2008년도 대한전기학회 전기기기 및 에너지변환시스템부문회 춘계학술대회 논문집 (2008. 4. 25 ~ 26)

Couple Particle Swarm Optimization for Multimodal functions

Minh-Trien Pham, Nyambayar Baatar, 고창섭
충북대학교 전기공학과

Couple Particle Swarm Optimization for Multimodal Functions

Minh-Trien Pham, Nyambayar Baatar, Chang Seop Koh
School of Electrical & Computer Engineering, Chungbuk National University

Abstract - This paper proposes a new couple particle swarm optimization (CPSO) for multimodal functions. In this method, main particles are generated uniformly using Faure-sequences, and move accordingly to cognition only model. If any main particle detects the movement direction which has local optimum, this particle would create a new particle beside itself and make a couple. After that, all couples move accordingly to conventional particle swarm optimization (PSO) model. If these couples tend toward the same local optimum, only the best couple would be kept and the others would be eliminated. We had applied this method to some analytic multimodal functions and successfully locate all local optima.

1. Introduction

Nowadays, multimodal functions optimization problem are drawn not only in the research community but also in many application fields.

The classical numerical optimization including genetic algorithm, evolutionary algorithm and particle swarm optimization have been proven to be very effective techniques. Especially, by finding a single optimum solution, PSO algorithm has brought more advantages than the other. However, the original optimization algorithms are less effective in locating more than one optimum without any modifications.

This paper briefly reviews the basic particle swarm algorithm and then we will introduce a new method that can locate all optima. Finally is about our results when we apply this method for some test multimodal functions.

2. Particle Swarm Optimization

The PSO algorithm was proposed by James Kennedy and Russell C.Eberhart in 1995 [1]. It is a stochastic optimization technique that is similar to the behavior of a flock of birds or the sociological behavior of a group of people. The PSO is a population based optimization technique, where the population is called a swarm. A simple explanation of the PSO's operation is the following: Each particle represents a possible solution of the optimization task. During movement of particles, each particle accelerates in the direction of its own particle best solution as well as in the direction of the global best position discovered by any particles in the swarm. This means that if a particle discovers a promising new solution, all the other particles will move closer to it, explorer the region more thoroughly in the process.

For a maximization problem, the algorithm is explained as follows: A swarm of n particles in search space of d-dimension are generated randomly. For each particle i its velocity vector $\mathbf{v}_i$ is updated at iteration t according to the equation (1).

$$\mathbf{v}_i(t+1)=\omega\cdot\mathbf{v}_i(t)+c_1\cdot r_1\cdot\left(\mathbf{p}_i(t)-\mathbf{x}_i(t)\right) \\ +c_2\cdot r_1\cdot(\mathbf{g}(t)-\mathbf{x}_i(t)) \quad (1)$$

where $\mathbf{p}_i$ and $\mathbf{g}_i$ are the particle best position and global best position i-th respectively. The current position of the particle is denoted by $\mathbf{x}_i$. The inertia weight $\omega>0$ controls the influence of the previous velocity. Parameter $c_1>0$ controls the effect of the personal best position. Parameter $c_2>0$ determines the effect of the best position that has been found by any of the particles in the swarm. Usually c_1 and c_2 are set to the same value. Random values r_1 and r_2 are drawn with uniform probability from [0,1]. After velocity update the particles move with their new velocity to their new positions (2).

$$\mathbf{x}_i(t+1)=\mathbf{x}_i(t)+\mathbf{v}_i(t+1) \quad (2)$$

Then for each particle i, the objective function $f(.)$ is evaluated at its new position. The personal best $\mathbf{p}_i$ is updated according this formula:

$$\mathbf{p}_i(t+1)=\mathbf{p}_i(t) \quad \text{if } f(\mathbf{x}_i(t+1))\le f(\mathbf{p}_i(t)) \\ =\mathbf{x}_i(t+1) \quad \text{if } f(\mathbf{x}_i(t+1))> f(\mathbf{p}_i(t)) \quad (3)$$

