



Particle swarm optimization with opposition-based learning and near neighbor interactions

Jin Wang

School of Information, Linyi University, Linyi 276000, China.

Abstract

Particle swarm optimization (PSO) is recently proposed as population-based stochastic algorithm, which has shown excellent abilities in many optimization problems. In this paper, a hybrid PSO variant is presented to enhance its performance. The new algorithm is called OFDR-PSO which employs opposition-based learning (OBL) and fitness-distance-ratio (FDR). In order to verify the performance of OFDR-PSO, we test in on a set of well-known benchmark problems. Simulation results demonstrate that our proposed approach is effective and outperforms other four compared algorithms. ©2017 All rights reserved.

Keywords: Particle swarm optimization, evolutionary computation, opposition-based learning, global optimization.
2010 MSC: 68T05, 68W25, 90C26.

1. Introduction

Particle swarm optimization (PSO) was originally introduced by Kennedy and Eberhart in 1995 [2]. It is an evolutionary computational technique motivated by the social cooperative and competitive behavior of bird flocking or fish schooling. In the PSO algorithm, a swarm is defined as a population of interacting elements and a particle is a member in the swarm, representing a potential solution to the optimization process. Each particle of the swarm adjusts search patterns according to its experience and that of the other particles. A recent research reported that PSO has been turned out to be another powerful tool besides other evolutionary algorithms such as genetic algorithms (GAs) [1].

Recently, several variants of PSO have been developed. To balance local exploitation and global exploration, Shi and Eberhart [7] introduced a parameter called inertia weight w for the original PSO algorithm. The inertial weight is used to balance the global and local search abilities. A large inertia weight is more appropriate for global search, and a small inertia weight facilitates local search. A linearly decreasing inertial weight over the course of search achieves good performance. The version of PSO with inertia weight is also called standard PSO. Suganthan [8] proposed an improved PSO algorithm with neighborhood operator, in which a local best particle $lbest$ is used to replace the global best particle. Veeramachaneni et al. [9] developed a fitness-distance-ratio (FDR) mechanism to improve the search abilities of PSO. Liang et al. [3] proposed a comprehensive learning PSO, in which each particle

Email address: lywangj@126.com (Jin Wang)

doi:[10.22436/jmcs.017.02.10](https://doi.org/10.22436/jmcs.017.02.10)

Received 2016-12-10

can not only learn its previous best particle's flying experience, but also learns other particles'. Therefore, the particle is not attracted to fly to a fixed direction. This is helpful to avoid premature convergence. Rigel [6] proposed a diversity-guided PSO, called ARPSO, which modifies the original velocity model as follows. If the diversity of swarm is very low, then use a repulsion velocity model to repel particles. This is helpful to increase the current diversity and improve search abilities. If the diversity of swarm is large, then use the original velocity model which is regarded as an attraction phase. This operation is beneficial for decreasing diversity and accelerating convergence speed.

In this paper, we present a hybrid PSO variant, namely OFDR-PSO, which employs opposition-based learning and FDR to enhance the performance of PSO. Simulation studies on a set of benchmark problems show that OFDR-PSO achieves better results than PSO, GA, FDR-PSO, and ARPSO.

2. Particle swarm optimization

In PSO, each individual, called particle, has a velocity and a position vector. Particles fly in the search space according to its current position, previous best position, and the global best position. For a particle, its velocity and position are updated according to the following equations [7]:

$$\begin{aligned} V_i(t+1) &= w * V_i(t) + c_1 * r_1() * (pbest_i - X_i(t)) + c_2 * r_2() * (gbest - X_i(t)), \\ X_i(t+1) &= X_i(t) + V_i(t+1), \end{aligned} \quad (2.1)$$

where X_i is the position of the i th particle, V_i is the velocity of the i th particle, $pbest_i$ and $gbest$ are previous best particle of the i th particle and the global best particle, respectively, the parameter w is called inertia weight [7], r_1 and r_2 are two random numbers independently generated between 0 and 1, and c_1 and c_2 are two acceleration factors.

3. Proposed approach

Opposition-based learning (OBL) has been proven to be an effective method to DE [5], PSO [10] and ant colony optimization (ACO) [4] in some optimization problems. When evaluating a solution x to a given problem, we can improve our chance of finding a better solution by simultaneously checking the opposite solution. By doing this, the fitter one (current solution or opposite solution) can be chosen as the current solution. In fact, according to probability theory, 50% of the times, a random solution is further from the current solution than its opposite one [5].

