NOLTA, IEICE

Paper

# Learning a simple multilayer perceptron with PSO

*Riku Takato* [1] *and Kenya Jin'no* [1]

[1] *Graduate School of Integrative Science and Engineering, Tokyo City University*
*1-28-1 Tamazutumi, Setagaya, Tokyo 158-8557, Japan*

Received October 13, 2022; Revised December 29, 20XX; Published July 1, 20XX

**Abstract:** In this study, we attempt to learn the parameters of a multilayer perceptron (MLP) using the particle swarm optimization (PSO) method, which is an approximate solution method for optimization problems without requiring the derivative information of the evaluation function. We used the gradient method and PSO to learn to classify a linearly inseparable dataset with an MLP in the middle layer with a few neurons. We experimentally confirmed that PSO outperformed gradient-based learning.

**Key Words:** particle swarm optimization, multilayer perceptron, learning, accuracy, gradient method

## 1. Introduction

In recent years, multilayer perceptrons (MLPs) have garnered attention for their excellent function approximation ability. In fact, MLPs have been used to solve many problems in image processing and natural language processing. An MLP comprises several layers, with each comprising multiple neurons with nonlinear activation functions. Learning weight parameters using input–output data for training affords the function approximation ability of MLP. The weight parameters are learned by minimizing the loss function, which is defined as the difference between the output of the MLP and the expected output, as in other machine-learning methods. In general, the loss function of an MLP is nonconvex. The gradient method is used to optimize such a nonconvex function, where a solution is searched iteratively based on the gradient of the loss function. In particular, stochastic gradient descent is well known to be effective and is currently used in MLP learning. However, the stochastic gradient descent method does not guarantee convergence to the minimum solution of the nonconvex loss functions and is sensitive to the initial parameter values of the loss functions. In addition, achieving the optimal solution of the evaluation function using the gradient method is difficult owing to the numerous local solutions, plateaus, and saddle points. However, local solutions near the optimal solution has been suggested [1], and the convergence of the stochastic gradient method to the optimal solution when the network comprises a sufficiently large number of neurons has been reported [2]. Because these results suggest that gradient-based learning is successful in systems composed of many neurons, MLPs have recently become deep and exhibit more constituent neurons. As the number of neurons increases, the amount of memory required by the system increases, and hence the amount of computation. However, from an edge computing perspective, MLPs should achieve excellent recognition functions


  DOI: 10.1587/nolta.2.1101

using a few neurons. Therefore, to converge to the optimal solution of the evaluation function, even with a few neurons, we attempt to learn the parameters of an MLP via PSO [3], which does not require information regarding the derivative of the evaluation function.

Many attempts to train MLPs by such metaheuristic optimization methods have been reported[4]. In this study, we consider learning MLPs using PSO with a small-perturbations based on chaotic oscillation. We examine how well such a method can learn when the number of elements in the intermediate layer is small, which is difficult to learn with the gradient-based method.

## 2. MLP & Adam

A perceptron is a pattern recognition model proposed by Rosenblatt in 1958 [5]. The perceptron consisting of only two layers, an input layer and an output layer, is called a simple perceptron, and a binary step function is applied as an activation function of the output layer. If the input data are linearly separable, the simple perceptron can determine the weight parameters between the input and output layers within a finite number of iterations and can discriminate between the input data. However, the simple perceptron cannot discriminate inputs that are linearly inseparable [6]. Although a multilayer perceptron (MLP) [7] with intermediate layers between the input and output layers can discriminate linearly inseparable inputs, when the multilayer perceptron was first proposed, the weight parameters between the input and intermediate layers could not be determined by learning [7]. To overcome this problem, the error back propagation algorithm [8] was proposed to learn these weight parameters by applying a differentiable monotone increasing function such as a sigmoid function to the activation functions of the intermediate and output layers. In the error back propagation algorithm, learning is performed by minimizing the squared error between the output of the multilayer perceptron and a teacher signal. Stochastic gradient descent method [9] has long been used as an algorithm for minimizing the squared error. However, the stochastic gradient descent method has a problem that it tends to converge to a local solution because the same learning coefficients are used for all parameters. Many researchers propose many improvements have been proposed to solve this problem [10]-[14]. These improved methods have been widely used in recent years, and among them, Adam [14] provides highly accurate estimation results by bias-correcting the estimation of the first and second order moments of the gradient. However, since all of these methods minimize based on the gradient method, there is no guarantee that they will converge to the minimum solution if there are local solutions trapped by them. A MLP for 2-input 1-output exclusive OR can be realized with a very simple system consisting of 2 input layers, 2 intermediate layers, and 1 output layer. However, even with these gradient-based optimization methods, if suitable initial values are not given, the weight parameters that can correctly separate the inputs cannot be learned. It is reported that when a very large number of elements or layers exist in the intermediate layer, the existence of local minimum, plateaus, or saddle points has little effect on optimization by stochastic gradient descent due to the existence of the Star-convex Path [15]. However, when the number of elements in the middle layer is small, stochastic gradient descent and Adam cannot provide an optimal solution because the presence of local minimum, plateaus, and saddle points significantly affect the optimization. To reduce the size of MLP, it is necessary to reduce the number of elements in the intermediate layer. For this reason, we consider learning by a method that does not rely on the gradient methods. The MLP is a complex version of the simple perceptron structure. The simple perceptron is expressed as follows:

