2017 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2017, Cancun, Mexico, December 4-7, 2017

Study of SI algorithm that individual to dropout

Takuya Shindo† and Kenya Jin'no†

†Department of Electrical and Electronics, Nippon Institute of Technology
4–1 Gakuendai, Miyashiro, Saitama 345-8501, Japan
Email: t-shindo@nit.ac.jp, jinno@nit.ac.jp

Abstract—The PSO has a network structure to exchange information with each other among individuals. The optimal solution search performance change by this network topology has been investigated and reported. The network structure is also considered to be an important factor in dropout. Therefore, in this paper, we investigate the performance change due to the network structure of PSO in using dropout. In addition, we confirmed the performance of the proposed method by numerical simulation.

1. Introduction

A swarm of creatures such as birds, fish and ants may behave like having intelligence. The intelligence found in the behavior of such swarm is called swarm intelligence (SI). Many of the optimization algorithms such as Particle Swarm Optimization (PSO) [1][2] and Ant Colony Optimization (ACO) [3] using SI are inspired by the behavior of actual swarm of organisms. In the framework of optimization, SI is thought to be closely related to evolutionary computation, neural networks, and the like. Therefore, we consider using dropout [**?**] as one of the neural network methods for individuals in the swarm of PSO. PSO has a network structure to exchange information with each other among individuals. The optimal solution search performance change by this network topology has been investigated and reported [4][5]. The network structure is also considered to be an important factor in dropout. Therefore, in this paper, we investigate the performance change due to the network structure of PSO in using dropout. In addition, we confirmed the performance of the proposed method by numerical simulation.

2. SPSO 2011

Since 2006, there are three standard versions of PSO on the web. In this article, we focus on SPSO2011 in this version. In the early simple PSO, its performance depended on the landscape of the evaluation function. In SPSO2011, dependency on the coordinate system of the evaluation function has been modified.

Outline of PSO algorithm is shown in Algorithm 1. At initialization, particles of PSO are randomly placed in the search space. At the same time, an appropriate speed is given. From this position, an initial evaluation value, that

```
Swarm initialization;
while Evaluation value > Criteria || t < Max Iteration
 do
    Update particles;
    while i < Swarm size do
        Update velocity v_i;
        Update position x_i;
        Calcuration of evaluation value f(x_i);
        if f(x_i) < personal best p then
            p = f(x_i);
        end
    end
    Update global best g;
    if p < g then
        g = p;
    end
end
```

Algorithm 1: Particle swarm optimization algorithm

is, personal best p is calculated. Updating of the solution is done by these particles repeating movement.

2.1. Initialization

In initialization, position $\boldsymbol{x}_i(t)$ and velocity $\boldsymbol{v}_i(t)$ are set by eq. (1). The component of dimension D is d. max and min represent the maximum and minimum of the search space of each dimension.

$$\begin{cases} \boldsymbol{x}_i(0) = U(min_d, max_d) \\ \boldsymbol{v}_i(0) = U(min_d - x_{i,d}(0), max_d - x_{i,d}(0)) \end{cases} \tag{1}$$

Where $U(min_d, max_d)$ is a uniform random number in $[min_d, max_d]$.

2.2. Velocity update

In conventional PSO, the particle velocity is updated by synthesis of three vectors as in the eq. (2).

$$\boldsymbol{v}_i(t+1) = w\boldsymbol{v}_i(t) + c_1\boldsymbol{r}_1(\boldsymbol{p} - \boldsymbol{x}_i(t)) + c_2\boldsymbol{r}_2(\boldsymbol{g} - \boldsymbol{x}_i(t)) \tag{2}$$

w is an inertia coefficient, and c_1 and c_2 are parameters called acceleration coefficients. Also, r_1 and r_2 are uniform random numbers of $[0, 1]$.

On the other hand, in SPSO 2011, the velocity updating formula is as shown in eq. (3).

$$\boldsymbol{v}_i(t+1) = w\boldsymbol{v}_i(t) + \boldsymbol{x}'_i(t) - \boldsymbol{x}_i \tag{3}$$

Where, $\boldsymbol{x}'$ uses the center of gravity $\boldsymbol{G}_i$ determined from the current position x, p and g.

$$\boldsymbol{G}_i = \frac{\boldsymbol{x}_i + c(\boldsymbol{p}_i - \boldsymbol{x}_i) + (\boldsymbol{x}_i + c(\boldsymbol{g} - \boldsymbol{x}_i))}{3} \tag{4}$$

$$\boldsymbol{x}'_i = \mathcal{H}_i(\boldsymbol{G}_i, \|\boldsymbol{G}_i - \boldsymbol{x}_i\|) \tag{5}$$

$\boldsymbol{x}'$ is generated in the hypersphere of the appropriate distribution $\mathcal{H}$.