Finally, the global best of the swarm is updated as follows:

$$\mathbf{g}(t+1)=\arg\min_i f(\mathbf{p}_i(t+1)) \; i=1,\cdots,n \quad (4)$$

Beside the conventional PSO model was introduced above. There are two difference PSO models, known as the cognition only model and social only model. The velocities of them are expressed as follow respectively:

$$\mathbf{v}_i(t+1)=\omega\cdot\mathbf{v}_i(t)+c_1\cdot r_1\cdot(\mathbf{p}_i(t)-\mathbf{x}_i(t)) \quad (5)$$

$$\mathbf{v}_i(t+1)=\omega\cdot\mathbf{v}_i(t)+c_2\cdot r_2\cdot(\mathbf{g}_i(t)-\mathbf{x}_i(t)) \quad (6)$$

The cognition only model only takes into account the particle's own experiences. Kenedy found that the performance of this model was inferior that of the conventional PSO model. And the performance of social only model was superior to that of conventional PSO on the specific problem [2].

3. Couple Particle Swarm Optimization

In this section, the couple PSO algorithm can locate all optima for multimodal function is discussed. There are n main particles were assigned uniformly. All main particles aim to define the direction which has local optima. After that a couple of particles will locate the local optimum exactly. The scheme of couple PSO algorithm is shown in Fig 1. All issues are now discussed:

---

1. Initialization.
2. Main particles searching.
3. Update fitness for main particles.
4. Create a new Couple.
5. For each Couple:
    5.1. Move using PSO
    5.2. Update each particle's fitness and the couple best
    5.3. Update Couple radius.
6. If possible, eliminate main particles and couples.
7. Repeat from step 2 until stopping conditions are reached.

Fig.1. CPSO algorithm

Initialization: At the first step, the main particles are generated random positions within the search space. To ensure uniform distribution, Faure-sequences were used to generate initial particle positions. Faure-sequences are distributed with high uniformly within a d-dimensional unit cube [3].

Main particles searching: As explain in section 2, the performance of cognition only model was inferior that of conventional PSO as well as social only model in locating global best. But the cognition only model has advantage in locating local optimum. Further more, we aim to locate local optimum as many as possible. That's why the main particles are trained using the cognition only model. Even both the social only model and conventional PSO model was proven that they move toward optimum faster than cognition only model. The velocity of main particle i-th is defined by equation (5).

$$\mathbf{v}_i(t+1)=\omega\cdot\mathbf{v}_i(t)+c_1\cdot r_1\cdot(\mathbf{p}_i(t)-\mathbf{x}_i(t)) \quad (5)$$

And the position at the next iteration should be:

$$\mathbf{x}_i(t+1)=\mathbf{x}_i(t)+\mathbf{v}_i(t+1) \;\; i=1,\cdots,n \quad (2)$$

where $\mathbf{x}_i$, $\mathbf{v}_i$, $\mathbf{p}_i$ describe position, velocity and particle best of main particle i-th respectively. To distinguish main particle and the particle in couple, we emphasize that there are not super scripts on symbols which show position, velocity and particle best of main particle.

Update fitness for main particles: Each main particle update itself fitness, if the position of the next iteration is better than previous one. The best position of this particle should be the next position other wise the best particle position is the current best position.

$$\begin{aligned}\mathbf{p}_i(t+1)&=\mathbf{x}_i(t+1) \;\text{ if }\; f(\mathbf{x}_i(t+1))>f(\mathbf{p}_i(t))\\ &=\mathbf{p}_i(t) \;\text{ if }\; f(\mathbf{x}_i(t+1))\le f(\mathbf{p}_i(t))\end{aligned} \quad (7)$$

where $f(\mathbf{p}_i(t))$ is the function value of particle best i-th at iteration t.

Identification local optimum: To define the direction which has local optimum, a prediction method was used. A main particle can move toward local optimum (Fig 2) or opposite direction (Fig 3). If this particle moves toward local optimum, we can conclude that following that direction, the local optimum will be found. That mean this method can predict the local optimum far from the exact value.

Create a new Couple: As discussion in section 2, the performance of conventional PSO was superior to that of cognition only model in locating optimum. To take full advantage of conventional PSO, a new particle is created beside the main particle which defined the local optimum direction. The new particle should be closely with the main particle in this couple. The distance between them should not so far and not so closely. In our experience, this distance δ equal 10^{-3} for all analytic functions in this paper. Fig 4 shows a new couple was created.

