International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 7, No. 6 (2014), pp. 285-296
http://dx.doi.org/10.14257/ijsip.2014.7.6.24

DEGSO: Hybrid Group Search Optimizer with Differential Evolution Operator

Yu Xie[1] Chunxia Zhao[1], Haofeng Zhang[1] and Debao Chen[2]

[1] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*
[2] *School of Physics and Electronic Information, Huaibei Normal University, Huaibei, China*
xieyuwh@163.com

Abstract

In standard group search optimizer (GSO) algorithm, scroungers will converge to the similar position if the producer cannot find a better position than the old one in a number of successive iterations and the group may suffer from the premature convergence. In this paper, a hybrid GSO with differential evolution (DE) operator named DEGSO is proposed to enhance the diversity of standard group search optimizer. In this method, the standard GSO algorithm and the DE operator alternate at the odd iterations and at the even iterations. The results of the experiments indicate that DEGSO is competitive to some other evolutionary computation (EA) algorithms.

Keywords: *Group Search Optimizer (GSO); differential evolution (DE); evolutionary computation (EC); function optimization*

1. Introduction

In the past few decades, nature-inspired computations have been widely employed to solve global optimization problems. Particle swarm optimizer (PSO) [1] gets inspiration from the social behavior of bird flocking or fish schooling. Ant colony optimization (ACO) [2] inspired by the behavior of real ant colonies. Artificial bee colony algorithm (ABC) [3] motivated by the intelligent behavior of honey bees. Inspired by animal searching behavior, a novel optimization algorithm which is called group search optimizer (GSO) [4] was proposed recently, primarily for continuous optimization problems.

In GSO algorithm, all individuals are divided into three parts; they are called producer, scroungers and rangers, respectively. The producer searches the food and shares information with the rest of the population, the scroungers keep searching for opportunities to join the resources found by the producer, and rangers walk randomly to find its new positions in the searching space. Except the producer, more than 80% of the rest of individuals are used as scroungers to maintain convergence speed of the algorithm. The scroungers will converge to the similar position if the producer cannot find a better position than the old one in a number of successive iterations. To improve the convergence performance of GSO, [5] proposed quantum-behaved operator for scroungers with a certain probability, [6] introduced randomness in determining the coefficients of individuals.

ISSN: 2005-4254 IJSIP

Differential evolution (DE), proposed by Storn and Price [7, 8], is one of the most recent evolutionary algorithms for solving optimization problems. It is a simple and powerful population-based global optimizer in continuous search space. In this work, a hybrid group search optimizer with differential evolution operator is proposed, named DEGSO. In DEGSO, the standard GSO algorithm and the DE operator alternate at the odd iterations and at the even iterations. Simulation results over several test functions validate DEGSO method is more accurate compared to some other evolutionary computation methods.

The remainder of this paper is organized in the following way. General descriptions of GSO are given in Section 2. Section 3 introduces a description of DE. Section 4 describes the proposed DEGSO technique. Section 5 presents the experimental results conducted on several test functions and the comparisons with some other EA methods. Finally, Section 6 concludes this paper.

2. Group Search Optimizer

Group search optimizer (GSO) is a population based optimization algorithm, which employs the producer-scrounger model and the animal scanning mechanism. The population of GSO algorithm is called group and each individual in the group is called a member [9]. In an n-dimensional search space, the i th member at the k th searching iteration has a position $X_i^k \in R^n$ and a head angle $\varphi_i^k = (\varphi_{i,1}^k, ..., \varphi_{i,n-1}^k) \in R^{n-1}$. The search direction of the i th member $D_i^k(\varphi_i^k) = (d_{i,1}^k, ..., d_{i,n}^k) \in R^n$ which can be calculated from φ_i^k via a polar to Cartesian coordinate transformation [10]. It can be expressed in equations (1)-(3):

$$d_{i,1}^k = \prod_{q=1}^{n-1} \cos(\varphi_{i,q}^k) \tag{1}$$

$$d_{i,j}^k = \sin(\varphi_{i,j-1}^k) \cdot \prod_{q=j}^{n-1} \cos(\varphi_{i,q}^k) \quad j = 2, ..., n-1 \tag{2}$$

$$d_{i,n}^k = \sin(\varphi_{i,n-1}^k) \tag{3}$$

In GSO, the scanning field is simplified to an n-dimensional space, which is characterized by maximum pursuit angle $\theta_{\max} \in R^1$ and maximum pursuit distance $l_{\max} \in R^1$. In GSO, a group consists of three types of members: producers, scroungers and rangers. The GSO algorithm can be simple described as four steps [9].

