2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

A Ring-type Growing Particle Swarm Optimizer for Identification of Multi-Solution

Takuya Takemura† and Toshimichi Saito†

†Department of Electronics and Electorical Engineering, Hosei University
3–7–2 Kajino-cho, Koganei, Tokyo 184-8584, Japan
Email: takuya.takemura.r5@stu.hosei.ac.jp,tsaito@hosei.ac.jp

Abstract—This paper studies the ring-type growing particle swarm optimizer (RGPSO) for multi-solution problems where the number of solutions is unknown. This algorithm uses ring-topology and has no random parameter. The number of particles can increase and the swarm can grow. The RGPSO can identify all the solutions and can clarify the number of solutions. The necessary number of particles depends on the number of solutions and is estimated based on the increasing number of particles.

1. Introduction

The particle swarm optimizer (PSO) is a population-based optimization method inspired by flocking behavior of living beings [1]-[3]. The particle positions correspond to potential solutions and is evaluated by an objective function. The particles search desired optimal solution(s) based on inter-particle communication. The PSO is simple in concept, is easy to implement and has been applied to optimization problems in various systems, e.g., signal processors, filters, switching power converters, renewable energy systems, and nonlinear dynamical systems [4]-[11].

For single solution problems, the PSO is suitable for global search. Because it can find an optimal solution by few particles even if the search apace is vast. However, standard PSOs are not suitable for multi-solution problems (MSP [12]-[16]) where particles are often trapped into partial/local solutions.

This paper studies ring-type growing particle swarm optimizer (RGPSO) for the MSP. Especially, we consider the case where the number of solutions is unknown. The RGPSO is defined on a particle swarm of ring-topology and the swarm can grow by generation of new particles. If parameter values are selected suitably, the RGPSO can identify all the approximate solutions. Also, the RGPSO includes no random parameters: it is deterministic. Such a deterministic system is convenient in motion analysis and reproducibility performance evaluation.

2. Algorithm

The objective function for the RGPSO is defined by

$$\begin{aligned} &F_A : S_A \rightarrow R_+, \\ &S_A = \{(x_1, x_2)|X_L \leq x_i \leq X_R, i = 1, 2\} \end{aligned} \tag{1}$$

where S_A is a search space and R_+ denotes positive reals. Assuming F_A has plural minima, the solutions x_s^i are defined by

$$\begin{aligned} &F_A(x_s^i) = 0, \\ &x_s^i \equiv (x_{s1}^t, x_{s2}^t) \in S_A, i = 1 \sim N_A \end{aligned} \tag{2}$$

where $i = 1 \sim N_A$ and N_A is the number of solutions. The RGPSO uses N particles. For the objective function F. The i-th particle P_i is characterized by its position x_i and velocity v_i. The update of the particle is based on the personal best ($Pbest_i$) and local best ($Lbest_i$). The $Pbest_i$ gives the best value in the past history of P_i. $Lbest_i$ is the best of the personal best in the neighbor of P_i. The neighbor particles are given depending on the structure of the particle swarms. We use the ring structure where the both sides particles are the neighbors of a particle. In order to defined the algorithm, let t be a search step and let P^t denote the particle swarm at time t. Let P_i^t be the i-th particle, let x_i^t be its position and v_i^t be its velocity where $i = 1 \sim N$.

In this paper, we assume that the number of solutions N_A is unknown. Our purpose is to identify positions of all the approximate solutions and to clarify the the number solutions. The RGPSO is defined as the following.

STEP 1 (Initialization 1): The number of approximate solutions is initialized: $k = 0$. The number of areas of approximate solutions is initialized: $S = 0$.
Let the number of particles be N.

STEP 2 (Initialization 2): Let search step $t = 0$. Particle positions x_i^t and velocities v_i^t are initialized where $i = 1 \sim N$. Personal bests and local bests are initialized: $\vec{x}_{pbest_i} = \vec{x}_{lbest_i} = \vec{x}_i^t$.

STEP 3 (Approximate solutions): If the i-th particle position satisfies

$$F(\vec{x}_i^t) < C_A \tag{3}$$

then x_i^t is declared as an approximate solution. The approximate solution is labelled by a_k. (If this is the first approximate solution then $S = 1$).

STEP 4 (Area judgement): If a_k is not included in an area of existing approximate solutions then a new area is generated.

$$S \leftarrow S + 1 \text{ if } |\vec{a}_k - \vec{a}_j| > r \text{ for } j < k \tag{4}$$

where $|\cdot|$ denote the Euclidean distance and the parameter r decides the approximate solution area. We have used the descending sort algorithm in the judgement. Let $k = k + 1$.

STEP 5 Personal and local bests are updated:

$$\vec{x}^t_{pbest_i} \leftarrow \vec{x}^t_i \qquad \text{if } F(\vec{x}^t_i) < F(\vec{x}^t_{pbest_i})$$

$$\vec{x}^t_{lbest_i} \leftarrow \vec{x}^t_{pbest_i} \qquad \text{if } F(\vec{x}^t_{pbest_i}) < (\vec{x}^t_{lbest_i})$$

Position and velocities are updated:

$$\begin{aligned} \vec{v}_i^{t+1} &\leftarrow w \times \vec{v}^t + c \times (\vec{x}^t_{lbest_i} - \vec{x}^t_i) \\ \vec{x}_i^{t+1} &\leftarrow \vec{x}^t + \vec{v}_i^{t+1} \end{aligned} \tag{5}$$

where w and c are deterministic parameters. Note that the RGPSO includes no random parameters.

