Particle Swarm Optimization Algorithms Inspired by Immunity-Clonal Mechanism and Their Applications to Spam Detection

Ying Tan, Peking University, China

ABSTRACT

Compared to conventional PSO algorithm, particle swarm optimization algorithms inspired by immunity-clonal strategies are presented for their rapid convergence, easy implementation and ability of optimization. A novel PSO algorithm, clonal particle swarm optimization (CPSO) algorithm, is proposed based on clonal principle in natural immune system. By cloning the best individual of successive generations, the CPSO enlarges the area near the promising candidate solution and accelerates the evolution of the swarm, leading to better optimization capability and faster convergence performance than conventional PSO. As a variant, an advance-and-retreat strategy is incorporated to find the nearby minima in an enlarged solution space for greatly accelerating the CPSO before the next clonal operation. A black hole model is also established for easy implementation and good performance. Detailed descriptions of the CPSO algorithm and its variants are elaborated. Extensive experiments on 15 benchmark test functions demonstrate that the proposed CPSO algorithms speedup the evolution procedure and improve the global optimization performance. Finally, an application of the proposed PSO algorithms to spam detection is provided in comparison with the other three methods.

Keywords: *Advance-and-Retreat, Black Hole, Clonal Particle Swarm Optimization, Immune Clonal Strategy, Particle Swarm Optimization, Spam Detection*

INTRODUCTION

Particle swarm optimization (PSO) is a stochastic global optimization technique inspired by social behavior of bird flocking or fish schooling. In the conventional PSO suggested in Kennedy and Eberhart (1995) and Eberhart and Kennedy (1995), each particle in a population adjusts its position in the search space according to the best position it has found so far, and the position of the known best-fit particle in the entire population. Compared to other population-based algorithms, i.e., genetic

DOI: 10.4018/jsir.2010010104

algorithms, the PSO does not need genetic operators such as crossover and mutation. Thus it has advantages of easy implementation, fewer parameters to be adjusted, strong capability to escape from local optima as well as rapid convergence. As a result, the PSO outperforms other population-based algorithms in many real-world application domains.

In recent years, the PSO has been increasingly used as an efficient technique for solving complicated and hard optimization problems, such as function optimization, evolving artificial neural networks, fuzzy system control, optimization in dynamic and noisy environments, blind source separation, machine learning, games, to name a few. Furthermore, the PSO has also been found to be robust and fast in solving non-linear, non-differentiable and multi-modal problems (Ge & Zhou, 2005). Therefore, it is very important and necessary to exploit some new mechanisms and principles to improve and promote the performance of the conventional PSO for a variety of problems in practice. In this article, the clonal mechanism found in natural immune system of creatures is introduced into the PSO, resulting in a novel clonal PSO (CPSO, for short). In addition, in order to improve the CPSO further, an advance-and-retreat(AR) strategy and the concept of random black hole(RBH) are then introduced into the CPSO, resulting in two variants of the CPSO, called CPSO with AR strategy (AR-CPSO, for short) and RBH model (RBH-PSO, for short).

This article is an extended version of our earlier short paper (Tan & Xiao, 2007), in which a basic idea of the CPSO is briefly presented. Here, we have extended it substantially and included two variants with some deep discussions, comprehensive experimental studies as well as our application to spam detection.

The remainder of this article is organized as follows. Section II describes the conventional PSO algorithm and its related modification versions. Section III presents the proposed CPSO by introducing the clonal mechanism in NIS into the conventional PSO and its implementation. Section IV improves the CPSO by introducing the AR strategy and the RBH model. Section V gives several experimental results to illustrate the effectiveness and efficiency of the proposed algorithms in comparison with the conventional PSO. An application of spam detection is also given in details in section VI. Finally, concluding remarks are drawn in Section VII.

RELATED WORKS

Conventional PSO

In the conventional PSO algorithm, each potential solution to an optimization problem is considered as a particle in the search space, and a population of particles called a swarm is used to explore the search space. All of particles in the swarm have their fitness values which are evaluated by a fitness function related to the optimization problem to be solved. Therefore, the PSO algorithm is originally initialized with a swarm of particles randomly placed on the search space. Then the randomly initialized swarm is getting to start to search for the optimal solution to the optimization problem by evolving iteratively. In each iteration, the position and the velocity of each particle are updated according to its own previous best position ($Pi_{Bd}(t)$) and the current best position of all particles($PgBd(t)$) in the swarm. The update formula for the velocity and position of each particle in the conventional PSO is written as

$$V_{id}(t+1) = wV_{id}(t) + c_1 r_1 (P_{iBd}(t) - X_{id}(t))$$
$$+ c_2 r_2 (P_{gBd}(t) - X_{id}(t)), \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1), \quad (2)$$

where i =1, 2, ⋯, n, n is the number of particles in the swarm, d =1, 2, ⋯, D, and D is the dimension of solution space.

In Eqs. (1) and (2), the learning factors $c1$ and $c2$ are nonnegative constants, $r1$ and $r2$ are random numbers uniformly distributed in the interval [0,1], $Vid \in [-V_{max}, V_{max}]$, where V_{max} is a designated maximum velocity which is a constant preset by users according to the objective function of optimization. The velocity on one dimension which exceeds the maximum will be set to V_{max}. This parameter controls the convergence rate of the PSO and can prevent it from growing too fast. The parameter $w \in [0, 1]$ in Eq. (1) is the inertia weight used to balance the global and local search abilities. A large inertia weight is more appropriate for global search while a small inertia weight facilitates local search.

The termination criterion for iterations in the PSO is determined by whether reaching the fixed maximum number of fitness evaluations or a designated value of the fitness.

For convenience, we call the PSO in Eqs. (1) and (2) as standard PSO (abbreviated as SPSO) in the remainder of this article.

Improvements of the PSO

Since its invention, the PSO has attracted an extensive attentions and interests of researchers from different scientific and engineering domains. Many researchers have worked on improving its performance in various ways, thereby deriving many interesting improvements of the PSO.

One of the improvements introduced a linearly decreasing inertia weight over the course of search by Shi and Eberhart (1998) and gave a good convergence performance. A smart technique for creating a binary PSO suggested by Kennedy and Eberhart in 1997 used the concept of velocity as a probability that a bit takes on one or zero. Furthermore, by analyzing the convergence behavior of the PSO, a variant of the PSO with a constriction factor was introduced by Clerc and Kennedy (2002), which guarantees the convergence and improves the convergence speed sharply simultaneously. Parsopoulos and Vrahatis proposed a unified particle swarm optimizer (UPSO) which combined both the global version and local version together (Parsopoulos, & Vrahatis, 2004). A cooperative particle swarm optimizer was also proposed in (Bergh & Engelbrecht, 2004). Furthermore, El-Abd and Kamel proposed a hierarchal cooperative particle swarm optimizer (El-Abd & Kamel, 2006). In Peram, Veeramachaneni and Mohan (2003), proposed the fitness-distance-ratio based particle swarm optimization (FDR-PSO),by defining the "neighborhood" of a particle as the n closest particles of all particles in the population. In Pereira and Fernandes (2005) and Ismael and Fernandes (2005), a SAPSO algorithm combined the particle swarm optimization with the simulated annealing. The SAPSO can narrow the field of search and speedup the rate of convergence continuously during the optimizing process. Recently, a comprehensive learning particle swarm optimizer (CLPSO) was proposed to improve the performance of the conventional PSO on multi-modal problems by a novel learning strategy (Liang, Qin, Suganthan & Baskar, 2006). A stretching technique was introduced into the PSO by Parsopoulos, Plagianakos, Magoulas and Vrahatis (2001), which applied a two-stage transformation to the shape of the fitness function that eliminates undesired local minima but preserves the global minimum.