Couple moving: The conventional PSO was applied for couples as follow. Equation (8) shows the velocity update of each particle in couple k-th.

$$\begin{aligned}\mathbf{v}_j^k(t+1)=\omega\cdot\mathbf{v}_j^k(t)+c_1\cdot r_1\cdot(\mathbf{p}_j^k(t)-\mathbf{x}_j^k(t))\\ +c_2\cdot r_2\cdot(\mathbf{c}^k(t)-\mathbf{x}_j^k(t))\end{aligned} \quad (8)$$

and each particle updates position:

$$\mathbf{x}_j^k(t+1)=\mathbf{x}_j^k(t)+\mathbf{v}_j^k(t+1) \quad (9)$$

where $\mathbf{x}_j^k(t)$, $\mathbf{v}_j^k(t)$, $\mathbf{p}_j^k(t)$ are position, velocity and particle best of j-th particle in couple k-th at iteration t. And the best position in couple k-th at iteration t is $\mathbf{c}^k(t)$. The particle best and couple best are updated at next step.

Update particle's fitness and the couple best: Like the update fitness in conventional PSO, the particle best will be selected as follow:

$$\begin{aligned}\mathbf{p}_j^k(t+1)&=\mathbf{x}_j^k(t+1) \;\text{ if }\; f(\mathbf{x}_j^k(t+1))>f(\mathbf{p}_j^k(t))\\ &=\mathbf{x}_j^k(t) \;\text{ if }\; f(\mathbf{x}_j^k(t+1))\le f(\mathbf{p}_j^k(t))\end{aligned} \quad (10)$$

And the couple best will be the better particle position in couple:

$$\mathbf{c}^k(t+1)=\arg\min_j f(\mathbf{p}_j^k(t+1)) \;\; j=1,2 \quad (11)$$

Update couple radius:

$$R^k=\text{distance}(x_1^k,x_2^k);\text{distance}(x_1,x_2)=\|x_1-x_2\| \quad (12)$$

where R^k is couple radius of k-th couple. Because there are two particles in couple, so radius will be distance between them and the center of k-th couple is $\mathbf{c}^k$.

Eliminate particles:

$$\text{distance}(\mathbf{c}^k,\mathbf{x}_i)<R^k \quad (13)$$

If any main particle moves inside couple area, or distance between couple best $\mathbf{c}^k$ and main particle $\mathbf{x}_i$ is smaller than radius of k-th couple (13), this main particle should be eliminated. Because if a new couple is created by this main particle which will locate the same local optimum with the k-th couple.

Eliminate couples:

$$\text{distance}(\mathbf{c}^k,\mathbf{c}^m)<(R^k+R^m) \quad (14)$$

If the distance between of any two couple best is smaller than total radius of them as equation (14), the couple has worse function value will be eliminated. Eliminate method is applied to reduce number of function call, that means it reduces computation time.

Stopping conditions: There are two stop conditions, Maximum iteration equal 2000 and all couple radii smaller than epsilon after 3 iterations. Actually, epsilon represents how accuracy of results. In this paper, epsilon equal 10^{-6} was chosen. If one of them is reached, the program will stop.

4. Experimental Results

In this section, we apply the CPSO algorithm to finding maxima of multimodal functions in 2-dimension (2D). From Fig 6 to 9 show how particles move and locate optimum for $F_1(x)$ in 2D design space. At the first figure, $n = 16$ initial main particles was generated in range [-1,1]. After several iterations, the number of couple will increase as Fig 7. But the number of couple will decrease when they move toward optima. Finally, there are 4 couples existence which relate with 4 optima found in Fig 9.

