitting is independent. After that, initialize Pbest and Lbest. **Step 5 (Fitness Judgment):** When best fitness evaluation of Lbest became less than the criteria C, finish the search. **Step 6 (Step Judgment):** Let $t \leftarrow t+1$, return Step 2 and repeat until $t = T_{max}$.

3. Numerical experiments

We apply RPSO and FRPSO to four benchmark functions.
1)Schwefwl function: This is a multimodal function and optimum solution exists on the edge of seach space.

$$f_1(\boldsymbol{x}) = \textstyle\sum_{d=1}^{D} -x_d sin(|\sqrt{x_d}|) + 418.98288727D \tag{5}$$

where D is number of dimensions ($d = 1 \cdots D$).
2)Salmon function: This function's characteristic is local-minima. These exist on circle line and consecutive.

$$f_2(\boldsymbol{x}) = -\cos(2\pi \sqrt{(\textstyle\sum_{d=1}^{D} x_d^2)}) + 0.1\sqrt{(\textstyle\sum_{d=1}^{D} x_d^2)} + 1 \tag{6}$$

3)Griewank function: This is a multimodal function.

$$f_3(\boldsymbol{x}) = 1 + \textstyle\sum_{d=1}^{D} \frac{x_d^2}{4000} - \prod_{d=0}^{D}(\cos(\frac{x_d}{\sqrt{d}})) \tag{7}$$

4)Ridge function:

$$f_4(\boldsymbol{x}) = \textstyle\sum_{d=1}^{D}(\sum_{k=1}^{d} x_k)^2 \tag{8}$$

We apply RPSO and FRPSO to these four benchmarks. Results is shown in Figure 1-5. The parameters are selected after trial-and-errors:

$$\begin{aligned} N = 10, J = 10, T_{max} = 1000, \\ C = 1.0, E_{max} = 256, |\boldsymbol{x}| < 512 \end{aligned} \tag{9}$$

Figures 1 to 3 show the results of the Schwefel function. Figure 1 shows evolution of the fitness value and the number of particles. We can see that the fitness is improved as the number of particles increase. Figure 2 shows time evolution of the fitness averaged for 1000 trials. In the early stage, the RPSO can decrease the fitness faster than the FRPSO. However, at the time limit T_{max}, the FRPSO can realize better fitness value than the RPSO. Figure 3 shows snapshots of the particles locations. We can see that the particle's diversity of the FRPSO is wider than that of the RPSO.

Figure 4 and 5 show results of the FRPSO in the four functions. The algorithm performance is evaluated by the SR and the number of iterations (#ITE) in successful runs. In the Schwefel, Salmon and Ridge functions, the SR decreases and lifetime L increases. In the Griewank function, the SR decreases as L increases, however, it increases as L approaches 1000. The case $L = 1000$ is almost the same as the standard PSO without insensitivity. The Griewank function is unimodal function and the standard PSO can work efficiently for this function. Roughly speaking, there exist a trade-off between the SR and #ITE: #ITE increases as SR decreases. For the multimodal functions, diversity of particles in the FRPSO seems to be work efficientively.

4. Conclusions

The FRPSO is presented and its capability is investigated in several benchmarks functions. The lifetime can work effectively to multimodal function.

Future problems include setting of suitable parameter values, control of particles diversity and applications to practical problems.

References

[1] A. P. Engelbrecht, Fundamentals of computational swarm intelligence, Willey, 2005.

[2] M. P. Wachowiak, R. Smolikova, Y. Zheng, J. M. Zurada and, A. S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3. pp. 289-301, 2004.

[3] A. B. van Wyk and A. P. Engelbrecht, Overfitting by PSO trained feedforward neural networks, in Proc. IEEE Congress Evol. Comput., pp. 2672 - 2679, 2010.

[4] H. Matsushita and T. Saito, Application of particle swarm optimization to parameter search in dynamical systems, NOLTA, IEICE, E94-N, 10, pp. 458-471, 2011.

[5] Z. Sevkli and F. E. Sevilgen, Discrete particle swarm optimization for the orienteering problem, in Proc. IEEE Congress Evol. Comput., pp. 1937-1944, 2010.

[6] M. Kubota and T. Saito, A discrete particle swarm optimizer for multi-solution problems, IEICE Trans. Fundamentals, E95-A, 1, pp. 406-409, 2012.

[7] K.-B. Lee and J.-H. Kim, Mass-spring-damper motion dynamics-based particle swarm optimization, in Proc. IEEE Congress Evol. Comput., pp. 2348-2353, 2008.

[8] T. Saito and E. Miyagawa, Growing-tree particle swarm optimizer with simple tabu search function, Proc. of NOLTA, pp. 376-379, 2009.

[9] R. Sano, T. Shindo, K. Jin'no, and T. Saito, PSO-based multiple optima search systems with switched topology, in Proc. IEEE Congress Evol. Comput., pp. 3301-3307, 2012.