Although there are numerous improved versions of the PSO, they almost need much time to accomplish the evaluations of fitness function, and give similar results in the early phase of convergence. Hence, we here choose the improvement of the PSO with the inertia weight as a foundation of our standard PSO for further comparisons in the rest of this article.

CLONAL PARTICLE SWARM OPTIMIZATION

Clonal Expansion Process in Nature Immune System

Artificial immune system (AIS) is a novel computational intelligence paradigm inspired by the natural immune system (NIS). Like artificial neural networks and genetic algorithm, AIS are highly abstract models of their biological counterparts applied to solve a number of complex problems in different domains. Some work processes in NIS are used as metaphors to develop novel computing models in computational intelligence, such as negative selection, clonal selection, to name a few, to solve many complex problems in science and engineering domain (Dasgupta & Attoh-Okine, 1997; Castro & Timmis, 2003; Castro, 2002).

Originally, according to clonal selection theory, when the B-and T-lymphocytes in NIS recognize an antigen as non-self, the NIS will start to proliferate by cloning upon recognition of such antigen. When a B cell is activated by binding an antigen, many clones are produced in response, via a process called clonal expansion. The resulting cells can undergo somatic hyper mutation, creating offspring B cells with mutated receptors. The higher the affinity of a B cell to the available antigens, the more likely it will clone. This is called as a Darwinian process of variation and selection, i.e., affinity maturation (Dasgupta & Attoh-Okine, 1997; Castro & Timmis, 2003).

The essence of the SPSO is to use these particles with best known positions to guide the swarm or the population to converge to a single optimum in the search space. However, how to choose the best-fit particle to guide each particle in the swarm is a critical issue. This becomes even more acute when the problem to be solved has multiple optima since the entire swarm could potentially be misled to local optima. In order to deal with this case, a clonal expansion in NIS is probably a good way to guide or direct the SPSO escaping from local optima whilst searching for the global optima efficiently. Therefore, here we want to introduce the clonal expansion process in NIS into the SPSO to strength the interaction between particles in a swarm for improving its convergent performances and global optimization capability greatly.

Clonal Particle Swarm Optimization Algorithm

According to the clonal expansion process in NIS discussed above, we propose a clonal operator for the SPSO. The clonal operator is at first to clone one particle as N same particles in the solution space according to its fitness function, then generate N new particles via clonal mutation and selection processes which are related to the concentration mechanisms used for antigens and antibodies in NIS. Here we call the SPSO with such clonal operator as clonal particle swarm optimization (for short, CPSO) algorithm. For simplification in presentation, we will use the abbreviated CPSO algorithm directly later on.

As indicated in Liang, Qin, Suganthan and Baskar (2006), CLPSO's learning strategy abandons the global best information, the past best information of other particles is used to update the particles' velocity instead. In such a way, the CLPSO can significantly improve the performance of the SPSO on multi-modal problems.

Here in order to present our CPSO clearly and efficiently, we adopt the similar definitions used in AIS paradigms. Antigen, antibody, and the affinity between antigen and antibody are corresponding to objective optimization function, solution candidate, and the fitness value of the solution on the objective optimization function, respectively. The clonal operator is used to duplicate one point as N same points according to its fitness function, and then generate N new particles by undergoing mutation and selection operations. In general, the state transition process of a swarm of particles in the CPSO can be schematically expressed as follows:

$$P(t) \xrightarrow{clone} C(t) \xrightarrow{mutation} M(t) \xrightarrow{sel} P(t+1) \tag{3}$$

where the arrow represents the transition process between two states while symbols over the arrows show the operations needed for the transition processes.

Notice that the population of particles $P(t)$ at time t can be transited as $C(t)$ via a clonal process, then next generation population $P(t+1)$ can be generated by using mutation and selection processes for the cloned population $C(t)$.

Briefly, the CPSO algorithm can be summarized in Algorithm 1.

Algorithm 1 CPSO Algorithm

- **Step 1: Initialization.** **Assume** a =1, $c1$ =2, $c2$ =2, and w be from 0.9 to 0.4 linearly.
- **Step 2:** The state evolution of particles is iteratively updated according to Eqs. (1) and (2).
- **Step 3:** Memory the global best-fit particle of each generation, P_{gB}, as a mother particle of the clonal operator in Step 4.
- **Step 4:** After M generations, clone the memorized M global best particles, $P_{gB}(i)$, i =1, ⋯, M.
- **Step 5:** Mutation Process. All of the cloned particles are mutated to some extents to differentiate with original or mother particle by using some random disturbances such as Gaussian noise. Assume P_{gBk} be the k-th entry of the vector P_{gB} and μ is an Gaussian random variable with zero mean and unity variance, then one can have the following random mutation process

$$P_{gB_k} = P_{gB_k} + s * (1 - \mu) * V_{\max} \tag{4}$$

  where s is the scale of mutation and V_{max} is the max velocity.
- **Step 6:** Selection Process. We **store** the current P_{gB} in memory, but the other particles are selected according to a strategy of the diversity keeping of the concentration mechanism so that in next generation of particles, a certain concentration of particles will be maintained for each fitness layer. Here the concentration of i-th particle are defined as follows:

$$D(x_i) = (\sum_{j=1}^{N+M} | f(x_i) - f(x_j) |)^{-1},$$

$$i = 1,2,\ldots,N + M \tag{5}$$

  where xi and f(xi) in Eq. (5) denote the i-th particle and its fitness value, respectively.

  According to above Eq. (5), one can derive a selection probability in terms of the concentration of particles as

$$p(x_i) = \frac{\frac{1}{D(x_i)}}{\sum_{j=1}^{N+M} \frac{1}{D(x_j)}}, i = 1,2,\ldots,N + M \tag{6}$$

- **Step 7:** Termination. The algorithm can be terminated by some common stop criteria such as a given maximum number of fitness evaluations or a presetting accuracy of the solution. In our experiments in the article, we adopt the former stop criterion, i.e. a maximum number of fitness evaluations, which is 1,200,000.

It can be seen from Eqs. (5) and (6) that the more the particles are similar to the antibody i, the less the probability the particle i can be chosen, and vice versa. In such a way, the particle with low fitness value also has an opportunity to evolve. Therefore, this kind of probability selection mechanism in terms of the concentration of particles in the swarm is able to guarantee the diversity of antibodies theoretically and endows the method with the ability of escaping from local minima.

Through keeping current global optima, the proposed CPSO algorithm can guarantee to maintain the good convergent performance of original SPSO. In the meantime, the essence of the clonal operator is to generate a new particle swarm near the promising candidate solution according to the value of the fitness function such that the search space are enlarged greatly and the diversity of clones is increased to avoid trapping in local minima. So, the speed of convergence and the global optimization capability can be raised rapidly.

Analysis of the CPSO

The essence of the CPSO is making full use of the area around the current best particle, denoted as x_{gB}, in depth, which works well for two reasons. First of all, we assume the probability that the actual global solution lies in the range around the current best particle would be probably greater than that in the other space. Secondly, when the SPSO converges to one solution, it is supposed to be the current best-fit position x_{gB}. If x_{gB} is the global best particle, the CPSO will speed up the convergence of the evolving swarm, because we have cloned more particles which are very close to x_{gB}, and search the area around x_{gB} more thoroughly and completely. If x_{gB} f alls into a local optimum, which means a premature for the SPSO, the CPSO can give x_{gB} another chance to escape from trapping in the local minima by using the mutation and selection operations that keep the diversity of the swarm.

However, the clones may be not efficient enough to find nearby minima in an enlarged unknown space after the clonal operation. Moreover, as can be seen, the above CPSO algorithm has complex operations which lead to much more computational time and preserve more memory. In addition, the clonal selection cannot be tuned easily for a specific task. So, by introducing the advance-and-retreat (AR) strategy, and random black hole (RBH) model into the CPSO, we propose two variants, i.e., AR-CPSO and RBH-PSO (Zhang, Xiao, Tan & He, 2008; Zhang, Liu, Tan, & He, 2008), to overcome these two limitations.

