IEICE Proceeding Series

A hybrid Algorithm based on Particle Swarm Optimization and Differential Evolution for Global Optimization Problems

Jun-ichi Kushida, Akira Hara, Tetsuyuki Takahama

Vol. 2 pp. 201-204
Publication Date: 2014/03/18
Online ISSN: 2188-5079

2013 International Symposium on Nonlinear Theory and its Applications
NOLTA2013, Santa Fe, USA, September 8-11, 2013

A hybrid Algorithm based on Particle Swarm Optimization and Differential Evolution for Global Optimization Problems

Jun-ichi Kushida†, Akira Hara† and Tetsuyuki Takahama†

†Department of Intelligent Systems Hiroshima City University
3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima, 731-3194 Japan
Email: kushida@hiroshima-cu.ac.jp

Abstract—Particle swarm optimization (PSO) and differential evolution (DE) are the evolutionary algorithms, and both have been successfully applied to various optimization problems. In this paper, we propose a hybrid algorithm based on PSO and DE. The design of the proposed method is mainly constructed from the algorithm of PSO, and DE operator is used supplementarily. In order to preserve diversity of population and have a rotation-invariant, we modify velocity update equation and add perturbation by difference vector. In addition, we introduce the uneven movement of particles, by updating position of some particles in the swarm. The proposed method was compared with basic PSO and DE. The simulation results showed the effectiveness of the proposed method on several test functions.

1. Introduction

Evolutionary algorithms (EA), which include population based stochastic search and optimization algorithms, are based on the natural evolution principles. As a typical EA, Genetic algorithm (GA), Particle swarm optimization (PSO) and Differential evolution (DE) are have been successfully applied to various problems including test and real world problems. PSO is based on a social behavior of bird flocking or fish schooling. The population is called a swarm and the individuals are called particles. Each particle represents a potential solution to the problem expressed by the objective function. DE was proposed for optimization with continuous variables. The structure of DE is similar to GA. As a genetic operator, mutation and crossover are used to generate new candidate solution. In mutation, distance and direction information from the current population is used. DE and PSO have been successfully applied to various problems including test and real world problems.

However, like most other stochastic optimization techniques DE and standard PSO often confront premature convergence and converge into a local minimum in multimodal function. In addition DE's crossover operation (binomial and exponential crossover) is not a rotation-invariant [7]. Rotation does not alter the objective function topology, however, the parameter becomes dependent. Thus, dependance between parameters caused by rotation degrades the performance of DE. In order to improve the performance of these algorithms, many version of DE and PSO have been developed. Among these improved methods, hybrids between PSO and DE have been showed better performance. These methods combine the advantages of the two approaches. DE and PSO's hybrid algorithm could be classified into two categories. The one is utilizing DE to improve PSO [3, 4]. the other is utilizing PSO to improve DE [5].

In this paper, we propose a hybrid algorithm of DE and PSO. Our proposed algorithm is based on PSO, and DE operator is used supplementarily to enhance the performance of PSO. The proposed method emphasizes an ability of global search and a rotation invariant. Due to this, velocity update equation is modified and perturbation by difference vector is added. In addition, we introduce the uneven movement of particles by updating position of a part of particles. As a result, the distribution of particles become spread and diversity of swarm is preserved.

2. Particle swarm optimization

In the PSO algorithm, each particle has two information; position and velocity. Algorithm 1 shows pseudocode of PSO. For N_{D}-dimensional search space, the position and velocity for the i-th particle is represented as $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \cdots, x_{iN_{\mathrm{D}}}]$ and $\boldsymbol{v}_i = [v_{i1}, v_{i2}, \cdots, v_{iN_{\mathrm{D}}}]$, respectively. The best previous position of the i-th particle is recorded and presented as $\boldsymbol{p}_i = [p_{i1}, p_{i2}, \cdots, p_{iN_{\mathrm{D}}}]$ which is also called pbest. The index of the best pbest is represented by the symbol g, and $\boldsymbol{p}_g$ is called gbest.

At k iteration step, the velocity and position update equations for the j-th dimension of the i-th particle in the swarm are given as follows:

$$\begin{aligned} v_{ij}^{k+1} &= wv_{ij}^{k} + c_1 rand_1(p_{ij} - x_{ij}) \\ &+ c_2 rand_2(p_{gj} - x_{ij}) \end{aligned} \quad (1)$$

$$x_{ij}^{k+1} = x_{ij} + v_{ij}^{k+1} \quad (2)$$

where, $rand_1$ and $rand_2$ are uniformly distributed numbers between 0 and 1, c_1 and c_2 are positive constants referred to as cognitive and social paramters respectively, w is called inertia weight.