## 3. Learning by PSO

PSO is an optimization algorithm [3] that searches for parameters that provide the optimal value of the evaluation function in the parameter space, where multiple particles exchange information with each other. It is a meta-heuristic solution method that does not require the derivative of the evaluation function. PSO was used to learn the parameters of the MLP. In this study, we focused on linearly inseparable classification problems. We considered the following three types of linearly inseparable classification problems:1) 3D ExOR, 2) quadruple circles, and 3) MNIST. Each problem features different input and output dimensions: 3D ExOR is 3 to 2, quadruple circle is 2 to 4, and

MNIST is 2 to 10.

### 3.1 3D ExOR

The 3D ExOR problem is a linearly inseparable problem that classifies eight different inputs that add noise into two classes in three dimensions, as illustrated in Fig. 1.

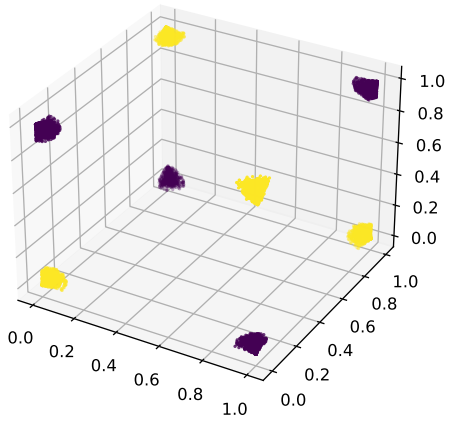


**Fig. 1.** 3D ExOR: Input is three-dimensional, and the number of labels is four.

### 3.2 Quadruple Circle

The quadruple circle problem is a linearly inseparable problem that classifies two-dimensional inputs into four classes as shown in Fig. 2.

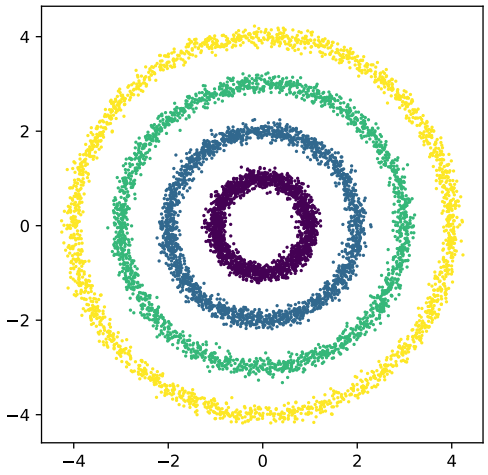


**Fig. 2.** Quadruple circle: Input is two-dimensional, and the number of labels is four.

### 3.3 MNIST

An auto encoder (AE) is used to map high-dimensional data to a low-dimensional feature space through nonlinear processing. We transformed a $28 \times 28 \times 1$ input grayscale image into a two-dimensional latent variable using an encoder, which constitutes an AE. We realize a classifier that

classifies 10 different classes from this two-dimensional latent variable into 10 outputs using the softmax function. The system was trained on 10 different handwritten numeral images from MNIST [16]. Consequently, we were able to build a classifier system that can identify 100% of the handwritten characters. The two-dimensional output of the encoder with the MNIST input images presents the distribution shown in Fig. 3 results in Fig. 3 indicate that the handwritten digit images in MNIST can be classified into clusters in a two-dimensional latent space. The two-dimensional latent variable space contained 10 clusters. The classifier performs discrimination between these clusters; however, these clusters are linearly inseparable. Therefore, we name the problem of classifying this two-dimensional data into 10 classes as “MNIST.”

