

LETTER

Improved Wolf Pack Algorithm Based on Differential Evolution Elite Set

Xiayang CHEN[†], Chaojing TANG[†], Jian WANG^{†a)}, Lei ZHANG[†], *Nonmembers*, and Qingkun MENG[†], *Member*

SUMMARY Although Wolf Pack Algorithm (WPA) is a novel optimal algorithm with good performance, there is still room for improvement with respect to its convergence. In order to speed up its convergence and strengthen the search ability, we improve WPA with the Differential Evolution (DE) elite set strategy. The new proposed algorithm is called the WPADEES for short. WPADEES is faster than WPA in convergence, and it has a more feasible adaptability for various optimizations. Six standard benchmark functions are applied to verify the effects of these improvements. Our experiments show that the performance of WPADEES is superior to the standard WPA and other intelligence optimal algorithms, such as GA, DE, PSO, and ABC, in several situations.

key words: Wolf Pack Algorithm (WPA), Differential Evolution (DE), swarm intelligence, evolutionary computation

1. Introduction

An optimization problem is the mathematical programming problem that is frequently encountered in scientific research and engineering applications. In recent years, many evolutionary algorithms have been successfully applied to the optimization problems. Such as Genetic algorithm (GA) [1], Particle swarm optimization (PSO) [2], Differential Evolution algorithm (DE) [3], and Artificial Bee Colony Algorithm (ABC) [4].

Wolf Pack Algorithm is a novel optimal algorithm with outstanding performance proposed by Wu [5]. Although WPA is a powerful algorithm with strong global search ability, there is still room for improvement with respect to its convergence. In order to strengthen its local search ability and make the WPA find the optimal value more efficiently, an elite set strategy is introduced into WPA. And the solutions kept in the elite set are reconstructed by the DE algorithm during every generation. So, the new optimal algorithm is called WPADEES for short. With the help of the DE elite set, the WPADEES achieves faster convergence and higher accuracy as well as preserving the attractive characteristics of the basic WPA. Six standard benchmark functions are used to evaluate our approach. These functions are applied to verify optimization algorithms on continuous optimization problems. Experimental results show that the WPADEES has a more efficient and accurate performance than basic WPA, GA, DE, PSO, and ABC on most standard

benchmark functions.

The rest of this paper is structured as follows: Sect. 2 reviews the basic WPA in brief. Section 3 presents the WPADEES in detail. WPADEES is evaluated by six benchmark functions in Sect. 4. In addition, our method is also compared with original WPA, GA, DE, PSO and ABC. Finally, Sect. 5 outlines plans for future work and offers a conclusion.

2. Wolf Pack Algorithm

We provide a brief introduction on WPA in this section. For more detail, readers can refer to the paper [5]. Wolves are typical social animals that have a clear social work division. Wolves in WPA can be divided into three categories: the lead wolf, scout wolves and ferocious wolves. The predation behavior of the wolf pack is formalized into three intelligent behaviors, scouting, summoning, and besieging, and two intelligent rules, the winner-take-all generating rule for the lead wolf and the stronger-survive renewing rule for the wolf pack.

The lead wolf is the artificial wolf with the best objective function value. It does not need to follow the three intelligent behaviors. But once there is any wolf that is better than the lead wolf, the lead wolf is replaced by the better wolf. Only the best wolf can be the lead wolf. This is the winner-take-all generating rule.

Except the lead wolf, several elite wolves act as scout wolves. The scout wolves have better function values than the ferocious wolves. They scout around the lead wolf and hunt for solutions in the predatory space. Their scouting follows Eq.1 in the paper [5].

The lead wolf howls and calls the ferocious wolves. And the ferocious wolves gather towards the lead wolf according to Eq.2 in the paper [5]. When ferocious wolves are close to the prey, they change their behavior from summoning to besieging, following Eq.4 in the paper [5].

In WPA, a weak wolf will be eliminated and a new wolf will be generated randomly. Only the strong wolf can survive. This is the stronger-survive renewing rule.

3. Our Approach: WPADEES

When the WPA iterates over a certain number of instances, in particular at the end of the optimization process, a large number of wolves may concentrate around the extremum value, which leads to slow convergence and difficulty jump-

Manuscript received September 15, 2017.

Manuscript revised March 7, 2018.

Manuscript publicized March 30, 2018.

[†]The authors are with the College of electronic sciences, National University of Defense Technology, China.

a) E-mail: jwang_2017@163.com

DOI: 10.1587/transinf.2017EDL8201

ing out of the local optimum. In order to strengthen WPA's local search ability and to make it possible for the WPA to jump out of the local optimum, the elite set strategy is introduced into the WPA. The DE algorithm is operated on the elite set as a second optimization process and reconstructs the solutions kept in the elite set.