Algorithm 1 PSO

```
Generate population and velocity {x_i | i = 1, 2, ..., N_P}, {v_i | i = 1, 2, ..., N_P}
for k = 1 to K_max do
  for i = 1 to N_P do
    Update the velocity of v_i by Eq. (1);
    Update the position of x_i by Eq. (2);
    Evaluate x_i

    /*Update pbest and gbest*/
    if f(x_i) < f(p_i) then
      p_i = x_i
      if f(x_i) < f(p_g) then
        g = i
      end if
    end if

  end for
end for
```

3. Differential evolution

DE is a population-based stochastic search technique for solving optimization problems in a continuous space. In the DE algorithm, at the beginning all members of the population are initialized randomly. At each generation, new candidate solutions are generated by genetic operator. For each parent (target vector) $\boldsymbol{x}_i$, mutant vector $\boldsymbol{v}_i$ is generated by the following equation:

$$\boldsymbol{v}_i = \boldsymbol{x}_{r_1,i} + F(\boldsymbol{x}_{r_2,i} - \boldsymbol{x}_{r_3,i}) \quad (3)$$

where, r_1, r_2, r_3 are randomly chosen integers, must be different from each other and also different from the running index i. F is the scaling factor which controls the mutation step size. The scaling factor influences the perturbation applied to the base vector $\boldsymbol{x}_{r_1,i}$ and has an important role in ensuring the population diversity. Next, crossover operator combines component of target vector and mutant vector and generates trial vector $\boldsymbol{u}_i$. Where $C_R \in [0, 1]$ is an input parameter influencing the number of elements to be exchanged by crossover. In the selection for survival, the objective function value of $\boldsymbol{x}_i$ with $\boldsymbol{u}_i$ are compared and better one is selected to next generation.

4. Proposed method

The proposed method as mentioned earlier is a hybrid algorithm of DE and PSO. The construction of the proposed method is based on PSO. The pseudo code of the proposed method is shown in Algorithm 2. In order to preserve diversity of population of PSO, we add perturbation using difference vector to velocity update equation. Additionally, the uneven movement of particles is introduced. At each iteration, moving particles, which are selected with a probability of moving (p_m), update velocity and position. Remaining particles, called stopping particles, update only their velocity. Due to this, the distribution of swarm become uneven and swarm can avoid too fast convergence.

Algorithm 2 Proposed method

```
Generate population and velocity {x_i | i = 1, 2, ..., N_P}, {v_i | i = 1, 2, ..., N_P}
for k = 1 to K_max do
  for i = 1 to N_P do
    if p_m > rand(0, 1) then
      Update the velocity of v_i using Eq. (5);
      Update the position of x_i by Eq. (2);
      Evaluate x_i

      /*Update pbest*/
      if f(x_i) < f(p_i) then
        p_i = x_i
      end if

    else
      Update the velocity of v_i by Eq. (6);
    end if
  end for
end for
```

If particle i is selected as a moving particle, a difference vector $\boldsymbol{\delta}$ is calculated as follow:

$$\boldsymbol{\delta} = \boldsymbol{p}_{r1} - \boldsymbol{p}_{r2} \quad (4)$$

where, r_1 and r_2 are selected randomly in a similar way of DE. Next, the velocity for the j-th dimension of the i-th particle is updated as follow:

$$\begin{aligned} v_{ij}^{k+1} &= w v_{ij}^{k} + c_1(p_{ij} - x_{ij}) \\ &+ c_2(p_{tj} - x_{ij}) + F\delta_j \end{aligned} \quad (5)$$

where, $t \in \{1, 2, \cdots, N_P\}$ is selected by tournament selection. Basic PSO uses gbest in third term, however, we consider that usage of gbest becomes a cause of too fast convergence. Therefore, by selecting the particle of the third term using tournament selection, the convergence speed of particles toward the best solution is reduced. Additionally, different from equation (1), $rand_1$ and $rand_2$ are not used. Instead, the difference vector is added as the fourth term in order to perturb velocity of particles.

