A Novel Atmosphere Clouds Model Optimization Algorithm

Gao-Wei Yan[1] Zhan-Ju Hao[1] Jun Xie[1]

[1] College of Information Engineering, Taiyuan University of Technology

Taiyuan 0300024, Shanxi, PRC

yangaowei@tyut.edu.cn, haozhanju_2008@126.com, xiejun@tyut.edu.cn

Received on 18 June 2013; Revised on 5 July 2013; Accepted on 20 July 2013

Abstract. This article introduces a novel Atmosphere Clouds Model Optimization algorithm (ACMO), which is inspired by the generation behavior, move behavior and spread behavior of clouds in the natural world. As the global search method of ACMO algorithm, the reverse search method composed by the move behavior and spread behavior of clouds disperses the whole population to the search space. This method can enhance the diversity of population; the generation behavior is mainly used to search in the vicinities of current global optimal, keeping the convergence of ACMO algorithm. And the proposed algorithm has been tested on a set of multimodal functions in comparison with Particle Swarm Optimization algorithm (PSO) and Genetic Algorithm (GA). The results demonstrate that the proposed algorithm has a certain advantage in solving multimodal functions.

Keywords: numerical optimization; evolutionary algorithm; swarm intelligence; cloud model

1 Introduction

Evolutionary algorithm (EA) is a very important research area in applied mathematics, which aims to find the best values for a system's parameters under various conditions. With the development of science and technology, more and more industrial areas are inseparable from the optimization design. Especially, when the objective function for an optimization problem is non-linear and non-differentiable, evolutionary algorithm techniques are typically used to find the global optimum.

In the past few decades, nature-inspired computation has attracted more and more attention. For instance, as a standard optimization tool in engineering, Genetic Algorithm (GA) is a stochastic search procedure based on the mechanics of natural selection, genetics and evolution [1]; Particle Swarm Optimization algorithm (PSO) is inspired by swarm intelligence and theory in general such as bird flocking, fish schooling and even human social behavior [2]; Tabu Search (TS) [3] is a local search algorithm, which simulates the intellectual processes of human memory process; Ant Colony Optimization algorithm (ACO) is developed based on ants' foraging behavior [4][5]; K. M. Passino proposes Bacterial foraging optimization (BFO) algorithm [6], by mimics the behavior of E. coli bacteria foraging. Hsuan et al. [7] introduce a novel bacterial foraging particle swarm optimization (BFPSO) algorithm to design vector quantization (VQ)-based fuzzy-image compression systems. Invasive Weed Optimization algorithm (IWO) [8] is inspired from colonizing weeds. Recently, Oftadeh et al. [9] propose a new hunting search (HuS) metaheuristic algorithm inspired by group hunting of animals such as lions. Inspiring from the echolocation behavior of micro bats, Yang [10][11] developed bat algorithm (BA). By idealizing behavior of the flashing characteristics of fireflies, Yang [12] conceptualized Firefly Algorithm (FA). Also, there are the nature-inspired algorithms that mimic physical phenomena, such as simulated annealing (SA) [13], big bang–big crunch algorithm (BB–BC) [13] and charged system search (CSS) [14]. Geem et al. [15] developed a harmony search (HS) meta-heuristic algorithm that was conceptualized using the musical process of searching for a perfect state of harmony. More recently, Ali Sadollah et al. [16] introduce a new mine blast algorithm (MBA) whose concepts are inspired form the explosion of mine bombs in real life situations. These bio-inspired and nature-inspired algorithms have been successfully employed to optimize some numerical problem and engineering optimization problems.

Many algorithms have one thing in common: the whole population updates their positions towards to the global optima positions in the process of optimization. This search method allows the algorithm converge to the optimal solution quickly, and then ensures the convergence accuracy of the algorithm. For unimodal functions this method can find the global optimum quickly, while it is easy to make algorithm appear premature convergence phenomenon when solving multimodal functions. In PSO algorithm, for example, all particles move quickly toward the optimum points with the guiding force of *pbest* and *gbest*. During the later stages of the search, all particles tend to cluster, and the diversity of the particles decreases quickly with the increasing of iteration, so that if the population cannot find the global optimum in the early stage of the search, then algorithm will easily get trapped into local optimum and is difficult to escape from it.

In this study, the authors propose a novel numerical stochastic optimization algorithm inspired by floating clouds. Cloud is a ubiquitous feature of our natural world [17]. Generally, with colorful appearance and ever-changing features, cloud is generated in the areas with high humidity, while the move behavior and spread behavior of cloud are affected by the atmospheric pressure. Based on the behavior rules of cloud this article proposes a novel atmosphere clouds model optimization algorithm (ACMO). In ACMO, a novel optimization method-Reverse Search Method is presented, which is totally different from traditional optimization methods. In this method, the whole population spread from the current optimal points to the whole search space in a 'cloud' existence pattern, instead of clustering from all directions to the optimum points. This special optimization method can make the ACMO algorithm maintain large population diversity and prevent the algorithm from trapping into local optima. At the same time the generation behavior of cloud ensures the algorithm can search carefully around optimum positions and enhances local search ability of the ACMO algorithm.

This article is organized as follows: Section 2 describes the proposed algorithm and the details of its implementation. Section 3 introduces the simulation study of ACMO algorithm; the simulation experiment has two parts: the first part tested the effects of several control parameters to the ACMO algorithm; the second part compared the results obtained by ACMO algorithm with Particle Swarm Optimization algorithm (PSO) and Genetic Algorithm (GA). Section 4 is the conclusion of this article.

2 ACMO algorithm

The unconstrained global optimization problem can be formulated as a *D*-dimensional maximization problem as follows:

$$\text{Maximize } f(\mathbf{x}),\ \mathbf{x}=[x_1,x_2,\dots,x_D] \tag{1}$$

where *D* is the dimension of the parameter space, which *f*(**x**) can be optimized.

The ACMO algorithm is inspired by the generation behavior, move behavior and spread behavior of cloud in the search space.