(1) The producer will scan at zero degree , then scan laterally by randomly sampling three points in the scanning field [11]: one point at zero degree, one point in the right hand side hypercube and one point in the left hand side hypercube, as follows:

$$X_z = X_p^k + r_1 \cdot l_{\max} \cdot D_p^k(\varphi^k) \tag{4}$$

$$X_r = X_p^k + r_1 \cdot l_{\max} \cdot D_p^k(\varphi^k + r_2 \cdot \theta_{\max} / 2) \tag{5}$$

$$X_l = X_p^k + r_1 \cdot l_{\max} \cdot D_p^k(\varphi^k - r_2 \cdot \theta_{\max} / 2) \tag{6}$$

where $r_1 \in R^1$ is a normally distributed random number with mean 0 and standard deviation 1, $r_2 \in R^{n-1}$ is a uniformly distributed random sequence in the range (0, 1).

The producer will then find the best point among the three points. If the best point has a better fitness value than its current position, then it will fly to this point. Otherwise, it will stay in its current position and turn its head to a new randomly generated angle

$$\varphi^{k+1} = \varphi^k + r_2 \cdot \alpha_{\max} \tag{7}$$

where $\alpha_{\max} \in R^1$ is the maximum turning angle.

If the producer cannot find a better area after a iteration, it will turn its head back to zero degree

$$\varphi^{k+a} = \varphi^k \tag{8}$$

where $a \in R^1$ is a pre-defined constant.

(2)When the producer is determined, the scroungers will keep searching for opportunities to join the resources found by the producer. At the *k* th iteration, the behavior of the *i* th scrounger can be modeled as a random walk toward the producer

$$X_i^{k+1} = X_i^k + r_3 \circ (X_p^k - X_i^k) \tag{9}$$

where $r_3 \in R^n$ is an uniform random sequence in the range (0,1).

(3) The third step of GSO is to renew the positions of rangers. At the k th iteration, if the *i* th member is selected as a ranger, it generates a new random head angle φ_i using equation (7); and then it chooses a random distance l_i and move to the new point

$$l_i = a \cdot r_1 l_{\max} \tag{10}$$

$$X_i^{k+1} = X_i^k + l_i \cdot D_i^k(\varphi^{k+1}) \tag{11}$$

At the end of *k* th iteration, all the members are re-evaluated. The member with the best fitness value is chosen as a producer in the next searching iteration. All the other members, including the producer in the previous searching iteration, will switch to be scroungers.

3. Differential Evolution Algorithm

Differential evolution (DE) is a relatively recent heuristic designed for optimization problems over continuous domains. In DE algorithm, each decision variable in the vector is represented by a real number. As in any other evolutionary algorithm, the *NP* vectors of initial population of DE is generated randomly, and then evaluated. The *i* th individual vector of the population at generation *G* has *D* components [12, 13].

$$\vec{X}_{i,G} = \left[x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \cdots\cdots, x_{D,i,G} \right] \tag{12}$$

Donor vector $\vec{V}_{i,G}$ is then generated by mutation as a temporary offspring.

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \tag{13}$$

where F is scale factor. It is a real and constant factor that controls the rate at which the population evolves. The indices r_1^i, r_2^i and r_3^i are mutually different random integers chosen from the range [1, NP] and they are also different from the base vector index i. These indices are randomly generated once for each mutant vector. DE employs crossover to increase the diversity of the perturbed parameter vectors. The donor vector $\vec{V}_{i,G}$ exchanges its components with the target vector $\vec{X}_{i,G}$ to generate the trial vector $\vec{U}_{i,G}$.

$$\vec{U}_{i,G} = \left[u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, \cdots\cdots, u_{D,i,G}\right] \tag{14}$$

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & if\ (rand_{i,j}[0,1] \le Cr\ \ or\ \ j = j_{rand}) \\ x_{j,i,G} & otherwise \end{cases} \tag{15}$$

where $rand_{i,j}[0,1]$ is a uniformly distributed random number, which can update every j th component of the i th parameter vector. $j_{rand} \in [1,2,\ldots,D]$ is a randomly chosen index, which ensures that trial vector $\vec{U}_{i,G}$ gets at least one component from donor vector $\vec{V}_{i,G}$. Cr is a predetermined crossover parameter lying in the range [0,1]. In selection procedure of DE, the following condition decides which vector should become a member of the next generation (G+1):

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & if\ \ f(\vec{U}_{i,G}) \le f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & otherwise \end{cases} \tag{16}$$

where f(.) is the objective function to be minimized. After installing the new population, the process of mutation, crossover and selection is repeated until the optimum is obtained, or the number of generations reaches a pre-specified maximum g_{max}.