In the AR-CPSO, the AR strategy endows the clones with faster speed to find nearby local basins by using the history information of each particle's last performance of "flying" after each clonal operation. In the next clonal operation, clonal mutation and selection of the best individual of a number of succeeding generations enlarge the search space greatly and increase the diversity of clones to avoid being trapped in local minima. Thus, the clones have more chances to find and flee the nearby local basins with fast speed.

Black hole model in physics is inspired by the concept of black holes in the outer space. A black hole is a highly dense star that exerts a strong force on other stars and matter around it. It is impossible to see a black hole directly because no light can escape from it, so it is always black. But when it passes through a cloud of interstellar matter, or is close to another "normal" star, the black hole can accrete matter into itself. So we can estimate the position of the black hole according to the X-ray emission curves of matter which is being magnetized by it (NASA, n.d.). Here we still use the clonal operation, but in each generation, we clone one particle of x_{gB}, which is set to be more powerful but more stochastic, like a black hole in outer space in physics. In such a way, we can accelerate the convergence rate considerably. On the other hand, instead of the mutation and selection operations in the CPSO, our black hole model employs the randomness to keep the diversity of the swarm and enlarge the search space in the meantime, which is very simple and effective.

TWO VARIANTS OF CPSO

CPSO with AR Strategy

Many researches focus on improving the convergent capability of the PSO by a variety of methods. In Liu, Qin and Shi (2004), golden division algorithm is introduced into the particle swarm optimization algorithm. In PSO-LS (Chen, Qin, Liu & Lu, 2005), each particle has a chance of self-improvement by applying local search algorithm before it communicates with other particles in the swarm. Hybrid Gradient descent PSO (HGPSO) (Noel & Jannett, 2004) algorithm makes use of gradient information to achieve a fast convergence. In this combination, the third part of the original evolving equation, i.e., local best solution is replaced by a gradient term. Multi-Local PSO (MLPSO) algorithm (Vaz & Fernandes, 2005) uses gradient descent directions to drive each particle to a nearby local minimum for locating multiple solutions. The second part of the original PSO equation, called global best solution, is replaced by the steepest descent direction evaluated at the best ever particle position. In these two methods, the gradient or the approximate gradient is used to increase the convergent ability in the PSO.

1. **AR strategy:** The AR strategy is a simple and effective method for the problem of one-dimensional search. One-dimensional search is also called linear search for optimization of a single-variable objective function. The iterative formula in one-dimension search is as follows.

$$x_{k+1} = x_k + v_k d_k \tag{7}$$

where *xk* denotes the position of a solution, v*k* the velocity of a solution and *dk* the direction of the velocity. The bottleneck problem in Eq. (7) is how to determine the search direction dk and the step-size *vk*. Let

$$\varphi(v_k) = f(x_k + v_k d_k) \tag{8}$$

where $\varphi(.)$ denotes the function value of the velocity *vk*, *f* is the objective function.

The problem of how to determine the step-size *vk* and the search direction *dk* in Eq.(8) is just an one-dimensional search problem such that makes

$$\varphi(v_k) < \varphi(0) \tag{9}$$

The step-size *vk* could be optimal if the step-size *vk* minimizes the objective function along the search direction *dk* as in Eq. (10).

$$\varphi(v_k) = \min_v \varphi(v), v > 0 \tag{10}$$

In practice, the optimal step-size is hard to determine analytically, and often requires expensive computational cost. Therefore, an approximate one-dimensional search with less cost becomes increasingly popular.

It is well known that the AR strategy is a simple and effective method in one-dimensional search, whose main principle is to start a particle from one point with a certain step-size to determine three points of 'high-low-high', then calculate the distance from the point of 'low', i.e., the approximate optimal step-size. If a particle succeeds in one direction, its search direction remains unchanged. Otherwise, it will return and search along its opposite direction. Finally, algorithm 2 outputs an interval which contains the minimum of an unimodal function. In summary, Algorithm 2 shows steps of the AR strategy in detail.

Algorithm 2 Advance-and-Retreat Algorithm

```
Step 1: Initialization: v0 ∈ [0, ∞), h0 > 0, acceleration
factor α > 1, compute φ(v0), k=0.
Step 2: Compare Fitness Values:
vk+1 = vk + hk
φk+1 = φ(vk+1)
if φk+1 < φk then
      go Step3
else
      go Step4
end if
Step 3: Advance:
hk+1 = αhk
v = vk
vk = vk+1
φk = φk+1
k = k +1
go Step2
Step 4: Retreat:
if k =0 then
      hk = −hk //reverse the
search direction vk = vk+1
      go Step2
else
      stop
end if
Step 5:
a = min{v, vk+1}
b = max{v, vk+1}
output [a,b]
```

2. AR-CPSO: In each iteration, we use the AR strategy to replace the first part (i.e., the previous velocity of a particle) of Eq. (1) in SPSO just for the cloned particles. When the fitness value

turns better after the last ”flying”, the cloned particle advances according to Eq. (1). On contrary, when the fitness value turns worse after the last ”flying”, the cloned particle then retreats the searches in the reverse direction of the last ”flying” with a smaller step-size of the previous velocity, which can be formulated as

$$V_{id}(t+1) = w(-\alpha V_{id}(t)) + c_1 r_1 (P_{iBd}(t) - X_{id}(t)) + c_2 r_2 (P_{gBd}(t) - X_{id}(t)) \quad (11)$$

where $\alpha < 1$.

The AR-CPSO algorithm is the same as the CPSO algorithm in Algorithm 1 except for replacing its Step2 with the following step.

Step2: The state evolution of particles is iteratively updated according to Eqs. (1), (2) and (11).

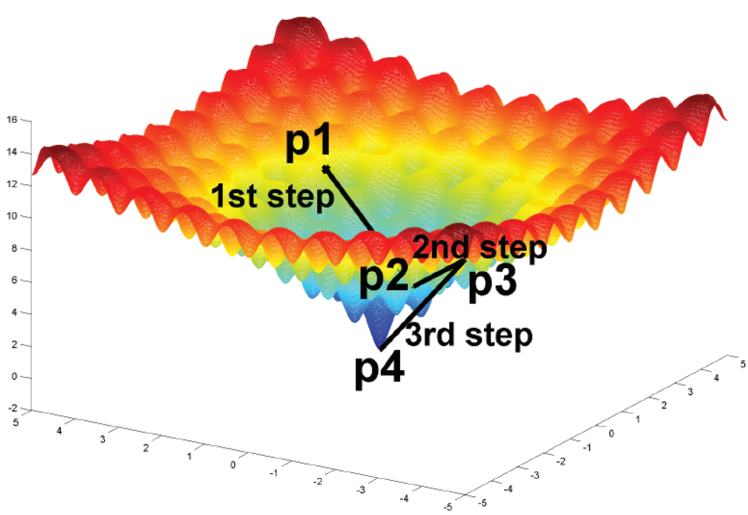
With the inertia weight w decreasing with the evolution of the swarm, clones may be restricted in a decreasing local area for searching nearby local minima. Due to the influence of the global best position and the local best positions, clones change their tracks randomly. Noticeably, the AR strategy is just applied to the cloned particles.

The AR strategy in the AR-CPSO on Ackley benchmark test function is schematically shown in Figure 1. As can be seen, a particle starts from $p1$, after the 1st step, the particle advances in the 2nd step and retreats in the 3rd step. Thus, the clones do not scatter over the search space, but fly toward the nearby local basin quickly. Therefore, the AR strategy enables each clone to predict the next direction to the local optima according to its own history information rather than just memorizing the last velocity without any judgement of the last ”flying”. In such a way, the clones are restricted to the search space around nearby local optima, so the individual convergent capability of each clone is able to be enhanced greatly.