Our target is to find all maxima of the following functions.

$$F_1(x) = \sum_{i=1}^{N}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right) \quad N = 2 \tag{15}$$

$$F_2(x,y) = 200 - \left(x^2 + y - 11\right)^2 - \left(x + y^2 - 7\right)^2 \tag{16}$$

$$F_3(x,y) = \sum_{k=1}^{m} \frac{b_k}{1 + \left((x - x_{pk})^2 + (y - y_{pk})^2\right)/a_k} \tag{17}$$

$$F_4(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad N = 2 \tag{18}$$

$$F_5(x,y) = 900 + (x-5)^2 + 10\cos(2\pi(x-5)) + (y-5)^2 + 10\cos(2\pi(y-5)) \tag{19}$$

Both $F_1(x)$ and $F_2(x)$ have 4 global optima, with function value of optima of $F_1(x)$ equal 40.503059 in range [-1,1] and function value of optima of $F_2(x)$ equal 200 in range [-5,5]. Our method can locate all maxima of both functions by using 9 main particles, $c_1 = c_2 = 1.5$, inertia weight $\omega = 0.77$.

Unlike $F_1(x)$ and $F_1(x)$, $F_3(x)$, $F_4(x)$ and $F_5(x)$ have many optima with difference function values of optima. $F_3(x)$ has 7 optima in range [-60,60], $F_4(x)$ has 18 optima in range [-10,10] and $F_5(x)$ is Rastrigin modified function which has 25 optima in range [3,7].

Summarizes of 5 functions are shown in Table I. For each of 5 functions, 100 experiments were done with CPSO algorithm.

5. Conclusion

This paper introduced a new algorithm based on conventional PSO algorithm which called CPSO. The CPSO algorithm can locate multiple optima solutions for multimodal optimization problems. Our experiment result shows that the new algorithm successfully located all optima for all the simulation runs.

[Reference]

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in Proc. IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 19421948.

[2] J.Kennedy, The particle swarm: Social adaptation of knowledge, in Proceedings of the International Conference on Evolutionary Computation, p 303308, 1997.

[3] E. Thi ′ emard Economic Generation of Low-Discrepancy Sequences with a b-ary Gray Code, Department of Mathematics, Ecole Polytechnique F ′ ed ′ erale de Lausanne, CH-1015 Lausanne, Switzerland

TABLE I
VARIABLE RANGES AND VAUES USED

Functions	Number of main particle	Range	Number of optimum found	% found
$F_1(x)$	9	[-1.1]	4	100%
$F_2(x)$	9	[-5,5]	4	100%
$F_3(x)$	16	[-60,60]	7	100%
$F_4(x)$	25	[-10,10]	18	100%
$F_5(x)$	64	[3,7]	25	100%

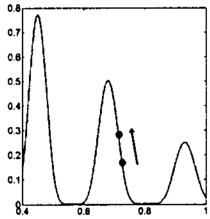
Fig.2. Main particle move up

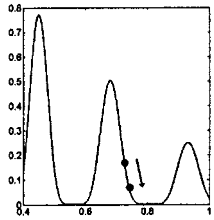
Fig.3. Main particle move down

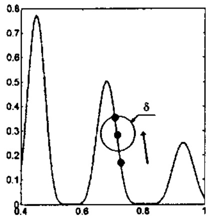

Fig.4. Creating a new couple

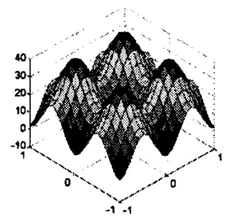
Fig.5. Function $F_1(x)$

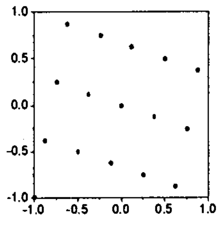
Fig.6. Initial main particles

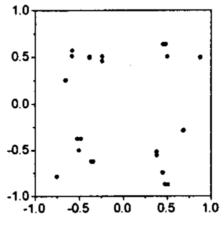
Fig.7. Iteration 50

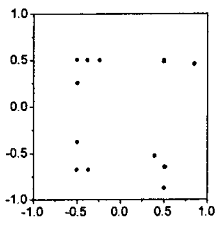
Fig.8. Iteration 100

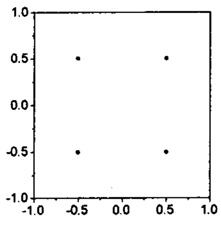
Fig.9. Converged