2.3. Position update

Updating the position is done by the following equation like the conventional PSO.

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1) \tag{6}$$

2.4. Confinement

In SPSO 2011, the following conditions are given to confine particles in the search space.

$$\begin{cases} \text{if } x_{i,d} < min_d & \text{then} \begin{cases} x_{i,d}(t+1) = min_d \\ v_{i,d}(t+1) = -0.5v_{i,d}(t+1) \end{cases} \\ \text{if } x_{i,d} > max_d & \text{then} \begin{cases} x_{i,d}(t+1) = max_d \\ v_{i,d}(t+1) = -0.5v_{i,d}(t+1) \end{cases} \end{cases} \tag{7}$$

3. Proposed method

In SPSO2011, the network structure between particles is set to ring or adaptive. Therefor, g is generally replaced with the local best l among the connected particles. As an example, the network of full connection and ring topology is shown in Figs. 1 and 2.

3.1. Dropout

Dropout[6] was proposed by Hinton et al. In order to optimize the deep neural network of the hierarchy with high accuracy. In the proposed method, initial solutions $p(0)$ and $g(0)$ are generated by dropout the particles of SPSO2011. Therefore, the meaning differs from dropout used in Neural Networks. The concept of Dropout in this paper is shown in the fig. 3. As shown in the fig. 3, the number of particles dropped from the swarm is determined by the probability α, and the search is performed with the remaining particles. In the proposed method, this search is repeated an arbitrary number of times. At this time, the position of each particle is reset every search. However, it holds p.

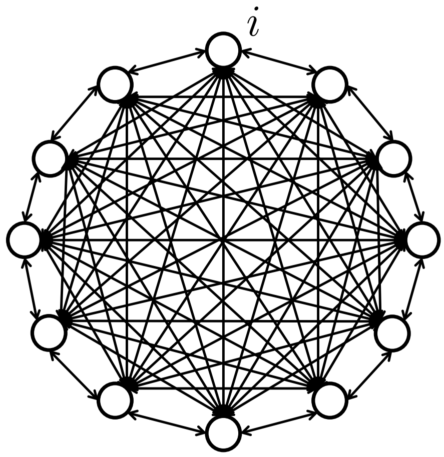

Figure 1: Fully connected network

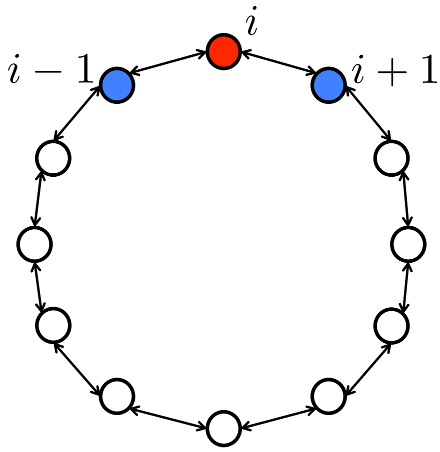

Figure 2: Ring network

4. Numerical Simulations

In order to confirm the effect of the proposed method, numerical simulation is executed. Common conditions of numerical simulation are shown below.

- Function: Sphere $f(\boldsymbol{x}_i) = \sum_{d=1}^{D} x_{i,d}^2$
- Search range: $[-5.0, 5.0]$
- Parameters: $w = 0.1$, $c = 0.1$
- Trial: 100
- Max iteration: 1000
- Swarm size: 40