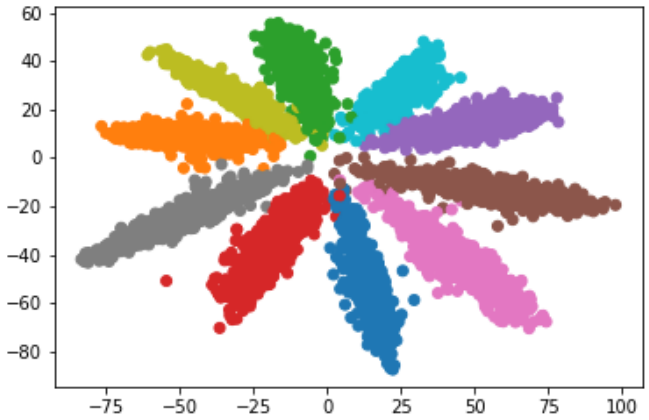


**Fig. 3.** MNIST:Input is two-dimensional, and the number of labels is 10.

### 3.4 Structure of MLP

We trained a three-layer MLP via PSO to classify the three types of linearly inseparable problems. For the 3D ExOR problem, the numbers of neurons in the input, middle, and output layers were three, two, and two, respectively. For the quadruple circle problem, the numbers of neurons in the input, middle, and output layers were two, two, and four, respectively. For the MNIST problem, the numbers of neurons in the input, middle, and output layers were two, two, and ten, respectively. For all problems, the ReLU function was applied as the activation function in the middle layer, and the softmax function was applied as the activation function in the output layer. In this study, we considered a case in which the connection coefficients between each layer and the bias value for each neuron were trained via PSO. For comparison, we considered a case where the gradient method was used for learning.

## 4. PSO

PSO is one of the most effective meta-heuristic optimization methods available. The original PSO was proposed by Eberhart and Kennedy (1995) [3]. The original PSO is expressed as follows:

$$\begin{aligned} V_i(t+1) &= wV_i(t) + c_1R_1(t)V_i^p(t) + c_2R_2(t)V_i^g(t) \\ X_i(t+1) &= X_i(t) + V_i(t+1) \\ V_i^p(t) &= Pbest_i(t) - X_i(t) \\ V_i^g(t) &= Gbest(t) - X_i(t) \end{aligned} \tag{1}$$

where $X_i(t)$ and $V_i(t)$ denote the location and the velocity of the $i$-th particle at $t$-th iteration, respectively; $Gbest(t)$ is the global best, which is the parameter that provides the best value of the evaluation function within the swarm at the t-th iteration; $Pbest_i(t)$ is the personal best, which is the parameter information that provides the best value of the evaluation function for the $i$-th particle at the $t$-th iteration; $V_i^p(t)$ and $V_i^g(t)$ are vectors toward $Pbest_i(t)$ and $Gbest(t)$, respectively; $w$

represents the inertia parameter; $c_1$ and $c_2$ are acceleration parameters; and $R_1(t)$ and $R_2(t)$ are time-variant random number parameters.

Although PSO is an extremely simple system, it can efficiently search for an optimal solution without the gradient information of the evaluation function. In practice, however, improving the accuracy of the solution is difficult, although the search in the vicinity of the optimal solution is rapid[17]-[19]. We also have proposed nonlinear map optimization (NMO) [18][20] to solve this problem by adding chaotic oscillating perturbations to each particle. Hence, we propose to improve PSO by perturbing the positional information of each particle. These methods aims to improve the ability of each particle to search in the neighborhood of the best solution identified and to solve the problem of all particles converging at the end of the search, which limits the scope of the search.

These methods are not specific to learning the weight parameters of MLPs, but are aimed at improving the ability to find the optimal solution to multimodal objective functions. Since the mean-square error between the output of an MLP that discriminates linearly inseparable inputs and the teacher signal is most often a multimodal function, we applied perturbation methods to find optimal solutions for such functions.