[10] K. Maruyama, R. Sano and T. Saito, Deterministic discrete particle swarm optimizers with collision and insensitivity, Proc. NDES, pp. 245-248, 2012.

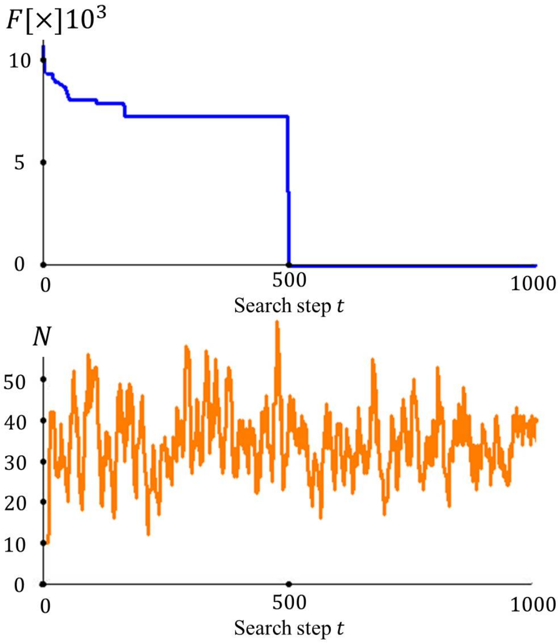

Figure 1: Typical example of fitness and the number of particles on FRPSO

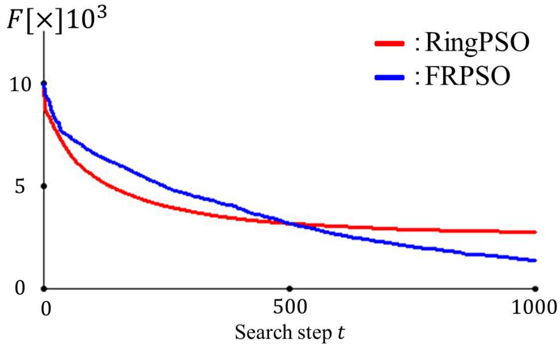

Figure 2: Comparison between FRPSO and RingPSO on schwefel function

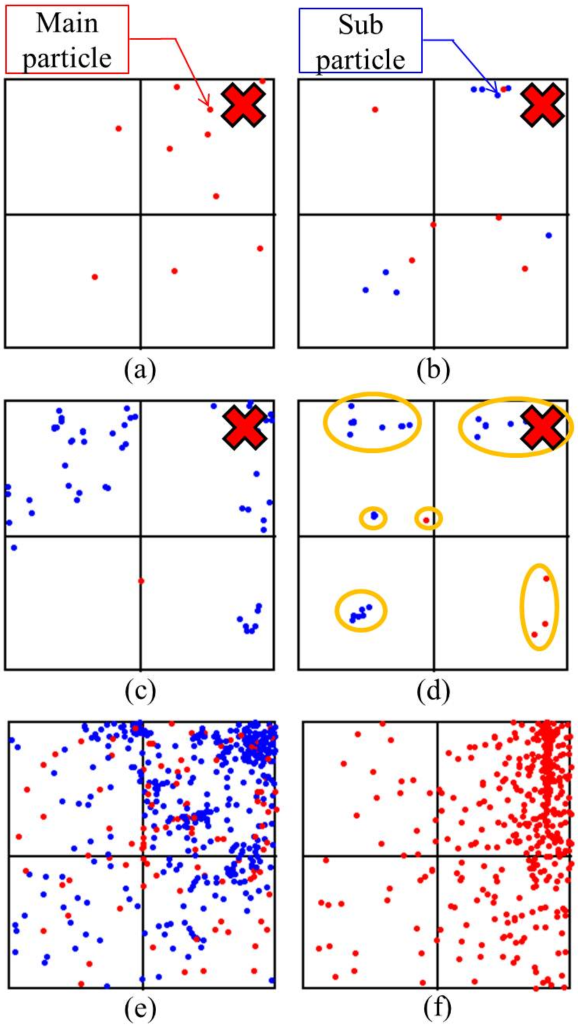

Figure 3: (a), (b), (c) and (d) are snapshot of seaching process. (e)(f) are overlapped drawing of seaching.