The core idea of ACMO algorithm is illustrated as follows:

1) The whole search space is divided into many disjoint regions according to the certain rules, and each region has its respective humidity value and atmospheric pressure value.
2) Another important concept in ACMO algorithm is that the behavior of the cloud must follow the rules below:
   a) Cloud can only be generated in the regions where the humidity value is greater than a certain threshold value;
   b) Under the action of wind, clouds move from regions with higher atmospheric pressure to regions with lower atmospheric pressure;
   c) In the move process, the droplets of one cloud would spread or gather according to the difference of the atmospheric pressure between the regions.
   d) One cloud is regarded to disappear when meeting some criteria.
3) The humidity value and atmospheric pressure value of all regions are updated after every behavior of clouds.

Definitions of important concepts in ACMO algorithm are as follows.

Definition 1 The region is the subspace after division of the search space *U*. Suppose $U\in R^D$, each dimension of *U* is divided into *M* small intervals

$$I_j=(u_j-l_j)/M,\ j=1,2,\dots,D \tag{2}$$

where I_j is the length of interval in *j*th dimension, u_j and l_j express the upper boundary and lower boundary of *j*th dimension respectively. Then the whole search space would be divided into M^D regions, each region is denoted as U_i.

Definition 2 The humidity value of region U_i, denoted by H_{U_i}, expresses the best fitness value found in U_i.

$$\mathbf{x}_i^*=\underset{\mathbf{x}\in U_i}{arg\,max}\ f(\mathbf{x}),\quad H_{U_i}=f(\mathbf{x}_i^*) \tag{3}$$

where **x** expresses the droplet which falls into region U_i, and $\mathbf{x}_i^*$ is a feasible solution which indicates the position with the maximum fitness value.

Definition 3 The atmospheric pressure value of region U_i , denoted by P_{U_i} , is defined as cumulative sum of the searched number of U_i.

Definition 4 Cloud C is defined as a set composed droplets.
The specific optimization process of ACMO algorithm is elaborated in details as Algorithm 1.

Algorithm 1: Atmosphere clouds model optimization algorithm

Input: Problem and Parameter of ACMO
Output: Optimum results

```
1 Step1: Initialization;
2 while $t < t_{max}$ do
3     Step2: generation of cloud;
4     Step3: Update the humidity value $H_{Ui}$ and air pressure value $P_{Ui}$ of region;
5     Step4: The movement phase of cloud;
6     Step5: The spread phase of cloud;
7     Step6: Update the humidity value $H_{Ui}$ and air pressure value $P_{Ui}$ of region;
8     $t \leftarrow t + 1$;
9 end while
```

2.1 Initialization Phase

In the initialization phase, the main task of the ACMO algorithm is the region division of the search space, the initialization of the humidity values and atmospheric pressure values of regions, and the set of population, max iterations t_{max} and other related parameters. And the parameters of algorithm such as threshold factor λ, contract factor ζ, weaken rate γ etc are set to the suitable value.

2.2 The generation of cloud

Normal distribution is universal and it is widespread in natural phenomena, so we adopt the normal cloud model [18] to describe the concept of the cloud.

The center position $Center_{Ck}$, entropy En_{Ck} and hyper entropy He_{Ck} of the cloud C_k are defined as the same as Expectation Ex, Entropy En, Hyper entropy He of the normal cloud model respectively.

According to the theoretical knowledge of normal cloud model, the normal cloud can be generated if the three numerical characteristics (Ex, En, He)[19] and the number of droplets are given. Therefore four parameters are required to be confirmed before the generation process: 1) $Center_{Ck}$, the region where the cloud can be generated; 2) En_{Ck}, the entropy which is used to determine the cover range of cloud; 3) He_{Ck}, the hyper entropy which is used to determine the thickness of cloud; 4) the number of droplets.

2.2.1 Determinate the *Center*$_{Ck}$

If the humidity value of a region is greater than a threshold, then this region can generate one cloud. The threshold value Ht is calculated dynamically according to Equation (4).

$$Ht = H_{min} + \lambda * (H_{max} - H_{min}) \tag{4}$$

where H_{min} and H_{max} are the minimum and maximum humidity values in the whole search space respectively; λ is the threshold factor, which we let λ be 0.7 through the experimental tests in the article.

The set of regions which can generate cloud is expressed as $S = \{U_i | H_{Ui} > Ht, i=1,2,\ldots,M^D\}$. Suppose that the length of S is q and one cloud can be generated in each region of S, then there will be cloud set $\{C_1, C_2,\ldots, C_k,\ldots,C_q,\}$, the center of cloud C_k is calculated by Equation (5) in region U_i .

$$Center_{C_k} = \mathbf{x}_i^* \ , \mathbf{x}_i^* \in U_i \tag{5}$$

2.2.2 Compute entropy En_{Ck} and hyper entropy He_{Ck}

In ACMO algorithm the initial entropy *EnM* is defined as

$$EnM_j^0 = \frac{I_j / M}{A} \tag{6}$$

where I_j is the length of region in *j*th dimension, $j=1,2,\ldots,D$; *A* determines the cover range of the cloud generated in the first iteration. According to the 3*En* rule of the cloud model [19], we set *A* be 6 in this article and it means that the cloud generated in the first iteration can cover one whole region approximately.

The value of entropy *EnM* decreases with the search process and it is given as:

$$EnM_j^t = EnM_j^0 \times \zeta \tag{7}$$

where ζ is the contract factor. In this article, the effect of the three different decreased models of *EnM* is compared to that of ACMO algorithm. According to the results, the entropy decreases in the sigmoid function model.

According to ref.[20], a large hyper entropy will cause a little condensation degree of cloud and a large search radius of cloud. It is beneficial for ACMO algorithm to escape from the local optima. The hyper entropy of clouds newly generated increases with the iteration in sigmoid function model as follows:

$$HeM^t = HeM^0 \times \left(\frac{1}{1+e^{8-16\times(t/t_{max})}} \right) \tag{8}$$

where HeM^0 is the initial hyper entropy and let HeM^0 be 0.5 after experiment tests; *t* is the current iteration; t_{max} expresses the maximum iteration times.