Particle Swarm Optimization with Random Black Hole

For each dimension in every generation, we randomly generate a particle close to the current best particle. We regard it as a black hole by giving a threshold p drawn in the interval [0, 1], to decide its capability of magnetism. In each dimension of a particle in the swarm, we randomly generate a value l, which is drawn

Figure 1. Convergent performance of one particle on Ackley benchmark function within 4 generations using the AR-CPSO



from an uniform distribution over interval [0, 1]. If l is smaller than p, we let the particle be arrested by the black hole, i.e. the coordinate of the particle in this dimension is directly set to the coordinate of the black hole. At the same time, the velocities of other particles in the swarm in calculation of the SPSO are kept unchanged, which will be used in the next generation. The transition of a particle x from the t-th to the (t+1)-th generation in the RBH-PSO is schematically shown in Figure 2, where x_{gB} is the position of current best-fit particle in the entire population. $x(t)$ is the position of x in current t-th generation, $x(t+1)$ is the position that x is supposed to be in the next generation in the SPSO, and $\tilde{x}(t+1)$ is its actual position in the next generation after using our random black hole operation. s is randomly drawn from an uniform distribution over the interval [-r, r], and r is the radius of the area around x_{gB}, in which the black hole is generated randomly. r is determined according to the attributes of test functions which will be discussed in section IV in detail.

As shown in Figure 2, if the particle $x(t)$ is randomly chosen to be magnetized to the black hole, i.e., $l < p$, the actual position $\tilde{x}(t+1)$ in our RBH-PSO would be calculated by Eq. (12). Otherwise, $x(t+1)$ will be as same as that in the SPSO according to Eq. (13). This operation is carried out in each dimension for all particles in the swarm. So, from the point of view of high-dimensionality, the black hole gives another direction for some particles in some dimensions to converge in a probability threshold p.

$$\tilde{x}(t+1) = x_{gB} + s, if\ l < p \tag{12}$$

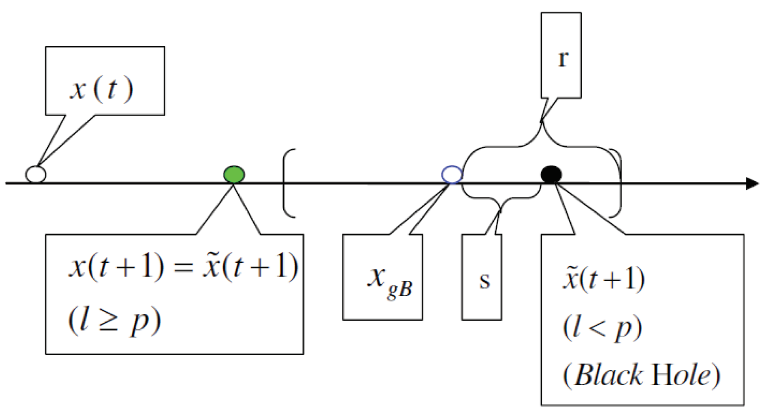
$$\tilde{x}(t+1) = x(t+1), if\ l \geq p \tag{13}$$

Briefly, the RBH-PSO algorithm can be summarized in Algorithm 3.

Algorithm 3 RBH-PSO Algorithm

- **Step 1:** Initialization. Assume $c1$ =2, $c2$ =2, and w be from 0.9 to 0.4 linearly.
- **Step 2:** The state of particles evolves iteratively according to Eqs. (1) and (2).
- **Step 3:** Find the current best-fit particle xgB, and generate a new particle close to xgB as a random black hole in the range of r. Determine r as the radium of the range, and then choose s randomly from an uniform distribution in interval [-r,r].
- **Step 4:** For each particle x in the swarm, randomly give it an evaluation l, and

Figure 2. Schematic Graph of the Position Transformation of $x(t)$ in the RBH-PSO



determine the threshold of the black hole p. Then update the position of x according to Eqs. (12) and (13). Accordingly, the velocity is updated by Eq. (1).

- **Step 5:** Termination. The algorithm can be terminated by a given maximum number of fitness evaluations, i.e., 1,200,000 in this study. If the termination condition is not met, go to step 2.

The essence of the RBH-PSO is to randomly clone and mutate another new best particle to guide all particles in the swarm. This new guide is considered to represent the actual best-fit position which exists but not found so far, just like a black hole in physics which has a huge quality (Black hole). We do not know where the real solution should be, but according to our analysis and knowledge so far, the small range around the current best-fit particle is considered to be the best candidate of the real solution, and randomness is employed to enhance the feasibility. As the SPSO evolves, we expect xgB in certain generations would converge to the solution of the problem at hand. If this is achieved, our black hole model will enhance the convergent speed because it is right next to xgB in each generation, and help xgB to magnetize other particles strongly. If the SPSO would converge to local optima, the black hole will give all particles another chance to fly out of the trapped local optima and keep evolving continuously.

Comparisons Among CPSO and Its Variants

For the AR-CPSO, a clonal operator is used to generate a new particle swarm near the promising candidate solution according to the value of the fitness function so that the search space is enlarged greatly and the diversity of clones is increased to avoid being trapped in local minima. Meanwhile, the essence of the AR strategy is to speed up clone for greatly finding nearby minima in an enlarged unknown space. Convergent rate and global optimization performance could be raised significantly. The RBH-PSO is also inspired by the clone and selection mechanism, which not only keep the diversity of the swarm but also accelerate the local search at the same time. Because the two clonal operations are complicated highly and need more computational time and preserve much more memory, therefore, a simple model, RBH-PSO, is proposed to find a more reasonable tradeoff between the convergent speed and global optimization capability.

In summary, during the iteration procedure of the three proposed algorithms, the local search space is enlarged significantly by the corresponding clonal operations around xgB, which accelerate the local search greatly. Meanwhile, we not only keep the velocity of the original particles but also keep the diversity of the swarm for global search.

EXPERIMENTS AND ANALYSIS

Experimental Setup

1. **Fifteen Benchmark Test Functions:** To test and verify the performance of the proposed CPSO and its variants, fifteen benchmark functions and their corresponding parameters listed in Figure 3 are used for our following simulations. Besides the global optimum of Shaffer f6 function is 1, the global optimum of the other fourteen benchmark test functions are 0. Since the optimization cost in real-world applications is usually dominated by the evaluations of the objective function, so the presetting expected number of fitness evaluations ($F\,Es$) is retained as the main algorithmic performance measure. The stop criterion in algorithms, i.e. the maximum number of fitness evaluations, is set to 1,200,000 in our simulations. In addition, we fix the number of particles in a swarm to be 40 for the convenience of comparisons later on. In Figure 3, $F\,Es$ denotes the number of the fitness evaluations and D the dimension of test functions.