Let $x \in [a, b]$ be a real number. The opposite number of x^* is defined by [5]:

$$x^* = a + b - x.$$

Let $X = (x_1, x_2, \dots, x_D)$ be a point in D -dimensional space, where $x_1, x_2, \dots, x_D \in \mathbb{R}$ and $x_j \in [a_j, b_j]$, $\forall j \in \{1, 2, \dots, D\}$. The opposite point $X^* = (x_1^*, x_2^*, \dots, x_D^*)$ is defined by [10]:

$$x_j^* = a_j + b_j - x_j.$$

In order to control the step size of opposition, the calculation of opposition is based on a dynamic interval boundaries $[a_j(t), b_j(t)]$, the opposition is computed as:

$$OP_{ij} = a_j(t) + b_j(t) - P_{ij}, \quad (3.1)$$

$$a_j(t) = \min(x_{ij}(t)), b_j(t) = \max(x_{ij}(t)), \quad (3.2)$$

where $i = 1, 2, \dots, ps$, $j = 1, 2, \dots, D$, P_{ij} is the j th element of the i th individual in the population, OP_{ij} is the opposite individual of P_{ij} , $[a_j(t), b_j(t)]$ are the boundaries of the j th dimension in current search space, respectively.

It has been pointed in [9] that the particle positions in PSO oscillate in damped sinusoidal waves until they converge to points between their previous best positions and the global best positions discovered by all particles so far. If these best particles are trapped, particles will quickly converge to the local optima. To improve this case, the literature [9] explored an alternative in which each particle is influenced by several other particles, not just moving towards or away from the best position discovered so far. A new velocity updating model, called Fitness-Distance-Ration (FDR), is proposed as follows, in which particles are not always attracted by their previous best particles and the global best particle [9].

$$V_i(t+1) = w * V_i(t) + \psi_1 * (pbest_i - X_i(t)) + \psi_2 * (gbest - X_i(t)) + \psi_3 * (nbest - X_i(t)), \quad (3.3)$$

where the parameter ψ_1 , ψ_2 and ψ_3 are weight factors, and the particle $nbest$ is called neighborhood best particle, which is chosen by maximizing the following expression [9]:

$$FDR(i, j, h) = \frac{\text{Fitness}(X_i) - \text{Fitness}(pbest_h)}{|pbest_h - X_{ij}|}, \quad (3.4)$$

where $h = 1, 2, \dots, ps$ and $|\cdot|$ denotes the absolute value.

In this paper, we combine opposition-based learning and FDR to enhance the performance of PSO. The main steps of our approach are described in Algorithm 3.1, where $\text{rand}(0,1)$ is a random value in $[0,1]$, p_o is the opposite rate, P_i is the i th particle in current population, OP_i is the opposite particle of P_i , P is the current population, OP is the opposite current population, gen is the number of generations, and MAX_GEN is the maximum number of generations.

Algorithm 3.1 (OFDR-PSO).

```

Begin
while  $gen < \text{MAX\_GEN}$  do
  if  $\text{rand}(0,1) < p_o$  then
    update intervals according to (3.2);
    for  $i = 1$  to  $ps$  do
      generate opposite particle  $OP_i$  according to (3.1);
      calculate the fitness value of particle  $OP_i$ ;
    end for
    select  $ps$  fittest particles from  $P$  and  $OP$  as new  $P$ ;
  end if
  else
    for  $i = 1$  to  $ps$  do
      choose  $nbest$  according to (3.4);
      update the velocity of particle  $P_i$  according to (3.3);
      update the position of particle  $P_i$  according to (2.1);
      calculate the fitness value of particle  $P_i$ ;
      if the fitness value of  $P_i$  is better than  $pbest_i$  then
        replace  $pbest_i$  with  $P_i$ ;
        if  $P_i$  is better than  $gbest$  then
          replace  $gbest$  with  $P_i$ ;
        end if
      end if
    end for
  end while
end.

```

4. Experimental studies

To compare the performance of OFDR-PSO and FDR-PSO, we use the same test problems considered in [9]. All the problems are to be minimized in this paper. The dimensions of the test problems are 20. All the benchmarks have global minima at the origin. The detailed descriptions of the test problems are presented as follows.

f_1 : DE Jong's function 1

$$f_1(x) = \sum_{i=1}^D x_i^2,$$

where $-5.12 \leq x_i \leq 5.12$.

f_2 : Axis parallel hyper-ellipsoid

$$f_2(x) = \sum_{i=1}^D i \cdot x_i^2,$$

where $-5.12 \leq x_i \leq 5.12$.

f_3 : Rotated hyper-ellipsoid

$$f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2,$$

where $-65.536 \leq x_i \leq 65.536$.

f_4 : Rosenbrock's Valley (banana function)

$$f_4(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (1 - x_i^2)^2],$$

where $-2.048 \leq x_i \leq 2.048$.

f_5 : Griewangk's function

$$f_5(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

where $-600 \leq x_i \leq 600$.

f_6 : DE Jong's function 1

$$f_6(x) = \sum_{i=1}^D |x_i|^{i+1},$$

where $-1 \leq x_i \leq 1$.

In this paper, we compare OFDR-PSO with standard PSO and FDR-PSO. The population size and maximum number of generations are set to 10 and 1000, respectively. For OFDR-PSO, the opposite probability is set to 0.3. For other parameter settings, please refer to [9]. All experiments have been run 30 times.