2.2.3 Compute the number of droplets

Assuming the maximum number of droplets in the search space at one iteration is a constant value and the droplets number of the cloud must be larger than the threshold value *dN*, otherwise the cloud is regarded to dissolve. The droplets numbers of the clouds generated in different regions are related with the humidity values of their respective regions: the higher humidity with the more droplets, the lower humidity with the fewer droplets.

Suppose that the maximum of droplets in the search space at each iteration is *N*. The number of the droplets can be generated in one iteration *nMax* is expressed as:

$$nMax = N - \sum_{k=1}^{p} n_k \tag{9}$$

where *p* is the number of the existed cloud; n_k is the droplets number of the cloud C_k.

If *nMax* calculated by Equation (9) is less than *dN*, then there is no cloud generated in this iteration, otherwise the droplets number of each cloud can be determined.

Suppose that the set *S* has *q* elements, the *k*th cloud newly generated belongs to the region U_i. The droplets number n_k has a linear relation with the humidity of the region U_i

$$n_k = \frac{H_{U_i}}{\sum_{k=1}^{q} H_{U_i}} \times nMax \quad , \quad U_i \in S \tag{10}$$

where $k = 1, 2, \ldots, q$; H_{U_i} is the humidity value of region U_i.

After computing the droplets number of each cloud, check out all the clouds newly generated and guarantee that the droplets numbers of each cloud is greater than *dN*. Algorithm 2 illustrates the procedure of generating of cloud.

2.3 The move behavior of cloud

As presented in the core idea of ACMO algorithm, the generated clouds move from regions with higher atmospheric pressure to regions with lower atmospheric pressure until arriving at the destinations or going extinct.

Suppose that the region where the cloud C_k located is region *E*, randomly select one region as the target region *F* where the atmospheric pressure value is lower than that in region *E*. The pressure difference between *E* and *F* is $\Delta P=P_E-P_F$.

Algorithm 2: Procedure of generating of cloud

```
Input: humidity value $H$
Output: clouds $C$
1  procedure Getregions($H$,$U$)   //Get regions can generate cloud;
2      $Ht \leftarrow H_{min} + \lambda\,(H_{max} - H_{min})$   // Calculate threshold value;
3      $S \leftarrow \Phi$;
4      for $i \leftarrow 1, M^D$ do
5          if $H_{Ui} > Ht$ then
6              $S \leftarrow S + U_i$;
7          end if
8      end for
9  end procedure
10 procedure GetEnHe($EnM^0$,$HeM^0$,$t$)   // Calculate entropy and hyper entropy;
11     for $j \leftarrow 1, D$ do
12         $EnM_j^t \leftarrow EnM_j^0 \times \zeta$;   // Calculate entropy;
13     end for
14     $HeM^t \leftarrow HeM^0 \times \left( \frac{1}{1+e^{8-16\times(t/t_{max})}} \right)$;   // Calculate hyper entropy;
15 end procedure
16 procedure Getndrops($H$,$C$)   // Compute the number of droplets;
17     $nMax \leftarrow N - \sum_{k=1}^{p} n_k$;
18     if ($nMax > dN$) then
19         for $k \leftarrow 1, q$ do
20             $n_k \leftarrow nMax \times H_{U_i} \Big/ \sum_{i=1}^{q} H_{U_i}$; //Calculate the number of droplets of cloud $C_k$;
21         end for
22         while (the set $H = \{C_k \mid n_k < dN, k = 1,2,...,q\} \neq \varnothing$) do
23             $C \leftarrow C - C_k$, $\left( C_k \in U_i \middle| _{H_{U_i} = \overset{k}{\underset{j=1}{min}} (H_{U_i})} \right)$; //Remove the lowest humidity region from $C$;
24             $q \leftarrow q - 1$;
25             Recalculate the number of cloud droplets $n_k$;
26         end while
27     end if
28 end procedure
```

The move velocity equation of the cloud is calculated as follows.

$$V_{C_k} = e \times 6 \times En_{C_k} \tag{11}$$

where *e* expresses the direction of movement and can be calculated by the Equation (12):

$$e = \frac{(1-\beta)\times V_{C_k} + \beta \times \left(\mathbf{x}_F^* - Center_{C_k} \right)}{\left\| (1-\beta)\times V_{C_k} + \beta \times \left(\mathbf{x}_F^* - Center_{C_k} \right) \right\|},\ \ \beta = \frac{\Delta P}{P_{Max} - P_{Min}} \tag{12}$$

where β is the atmospheric pressure factor, and the value of it indicates the influence degree of the atmospheric pressure on the move velocity of cloud; P_{Max} and P_{Min} are the maximum and minimum atmospheric pressure of the search space respectively; $\mathbf{x}_F^*$ is the location with the best fitness value in the region *F*. The $6 \times En_{C_k}$ in Equation (11) expresses the move speed. For simplifying the searching in the move process, the move speed is set to be equal to the length of the cover scope approximately.

The energy of cloud would reduce as a result of evaporation or clash of clouds in the move process, so the weaken rate γ is proposed as an important concept in the article, which means the droplets number of each cloud would decrease γ*100% after each iteration. The γ is used to determine the dissolve rate of the clouds, and let γ be 0.2 hereafter. Algorithm 3 illustrates the movement phase of cloud.