4. The Improved GSO with DE

GSO is inspired by animal behavior, especially animal searching behavior. In standard GSO, if the members of group cannot find a better position than that of the old producer in some successive iteration, the positions of all scroungers will almost the same and the group may suffer from the premature convergence. The diversity of the group will be decreased because the number of scroungers is very large. To overcome its drawback, the DE algorithm can be used to increase the diversity of group.

DE is a kind of stochastic real-parameter evolutionary algorithm for global optimization over continuous spaces. Unlike traditional EAs, DE employs difference of the vectors of the population at the current generation to explore the objective function space [13]. Compared to most other EAs, no separate probability distribution has to be used for generating the offspring, DE is much more simple and straightforward to implement. So DE can be easily applied with GSO to enhance the ability of global search.

The procedure for the implementation of DEGSO involves the following basic steps:

(1) In every odd iteration, performs a classical GSO algorithm on each members of the group.

(2) For every even iteration, executes the DE algorithm.

(3) The terminating conditions are defined as obtaining a pre-specified objective function value or attaining a preset maximum number of iterations.

By doing this, DEGSO can obtain the animal scanning mechanism of GSO and the powerful stochastic real-parameter optimization of DE. It can significantly reduce the risk of premature convergence.

5. Simulation and Results

In this study, eight benchmark functions are used to test the efficiency of DEGSO. To compare the performance of DEGSO with some other methods, PSO, DE and GSO are also simulated in this research. The initial range of the population is symmetric as shown in Table 1. The other parameters are depicted as follows.

Table 1. Test Functions

Test function	Formulation	Initialization	f_{min}
Ackley's Function	$f_1(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i) + 20 + e$	$[-32,32]^n$	0
Generalized Rosenbrock's Function	$f_2(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$	0
Generalized Rastrigin's Function	$f_3(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]^n$	0
Generalized Griewank Function	$f_4(x) = \frac{1}{400}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^n$	0
Sphere Functionl	$f_5(x) = \sum_{i=1}^{n}x_i^2$	$[-100,100]^n$	0
Schwefel's Problem 2.26	$f_6(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500,500]^n$	-12569.5 (n=30)
Generalized Penalized Functions	$f_7(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{29}(x_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{30}u(x_i, 10, 100, 4)$, $y_i = 1 + \frac{1}{4}(x_i + 1)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50,50]^n$	0
Generalized Penalized Functions	$f_8(x) = 0.1\{\sin^2(\pi 3x_1) + \sum_{i=1}^{29}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_{30})]\} + \sum_{i=1}^{30}u(x_i, 5, 100, 4)$	$[-50,50]^n$	0

Table 2. The mean fitness value, standard deviation and average execution time of CPU for f1

M	D	PSO			GSO			DE			DEGSO		
		Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)
20	10	1.5231E+00	1.1423E+00	6.28	1.7142E-05	1.9450E-05	5.88	2.3103E-02	1.6336E-01	0.98	**9.7294E-07**	5.5228E-06	4.14
	20	3.6584E+00	9.0922E-01	6.52	4.0276E-04	3.1893E-04	7.00	1.0957E+00	1.5455E+00	2.42	**1.9006E-04**	1.2510E-03	4.72
	30	4.6609E+00	9.4565E-01	6.84	**2.5298E-02**	9.3308E-02	8.20	3.4163E+00	1.5459E+00	2.96	3.5431E-02	2.4913E-01	5.36
40	10	8.1048E-01	1.0475E+00	12.74	1.3163E-06	2.0799E-06	9.74	**4.9383E-15**	1.2453E-15	1.82	2.3807E-07	1.1246E-06	7.16
	20	2.4574E+00	9.6576E-01	13.96	2.3328E-05	1.8716E-05	11.60	**1.0054E-14**	3.3735E-15	4.10	2.6389E-05	1.7955E-04	8.14
	30	3.3104E+00	7.5577E-01	15.32	2.2935E-04	1.3432E-04	13.46	1.7062E-01	3.9932E-01	5.68	**9.8255E-05**	5.0442E-04	9.16
80	10	3.4351E-01	6.3423E-01	25.92	8.7901E-08	1.7627E-07	17.52	**4.5830E-15**	7.0325E-16	3.62	8.8750E-09	5.7864E-08	13.46
	20	1.8170E+00	8.7559E-01	31.92	2.7001E-06	4.2011E-06	20.78	**8.2068E-15**	1.5073E-15	9.62	2.4335E-06	1.4101E-05	15.20
	30	2.6447E+00	9.2839E-01	38.64	2.0786E-05	2.0306E-05	24.16	**7.2305E-10**	3.6267E-10	11.16	8.3229E-06	5.6854E-05	17.26

Table 3. The mean fitness value, standard deviation and average execution time of CPU for f2