The column 'Ini. Space' in the figure shows the spaces where the initializations lie in. For concrete expressions of the fifteen benchmark

Figure 3. List of fifteen benchmark test functions and their parameters for our following simulations

Functions	**Exp.**	D	Search Space	V_{max}	Ini. Space
Shaffer f6	F_1	30	$[-100, 100]^D$	100	$[15, 30]^D$
Sphere	F_2	30	$[-100, 100]^D$	100	$[15, 30]^D$
Rosenbrock	F_3	30	$[-100, 100]^D$	100	$[15, 30]^D$
Griewangk	F_4	30	$[-100, 100]^D$	100	$[15, 30]^D$
Ellipse	F_5	30	$[-100, 100]^D$	100	$[15, 30]^D$
Cigar	F_6	30	$[-100, 100]^D$	100	$[15, 30]^D$
Tablet	F_7	30	$[-100, 100]^D$	100	$[15, 30]^D$
Sumcan	F_8	30	$[-0.16, 0.16]^D$	0.16	$[0.03, 0.1]^D$
Schwefel	F_9	30	$[-100, 100]^D$	100	$[15, 30]^D$
Ackley	F_{10}	30	$[-32, 32]^D$	32	$[10, 20]^D$
Non. Rastrigin	F_{11}	30	$[-100100]^D$	100	$[15, 30]^D$
Rastrigin RT	F_{12}	30	$[-100100]^D$	100	$[15, 30]^D$
Griewangk RT	F_{13}	30	$[-100, 100]^D$	100	$[15, 30]^D$
Schwefel RT	F_{14}	30	$[-100, 100]^D$	100	$[15, 30]^D$
Ncrastrigrin RT	F_{15}	30	$[-100100]^D$	100	$[15, 30]^D$

test functions used in our experiments, and more complex and compound benchmark test functions (see Liang, Qin, Suganthan & Baskar, 2006; Kennedy & Mendes, 2002).

2. **Experimental Platforms:** All experiments in this article are conducted on two PCs with AMD Athlon 3200+ CPU and 1G RAM under Windows XP OS. Accuracy, precision, recall and miss rates are used as performance indices for spam detection.

LIBSVM software package is used for implementing our SVM under an environment of MATLAB version R2007a.

Determination of Parameters in Algorithms

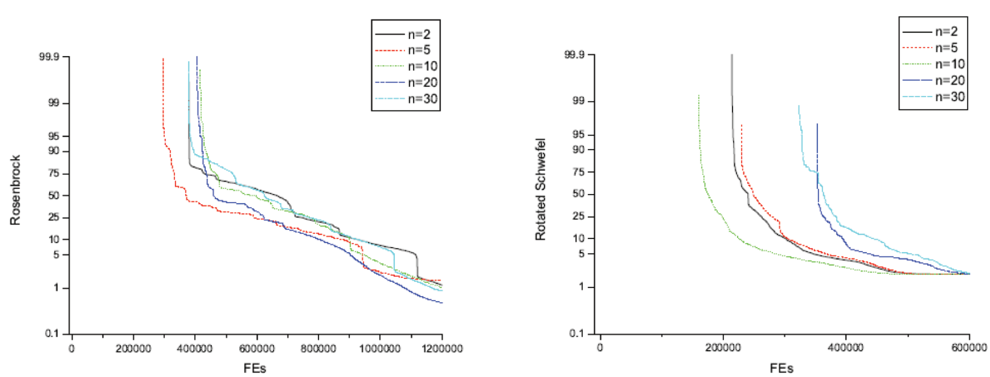
A best trade-off between exploration and exploitation strongly depends on properties of objective functions to be optimized, such as the number of local optima, the distance to the global optimum, the position of the global optimum in the search space (for example, at center, near borders, etc.), the size of the search area, the accuracy required in location of the optimum, etc. It is probably impossible to find a unique set of algorithmic parameters that work well in all cases (Trelea, 2003). Therefore, usually, a tentative method is adopted to determine the parameters in terms of three representative test functions, i.e., the Sphere function with only one optimum, the Rosenbrock function with slow slope, and the Schwefel function being rotated. Actually, all fifteen benchmark test functions are used to verify and test the validation of the parameters determined above in our experiments.

1. **Number of generations of clones versus performance:** For the number of generations of clones, denoted by symbol n, being 2, 5, 10, 20 and 30, respectively, the performances of the CPSO on Sphere and Rotated Schwefel functions are illustrated in Figure 4.

It can be seen from Figure 4 that the best performances of the proposed CPSO on both Sphere and Rotated Schwefel functions are obtained when 'n=10'.

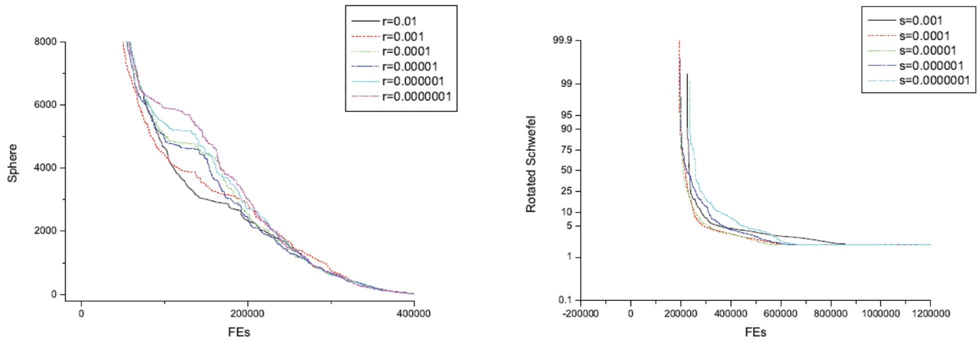
2. **Mutation scale of clones versus performance:** For the mutation scale of clones s from 0.001 to 0.0000001, when the retreat step a is equal to 0.4 and the generation of clones is set to 10, the performances of the CPSO on Rosenbrock and Rotated Schwefel functions are illustrated in Figure 5. It can be seen from Figure 5 that s =0.000001 has better performance. As a result, in the following experiments, let s be 0.000001 for the CPSO and the AR-CPSO.
3. **Retreat step-size versus performance of AR-CPSO:** For the retreat step-size$a$, in Eq. (11), being from 0.1 to 0.5, when the mutation scale of clones is equal to 0.000001 and the number of generations of clones is 10, the performances of the AR-CPSO on Rosenbrock and Rotated Schwefel functions are illustrated in Figure 6. It can be seen that the best performance is achieved as a =0.4. So, in the following experiments, a is assumed to be 0.4 for the AR-CPSO.
4. **Generations of clones versus performance for AR-CPSO:** For the number of generations of clones, denoted by n, from 10 to 50, when the mutation scale of clones is equal to 0.000001 and the retreat step is equal to 0.4, the performances of the AR-CPSO on Rosenbrock and Rotated Schwefel functions are illustrated in Figure 7. It can be seen that n = 10 has better performance. So in the following experiments, let n be 10 for the AR-CPSO and CPSO.
5. **Determination of parameters in the RBH-PSO:** Two parameters in the RBH-PSO need to be determined, i.e., p and r, which represent the probability of a black hole and the radius of the interval, respectively. For simplification and convenience, three benchmark functions are chosen

Figure 4. Convergent performances of the CPSO on Rosenbrock and rotated Schwefel functions with different generations of clones (Denoted by n)



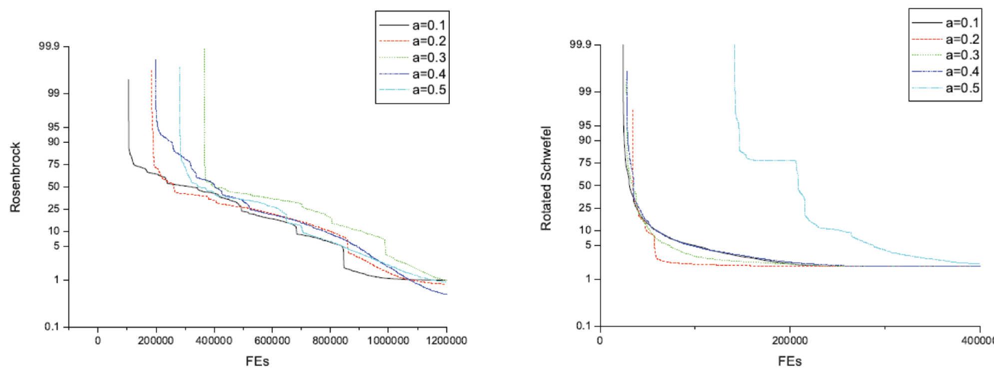
(a) F_3 (b) F_{14}

Figure 5. Convergent performances of the CPSO on Rosenbrock and rotated Schwefel functions with different mutation scales (Denoted by s)