### 4.1 PSO_dist

To prevent the premature convergence of the PSO, perturbation is applied to the particles in the PSO [20]. The perturbation is expressed as shown in Eq. (2).

$$\begin{aligned} V_i(t+1) &= wV_i(t) + c_1R_1V_i^p(t) + c_2R_2V_i^g(t) \\ X_i(t+1) &= X_i(t) + V_i(t+1) + A(t+1)\cos\theta_i(t+1) \\ \theta_i(t+1) &= \theta_i(t) + 0.01 * \cos\theta_i(t) + 0.01 \\ A(t+1) &= A(t) + B \end{aligned} \tag{2}$$

where $\theta(t)$ represents the internal state variable, $A(t)$ is a function that controls the magnitude of the microperturbation, and $B$ is a parameter that controls the magnitude of the tangential motion such that $A(t)$ increases over time but does not become excessively large. $\theta(t)$ in Eq. (2) is a nonlinear map [18][20]. This nonlinear map generates a chaotic series with a distribution similar to the normal distribution, where $\theta(t)$ is onto an invariant interval of $[0, 2\pi]$ and the point $\theta = \pi$ gives the highest value. The point $\theta = \pi$ corresponds to the best point found so far. $A(t)\theta(t)$ in Eq. (2) is a small perturbation to search around it. The magnitude of the perturbation is determined by the parameter $A(t)$ but the parameter $B$ gradually increases the magnitude of the perturbation. In other words, it searches the vicinity of the best point, but its magnitude is gradually increased. This is intended to perform global search using the dynamics of the PSO itself, but as the search range of the PSO gradually shrinks, the search is performed while gradually expanding the neighborhood search range using micro perturbations in the nonlinear map. However, $A(t)$ is not monotonically increasing, but is initialized to 0 when Gbest updates.

### 4.2 PSO_q

PSO_dist improves local search capability by applying micro perturbations to PSO, but it not only applies micro perturbations but also is used by nonlinear dynamics effectively. To verify the effect of the nonlinear dynamics experimentally, we also use a system with simply normally distributed perturbations for comparison. We named this system PSO_q. The dynamics of the PSO_q is described by Eq. (3).

$$\begin{aligned} V_i(t+1) &= wV_i(t) + c_1R_1V_i^p(t) + c_2R_2V_i^g(t) \\ X_i(t+1) &= X_i(t) + R_3 + V_i(t+1) \\ R_3 &\sim N(0, 0.0625) \end{aligned} \tag{3}$$

where $R_3$ is a random parameter that follows a normal distribution. We set the standard deviation $\sigma$ of the normal distribution as $\sigma = 0.0625$ from preliminary experiments.

### 4.3 Previous studies

Previously, we searched the parameter space using only microperturbations. In this experiment, we set one particle to be searched and then observed it. The unimodal function was identified and shifted in a direction similar to that of the steepest descent. Subsequently, in the multimodal function, another local solution was warped. This is also meant to compare the results of the perturbation added on PSO_dist, which is based on the deterministic chaos introduced by NMO, with the results of simply adding a random perturbation.

## 5. Experiments

To compare the results of optimizing with PSO with those of optimizing with Adam, experiments are conducted on the three types of problems introcuced in Section 3. The evaluation function is a binary cross-entropy function between the one-hot representation output of MLP and the teacher signal of the one-hot representation label. Note that Adam uses the derivative of this evaluation function to search for the optimal solution, while PSO uses only the value of this evaluation function and not the derivative. All three kinds PSO experiments are conducted for the case of 16 particles. For the PSO parameters, we apply $w = 0.729$, $c_1 = c_2 = 1.494$, which is generally considered a suitable value, and we apply uniform distributions from 0 to 1 for $R_1$ and $R_2$. The initial values of $X_i(t)$ and $V_i(t)$ are set by random numbers from the z distribution. The initial value of PSO_dist parameter $A(t)$ and $B$ are set to 0.001 and 0.001, respectively. The initial value of the nonlinear map $\theta_i(0)$ is determined by a uniform distribution of $[0, 2\pi]$. The blue, orange, green and red lines represent the average of the learning results obtained using Adam, the conventional PSO, PSO_dist, and PSO_q, respectively. All experiments are conducted for 10 trials for each method and each parameter, and the mean and variance are obtained. The thin band represents the variance of each method. The results of the change in accuracy at each epoch during the learning are shown in Figs. 4–6. First, we performed