M	D	PSO			GSO			DE			DEGSO		
		Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)
20	10	1.1546E+01	1.7491E+01	3.76	5.1253E+00	3.6029E+00	4.78	**3.1893E-01**	1.0925E+00	1.20	5.6390E-01	8.7364E-01	3.16
	20	3.6845E+02	3.4073E+02	4.06	3.0554E+01	3.0690E+01	5.86	**1.1960E+00**	1.8454E+00	1.92	1.0201E+01	2.9378E+00	3.70
	30	2.7411E+03	1.9843E+03	4.28	7.1655E+01	1.1491E+02	7.04	**8.9926E+00**	5.2142E+00	2.02	1.6470E+01	1.1647E+01	4.30
40	10	2.8805E+00	2.5815E+00	7.90	3.7812E+00	2.7910E+00	7.76	**3.1893E-01**	1.0925E+00	1.98	5.7397E-01	1.3843E+00	5.30
	20	5.8323E+01	4.0865E+01	9.00	2.5739E+01	2.6738E+01	9.72	**5.5813E-01**	1.3974E+00	3.58	8.3161E+00	5.6246E+00	6.20
	30	4.4597E+02	3.6112E+02	10.02	4.4311E+01	2.7591E+01	11.34	**9.1057E-01**	1.7493E+00	3.78	5.1447E+01	3.6420E+01	7.16
80	10	7.9971E-31	3.5201E-30	17.14	3.9324E+00	3.2255E+00	13.96	9.5678E-01	1.7199E+00	3.84	**7.0299E-01**	3.6712E-01	10.00
	20	1.9814E+01	1.4025E+01	22.00	2.4933E+01	2.7094E+01	16.92	**1.5946E-01**	7.8915E-01	7.96	7.7293E+00	4.4920E-01	11.52
	30	8.3747E+01	4.0796E+01	26.68	5.4573E+01	3.1163E+01	20.98	**3.3195E-01**	1.0890E+00	7.40	8.5004E+00	2.2902E+01	14.14

Table 4. The mean fitness value, standard deviation and average execution time of CPU for f3

M	D	PSO			GSO			DE			DEGSO		
		Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)
20	10	1.2074E-01	4.3283E-01	3.14	1.3978E-01	3.4856E-01	4.70	1.1556E+01	4.6288E+00	1.64	**1.9900E-02**	1.4071E-01	3.04
	20	**1.0656E+00**	2.2188E+00	3.46	3.6989E+00	1.6049E+00	5.84	3.2929E+01	1.2306E+01	1.80	1.2542E+00	1.6945E+00	3.56
	30	2.3118E+00	3.9011E+00	3.72	1.1844E+01	3.5266E+00	7.06	6.4076E+01	2.2807E+01	1.90	**7.7937E-01**	2.0952E+00	4.22
40	10	2.1974E-09	1.1321E-08	6.54	5.9698E-02	2.3869E-01	8.24	6.7801E+00	3.3543E+00	3.04	**3.4672E-10**	2.4514E-09	5.04
	20	3.8943E-01	1.4945E+00	7.74	9.6207E-01	9.1607E-01	9.44	2.2120E+01	6.8519E+00	3.32	**2.9850E-01**	1.0494E+00	5.98
	30	**7.7867E-01**	9.8435E-01	8.94	6.1053E+00	2.2008E+00	11.58	4.5144E+01	1.2664E+01	3.58	1.1182E+00	2.3572E+00	7.08
80	10	**4.8317E-15**	9.7826E-15	14.52	8.0945E-13	3.3066E-12	14.18	4.7758E+00	2.0793E+00	6.24	1.0772E-13	7.5964E-13	9.58
	20	7.9975E-02	3.9459E-01	19.60	2.5932E-01	4.4053E-01	17.58	2.1475E+01	8.2280E+00	6.42	**3.9798E-02**	1.9695E-01	11.62
	30	**1.3127E-01**	3.4991E-01	24.40	2.5897E+00	1.8762E+00	21.36	3.4891E+01	1.0314E+01	7.46	2.9905E-01	1.5075E+00	13.84

Table 5. The mean fitness value, standard deviation and average execution time of CPU for f4