(a) F_3, When a=0.4 (b) F_{14}, When a=0.4

Figure 6. Convergent performances of the CPSO on Rosenbrock and rotated Schwefel functions with different retreat step-size (Denoted by a)

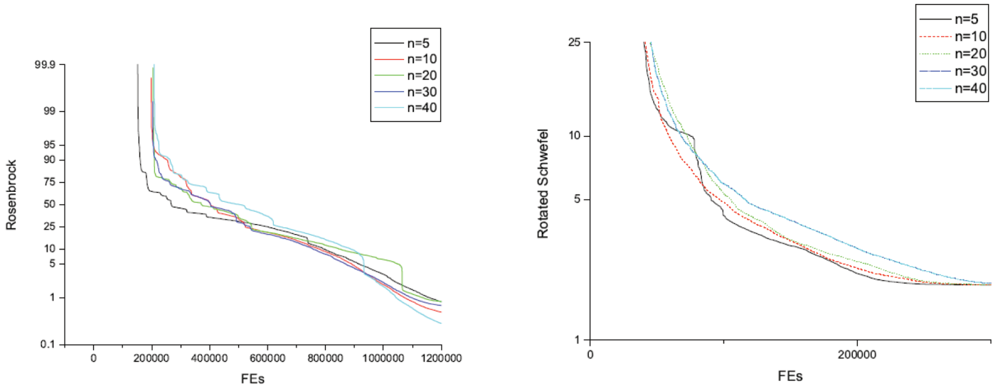


(a) F_3, When s=0.000001 (b) F_{14}, When s=0.000001

to test their effects. Sphere function and Rosenbrock function both have one single optimum, but the former has a "large scale" curvature while the latter has a flat bottom. Rotated Rastringrin function has multiple optima scattering over the entire search space.

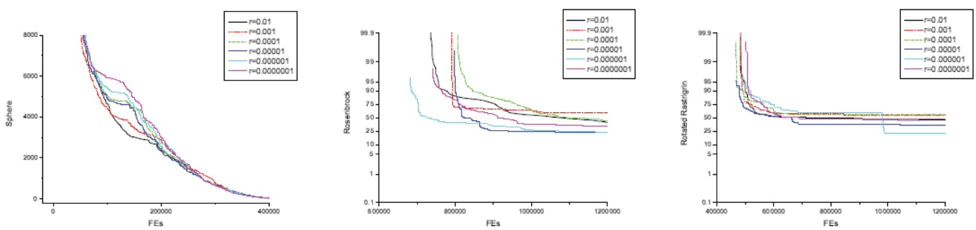
With a fixed p =0.05, the performance of the RBH-PSO on the three functions with different initial values of r are illustrated in Figures 8 (a)-8 (c). It can be seen from Figures 8 (a)-8 (c) that the best performance for all the three benchmark functions is achieved at r =0.000001. In Figures 8(d)-8(f), with a fixed r =0.000001, it is obvious that p =0.1 would

Figure 7. Convergent performances of the CPSO on Rosenbrock and rotated Schwefel functions with different generations of clones (Denoted as n)

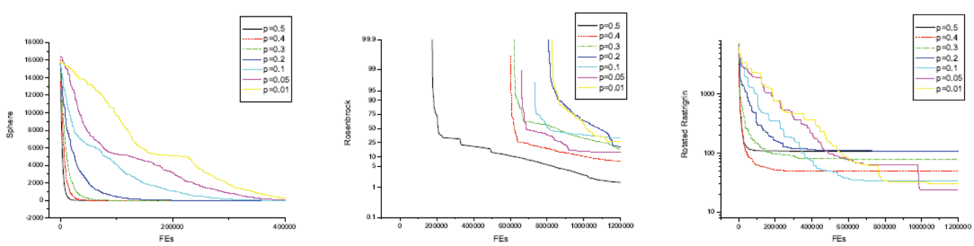


(a) F_3, When a=0.4, s=0.000001 (b) F_{14}, When a=0.4, s=0.000001

Figure 8. Convergent performances of the RBH-PSO on sphere, Rosenbrock and rotated Rastringrin functions with different parameters



(a) F_2, When p=0.05 (b) F_3, When p=0.05 (c) F_{12}, When p=0.05



(d) F_2, When r=0.000001 (e) F_3, When r=0.000001 (f) F_{12}, When r=0.000001

be a good choice. Therefore, the parameters in the RBH-PSO are finally chosen as p =0.1 and r =0.000001 which will be used in our following experiments.

Performance Comparisons Among CPSO, AR-CPSO, RBH-PSO and SPSO

The comparisons of performance among the CPSO, the AR-CPSO, the RBH-PSO and the SPSO on fifteen typical benchmark test functions are shown in Figure 9. These convergent curves are drawn from the averaged values of 50 independent runs. In such a way, these curves can give the stable performances of the CPSO, the AR-CPSO, the RBH-PSO and the SPSO completely and reliably. As can be seen from Figure 9, the proposed CPSO, AR-CPSO and RBH-PSO have much faster convergence speed and much more accurate solution than that of the SPSO on all fifteen benchmark test functions.

Furthermore, in order to verify the validation and efficiency of our proposed three algorithms, by 50 independent runs, we give the statistical means and standard deviations, in Figure 10, of our obtained solutions of the fifteen benchmark test functions listed in Figure 3, by using the proposed CPSO, AR-CPSO, RBH-PSO and the original SPSO, respectively. It turns out that the proposed CPSO, AR-CPSO and RBH-PSO has much more accurate solution than that of the SPSO on all fifteen benchmark test functions.

Specifically, the relationship of the convergent speed among the CPSO, the AR-CPSO, the RBH-PSO and the SPSO can be obviously observed from Figure 9, which could be expressed as follows:

$$AR-CPSO \succ CPSO \succ RBH-PSO \succ SPSO \tag{14}$$

where symbol $\succ$ denotes a relation of partial order such as ‘faster than’ in convergence speed.

According to Eq.(14), one can easily capture a clear picture of relations among the CPSO, AR-CPSO, RBH-PSO as well as the SPSO.

In a same way, the relationship among the proposed CPSO, AR-CPSO, RBH-PSO and the SPSO in global exploration capability is also easily observed from Figure 10, as follows:

$$RBH-PSO \succ CPSO \succ AR-CPSO \succ SPSO \tag{15}$$

where symbol $\succ$ denotes a relation of partial order such as ‘stronger than’ in global exploration capability.

As can be seen from Figure 10 and Eq. (15), the proposed CPSO, the AR-CPSO and the RBH-PSO has stronger global optimization capability than that of the SPSO on almost all fifteen benchmark test functions. Specifically, the CPSO achieves better global solution than the SPSO on twelve test functions. The AR-CPSO achieves better global solution than the SPSO on eleven test functions. The RBH-PSO achieves better global solution than the SPSO on all fifteen test functions.