Algorithm 3: Procedure of the movement phase of cloud

Input: clouds *C*
Output: clouds *C*

```
1   procedure Cloudmove(C)          // The movement procedure of cloud;
2       $k \leftarrow 1$;
3       while $k<q$ do
4           ir $\leftarrow randn(1,M^D)$ ;
5           $F \leftarrow U_{(ir)}$ ;                 // Select one region randomly;
6           $\Delta P \leftarrow P_E\text{-}P_F$ ;               // Calculate pressure difference between $E$ and $F$;
7           while ($\Delta P$=<0) do
8               ir $\leftarrow randn(1,M^D)$ ;
9               $F \leftarrow U_{(ir)}$ ;
10              $\Delta P \leftarrow P_E\text{-}P_F$ ;
11          end while
12          $V_{C_k} \leftarrow e \times 6 \times En_{C_k}$ ;      // Calculate the movement speed of cloud $C_k$;
13          $center_{C_k}^{t+1} = center_{C_k}^{t} + V_{C_k}$ ;  // Calculate the center position of cloud $C_k$ after movement;
14          $n_k \leftarrow n_k \times (1\text{-}\gamma)$;         // Calculate the droplets of cloud $C_k$ after movement;
15          if $n_k<dN$ then
16              $C \leftarrow C\text{-}C_k$;          // Cloud $C_k$ is dissolved;
17              $q \leftarrow q\text{-}1$;
18          end if
19      end for
20  end procedure
```

2.4 The spread behavior of cloud

As mentioned in the core idea of ACMO algorithm, the droplets of the clouds will gather or spread during the moving progress.

Suppose that the region where the cloud C_k currently located is E, and the region where the cloud C_k located after moving process is region G. The spread speed of the cloud is expressed in Equation (13)

$$En_{C_k} = En_{C_k} \times (1+\alpha) \tag{13}$$

where α is the spread factor which determines the spread speed of the cloud and can be calculated as

$$\alpha = \begin{cases} \dfrac{\Delta P}{\Delta P_{Max}}, & if \;\; E \neq G \\ 0.3, & others \end{cases} \tag{14}$$

where $\Delta P = P_E\text{-}P_G$ is the pressure difference between the regions E and G. ΔP_{Max} expresses the maximum pressure difference in the search space.

With the entropy increasing in the iteration process, let the hyper entropy of the cloud decrease to decline the discrete degree of the cloud in the iteration process so that the clouds can do the comprehensive search. The formula of the decreased hyper entropy is as follows

$$He_{C_k} = He_{C_k} \times (1\text{-}\alpha) \tag{15}$$

The clouds cannot exist forever. Here gives two criteria for the dissipation of the clouds: 1) the *En* of the cloud is greater than $5 \times EnM^0$; 2) the droplets number of the cloud is less than *dN*. Algorithm 4 illustrates the spread phase of cloud.

The ACMO algorithm updates the humidity value and the atmospheric pressure value of all regions in each iteration after the generation process, move process and spread process of the cloud. The results can confirm new regions to generated clouds and the move direction of the clouds.

Algorithm 4: Procedure of the spread phase of cloud;

Input: clouds *C*
Output: clouds *C*

```
1  procedure cloudspread ($C$)     // the spread procedure of cloud;
2      $k \leftarrow 1$;
3      while $k<q$ do
4          $E \leftarrow Getregion(center_{C_k}^{t})$;  //get current region;
5          $G \leftarrow Getregion(center_{C_k}^{t+1})$;  // get region after movement;
6          $\Delta P \leftarrow P_E - P_G$;       // calculate pressure difference between $E$ and $G$;
7          if ($E \neq G$) then
8              $\alpha \leftarrow \Delta P / \Delta P_{max}$;
9          else
10             $\Delta P \leftarrow 0.3$;
11         end if
12         $En_{C_k} \leftarrow En_{C_k} \times (1+\alpha)$;     // calculate entropy of cloud $C_k$ after movement;
13         $He_{C_k} \leftarrow He_{C_k} \times (1-\alpha)$;     // calculate hyper entropy of cloud $C_k$ after movement;
14         if $En_{C_k} > EnM^0 \times 5$ then
15             $C \leftarrow C - C_k$;       // cloud $C_k$ dissolved;
16             $q \leftarrow q-1$;
17         end if
18     end for
19 end procedure
```

3 Case Studies

A set of benchmark functions [21],[22],[23] is used to test the performance of the proposed algorithm, which is shown in Table 1. All these functions are the multimodal functions.

3.1 The performance analysis of ACMO algorithm with different values of the parameters

This section tests the performance of ACMO algorithm with different values of the parameters.

3.1.1 The influence of the contract factor to ACMO algorithm

The contract factor ζ controls the decreased speed of the entropy *EnM*. The performance of ACMO algorithm is tested based on ζ in accordance with the law of the sigmoid function, the exponential function and the linear function respectively, as shown in

Table 3.

Fig. **1** shows that three different contact factors ζ_1, ζ_2, ζ_3 decrease with iterations, in which ζ_1 keeps a large value for a long time in the early stage and a small value for a long time in the later stage, and a drastic change in the middle process of the iteration; ζ_2 keeps a same deceased speed in the whole iteration process; ζ_3 decreases in a fast speed in the early stage and in a slow speed in the later stage of the process. In this section the contract factor is given to be selected through the experiments in order to ensure the optimal performance of ACMO algorithm.

The convergence processes of ACMO algorithm based on ζ under the three mentioned cases are shown in Fig. 2, where ACMO algorithm is tested in Schaffer function and Needle in the haystack function. Compared with ζ_2 and ζ_3, ACMO algorithm with ζ_1 has a slow convergence speed in the early stage of the iteration process, while in the later stage the convergence ability will be raised up obviously so that the ACMO algorithm can converge to the global optimal solution. It is because that in the early stage of the iteration, the decreased speed of ζ_1 is slow, the *EnM* value of the newly generated cloud can make cloud cover a larger region, ACMO algorithm can search more scopes and the population diversity of the algorithm is maintained; in the later stage of the iteration process, ζ_1 keeps a small value for a long time, which means that the clouds do the accurate search around the global optimal, the convergence precision of ACMO algorithm is improved.