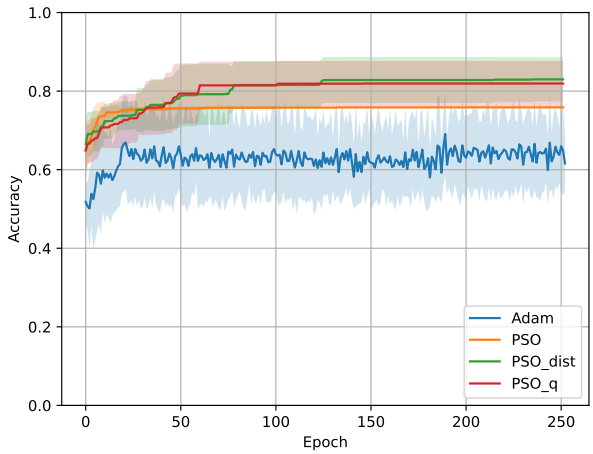


**Fig. 4.** Change in output accuracy at each training step when using "3D ExOR"

an experiment using 3D ExOR. The experimental results are shown in Fig. 4. The result of this experiment confirmed that PSO_dist yielded the best results, although the averages of all methods were similar. In this experiment, learning via the gradient method may be result in trapping by the local solutions in the first stage, and updating may not improve the accuracy. Next, we performed an experiment using a quadruple circle. The results of these experiments are shown in Fig. 5. In this experiment, we confirmed that PSO_dist yielded the best results. Using the gradient method, the value of the evaluation function fluctuated at each update, but the accuracy of the solution was not affected significantly.

Finally, we performed an experiment using MNIST. This problem can be regarded as learning the classifier of the MNIST handwritten digit recognizer using a CNN. We confirmed that the MNIST

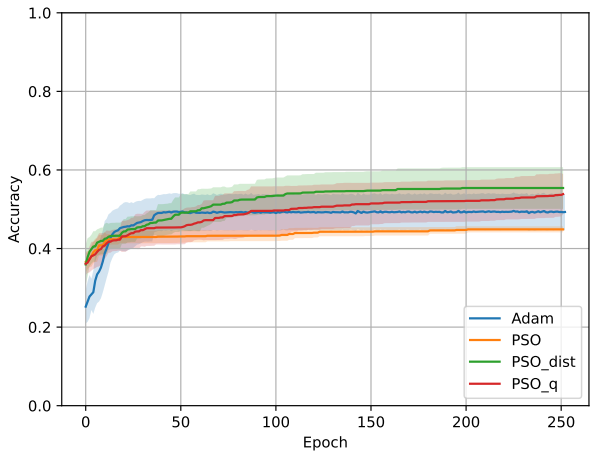


**Fig. 5.** Change in output accuracy at each training step when using "quadruple circle"

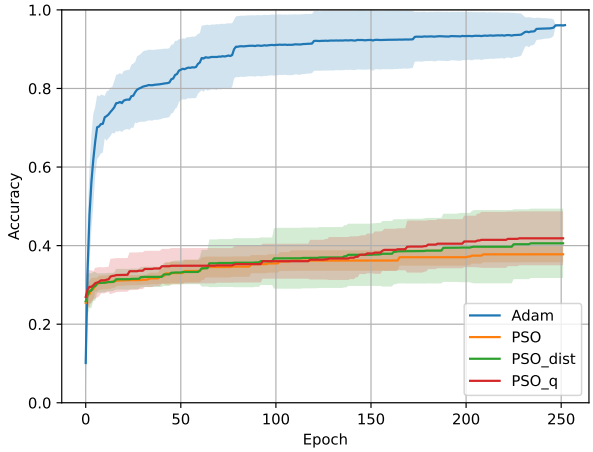


**Fig. 6.** Change in output accuracy at each training step when using "MNIST"

handwriting digit recognizer with a CNN can be trained using the gradient method and resulted in a discrimination accuracy of 100%. However, when only a classifier that classifies a two-dimensional latent variable dataset created for our MNIST problem into 10 classes was trained using the gradient method, a classification accuracy of 100% was not achieved. This problem has already been learned by Adam in creating the dataset; therefore, Adam can search easily but not accurately. We could not fully examine this result; therefore, we plan to analyze and clarify it in the future. These experimental results suggest that training with PSO on MLPs with a small number of elements in the middle layer provides better travel results than training with Adam based on the gradient method.