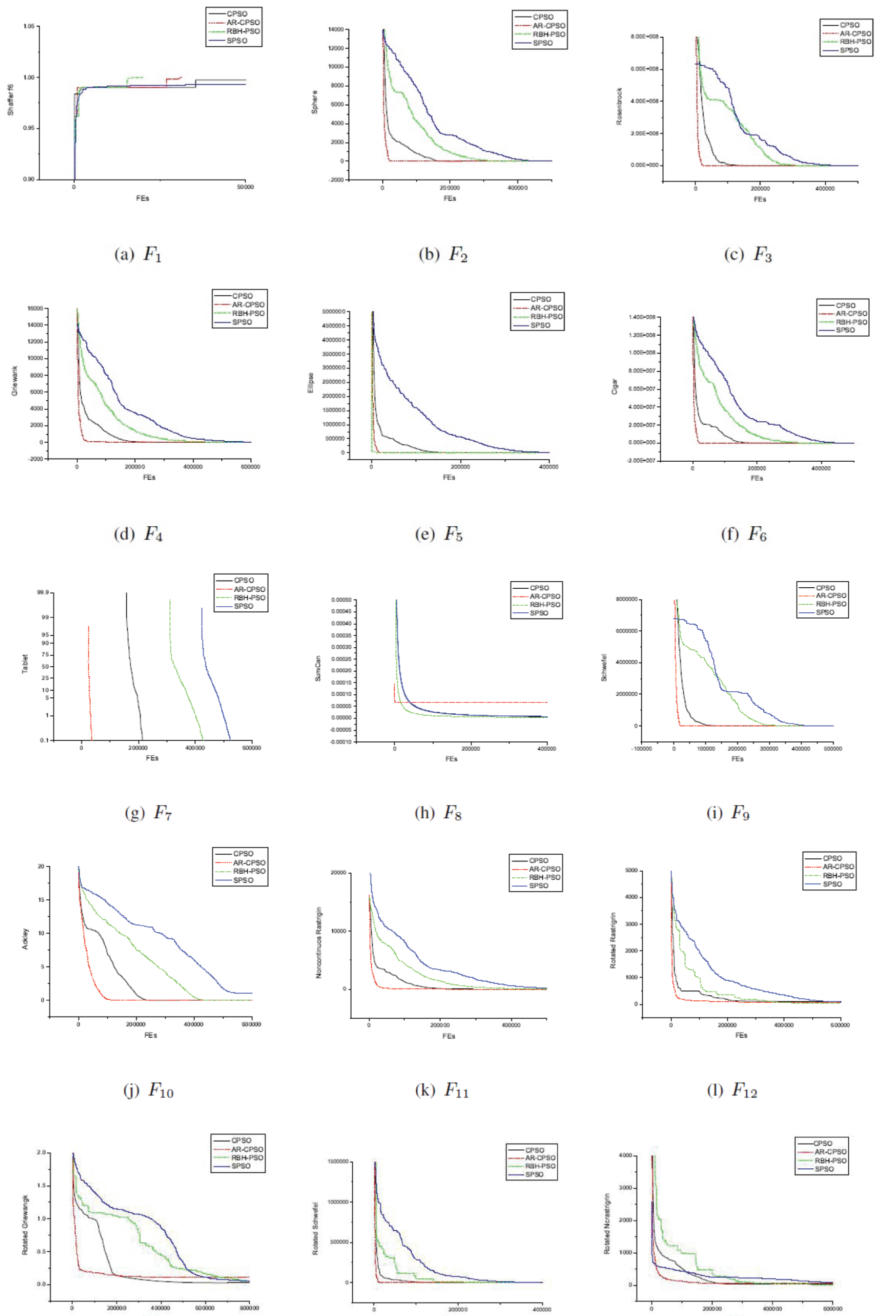
Therefore, we can conclude that the proposed CPSO and its two variants are able to accelerate the convergence tremendously whilst keeping a good global search capability with much more accuracy. All of the simulation results in experiments have shown that the introduction of the clonal mechanism in NIS into the PSO leads to a promising performance.

Analysis and Discussion

By inspired by immunity-clonal mechanism, the CPSO and its two variants, i.e., AR-CPSO and RBH-PSO, use a clonal operation to generate new particles near the promising candidate solution according to the fitness value so that the search space is able to be enlarged greatly and further the diversity of clones is increased to avoid being trapped in local minima. In such a way, convergent rate and global exploration capability can be greatly raised simultaneously.

As we know, those fifteen functions listed in Figure 3 are of very much different characteristics and properties. Some of them have a single minimum, and others have multiple local minima. Several functions are of a large-scale curvature which guides the search toward the global minimum, while others are essentially

Figure 9. The average performance of the RBH-PSO and SPSO on F1 − F15 $_i$n Figure 3 with 40 particles in a swarm over 50



(a) F_1 (b) F_2 (c) F_3

(d) F_4 (e) F_5 (f) F_6

(g) F_7 (h) F_8 (i) F_9

(j) F_{10} (k) F_{11} (l) F_{12}

(m) F_{13} (n) F_{14} (o) F_{15}

Figure 10. Statistical means (M)and standard deviations (STD)of the solutions of fifteen benchmark test functions, listed in Figure 3, given by the AR-CPSO, the CPSO, the RBH-PSO and the SPSO over 50 independent runs, where FEs denote the fitness evaluations

Functions	FEs.	CPSO's M ± Std	AR-CPSO's M ± Std	RBH-PSO's M ± Std	SPSO's M ± Std
Shaffer f6	20, 120	0.9949 ± 0.004	0.9922 ± 0.003	**1 ± 0**	0.9924 ± 0.003
Sphere	315, 331	0.000002 ± 0.000001	**0 ± 0**	81.361 ± 51.58	1025.55 ± 372.70
Rosenbrock	1, 200, 000	1.01 ± 1.72	**0.49 ± 1.23**	32.01 ± 35.87	45.64 ± 54.51
Griewangk	1, 200, 000	7.31 ± 3.52	**4.576811 ± 2.69**	**4.57 ± 2.69**	18.80 ± 4.95
Ellipse	492, 820	0.000001 ± 0	**0± 0**	0.000069 ± 0.00008	53 ± 38
Cigar	549, 000	0.000002 ± 0.000002	0.000001 ± 0.000001	**0 ± 0**	15.7 ± 17.63
Tablet	353, 010	0.000001 ± 0	**0 ± 0**	55.98 ± 24.79	857.36 ± 362.3
SumCan	1, 200, 000	0.000004 ± 0	0.000022 ± 0	**0.000002 ± 0**	0.0000030 ± 0
Schwefel	427, 999	**0 ± 0**	0.000001 ± 0	51.2 ± 40.1	11989.14 ± 8907
Ackley	594, 336	**0 ± 0**	0.000001 ± 0	0.000001 ± 0	1.02 ± 4.48
Nonc. Rastrigin	1, 200, 000	0.95 ± 0.97	**0.35 ± 0.65**	1.95 ± 3.29	5.9 ± 3.89
Rastrigin RT	1, 200, 000	82.12 ± 79.07	103.95 ± 74.58	**34.18 ± 1.33**	52.76± 24.56
Griewangk RT	1, 200, 000	0.025 ± 0.03	0.11 ± 0.13	**0.013 ± 0.000046**	0.034 ± 0.027
Schwefel RT	1, 200, 000	1.898378 ± 0.000001	1.898377 ± 0.000001	**1.898374 ± 0**	1.89839 ± 0.00002
Ncrastrigrin RT	1, 200, 000	31.06± 19.29	58.41 ± 51.52	**8.9 ± 3.92**	25.32 ± 7.16

flat except the area near the global minimum. So, these functions are usually used as powerful and useful benchmark test functions to test the newly developed algorithms thoroughly and objectively. In spite of the complexities of these functions, it turns out from the comparisons of performances among the CPSO, the AR-CPSO, the RBH-PSO and the SPSO, that the CPSO and its two variants not only has a faster convergence speed but also has more accurate optimal solution than that of the SPSO on almost all of the benchmark test functions, which strongly support our contributions introducing the clonal mechanism into the PSO.

In particular, the proposed CPSO, the AR-CPSO and the RBH-PSO do not need to stop the evolving of the swarm (Liu, Qin & Shi, 2004; Chen, Qin, Liu & Lu, 2005) for a local search. Furthermore, the CPSO and its two variants do not need to calculate the gradient of the objective function, which is computationally expensive cost, and change the structure of the conventional PSO (Noel & Jannett, 2004; Vaz & Fernandes, 2005). Most recently, we have developed parallelism implementations of the PSO algorithms based on graphics processing unit (GPU) in a personal computer and obtained a more than 20 times of speedup for the PSO algorithm for a specific task (Zhou & Tan, 2009).