Table 1. List of the benchmark functions

	Function name	Function	GM	*D*	*S*
f_1	Bohachev-sky1	$f_1=-\left(x_1^2+2x_2^2-0.3cos\left(3\pi x_1\right)-0.4cos\left(4\pi x_2\right)+0.7\right)$	0	2	$[-100,100]^n$
f_2	Schaffer	$f_2=-\left(0.5+\frac{sin^2\left(\sqrt{x^2+y^2}\right)-0.5}{\left(1+0.001*\left(x^2+y^2\right)\right)^2}\right)$	0	2	$[-5.12,5.12]^n$
f_3	Needle in haystack	$f_3=\left(\frac{a}{b+\left(x^2+y^2\right)}\right)^2+\left(x^2+y^2\right)^2, \quad a=3.0,b=0.05$	3600	2	$[-5.12,5.12]^n$
f_4	Yang	$f_4=-\left(\left[e^{-\sum_{i=1}^{n}\left(x_i/\beta\right)^{2m}}-2e^{-\sum_{i=1}^{n}\left(x_i-\pi\right)^2}\right]\times\prod_{i=1}^{n}cos^2 x_i\right),$ $m=5,\beta=15$	1	2	$[-20,20]^n$
f_5	Goldstein and Price	$f_5=-\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\times$ $\left[30+\left(2x_1-3x_2\right)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	-3	2	$[-2,2]^n$
f_6	Hertman 3	$f_6=\sum_{i=1}^{4}c_i\,exp\left(-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	3.862782	3	$[0,1]^n$
f_7	Power Sum	$f_8=-\left(\sum_{k=1}^{n}\left[\left(\sum_{i=1}^{n}x_i^k\right)-b_k\right]^2\right),$ $b=[8,18,44,114],n=4$	0	4	$[-4,4]^n$
f_8	Kowalik	$f_9=-\left(\sum_{i=1}^{11}\left(a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right)^2\right),$ $a=[0.1957,\ 0.1947,\ 0.1735,\ 0.1600,0.0844,0.0627,$ $0.0456,\ 0.0342,0.0323,0.0235,0.0246];$ $b=[4.0,2.0,1.0,0.5,0.25,0.167,0.125,0.1,0.0833,0.0714,0.0625]$	-3.0748e-04	4	$[-5,5]^n$

Notes: where GM is the optimum, *D* is the dimension, and *S* is the variable ranges

Table 2. The parameters of Hertman 3 function

i	c_i	a_{i1}	a_{i1}	a_{i1}	p_{i1}	p_{i2}	p_{i3}
1	1.0	3.0	10	30	0.3689	0.1170	0.2673
2	1.2	0.1	10	35	0.4699	0.4387	0.7470
3	3.0	3.0	10	30	0.1091	0.8732	0.5547
4	3.2	0.1	10	35	0.03815	0.5743	0.8828

Table 3. Three formulas of contact factor

Z	The formula of ζ
sigmoid function	$\zeta_1 = 1/\left(1+e^{-\left(8-16\times\left(t/t_{max}\right)\right)}\right)$
exponential function	$\zeta_2=0.97^t$
linear function	$\zeta_3=1\text{-}0.9999*\dfrac{t}{t_{max}}$

Notes: t expresses the current iteration; t_{max} is the iteration number.

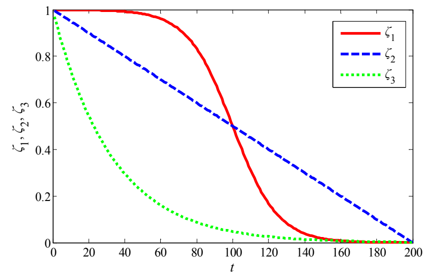

Fig. 1. Three different contact factor ζ decreases with the iteration

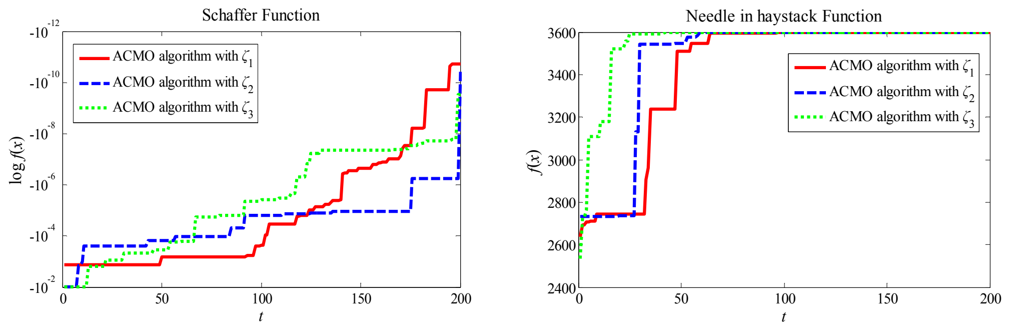

Fig. 2. The convergence process of ACMO algorithm with the different contact factor on f_2 f_3 functions

Table 4 shows the results of ACMO algorithm with ζ_1, ζ_2 and ζ_3, which includes the percentage of success (as represented by the number of trials required for the object function to reach its known target values, which is the absolute error value between optimum solution obtained and the true optimum is equal to or lower than 1e-4 in the all trials), the average value and the standard deviation of the solution obtained in all trials. The best results are bold.

In the trials, other parameters of ACMO algorithm are set as follows. Each dimension of the search space is divided into M small intervals. Here let M be 5; the total droplets number N in search space is equal to 100; the lowest droplets number of one cloud dN=5; the initial hyper entropy HeM^0=0.5; the threshold factor λ=0.7; the weaken rate γ=0.2. The iteration number t_{max}=200; the experiment is repeated to 50 times.

From the results it can be seen that for f_1~f_6 functions, no matter the contact factor is ζ_1, ζ_2 or ζ_3, ACMO algorithm can fully obtain the global optimum except that the success rate of the algorithm on the Schaffer function is 98%; the convergence precisions of ACMO algorithm with ζ_1 on all functions except Hertman 3 are all the best, while the results of ACMO algorithm with ζ_3 are the worst. For the complex functions f_7~f_8, the results obtained by ACMO algorithm with ζ_1 are the best in both terms of Suc. and Mean(SD), while the results obtained by ACMO algorithm with ζ_3 are also the worst.