The core idea of WPADEES is shown as follows: first, the population follows the rule of the original WPA. It subsequently implements the scouting, summoning and besieging behavior. Then it sorts the individuals according to fitness and selects the elite set. The elite set is selected from the elite individuals in the pack. In this paper, we choose the lead wolf and the scout wolves to construct the elite set, as they have the best fitness in the pack. The DE algorithm is operated on the elite set and reconstructs the solutions kept in it. Next, the wolf pack and the lead wolf are updated. Finally, repeat the above procedures until the termination condition is met. With the help of the mutation, the crossover and the selection operations of the DE, the new hybrid algorithm can extend the exploration to the new search space. The second optimization process on the elite set makes it possible for the WPA to achieve a stronger local search ability in the local search and to avoid being trapped in the local optimum.

The framework of WPADEES is displayed in Algorithm 1. In Algorithm 1, $L(i, lead)$ represents the distance between the lead wolf and the i th ferocious wolf. L_{near} is the

besieging distance, which can be treated as a judging condition. It determines whether i th wolf changes behavior from summoning around the lead wolf to besieging behavior.

4. Simulation Experiments

In this section, six benchmark functions for global numerical optimization are used to test the performance of the WPADEES. Our implementation was compiled via Python 2.7 running under Windows7. In order to explore the advantages of WPADEES, in this subsection, we compared its performance with the original WPA and several other optimization methods, which are GA [6], DE [3], ABC [4] and PSO [2].

In order to facilitate understanding of the WPA, Wu applied many benchmark functions to test its characteristic. However, the space is limited in this paper. Therefore, only six representative functions are selected from those benchmark functions used by Wu [5], [7] to evaluate the WPADEES. The experiments' benchmark functions are shown in Tables 1 and 2. The global extremum of the Easom functions is -1 . While the global extremums of the others are 0.

Although this test only contains six functions, these six functions are representative and sufficient to simulate different types of problems, including separable, nonseparable, unimodal, multimodal, low-dimensional and high-dimensional problems. In Table 2, D represents the dimension, U indicates that the function is unimodal, M means multimodal, S means separable and N means nonseparable.

The parameters of WPA, WPADEES, GA, DE, ABC and PSO are kept the same in each experiment. The parameters for WPADEES are set as follows: $S_{num} = M_{elite} = 6$,

Algorithm 1 The WPADE algorithm

```

1: Step 1. Initialization: Define the maximum iterations  $k_{max}$ 
   and initialize other coefficient used in the WPADEES.
2: while the iteration  $k < k_{max}$  do
3:   Step 2. Scouting behavior:
4:   Select the lead wolf and scout wolves
5:   while fail to meet the termination conditions of Scouting
   do
6:     Scout wolves search solution space according to the Eq.1
       in the paper [5]
7:   end while
8:   Step 3. Summoning and besieging behavior:
9:   if  $L_{i,lead} > L_{near}$  then
10:    ferocious wolves will gather towards the lead wolf ac-
       cording to the Eq.2 in the paper [5]
11:   else
12:    ferocious wolves will gather towards the lead wolf ac-
       cording to the Eq.4 in the paper [5]
13:   end if
14:   Step 4. DE elite set operation:
15:   Select the elite set
16:   Mutation: execute the mutation operators on elite set ac-
       cording to the Eq.2 in the paper [3]
17:   Crossover: execute the Crossover operators on elite set ac-
       cording to the Eq.4 in the paper [3]
18:   Selection: follow a greedy selection rule to select the new
       solutions according to the paper [3]
19:   Step 5: Updating
20:   Renew weak wolves in a randomly way according to the
       stronger-survive renewing rule in the paper [5]
21:    $k = k + 1$ 
22: end while

```

Table 1 Benchmark functions in experiments.

Function	Formulation
Easom	$f(X) = -\cos(x_1) \cos(x_2) \cdot \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$
Sphere	$f(X) = \sum_{i=1}^D x_i^2$
Griewank	$f(X) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Rstrigin	$f(X) = \sum_{i=1}^D [x_i^2 - 10 \cdot \cos(2\pi x_i) + 10]$
Quadric	$f(X) = \sum_{i=1}^D \left(\sum_{k=1}^i x_k\right)^2$
Ackley	$f(X) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$

Table 2 Details of benchmark functions.