Stopping particles update only velocity by following equation:

$$v_{ij}^{k+1} = w_{ac} v_{ij}^{k} \quad (6)$$

where, $w_{ac}(> 1)$ is a parameter for acceleration of velocity. In the proposed method, by acceleration velocity of some particles, distribution of particles is expanded and diversity is preserved.

Table 1: Definition of test functions

Name	Expression	Domain	Structure
Sphere	$f(\boldsymbol{x}) = \sum_{i=1}^{N_D} x_i^2$	$[-5.12 \le x_i \le 5.12]^{N_D}$	unimodal
k-tablet (k=$N_D/2$)	$f(\boldsymbol{x}) = \sum_{i=1}^{k} x_i^2 + \sum_{i=k+1}^{N_D} (100x_i)^2$	$[-5.12 \le x_i \le 5.12]^{N_D}$	unimodal, ill-scale
Rastrigin	$f(\boldsymbol{x}) = 10N_D + \sum_{i=1}^{N_D} x_i^2 - 10\cos(2\pi x_i)$	$[-5.12 \le x_i \le 5.12]^{N_D}$	strongly multimodal
Rosenbrock	$f(\boldsymbol{x}) = \sum_{i=2}^{N_D} \{100(x_1 - x_i^2)^2 + (x_i - 1)^2\}$	$[-2.048 \le x_i \le 2.048]^{N_D}$	nonlinear ridge
Schaffer	$f(\boldsymbol{x}) = \sum_{i=1}^{N_D-1} (x_i^2 + x_{i+1}^2)^{0.25} \times (\sin^2(50(x_i^2 + x_{i+1}^2 + 1.0))^{0.1})$	$[-5.12 \le x_i \le 5.12]^{N_D}$	strongly multimodal
Griewank	$f(\boldsymbol{x}) = 1 + \sum_{i=1}^{N_D} \frac{x_i^2}{4000} - \prod_{i=1}^{N_D} \left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$	$[-512 \le x_i \le 512]^{N_D}$	strongly multimodal

5. Experiment

5.1. Setup

In order to confirm the performance of the proposed method, we compare the proposed method, PSO and DE on standard test functions with the number of dimensions N_D= 20. Table 1 shows a list of test functions. All functions are chosen for the minimization problems and their optimal values are all 0. Table 2 shows control parameters of the proposed method, PSO and DE. In the proposed method, the parameters are detenumined by preliminary experiment and the tournament size is 2. The parameter setting of PSO is based on [6] and DE strategy is DE/rand/1/exp. In order to validate the effectiveness of C_R and p_m, several value was used. C_R was viried across the set $\{0.7, 0.9, 0.95\}$ and p_m was varied across the set $\{0.60, 0.65\}$.

Table 2: Control parameters

	N_P	c_1	c_2	w	w_{ac}	F
Proposed method	500	0.25	0.5	0.4	1.8	0.3
PSO	50	1.4955	1.4955	0.729		
DE	50					0.7

Each experiment for the function is carried out 50 times. When the best fitness of the population has become less than 10^{-7}, we assume that an optimal solution has been found. The termination criteria is 5×10^7 fitness evaluations in all functions.

5.2. Result

Table 3 shows the experimental results of proposed method, PSO and DE on some standard test functions. The top row shows the number of function evaluations until finding a near optimal value. The bottom row shows the number of runs (out of 50) that found the optimal solution. In this table, RT means 30 degree coordinate rotation.

PSO tends to converge into a local minimum, and failed to find the optimal solution except of sphere function and k-tablet function. DE is faster than other algorithms in multimodal functions. However, in whichever C_R, DE can not success completely in all trials. In rotate functions, the convergence speed and the success rate of DE degrade because crossover is not a rotation-invariant [7]. When C_R is a high value, population tend to move at angles to the search space's axes. Therefore, it is effective for problems which have dependency problem between the variables. Meanwhile, a higher C_R reduces population diversity rapidly and DE tend to get stuck in local minimum. Due to this, adequate setting of C_R is difficult in DE.

The proposed method using p_m=0.60 finds optimal solutions completely in all functions. Additionally, we confirm that in all rotated functions the convergence speeds are not degrade campared to not rotated functions. However, the convergence speed of proposed method is slow because many particles are required for stable convergence. When p_m is 0.65, the convergence speed become faster in all functions. Especially, in the Rosenbrock function, the Schaffer (RT) function and the Griewank (RT), the proposed method is faster than other methods. In contrast, the population diversity decreases and several trials failed in the Rastrigin function. From these results, a certain number of stopping particles are required in order to stable convergence.