The experiment results proof that contact factor in the sigmoid function model can enhance the global search ability of ACMO algorithm, therefore the sigmoid function is selected as the contact factor of ACMO algorithm.

Table 4. The simulation results of ACMO algorithm with the different contact factor ζ

	ζ_1		ζ_2		ζ_3	
	Suc.	Mean (SD)	Suc.	Mean (SD)	Suc.	Mean (SD)
f_1	**50/50**	**-3.99183e-008 (4.40892e-008)**	**50/50**	-1.2341e-07 (1.58504e-07)	**50/50**	-2.40108e-07 (2.35325e-07)
f_2	**50/50**	**-9.52418e-010 (7.55987e-010)**	**50/50**	-7.74867e-08 (1.99375e-07)	49/50	-0.000194325 (0.00137404)
f_3	**50/50**	**3600 (1.85428e-005)**	**50/50**	3600 (0.00137675)	**50/50**	3600 (0.000132644)
f_4	**50/50**	**1 (1.89865e-010)**	**50/50**	1 (4.60707e-09)	**50/50**	1 (1.40315e-09)
f_5	**50/50**	**-3 (2.82569e-010)**	**50/50**	-3 (3.29799e-09)	**50/50**	-3 (2.70427e-09)
f_6	**50/50**	3.86278 (3.5969e-006)	**50/50**	3.86277 (1.20004e-05)	**50/50**	**3.86278 (7.40473e-08)**
f_7	**14/50**	**-0.000311214 (0.000312732)**	4/50	-0.000841436 (0.000795419)	10/50	-0.000470356 (0.000594141)
f_8	**23/50**	**-0.000406317 (6.31383e-005)**	10/50	-0.000467008 (6.25066e-05)	16/50	-0.000419879 (5.27768e-05)

Notes: Suc. denotes the percentage of success; Mean(SD) means the average value and standard deviation of the solution obtained in the all trials.

3.1.2 The effect of the weaken rate γ to the performance of ACMO algorithm

The weaken rate γ is mainly to control the decreased speed of droplets number of the cloud. This section gives the value setting of the weaken rate through the following experiment.

First, the effect of the performance of ACMO algorithm is tested when γ =0.1, 0.2, 0.3 and 0.4 respectively. Table 1 shows the test functions. The setting of the other parameters is the same to Section 3.1.1 and the results are listed in Table 5. Fig. 3 shows the average convergence process of ACMO algorithm when γ =0.1, 0.2, 0.3 and 0.4 on Schaffer function and Kowailk function respectively.

From Table 5, the results are the worst when γ =0.1; the success numbers of ACMO algorithm decrease in turn when γ =0.2, 0.3 and 0.4, except the case that the results of when γ =0.3 and 0.4 are superior to the result of when γ =0.2 on Power Sum function. In summary, the performance of ACMO algorithm is the best when γ =0.2. If the weaken rate is too large or too small, it will weaken the global search ability of ACMO algorithm.

The reason is if γ is set too large, the clouds would dissolve before they leave the generation regions or spread out. It would result in narrowing the search space; if γ is set too small, the clouds survive so long that the droplets newly generated in the potential regions would be too few. It would lead to the lower convergence accuracy of ACMO algorithm. In this article, γ is set to 0.2, where ACMO algorithm can obtain the best performance.

Table 5. The simulation results of ACMO algorithm with different weaken rate γ

γ / Suc.	0.1	0.2	0.3	0.4
f_1	0/50	**50/50**	**50/50**	**50/50**
f_2	10/50	**50/50**	**50/50**	49/50
f_3	0/50	**50/50**	48/50	49/50
f_4	0/50	**50/50**	**50/50**	**50/50**
f_5	37/50	**50/50**	**50/50**	**50/50**
f_6	1/50	**50/50**	**50/50**	**50/50**

f_7	0/50	14/50	**17/50**	16/50
f_8	0/50	**23/50**	20/50	18/50

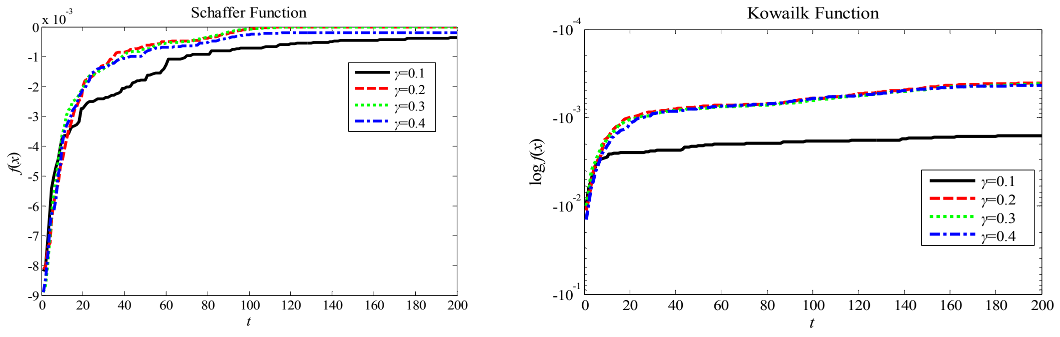

Fig. 3. The convergence process of ACMO algorithm with the different contact factor on f_2 f_8 functions

3.1.3 The effect of the different combination of parameters to the performance of ACMO algorithm

As presented in Equation (13), the cloud is currently located in the region U_S, which is the same as the target region U_T. Let the spread factor α be 0.3. And for the purpose to test the effect of the different parameters, the experiment results of ACMO algorithm is given on the functions f_2, f_3, f_4 f_5, f_7, f_8 when λ=0.6, 0.7; γ=0.2, 0.3; and α=0.3, 0.4 respectively.