M	D	PSO			GSO			DE			DEGSO		
		Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)
20	10	2.0295E-01	1.4277E-01	8.30	9.6236E-02	3.9372E-02	7.72	1.0633E-01	8.4114E-02	3.82	**3.9630E-02**	4.0753E-02	5.38
	20	1.0313E+00	2.1999E-01	8.78	6.3471E-02	5.4132E-02	8.98	**1.9629E-02**	1.9193E-02	3.88	4.1782E-02	1.0148E-05	6.00
	30	1.6789E+00	5.7527E-01	9.16	4.1533E-02	3.7290E-02	10.16	**2.8770E-02**	4.4365E-02	4.18	3.2452E-02	3.0957E-03	7.18
40	10	1.6814E-01	1.4083E-01	16.72	7.1230E-02	3.6559E-02	16.50	8.5182E-02	4.0855E-02	7.80	**5.2306E-02**	3.9508E-02	10.66
	20	4.1884E-01	2.2541E-01	18.26	4.8400E-02	4.5247E-02	19.00	1.4669E-02	1.7258E-02	8.82	**1.4199E-02**	1.6575E-02	12.28
	30	9.8749E-01	2.5658E-01	20.42	2.9168E-02	3.6542E-02	17.10	**6.6991E-03**	8.3684E-03	7.94	7.6162E-02	1.7293E-05	11.88
80	10	1.0574E-01	9.1158E-02	33.78	6.0513E-02	2.4783E-02	23.88	7.6942E-02	3.4219E-02	15.40	**3.9762E-02**	2.7902E-03	19.86
	20	5.0310E-02	4.3278E-02	41.46	4.3529E-02	3.5558E-02	27.64	**1.7333E-02**	1.7031E-02	15.32	4.6695E-02	5.8422E-11	21.90
	30	4.0846E-01	2.2129E-01	48.00	2.5881E-02	2.2588E-02	30.92	**7.0898E-03**	8.8151E-03	16.64	7.3961E-03	1.4345E-07	23.28

Table 6. The mean fitness value, standard deviation and average execution time of CPU for f5

M	D	PSO			GSO			DE			DEGSO		
		Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)
20	10	5.7572E-02	1.2249E-01	2.72	1.4135E-09	4.3248E-09	4.74	**7.2567E-91**	2.1061E-90	0.34	1.6240E-09	1.1462E-08	2.98
	20	2.6884E+01	1.6388E+01	2.98	1.4492E-06	5.3613E-06	5.94	**6.1419E-41**	3.1261E-40	0.74	2.6552E-06	1.8715E-05	3.58
	30	1.5762E+02	6.4478E+01	3.54	1.9069E-04	5.3363E-04	7.16	**9.8818E-22**	4.4187E-21	1.36	4.6465E-05	3.2694E-04	4.22
40	10	2.4052E-10	7.4005E-10	5.86	1.7829E-11	6.5200E-11	7.54	**5.2287E-84**	2.8862E-83	0.64	1.3881E-12	9.6553E-12	4.82
	20	1.5235E+00	1.2117E+00	7.00	7.7629E-09	1.7886E-08	9.50	**1.4159E-37**	2.7076E-37	1.50	6.1422E-09	4.2795E-08	5.94
	30	2.0298E+01	1.1155E+01	8.56	6.8958E-07	1.5297E-06	11.46	**6.3732E-24**	7.8747E-24	2.42	4.3681E-07	3.0809E-06	6.92
80	10	7.9542E-56	5.6242E-55	13.26	1.0754E-13	3.3786E-13	13.18	**2.8931E-83**	5.5237E-83	1.20	1.5527E-13	1.0895E-12	8.68
	20	3.2331E-03	5.1069E-03	18.76	7.0537E-11	1.9169E-10	17.32	**8.6800E-31**	1.1299E-30	3.48	9.2662E-11	6.5487E-10	10.68
	30	1.3120E+00	1.1408E+00	22.96	1.0225E-08	3.2283E-08	19.04	**6.5201E-18**	7.8074E-18	6.08	2.9277E-08	2.0586E-07	12.58

Table 7. The mean fitness value, standard deviation and average execution time of CPU for f6