## 6. Conclusions

We used PSO to learn the parameters of an MLP comprising a few neurons. We confirmed that PSO_dist and PSO_q can learn the parameters of neural networks better than the conventional gradient method, such as Adam, when the number of neurons in the hidden layer is small and many local solutions are available for the loss function. In particular, we experimentally confirmed that micro-perturbations generated using nonlinear maps are more effective than simply applying normally distributed random numbers in the solution search. PSO exhibited the lowest variance; therefore, it was considered stable. However, it could not identify the best parameters. In addition, we confirmed that PSO can be improved via perturbation, as slight perturbations can allow certain local solutions

to be avoided.

## Acknowledgments

This study was supported by a Japan Society for the Promotion of Sience KAKENHI Grant-in-Aid for Scientific Research (C) (number 20K11978).

This study was partially conducted under the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University.

We would like to thank Editage (www.editage.com) for English language editing

## References

[1] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, Y. LeCun, "The Loss Surfaces of Multilayer Networks," Proc. AISTATS 2015, pp. 192-204, 2015. https://arxiv.org/abs/1412.0233

[2] Z. A-Zhu, Y. Li, Z. Song, "A Convergence Theory for Deep Learning via Over-Parameterization," Proc. ICML, PMLR vol. 97, pp. 242-252, 2019. https://arxiv.org/abs/1811.03962

[3] J. Kennedy, R. Eberhart, "Particle Swarm Optimization". Proc. ICNN 1995, pp. 1942–1948, 1995. doi:10.1109/ICNN.1995.488968

[4] Hue Yee Chong, Hwa Jen Yap, Shing Chiang Tan, Keem Siah Yap, Shen Yuong Wong, "Advances of metaheuristic algorithms in training neural networks for industrial applications," Soft Computing - A Fusion of Foundations, Methodologies and Applications, vol. 25, no. 16, pp. 11209–11233, 2021.

[5] Frank Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Review 65 (6): 386-408, 1958.

[6] Marvin Minsky, Seymour Papert, Perceptrons, MIT Press, 1969.

[7] Frank Rosenblatt, Frank, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Spartan Books, 1961.

[8] David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams, "Learning representations by back-propagating errors,". Nature 323 (6088) pp. 533–536, 1986.

[9] Shunichi Amari, "A Theory of Adaptive Pattern Classifiers," IEEE Trans. on Electronic Computers, vol. EC-13, 3, pp. 299-307, 1967.

[10] Yurii Nestero, "A method of solving a convex programming problem with convergence rate O(1/k2)," Soviet Mathematics Doklady 27, pp. 372–376, 1983.

[11] John Duchi, Elad Hazan, Yoram Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". The Journal of Machine Learning Research 12, pp. 2121-2159, 2011.

[12] Tijmen Tieleman; G. Hinton, Lecture 6.5 - rmsprop, COURSERA: Neural Networks for Machine Learning, 2012.

[13] Matthew D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," 2012.

[14] Diederik P. Kingma, Jimmy Ba, "Adam: A Method for Stochastic Optimization," in Proc. the 3rd International Conference for Learning Representations, ICLR 2016, pp.1-15, 2016.

[15] Yi Zhou, Junjie Yang, Huishuai Zhang, Yingbin Liang, Vahid Tarokh, "SGD Converges to Global Minimum in Deep Learning via Star-convex Path," in Proc. International Conference for Learning Representations, ICLR 2019, arXiv:1901.00451 [cs.LG], 2019.

[16] The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/

[17] Keiji Tatsumi, Hiroyuki Yamamoto, Tetsuzo Tanino, "A Perturbation Based Chaotic Particle Swarm Optimization Using Multi-type Swarms," SICE Annual Conference 2008, pp. 1199-1203, 2008.

[18] Kenya Jin'no, Tomoyuki Sasaki, Hidehiro Nakano, "Search Strategy Based on a Nonlinear Map Optimization," in Proc. 2019 International Conference of Nonlinear Theory and its Applications (NOLTA 2019), pp. 565 - 568, 2019.

[19] Stephen Chen, Imran Abdulselam, Naeemeh Yadollahpour, Yasser Gonzalez-Fernandez, "Particle Swarm Optimization with pbest Perturbations," in Proc. 2020 IEEE Congress on Evolutionary

Computation (CEC2020), E-24220, 2020.
[20] Kenya Jin'no, "Analysis of particle swarm optimization by dynamical systems theory," NOLTA, IEICE, vol. 12 no. 2 pp. 118-132, April, 2021.