Table 6 shows the values of λ, γ and α when ACMO algorithm obtains the best results on the test functions. From the results it can be seen that when α=0.3, the performance of ACMO algorithm is always better than that when α=0.4. From the term of the threshold factor λ or the weaken rate γ, the results have no features. However, combining the both terms, it can be seen that ACMO algorithm obtains more global optima when λ=0.6, γ=0.3 or λ=0.7, γ=0.2. This result indicates that when the threshold factor is set to a small value, there will be more regions generating the clouds. Therefore ACMO algorithm requires a large weaken rate to reduce the survival time of clouds to make sure that the algorithm has enough droplets to generate clouds. When the threshold factor is set a large value, there will be fewer regions generating the clouds. Therefore, ACMO algorithm requires a small weaken rate to make sure that the clouds have enough time to spread to the new regions for the global search. Here let λ=0.7, γ=0.2 as the standard value of ACMO algorithm.

Table 6. The setting of the parameters when the performance of ACMO algorithm is the best

Functions	threshold factor λ	weaken rate γ	spread factor α
f_2	0.6	0.3	0.3
f_3	0.7	0.2	0.3
f_4	0.7	0.3	0.3
f_5	0.6	0.3	0.3
f_7	0.6	0.2	0.3
f_8	0.7	0.2	0.3

3.2 Comparison with PSO and GA

In this section, the performance of ACMO is compared to PSO and GA on each test function listed in Table 1.

3.2.1 Experiment setting

The parameters of the ACMO algorithm is set as summarized in Section 3.1. PSOt (a particle swarm optimization toolbox for MATLAB) is employed to execute the standard PSO. The maximum velocity *Vmax* in standard PSO is set as the half of the length of search space. The GA algorithm is accomplished by the GATBX toolbox. The chromosome type is binary coded. The selection operator is roulette wheel selection, the crossover operator is single-point crossover, and the mutation operator is uniform mutation. The chromosome length is set to 22, the crossover rate *Pc*=0.8, and the mutation rate *Pm*=0.02. To make it reasonable, the populations of PSO and GA are set the same and equal to the total droplets number in this article. The maximum number of the iterations is set to 200, and each experiment is repeated 50 times.

The experiments are carried out on a PC with a 2-GHz Intel Processor and 1.0-GB RAM. The operating system is Microsoft Windows 7. The algorithm is written and executed in MATLAB 7.1.

3.2.2 Experiment results

The results are summarized in Table 7. And the best convergence processes comparison of ACMO, PSO and GA algorithm on Needle in haystack function and Power sum function respectively are illustrated in Fig. 4.

As observed from Table 7, all the three algorithms can find the optimum 100% for f_1 function and f_6 function, while the results of PSO are the best in terms of Mean (SD); for f_2 function and f_5 function, both ACMO algorithm and PSO algorithm can find the optimum in probability of 100%, while the success rates of GA are 0% and 92% respectively, and the terms of Mean (SD) of PSO algorithm are still superior to the proposed algorithm; for the f_3, f_4, f_7, f_8 functions, ACMO algorithm is better than PSO algorithm and GA algorithm in terms of the success rate and Mean (SD). In a word, the proposed algorithm is the best in terms of the success rate, while the convergence accuracy of PSO algorithm is higher than ACMO algorithm with the same success rate.

From the convergence curves shown in Fig. 4, it can be seen obviously that ACMO algorithm has a slower convergence speed than the other two algorithms in the early stage of iteration. However, as the optimization process goes to the middle and later stages of the search, the PSO algorithm and GA algorithm are stalled gradually, while ACMO algorithm still has a good evolutionary ability to escape from the local optima. The combining results can be concluded in Table 7. Compared with PSO and GA, the ACMO algorithm has a better ability to solve the multimodal functions.

Table 7. Comparison with PSO and GA

functions	PSO		GA		ACMO	
	Sec.	Mean (SD)	Sec.	Mean (SD)	Sec.	Mean (SD)
f_1	**50/50**	**0 (0)**	**50/50**	-5.6974e-006 (8.28916e-006)	**50/50**	-3.99183e-008 (4.40892e-008)
f_2	**50/50**	**0 (0)**	0/50	-0.00958864 (0.00089179)	**50/50**	-2.79951e-010 (3.82982e-010)
f_3	27/50	3208.44 (428.552)	49/50	3599.97 (0.0782634)	**50/50**	**3600 (1.85428e-005)**
f_4	35/50	0.7 (0.46291)	0/50	0.0325908 (0.154231)	**50/50**	**1 (1.89865e-010)**
f_5	**50/50**	**-3 (1.53313e-015)**	46/50	-5.16001 (7.39929)	**50/50**	**-3** (2.82569e-010)
f_6	**50/50**	**3.86278 (3.0213e-015)**	**50/50**	3.86278 (2.24845e-007)	**50/50**	3.86278 (3.5969e-006)
f_7	10/50	-0.00515205 (0.00980218)	0/50	-0.305182 (0.367386)	**14/50**	**-0.000311214 (0.000312732)**
F_8	12/50	-0.000581606 (0.000246696)	0/50	-0.00262072 (0.00537353)	**23/50**	**-0.000406317 (6.31383e-005)**

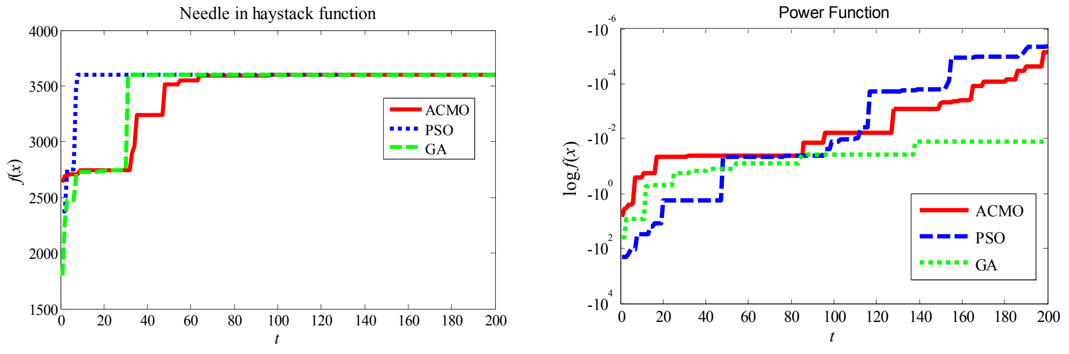

Fig. 4. Comparison of the three convergence curves among ACMO, PSO and GA algorithm.