M	D	PSO			GSO			DE			DEGSO		
		Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)
20	10	**-4.1898E+03**	1.8375E-12	3.50	-4.1898E+03	1.0673E-03	4.74	-4.1322E+03	8.5891E+01	0.01	-4.1898E+03	6.1787E-06	3.12
	20	**-8.3797E+03**	3.6749E-12	4.50	-8.3706E+03	3.5409E+01	6.06	-8.0847E+03	2.6952E+02	0.00	-8.3797E+03	4.2499E-03	3.70
	30	**-1.2569E+04**	7.3498E-12	4.14	-1.2494E+04	2.3464E+02	7.06	-1.2102E+04	2.9438E+02	0.00	-1.2558E+04	6.6810E+01	4.38
40	10	-4.1898E+03	1.8375E-12	7.54	-4.1898E+03	1.0114E-05	7.70	-4.1661E+03	5.3505E+01	0.01	**-4.1898E+03**	2.8922E-08	5.20
	20	**-8.3797E+03**	3.6749E-12	8.94	-8.3797E+03	7.8244E-03	9.60	-8.3252E+03	7.2617E+01	0.01	-8.3797E+03	3.7152E-02	6.22
	30	**-1.2569E+04**	7.3498E-12	9.44	-1.2555E+04	5.8200E+01	11.52	-1.2428E+04	1.4553E+02	0.01	-1.2569E+04	1.3931E+00	7.30
80	10	**-4.1898E+03**	1.8375E-12	17.96	-4.1898E+03	3.8393E-07	13.62	-4.1898E+03	1.9489E-12	0.01	-4.1898E+03	1.5242E-06	9.66
	20	**-8.3797E+03**	3.6749E-12	20.80	-8.3797E+03	2.2305E-04	16.90	-8.3702E+03	3.2458E+01	0.01	-8.3797E+03	1.5993E-05	11.48
	30	**-1.2569E+04**	7.3498E-12	25.62	-1.2569E+04	6.4514E-03	20.40	-1.2543E+04	6.0012E+01	0.01	-1.2569E+04	5.4165E-03	13.68

Table 8. The mean fitness value, standard deviation and average execution time of CPU for f7

M	D	PSO			GSO			DE			DEGSO		
		Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)
20	10	3.8300E-02	1.1163E-01	7.14	**2.1205E-12**	3.5550E-12	6.50	4.8419E-02	2.8473E-01	0.98	5.5858E-12	3.9050E-11	4.74
	20	8.2132E-01	7.3495E-01	8.22	2.0750E-03	1.4661E-02	8.06	3.6465E-01	5.8844E-01	2.66	**2.2189E-09**	1.5409E-08	5.68
	30	3.3490E+00	2.1222E+00	9.26	8.3008E-03	2.8408E-02	9.64	5.0213E-01	8.5805E-01	4.06	**2.4665E-06**	1.7412E-05	6.66
40	10	8.2937E-03	2.8410E-02	14.54	1.0259E-14	3.3371E-14	11.00	2.0734E-03	1.4661E-02	1.34	**6.5537E-15**	4.6139E-14	8.28
	20	2.9650E-01	3.7178E-01	17.14	**4.0219E-11**	1.6335E-10	13.48	2.9027E-02	8.6456E-02	3.80	3.3124E-10	2.3416E-09	9.90
	30	1.6551E+00	1.4229E+00	20.56	6.7376E-09	2.0990E-08	16.10	2.4717E-01	5.8791E-01	7.14	**5.4437E-09**	3.8451E-08	11.60
80	10	**1.5809E-32**	7.3017E-34	29.40	2.8692E-17	6.7727E-17	19.74	2.0734E-03	1.4661E-02	2.52	1.7133E-17	1.1847E-16	15.60
	20	8.6345E-02	1.6329E-01	38.56	**1.0624E-13**	2.5200E-13	24.24	2.0734E-03	1.4661E-02	8.04	1.1790E-13	8.2110E-13	18.40
	30	4.7818E-01	4.9935E-01	47.32	**1.1883E-11**	2.5539E-11	29.08	2.4896E-02	9.7329E-02	14.60	1.9981E-11	1.4036E-10	22.00

Table 9. The mean fitness value, standard deviation and average execution time of CPU for f8

M	D	PSO			GSO			DE			DEGSO		
		Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)	Mean best	St.var.	Time(s)
20	10	4.6105E-02	4.9079E-02	8.70	9.6286E-10	5.6029E-09	7.10	1.8838E+00	3.7016E+00	1.88	**4.8349E-11**	3.4020E-10	5.30
	20	9.2427E+00	1.1566E+01	9.70	2.4045E-04	1.5572E-03	8.56	3.2632E+01	1.6603E+01	4.28	**1.1676E-06**	8.2540E-06	6.22
	30	3.4329E+01	2.0749E+01	10.76	3.7501E-03	5.2524E-03	10.10	1.2142E+02	4.3948E+01	4.66	**4.4461E-04**	2.2002E-03	7.16
40	10	5.4550E-03	1.6801E-02	17.60	2.5429E-13	7.5248E-13	12.10	1.3185E-03	3.6067E-03	1.84	**6.9432E-14**	4.6852E-13	9.42
	20	1.5096E+00	2.5371E+00	20.18	2.1975E-04	1.5538E-03	14.56	5.6241E+00	6.3550E+00	6.54	**6.9378E-11**	4.8825E-10	11.04
	30	9.6587E+00	8.4852E+00	22.84	6.6988E-04	2.6342E-03	17.06	3.4443E+01	1.8371E+01	8.98	**3.7792E-07**	2.6702E-06	12.64
80	10	8.7899E-04	3.0111E-03	34.98	7.5522E-16	1.2416E-15	22.00	**1.3498E-32**	1.1059E-47	3.34	2.0871E-16	1.4666E-15	17.90
	20	1.0330E-01	3.1127E-01	45.00	**2.0380E-12**	7.4277E-12	26.46	2.4665E-01	1.2183E+00	10.90	6.0524E-12	4.2456E-11	20.86
	30	1.9570E+00	2.1462E+00	53.78	7.2950E-10	4.3099E-09	31.30	5.1491E+00	5.4615E+00	17.80	**3.4726E-10**	2.4533E-09	24.20