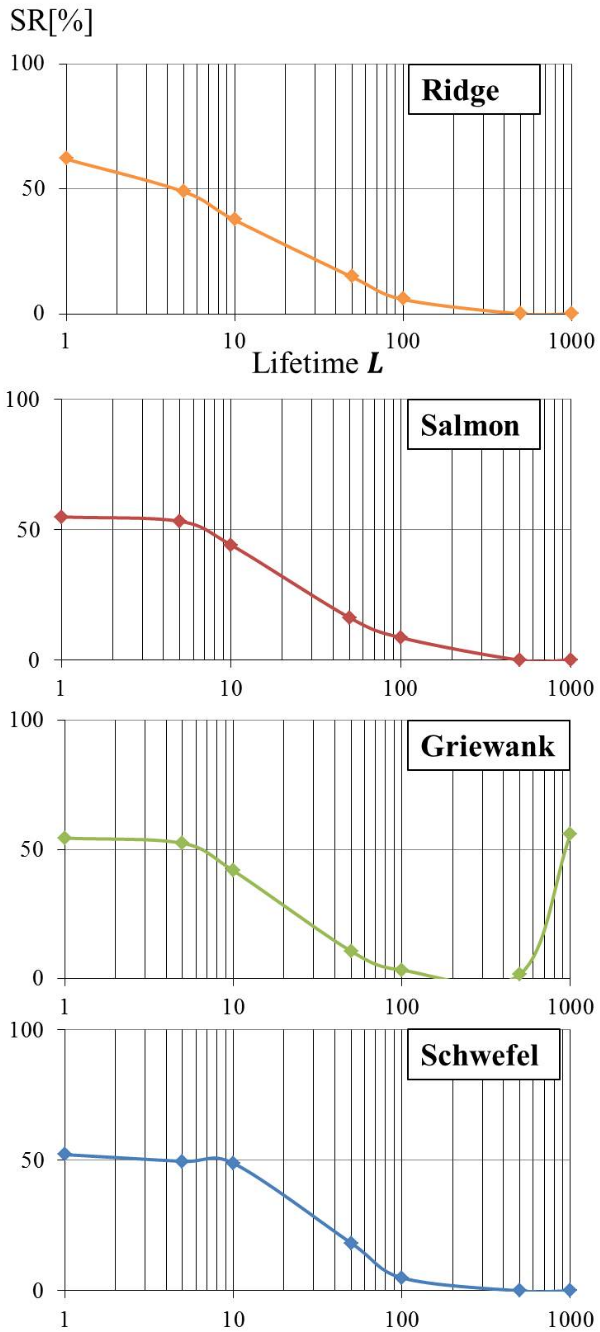

Figure 4: Lifetime vs SR

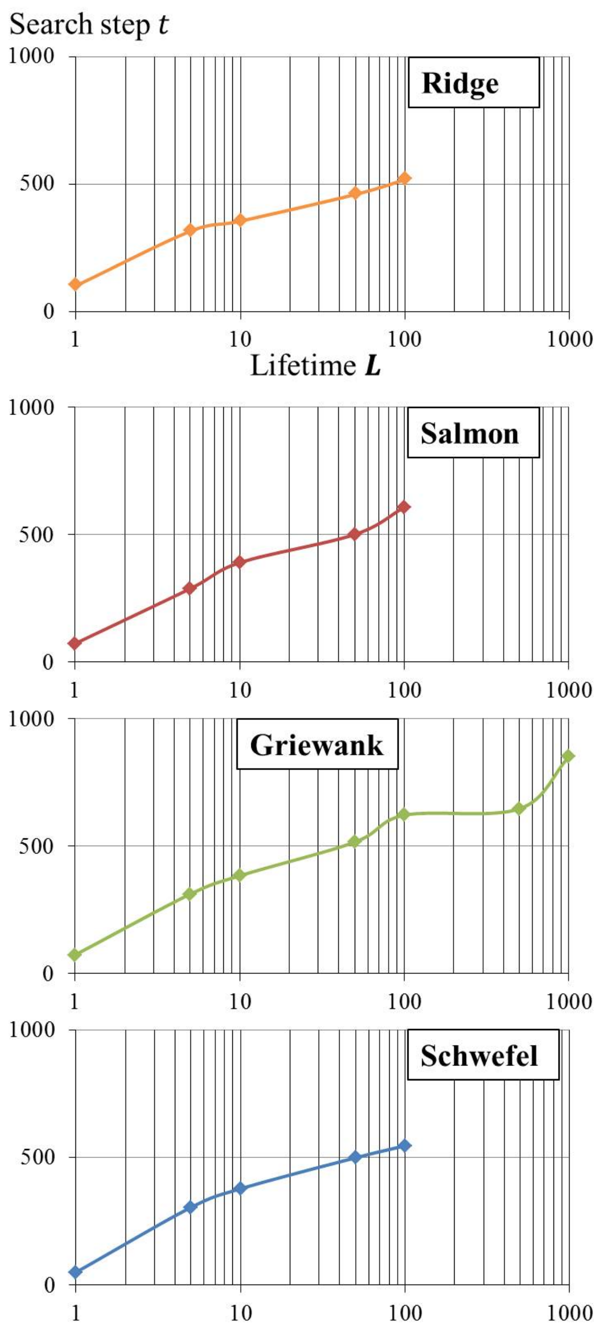

Figure 5: Lifetime vs Search step. Over $L = 100$, we cannot get the date of t.