Table 1 shows the computational results achieved by PSO, FDR-PSO1, FDR-PSO2, and OFDR-PSO, where FDR-PSO1 and FDR-PSO2 are two variants of FDR-PSO proposed in [9]. Results of PSO, FDR-PSO1 and FDR-PSO2 are taken from Table 1 in [9]. The best results among the four algorithms are shown in bold.

From the results in Table 1, it can be seen that OFDR-PSO outperforms other three algorithms on four problems, while FDR-PSO2 achieves better results on the rest two problems. OFDR-PSO significantly improves the results on two problems, Rosenbrock's and Griewangk's. On these problems, only OFDR-PSO could obtain promising results, while other algorithms fall into local minima.

Table 2 presents the comparison of PSO, GA, ARPSO [6], FDR-PSO, and OFDR-PSO on two problems. Results of PSO, GA, ARPSO, and FDR-PSO are taken from Table 2 in [9]. As seen, for $D = 10$ and $D = 20$, OFDR-PSO outperforms PSO, GA and FDR-PSO on the two problems. Especially for Rosenbrock's problem, only OFDR-PSO can achieve reasonable solutions, while other four algorithms are trapped in local optima. The above comparisons demonstrate that combining OBL and FDR method can effectively improve the performance of PSO and FDR-PSO.

Table 1: Results achieved by PSO, FDR-PSO1, FDR-PSO2, and OFDR-PSO

Test Problems	PSO	FDR-PSO1	FDR-PSO2	OFDR-PSO
De Jong's	2.39e-02	2.70e-03	1.00e-03	5.92e-05
Rosenbrock's	6.83	6.08	8.29	2.32e-02
Axis Parallel Hyper-Ellipsoid	1.25e-01	2.30e-02	3.50e-03	5.54e-01
Rotated Hyper-Ellipsoid	5.59e+01	2.06e+01	1.51e+03	1.89e+01
Griewangk's	5.01	3.69	2.17	8.07e-03
Sum of Powers	1.80e-07	7.32e-11	3.30e-12	4.73e-06

Table 2: Comparison of PSO, GA, ARPSO, FDR-PSO, and OFDR-PSO on two problems

Test Problems	D	MAX_GEN	Griewangk's	Rosenbrock's
PSO	20	2000	1.74e-02	11.2
GA	20	2000	1.71e-02	1.70e+02
ARPSO	20	2000	2.50e-02	2.34
FDR-PSO	20	2000	3.00e-03	1.72
OFDR-PSO	20	2000	1.20e-03	3.13e-04
PSO	10	1000	8.98e-02	43.05
GA	10	1000	2.83e+02	1.10e+02
FDR-PSO	10	1000	1.48e-02	9.44
OFDR-PSO	10	1000	1.50e-03	3.61e-05

5. Conclusions

In this paper, we combine opposition-based learning and FDR method to enhance the performance of PSO and FDR-PSO. Experimental verifications on a set of benchmark problems show that our approach outperforms other compared algorithms. The hybridization strategy could effectively improve the performance of PSO and FDR-PSO. The values of p_o may affect on the performance of OFDR-PSO. This will be investigated in our future work.

References

- [1] R. Eberhart, Y. Shi, *Comparison between genetic algorithms and particle swarm optimization*, The 7th Annual Conference on Evolutionary Programming, San Diego, (1998). 1
- [2] J. Kennedy, R. C. Eberhart, *Particle swarm optimization*, IEEE Int. Conference Neural Networks, Perth, Australia. (1995). 1
- [3] J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar, *Comprehensive learning particle swarm optimizer for global optimization of multimodal functions*, IEEE Trans. Evol. Comput., **10** (2006), 281–295. 1
- [4] A. R. Malisia, H. Tizhoosh, *Applying opposition-based ideas to the ant colony system*, Proc. IEEE Swarm Intelligence Symposium, (2007), 79–87. 3
- [5] S. Rahnamayan, H. R. Tizhoosh, M. M. A. Salama, *Opposition-based differential evolution*, IEEE Trans. Evol. Comput., **12** (2008), 64–79. 3
- [6] J. Riget, J. S. Vesterstom, *A diversity-guided particle swarm optimizer-the ALPSO*, Dept. Comput. Sci. Univ. Aarhus, Aarhus, Denmark, (2002). 1, 4
- [7] Y. Shi, R. C. Eberhart, *A modified particle swarm optimizer*, Proc. Conference Evol. Comput., IEEE Press, Piscataway, (1998), 69–73. 1, 2, 2
- [8] P. N. Suganthan, *Particle swarm optimizer with neighbourhood operator*, IEEE Congress Evol. Comput., (1999), 1958–1962. 1
- [9] K. Veeramachaneni, T. Peram, C. Mohan, L. A. Osadciw, *Optimization using particle swarms with near neighbor interactions*, Proc. Genetic Evol. Comput. Conference (GECCO), Berlin, (2003), 110–121. 1, 3, 3, 4, 4
- [10] H. Wang, H. Li, Y. Liu, C. H. Li, S. Y. Zeng, *Opposition-based particle swarm algorithm with Cauchy mutation*, Proc. Conference Evol. Comput., (2007), 4750–4756. 3