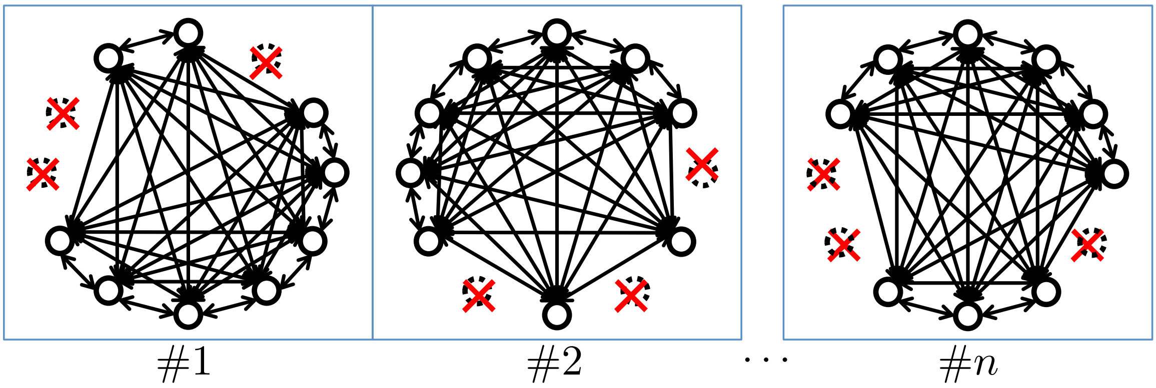

Figure 3: Dropout in proposed method

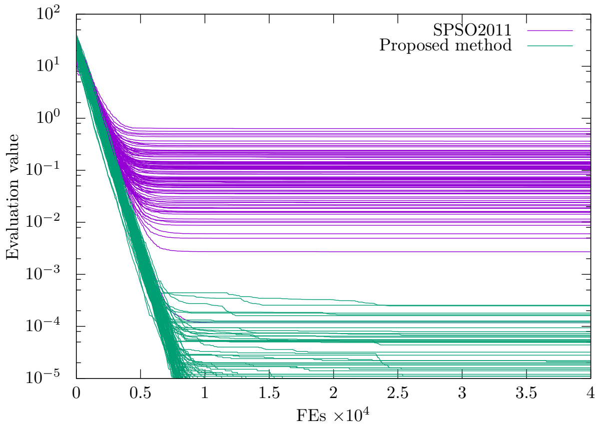

Figure 4: Search performance of proposed method

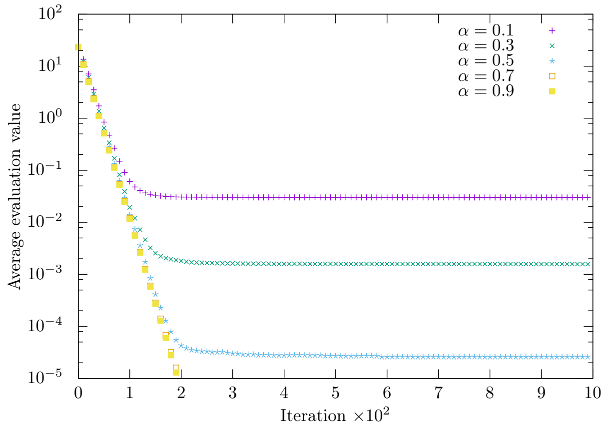

Figure 5: Effect of particle drop probability on performance change

4.1. Results

A comparison of the search performance of SPSO 2011 and the proposed method is shown in the fig. 4. The horizontal axis represents the number of evaluations of the evaluation function, and the vertical axis represents the evaluation value. In all trials, the performance of the proposed method exceeded SPSO 2011.

Figure 5 shows the performance change due to the drop probability. The horizontal axis represents the number of iterations and the vertical axis represents the average evaluation value of the trials. From the results, it can be confirmed that when the drop probability is high, the performance is also improved.

Figure 6 shows the performance change with the network change timing. The horizontal axis represents the number of iterations and the vertical axis represents the average evaluation value of the trials. When $iteration\%n = 0$, the network is changed. From the results it is expected that there will be appropriate network change timing.

5. Conclusions

We focused on the network structure of PSO and proposed a performance improvement method of solution search performance. We confirmed that the performance changes by dropout particles of swarm at an appropriate timing. On the other hand, we only check with a simple evaluation function. In the future, we will investigate the influence on the performance by parameters in more detail and confirm its performance by a more complicated problem.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95 - Int. Conf. Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[2] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proc. 1997 IEEE Int. Conf. Evol. Comput. (ICEC '97)*. IEEE, 1997, pp. 303–308.

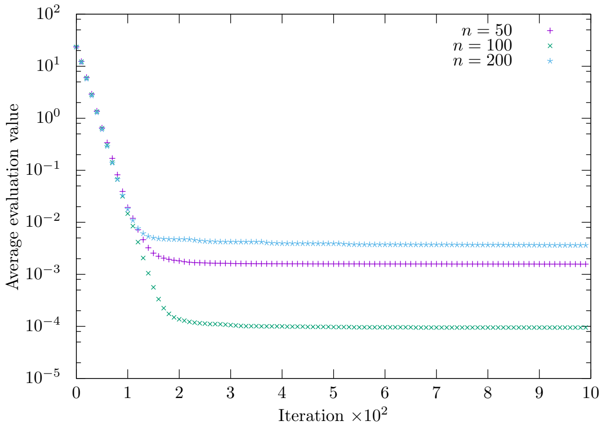

Figure 6: Performance change due to network change timing ($\alpha = 0.3$)

[3] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006.

[4] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. 2002 Congr. Evol. Comput. CEC 2002*, vol. 2, 2002, pp. 1671–1676.

[5] Q. Liu, W. Wei, H. Yuan, Z. H. Zhan, and Y. Li, "Topology selection for particle swarm optimization," *Inf. Sci. (Ny).*, vol. 363, pp. 154–173, 2016.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, jun 2014, vol. 15.