Table 3: Comparison of the proposed method, PSO and DE

	Proposed method		PSO	DE		
	p_m=0.60	p_m=0.65		C_R=0.7	C_R=0.9	C_R=0.95
Sphere	9.13E+04 (50/50)	6.66E+04 (50/50)	**1.32E+04** (50/50)	3.07E+04 (50/50)	4.00E+04 (50/50)	5.10E+04 (50/50)
k-tablet	1.30E+05 (50/50)	8.05E+04 (50/50)	**2.02E+04** (50/50)	3.94E+04 (50/50)	4.90E+04 (50/50)	6.11E+04 (50/50)
Rastrigin	1.11E+07 (50/50)	4.17E+06 (48/50)	(0/50)	5.89E+04 (50/50)	1.10E+05 (50/50)	**1.80E+05** (50/50)
Rastrigin (RT)	9.39E+06 (50/50)	4.03E+06 (48/50)	(0/50)	1.42E+06 (49/50)	4.60E+05 (50/50)	**9.80E+03** (20/50)
Rosenbrcok	1.66E+05 (50/50)	**1.39E+05** (50/50)	(0/50)	1.41E+06 (50/50)	2.70E+05 (50/50)	1.70E+05 (50/50)
Schaffer	4.79E+05 (50/50)	3.37E+05 (50/50)	(0/50)	**1.30E+05** (50/50)	1.77E+05 (50/50)	2.45E+05 (50/50)
Schaffer (RT)	4.79E+05 (50/50)	**3.37E+05** (50/50)	(0/50)	9.12E+06 (0/50)	6.65E+05 (50/50)	5.37E+05 (48/50)
Griewank	1.30E+05 (50/50)	9.42E+04 (50/50)	(0/50)	**5.47E+04** (50/50)	6.45E+04 (49/50)	8.09E+04 (47/50)
Griewank (RT)	1.30E+05 (50/50)	**9.33E+04** (50/50)	(0/50)	1.28E+06 (49/50)	1.28E+05 (43/50)	1.17E+05 (33/50)

6. Conclusion

PSO and DE, which belong to Evolutionary algorithms, have been successfully applied to various problems. In this paper, a hybrid method of PSO and DE is proposed for global optimization problems. The basic structure of the proposed method is PSO and mutation operator of DE is used as a supplementary. To enhance the diversity of swarm and avoid premature convergence, we modified velocity update equation and introduced uneveness by moving particles and stopping particles. Through simulation results using several test functions, we confirmed that the proposed method is not affected by the rotation which degrades performance of DE. The proposed method found optimal solutions in all test functions, however the convergence speed was slower than basic DE. In the future work, the reduction of population and adaptive control of parameters are required in order to reduce function evaluations and improve stability of convergence.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 24700232.

References

[1] Kennedy, J. and Eberhart, R. “Particle Swarm Optimization,” *in Proc of the IEEE International Conference on on Neural Networks* , vol. 4, pp. 1942-1948, 1995.

[2] Storn, R. and Price, K. “Differential evolution - a simple and effcient adaptive scheme for global optimization over continuous spaces,”Technical Report TR-95-012, 1995.

[3] Z. F. Hao, G. H. Guo and H. Huang, “A Particle Swarm Optimization Algorithm with Differential Evolution,” *in Proc. the Sixth International Conference on Machine Learning and Cybernetics*, pp. 1031-1035, 2007.

[4] W. J. Zhang and X. F. Xie, “DEPSO: Hybrid Particle Swarm with Differential Evolution Operator,” *in Proc. IEEE International Conference on Systems, Man and Cybernetics*, pp. 3816-3821, 2003.

[5] M. G. H. Omran, A. P. Engelbrecht and A. Salman, “Differential Evolution Based Particle Swarm Optimization,” *in Proc. the 2007 IEEE Swarm Intelligence Symposium*, pp. 112-119, 2007.

[6] Clerc, M. and Kennedy, J., “The particle swarm - explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.

[7] T. Takahama and S. Sakai, “Efficient Nonlinear Optimization by Differential Evolution with a Rotation-Invariant Local Sampling Operation,” *in Proc. of 2011 IEEE Congress on Evolutionary Computation (CEC2011)*, pp.2215-2222, 2011.