Functions	Range	iterations	Characteristic	D
Easom	$[-100, 100]$	1000	UN	2
Sphere	$[-100, 100]$	100	US	30
Griewank	$[-600, 600]$	500	MN	30
Rstrigin	$[-10, 10]$	100	MS	60
Quadric	$[-30, 30]$	500	MS	60
Ackley	$[-32, 32]$	500	MN	100

$T_{max} = 8$, $w = 500$, $S = 500$, $\beta = 6$, $h_{min} = 5$, $h_{max} = 10$, $F = 0.6$, $CR = 0.8$. S_{num} is the number of scout wolves. M_{elite} is the number of wolves in the elite set. T_{max} is the max scouting times. w is the distance determining coefficient. S is a step coefficient and represents the elaborate degree of artificial wolves hunting for prey in the solution space. The parameters for WPA are similar to the WPADEES except F and CR , while the DE has the same F and CR as the WPADEES. The parameters for GA are set as: $crossrate = 0.95$, $mutationrate = 0.1$, $\alpha = 0.5$. The ABC has the parameters of $trailimit = 100$ and $C = 0.5$, while the PSO has the parameters of $w = 0.8$ and $C_1 = C_2 = 2$. Every experiment was repeated for 50 runs. The results of the best optimal values, worst optimal values, and mean optimal values of each benchmark function are given in Table 3. In order to increase the identification, we highlighted the best values for each case in boldface.

From Table 3, on the whole, we can see that the WPADEES has the strongest optimizing ability among all six methods. Its best optimal values and worst optimal values are better than those of other algorithms on all the benchmark functions. Furthermore, the WPADEES algorithm has the best mean optimal value of all six bench-

Table 3 The comparison of six algorithms.

Functions	Algorithm	Worst value	Best value	Mean value
Easom	WPA	-9.034E-01	-9.998E-01	-9.813E-01
	WPADEES	-1.000E+00	-1.000E+00	-1.000E+00
	GA	-5.032E-06	-1.000E+00	-5.877E-01
	DE	-1.000E+00	-1.000E+00	-1.000E+00
	ABC	-5.181E-01	-9.997E-01	-8.990E-01
	PSO	-8.065E-05	-1.000E+00	-9.800E-01
Sphere	WPA	3.618E+02	1.259E+02	2.019E+02
	WPADEES	6.147E+00	5.638E-02	1.022E+00
	GA	4.633E+03	5.186E+02	1.945E+03
	DE	2.106E+01	3.075E+00	9.181E+00
	ABC	6.780E+03	1.340E+03	4.806E+03
	PSO	5.392E+04	9.385E+03	2.950E+04
Griewank	WPA	1.086E+00	1.025E+00	1.050E+00
	WPADEES	3.174E-01	1.426E-02	1.118E-01
	GA	4.215E+01	9.315E+00	1.994E+01
	DE	1.203E+00	1.032E+00	1.088E+00
	ABC	7.255E+01	2.262E+01	4.280E+01
	PSO	6.018E+02	5.008E+01	2.775E+02
Rstrigin	WPA	9.486E+02	7.501E+02	8.632E+02
	WPADEES	6.227E+02	2.192E+02	4.143E+02
	GA	8.280E+02	5.936E+02	7.034E+02
	DE	6.692E+02	5.450E+02	6.127E+02
	ABC	1.079E+03	7.640E+02	9.714E+02
	PSO	2.172E+03	9.008E+02	1.597E+03
Quadric	WPA	6.854E+05	3.946E+05	5.285E+05
	WPADEES	4.561E+04	9.554E+03	2.311E+04
	GA	1.609E+06	5.312E+05	9.568E+05
	DE	1.906E+05	7.763E+04	1.197E+05
	ABC	5.733E+06	2.498E+06	4.504E+06
	PSO	1.499E+07	3.826E+06	1.006E+07
Ackley	WPA	1.249E+01	1.122E+01	1.181E+01
	WPADEES	7.025E+00	5.206E+00	6.195E+00
	GA	1.866E+01	1.536E+01	1.675E+01
	DE	2.007E+01	1.972E+01	1.997E+01
	ABC	2.006E+01	1.955E+01	1.988E+01
	PSO	2.095E+01	1.870E+01	2.041E+01

mark functions. The DE is the suboptimum method of the six methods. It performs well in functions that are low-dimensional. For the sake of clarity, the convergence graphs of WPADEES, WPA, GA, DE, ABC and PSO are shown in Figs. 1–6. All of the figures behind are based on the mean optimal value of each benchmark function.

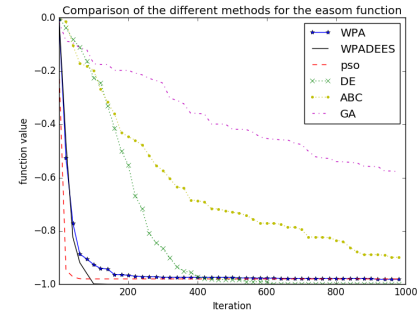


Fig. 1 The Easom function.

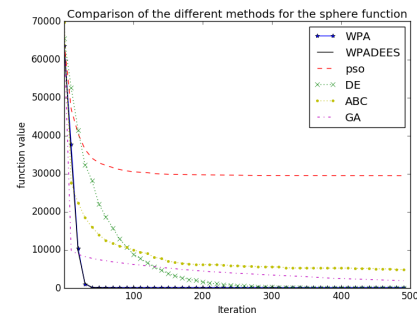


Fig. 2 The Sphere function.