4 Conclusion

In this article, a novel Atmosphere Clouds Model Optimization Algorithm is formulated based on the simulation of the generation behavior, move behavior and spread behavior of cloud in a simple way.

Numerous numerical simulations are performed to demonstrate the effectiveness of ACMO algorithm. The experiment studies suggest that the ACMO algorithm can prevent premature convergence and the performance of ACMO is comparable with other evolutionary algorithms. The application on the PID parameter tuning also proves the feasibility of ACMO algorithm.

The behaviors of cloud in nature is extremely complex, while the algorithm proposed in this article just mimes the behaviors of cloud in a simple way, which means that there are a lot of research spaces in this model. Therefore, we will do the further research of this model in the future.

Acknowledgment

This article is supported by The Science Foundation’s Project of Shanxi Province (No.2011011012-2), the Youth Science Foundation’s Project of Shanxi Province (No. 2011011011-1), the Program for the Top Young Academic Leaders of Shanxi Province(TYAL), and the Science and Technique Project of Shanxi Province (No. 20110321036).

References

[1] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, 1998.

[2] T. Krink, J.S. VesterstrOm, J. Riget, "Particle Swarm Optimisation with Spatial Particle Extension," in *Proceedings of 2002 Congress on Evolutionary Computation*, IEEE Press, pp. 1474-1479, 2002.

[3] F. Glover, M. Laguna, *Tabu Search*, Springer, 1998.

[4] E. Bonabeau, M. Dorigo, G. Theraulaz, "Inspiration for Optimization from Social Insect Behavior," *Nature*, Vol. 406, No. 6791, pp. 39-42, 2000.

[5] H. Al-Qaheri, A. Mustafi, and S. Banerjee, "Digital Watermarking Using Ant Colony Optimization in Fractional Fourier Domain," *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 1, No.3, pp. 179-189, 2010.

[6] K.M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control," *Journal of IEEE Control Systems*, Vol. 22, No. 3, pp. 52-67, 2002.

[7] H.-M. Feng, J.-H. Horng, S.-M. Jou, "Bacterial Foraging Particle Swarm Optimization Algorithm Based Fuzzy-VQ Compression Systems," *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 3, No.3, pp. 227-239, 2012.

[8] A.R. Mehrabian, C. Lucas, "A Novel Numerical Optimization Algorithm Inspired From Weed Colonization," *Ecological Informatics*, Vol. 1, No. 4, pp. 355-366, 2006.

[9] R. Oftadeh, M.J. Mahjoob, M. Shariatpanahi,"A Novel Meta-Heuristic Optimization Algorithm Inspired by Group Hunting of Animals: Hunting Search," *Computers & Mathematics with Applications*, Vol. 60, No. 7, pp. 2087-2098, 2010.

[10] X.-S. Yang, "A New Metaheuristic Bat-Inspired Algorithm," in *Proceedings of Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)*, Vol. 284 of Studies in Computational Intelligence, pp. 65–74, 2010.

[11] X.-S. Yang, A.H. Gandomi, "Bat Algorithm: A Novel Approach for Global Engineering Optimization," *Engineering Computations*, Vol.29, No.5, pp. 464-483, 2012.

[12] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," in *Proceedings of Stochastic Algorithms: Foundations and Applications (SAGA 2009)*, Vol. 5792 of Lecture Notes in Computer Science, pp. 169–178, 2009.

[13] O.K. Erol, I. Eksin, "A New Optimization Method: Big Bang–Big Crunch," *Adv Eng Software*, Vol. 37, No. 2, pp. 106–111, 2006.

[14] A. Kaveh, S. Talatahari, "A Novel Heuristic Optimization Method: Charged System Search," *Acta Mech*, Vol. 213, No. 3-4, pp. 267–289, 2010.

[15] Z.W. Geem, J.-H. Kim, G.V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *Simulation*, Vol. 76, No. 2, pp. 60–68, 2001.

[16] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, "Mine Blast Algorithm: A New Population Based Algorithm for Solving Constrained Engineering Optimization Problems," *Applied Soft Computing*, Vol. 13, No. 5, pp. 2592-2612, 2013.

[17] M.J. Harris, W.V. Baxter, T. Scheuermann, et al., "Simulation of Cloud Dynamics on Graphics Hardware," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, pp. 92-101, 2003.

[18] D.Y. Li, C.Y. Liu, L.Y. Liu, "Study on the Universality of the Normal Cloud Model," *Engineering Science*, Vol. 6, No. 8, pp. 28-34, 2004.

[19] C. Y. Liu, D.Y. Li, L.L. Pan, "Uncertain Knowledge Representation Based on Cloud Model," *Computer Engineering and Applications*, Vol. 40, pp. 32-35, 2004.

[20] Y. Liu, D.Y. Li, G.W. Zhang, et al, "Atomized Feature in Cloud Based Evolutionary Algorithm," *Acta Electronica Sinica*, Vol. 37, pp. 1651-1658, 2009.

[21] F. Kang, J. Li, Z. Ma, "Rosenbrock Artificial Bee Colony Algorithm for Accurate Global Optimization of Numerical Functions," *Information Sciences*, Vol. 181, No. 16, pp. 3508-3531, 2011.

[22] Huang C F, Bieniawski S, Wolpert D H, et al., "A Comparative Study of Probability Collectives Based Multi-Agent Systems and Genetic Algorithms," in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, IEEE Press, pp.751-752, 2005.

[23] X.-S. Yang, "Firefly Algorithm, Stochastic Test Functions and Design Optimization," *International Journal of Bio-Inspired Computation*, Vol. 2, No. 2, pp. 78-84, 2010.