Table 10. Comparisons between DEGSO and other algorithms on *t* test

Function	Population		PSO	GSO	DE
f1	20	*t*-value	-1.221717	2.042064	**11.422667**
		P-value	0.223744	0.042909	**0.000000**
	40	*t*-value	-1.205038	1.678673	**8.135061**
		P-value	0.230099	0.095312	**0.000000**
	80	*t*-value	-1.610776	0.118205	3.573577
		P-value	0.109345	0.906065	0.000475
f2	20	*t*-value	**24.335710**	-0.262026	**9.746473**
		P-value	**0.000000**	0.793663	**0.000000**
	40	*t*-value	**19.285473**	1.690112	2.866626
		P-value	**0.000000**	0.093098	0.004750
	80	*t*-value	**15.608053**	1.531470	-1.301203
		P-value	**0.000000**	0.127774	0.195197
f3	20	*t*-value	**8.966227**	2.268135	**5.772224**
		P-value	**0.000000**	0.024758	**0.000000**
	40	*t*-value	**7.212578**	0.159883	3.171373
		P-value	**0.000000**	0.873190	0.001842
	80	*t*-value	**6.287528**	-0.400623	2.041437
		P-value	**0.000000**	0.689272	0.042972
f4	20	*t*-value	**8.938207**	3.982711	**11.054743**
		P-value	**0.000000**	0.000106	**0.000000**
	40	*t*-value	**6.884858**	2.044084	**8.701576**
		P-value	**0.000000**	0.042706	**0.000000**
	80	*t*-value	**5.477293**	0.536656	**5.514230**
		P-value	**0.000000**	0.592306	**0.000000**
f5	20	*t*-value	**7.553896**	4.554961	-8.021940
		P-value	**0.000000**	0.000011	0.000000
	40	*t*-value	**6.724387**	1.631126	-7.813499
		P-value	**0.000000**	0.104976	0.000000
	80	*t*-value	**8.266626**	**8.750257**	-4.577576
		P-value	**0.000000**	**0.000000**	0.000010
f6	20	*t*-value	1.973585	**10.347095**	**16.587538**
		P-value	0.050279	**0.000000**	**0.000000**
	40	*t*-value	-0.623666	**7.805620**	**16.938624**
		P-value	0.533801	**0.000000**	**0.000000**
	80	*t*-value	-0.599915	**5.651904**	**17.166712**
		P-value	0.549474	**0.000000**	**0.000000**
f7	20	*t*-value	**9.549991**	1.578502	-1.060990
		P-value	**0.000000**	0.116571	0.290411
	40	*t*-value	**7.899535**	0.560098	-1.016586
		P-value	**0.000000**	0.576253	0.310998
	80	*t*-value	**5.958274**	-0.651119	-1.008762
		P-value	**0.000000**	0.515973	0.314724
f8	20	*t*-value	**16.108370**	**6.772946**	2.404045
		P-value	**0.000000**	**0.000000**	0.017443
	40	*t*-value	**14.845665**	0.430336	-2.748792
		P-value	**0.000000**	0.667573	0.006721
	80	*t*-value	**8.582779**	4.907611	0.868067
		P-value	**0.000000**	0.000002	0.386754
Better			18	5	10
Same			6	19	12
Worse			0	0	2
General merit over contender			18	5	8

The value of function is defined as the fitness function of algorithm in this study. For the purpose of reducing statistical errors, each function is independently simulated fifty times. To investigate the scalability of the algorithm, the population sizes of different algorithms are 20, 40 and 80 with maximum generations is 2000. The experiments were carried out on a PC with a 1.80-GHz CPU and 1.0-GB RAM. All the programmers were written and executed in MATLAB 7.0. The mean best fitness values, standard deviations and the average computation costs of CPU for every function are displayed from Tables 2 to 9.

Table 2 shows DE has better mean solutions and standard deviations than those of algorithms for function 1, but the average solutions of DEGSO are better than those of DE for 20 dimension function with population sizes were 10 and 20. Table 3 displays that the performance of DE is better than DEGSO on function 2.