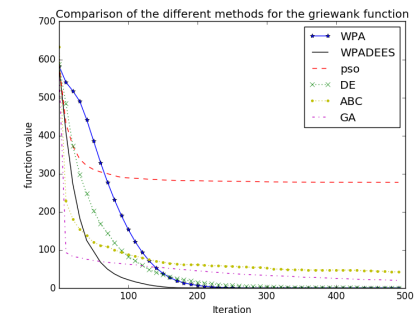


Fig. 3 The Griewank function.

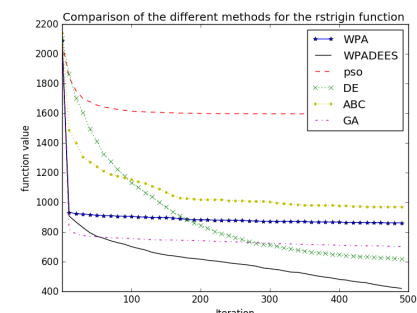


Fig. 4 The Rstrigin function.

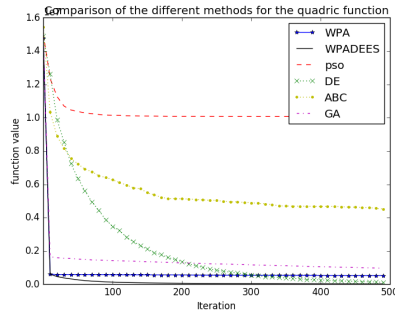


Fig. 5 The Quadric function.

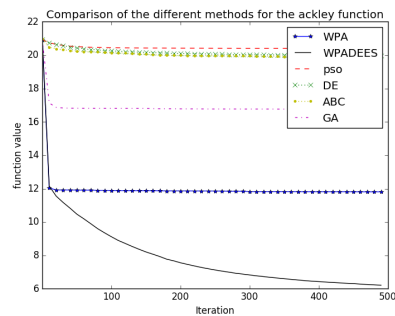


Fig. 6 The Ackley function.

From Figs.1–6, on the whole, we can see that the WPADEES has a relatively outstanding performance in terms of convergence speed. Although it is not the fastest method at the beginning of the iteration in Easom, Sphere, Rstrigin and Griewank, it is the fastest method converging to the global extremum in all six benchmark functions. Furthermore, other optimization algorithms, such as PSO, ABC and GA, always slow down and become trapped in the local optimum after several generations, while the WPADEES maintains a fast convergent tendency. We can see this clearly in Figs.4 and 6. In particular, in Fig.6, all the algorithms except the WPADEES are trapped in the local optimum. The performance of WPADEES is superior to the standard WPA and other intelligence optimization algorithms.

It is obvious that the WPADEES has better convergence accuracy and optimizing ability than WPA. In addition, based on the experiment results, we can see that WPADEES has the fastest convergence and a stronger ability to jump out of the local optimum than other intelligence optimization algorithms, such as GA, DE, PSO, and ABC.

5. Conclusion and Future Work

This paper proposed a hybrid metaheuristic WPADEES method for an optimization problem. We improved the WPA by introducing elite set strategy and the Differential Evolution algorithm. Based on the results of the test benchmark functions, we can conclude that the WPADEES significantly improves the performance of the WPA on optimal problems.

WPADEES is an improvement on WPA and achieves some success for global optimization, which can provide new ideas for solving science and engineering optimization problems. In the future, tests can be performed on different real-world tests, and different improvements can be made on the WPA algorithm. Thus, future work will focus on two issues: On the one hand, we will apply our proposed approach WPADEES to practical engineering optimization problems; on the other hand, we will develop a new meta-hybrid approach to optimization problems.

Acknowledgements

This work was supported by the National Science Foundation of China (Project No.61672527).

References

- [1] J.H. Holland, "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence," *Quarterly Review of Biology*, vol.6, no.2, pp.126–137, 1975.
- [2] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. IEEE International Conference on Neural Networks, ICNN'95*, pp.1942–1948, 1995.
- [3] R. Storn and K. Price, "Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol.11, no.4, pp.341–359, 1997.
- [4] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.
- [5] H. Wu, F. Zhang, and L. Wu, "New swarm intelligence algorithm wolf pack algorithm," *Journal of Systems Engineering & Electronics*, vol.35, no.11, 2013.
- [6] X.F. Zheng and W.U. Guo-Zhong, "The decimal-coded genetic algorithm and development of its emulational software," *Computer Simulation*, vol.22, no.1, pp.194–196, 2005.
- [7] H.-S. Wu and F.-M. Zhang, "Wolf pack algorithm for unconstrained global optimization," *Mathematical Problems in Engineering*, vol.2014, Article ID 465082, 17 pages, 2014.