STEP 6 (Increase of particles):

At $t = n_1 T_1$, N_1 pieces of new particles are added and are assigned randomly in the ring-topology, where n_1 denote integers and T_1 is a time interval.

$$N \leftarrow N + N_1 \text{ at } t = n_1 T_1$$

STEP 7 Let $t \leftarrow t + 1$, return to **STEP 3** and repeat until $t = t_{max}$. At $t = t_{max}$, go to STEP 8

STEP 8 If the number of solution regions is not change after repeating STEP 3 to STEP 7 M times then the algorithm is terminated. Otherwise, go to STEP 2.

3. Numerical Experiments

We have applied the RGPSO to MSPs defined by the following simple cost function

$$\begin{aligned} f_m(x_1, x_2) &= \cos\frac{4m\pi}{N}x_1 + \cos\frac{4m\pi}{N}x_2 + 2 \\ x_1 &\in \{-N, N\},\ x_2 \in \{-N, N\} \end{aligned} \tag{6}$$

where $N = 128$ and $m \in \{2, 3, 4, 5\}$ Depending on the parameter X, the search area and the number of solutions vary. For simplicity, we have selected four values of X as shown in Table 1. Figure 1 shows typical results. Table 1 summarize results of 100 trials.

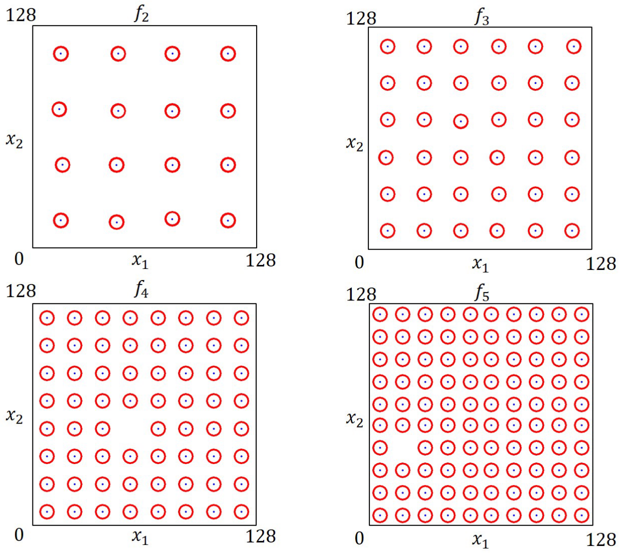

Figure 1: Typical search results

Table 1: Parameters and results. #SOL = the average number of solutions. #PLC = the average number of particles. #ASL = the average number of identified solutions.

X	2	3	4	5
#SOL	16	36	64	100
#PCL	73.5	206.5	328	713
#ASL	15.5	35.45	63.55	98.65

4. Conclusion

We have studied the RGPSO for MSPs where the number of solutions is unknown. Performing numerical experiments for fundamental MSPs, the algorithm efficiency is investigated.

References

[1] J. Kennedy and R. Eberhart, Particle swarm optimization, Proc of IEEE/ICNN, pp.1942-1948, 1995

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

[3] E. Miyagawa and T. Saito, Particle swarm optimizers with grow-and-reduce structure, Proc. of IEEE/CEC, pp. 3975-3980, 2008.

[4] S.-T. Hsieh, T.-Y. Sun, C.-L. Lin, and C.-C. Liu, Effective learning rate adjustment of blind source separation based on an improved particle swarm optimizer, IEEE Trans. Evol. Comput., 12, 2. pp. 242-251, 2008.

[5] R. A. Vural, T. Yildirim, T. Kadioglu and A. Basargan, Performance evaluation of evolutionary algo-

rithms for optimal filter design, IEEE Trans. Evol. Comput., 16, 1. pp. 135-147, 2012.

[6] H. Qin, J. W. Kimball and G. K. Venayagamoorthy, Particle swarm optimization of high-frequency transformer, Proc. Annual Conf. IEEE Ind. Electron. Soc., pp. 2908-2913, 2010.

[7] K. Ono and T. Saito, Application of particle swarm optimizers to two-objective problems in design of switching inverters, Proc. of IEEE-INNS/IJCNN, pp. 2353-2357, 2009.

[8] K. Kawamura and T. Saito, Design of switching circuits based on particle swarm optimizer and Hybrid Fitness Function, Proc. IEEE/IECON, pp. 1099-1103, 2010.

[9] H. Matsushita and T. Saito, Application of particle swarm optimization to parameter search in dynamical systems, NOLTA, IEICE, 2, 4, pp. 458-471, 2011.

[10] N. Ando, K. Tanakajima, and T. Saito, Switched ring particle swarm optimizer for trace of time-variant maximum power point, Proc. IEEE/SMC, 2015.

[11] K. Maruyama and T. Saito, Collision particle swarm optimizers and exploring periodic points, NOLTA, IEICE, 5, 4, pp. 523-534, 2014

[12] K. E. Parsopoulos and M. N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Trans. Evol. Comput., 8(3), pp. 211–224 (2004)

[13] S. Yang and C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, IEEE Trans. Evol. Comput., 14, 6. pp. 959-974, 2010.

[14] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Trans. Evol. Comput., 14(1), pp. 150–169, 2010.

[15] K. Jin'no, R. Sano, and T. Saito, Particle swarm optimization with switched topology NOLTA, IEICE, 6, 2, pp. 181-193, 2015

[16] T. Takemura, T. Sato, K. Maruyama, and T. Saito, A Growing insensitive particle swarm optimizer for identification of multi-solutions, Proc. NOLTA, pp. 140-143, 2014