Table 4, 8 and 9 indicate that DEGSO has higher convergence accuracy than those of other methods. Table 5 displays that the performances of DE and DEGSO are very closely on function 4. From Table 6, it can conclude that DE is more efficient than other three algorithms. Table 7 shows that the mean solutions and standard deviations of PSO with are better than those of other methods.

For a thorough comparison of classification accuracy, one-tailed paired *t* test with a significance level of 0.05 has also been carried out. In experiments, the number of that DEGSO performs significantly better than, almost the same as and significantly worse than the compared algorithms under different conditions is given in the Table 10. “General merit over contender” shows that the difference between the number of better results and the number of worse results, which is used to give an overall comparison between the two algorithms. Moreover, Table 10 indicates that the DEGSO generally offered better performance than those of other algorithms in this study.

6. Conclusion

In this paper, a DEGSO algorithm based on differential evolution and Group Search Optimizer was proposed for function optimizations in continuous space. The hybrid strategy provides animal scanning mechanism of GSO and the population diversity of DE operator. The comparison results show that DEGSO has better optimization capability than traditional DE and GSO. It is even more accurate compared to some other EA ensemble approaches.

Acknowledgements

This work was partially supported by the National Nature Science Foundation of China under Grant (61272220 and 61101197) and the Nature Science Foundation of Anhui Province of China under Grant 1308085MF82.

References

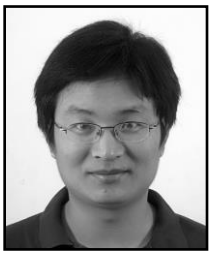
[1] J. Kennedy and R. Eberhart, "Particle swarm optimization", IEEE International Conference on Neural Networks, **(1995)** November 27- December 1, Perth, Australia.
[2] M. Dorigo, "Optimization, learning and natural algorithms, Ph.D. Dissertation (in italian)", Politecnico di Milano, Milano, Italy, **(1992)**.
[3] D. Karaboga, "An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University", Engineering Faculty, Computer Engineering Department, **(2005)**.
[4] S. He, Q. H. Wu and J. R. Saunders, "A Novel Group Search Optimizer Inspired by Animal Behavioural Ecology", IEEE Congress on Evolutionary Computation, **(2006)** July 16-21, Vancouver, Canada.
[5] D. Chen, J. Wang, F. Zou, W. Hou and C. Zhao, "An improved group search optimizer with operation of quantum-behaved swarm and its application", Applied Soft Computing, vol. 1, no. 12, **(2012)**.

[6] L. Wang, X. Zhong and M. Liu, "A novel group search optimizer for multi-objective optimization", Expert Systems with Applications, vol. 3, no. 39, **(2012)**.
[7] R. Storn and K. Price, "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces", International Computer Science Institute-Publications-TR, **(1995)**.
[8] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces", Journal of global optimization, vol. 4, no. 11, **(1997)**.
[9] S. He, Q. H. Wu and J. R. Saunders, "Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior", IEEE Transactions on Evolutionary Computation, vol. 5, no. 13, **(2009)**.
[10] L.-A. Giraldeau and L. Lefebvre, "Exchangeable producer and scrounger roles in a captive flock of feral pigeons: a case for the skill pool effect", Animal Behaviour., vol. 3, no. 34, **(1986)**.
[11] W. J. O'Brien, B. I. Evans and G. L. Howick, "A New View of the Predation Cycle of a Planktivorous Fish", Canadian Journal of Fisheries and Aquatic Sciences, vol. 43, **(1986)**.
[12] U. Maulik and I. Saha, "Automatic Fuzzy Clustering Using Modified Differential Evolution for Image Classification", IEEE Transactions on Geoscience and Remote Sensing, vol. 9, no. 48, **(2010)**.
[13] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art", IEEE Transactions on Evolutionary Computation, vol. 1, no. 15, **(2011)**.

Authors

Yu Xie, he received M.S. degree from Shanghai University, Shanghai, China, in 2008. Currently, he is pursuing the Ph.D. degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His current research interests include evolutionary computation and machine learning.

Chunxia Zhao, she received the Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 1998. She is professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. Her current research interests include pattern recognition, image processing, artificial intelligence and intelligent robot.



Haofeng Zhang, he received the B.S. and Ph.D. degrees from Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2007, respectively. He is Associate professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include pattern recognition, machine learning, and intelligent robot.

Debao Chen, he received the Ph.D. degree from Nanjing University of Science and Technology, Nanjing, China, in 2007. He is professor at the School of Physics and Electronic Information, Huaibei Normal University, Huaibei, China. His current research interests include evolutionary computation, pattern recognition, and artificial intelligence.