What follows is an application of the proposed CPSO-like algorithms on spam detection.

SPAM DETECTION APPLICATION

Spam, usually defined as unsolicited commercial e-mail, or unsolicited bulk e-mail, or uninterested e-mail from the perspective of individual e-mail user, has been regarded as an increasingly serious problem to the infrastructure of Internet. According to the statistics from International Telecommunication Union (ITU), about 70% to 80% of the present emails in Internet are spam. Numerous spams not only occupy valuable communications bandwidth and storage space, but also threaten the network security as it is often used as a carrier of viruses and malicious codes. Meanwhile, spam wastes much user's time to tackle with them so that the productivity is reduced considerably. Therefore, spam detection has attracted many attentions in Internet research community from academia as well as industry. Theoretical analysis and many practical algorithms, tools, and system level solutions have been successfully developed. In summary, they can be classified into three categories: simple approaches, intelligent approaches and hybrid approaches.

Simple approaches include munging, listing, aliasing and challenging. These techniques are easy to implement while are also prone to be deceived by tricks of spammers. Intelligent approaches play an increasingly important role in anti-spam in recent years for their ability of self learning and good performance, which include Naïve Bayes (Androutsopoulos, Koutsias, Chandrinos & Spyropoulos, 2000; Sahami, Dumais, Heckerman & Horvitz, 1998), Support Vector Machine (SVM) (Ruan & Tan, 2007; Drucker, Wu & Vapnik, 1999; Tan & Wang, 2004), Artificial Neural Network (ANN) (Clark, Koprinska & Poon, 2003; Stuart, Cha & Tappert, 2004), Artificial Immune System (AIS) (Ruan & Tan, 2008; Oda & White, 2003; Secker, Freitas & Timmis, 2003; Bezerra & Barra, 2006; Tan, 2006; Sirisanyalak & Sornil, 2007) and DNA Computing (Rigoutsos & Huynh, 2004). As an anti-spam shield with one technique alone can be easily intruded in practice, consequently, several hybrid approaches by combining two or more techniques together are proposed (Leiba & Borenstein, 2004; Wu, Huang, Lu, Chen & Kuo, 2005) for better overall performance.

Support Vector Machine (SVM) has already proved its superiority in pattern recognition for its generalization performance, which is based on the Structural Risk Minimization principle from statistical learning theory (Vapnik, 1995; Drucker, Burges, Kauffman, Smola & Vapnik, 1997). The goal of SVM is to find an optimal hyper plane for which the lowest true error can be guaranteed. In what follows, the SVM is used as the classifier for spam detection.

Natural immune system has some desirable properties for spam detection, including pattern recognition, dynamically changing coverage and noise tolerance, etc, some of which are drawn for our algorithm. So, inspired by human immune system, a concentration based feature construction (CFC) approach is constructed to characterize each e-mail through a two-element feature vector (Tan, Deng & Ruan, 2009). In the CFC approach, 'self' concentration and 'non-self' concentration are constructed by using 'self' gene library and 'non-self' gene library, respectively. Subsequently, they are used to form a two-element concentration vector which characterizes the e-mail efficiently and concisely.

Two corpus used to test the CFC approaches are the PU1 corpus (Androutsopoulos, Koutsias, Chandrinos & Spyropoulos, 2000) and Ling corpus1 (Androutsopoulos, Koutsias, Chandrinos, Paliouras & Spyropoulos, 2000). PU1 corpus consists of 1,099 m^{e}ssages, with spam rate 43.77%, Ling corpus consists of 2,893 messages, with spam rate 16.63%. All the messages in both corpora have header fields, attachment and HTML tags removed, leaving only subject line and mail body text. In PU1, each token is mapped to a unique integer to ensure the privacy of the content while keeping its original form in Ling. Each corpus is divided into ten partitions with approximately equal amount of messages and spam rate. The version with stop-word removal is used in our experiments.

LIBSVM software package is used as an implementation of the SVM (Chang & Lin, 2001). Polynomial kernel with three parameters, i.e., gamma, coef0 and degree, is adopted. Together with the cost parameter C, there are four parameters to be optimized.

The proposed CPSO, AR-CPSO, and RBH-PSO as well as three typical algorithms are used to tune the above four parameters. A corresponding test function model with four parameters as input and classification accuracy as output is established. The classification accuracy, measured by 10-fold cross validation, serves as the objective function. The CPSO-like algorithms terminate when the fitness value of the global best particle does not change in consecutive 50 generations.

Comparisons of performances among the CPSO, the AR-CPSO, the RBH-PSO are made and shown in Figure 11, where the accuracy of the CPSO, the AR-CPSO, the RBH-PSO, Nave Bayesian, Linger-V and SVM-IG on corpus PU1 and Ling are listed in digits (Androutsopoulos, Koutsias, Chandrinos & Spyropoulos, 2000; Clark, Koprinska & Poon, 2003; Androutsopoulos, Koutsias, Chandrinos, Paliouras & Spyropoulos, 2000; Koprinska, Poon, Clark & Chan, 2007). Where, Linger-V is a NN-based system for automatic e-mail classification. All these results are obtained by using 10-fold validation. For Naïve Bayesian, 50 words with the highest mutual information scores are selected. LINGER-V and SVM-IG uses variance (V) and information gain (IG) as the criteria of feature selection, respectively, and the best-scoring 256 features are chosen. It can be seen from Figure 11 that the proposed CPSO-like algorithms is indeed used to tune the parameters of SVM classifier and raise the accuracy of classification greatly.

Furthermore, the corresponding solutions of the four parameters, i.e., gamma, coef0, degree and C, optimized by the CPSO, the AR-CPSO and the RBH-PSO, respectively, are given in Figure 12 for corpus PU1 and in Figure 13 for corpus Ling. In one word, we succeed in applying the proposed PSO algorithms to optimize the parameters of the SVM, which gives much higher classification accuracy than that of current methods.

CONCLUSION

Inspired by immunity-clonal strategies, a clonal particle swarm optimization (CPSO) and its two variants are proposed and implemented in details in this article. By cloning the best individuals of every several successive generations, the proposed CPSO algorithms have better optimization solving capability and convergence performance than the conventional SPSO in terms of a number of experiments on fifteen benchmark test functions. Two variants to the

Figure 11. Performances of CPSO, AR-CPSO, RBH-PSO, Naïve Bayesian (NB), Linger-V and SVM-IG on corpus PU1 and Ling, by using 10-fold cross-validation

Dat Sets	CPSO (%)	AR-CPSO (%)	RBH-PSO (%)	NB (%)	Linger-V(%)	SVM-IG(%)
PU1	**99.09**	**99.09**	98.99	91.07	93.45	93.18
Ling	**99.82**	**99.82**	**99.82**	96.40	98.2	96.85

Figure 12. The corresponding solutions of the four parameters, i.e., gamma, coef0, degree and c, found by the CPSO, AR-CPSO and RBH-PSO in Figure 11 on corpus PU1,using 10-fold cross-validation

Parameters	CPSO	AR-CPSO	RBH-PSO
C	1	3	169.38
gamma	6.21	0.19	0.87
coef0	178.94	21.19	142.76
degree	8	4	10

Figure 13. The corresponding solutions of the four parameters, i.e., gamma, coef0,degree and c, found by the CPSO, AR-CPSO and RBH-PSO in Figure 11 on corpus Ling, using 10-fold cross-validation

Parameters	CPSO	AR-CPSO	RBH-PSO
C	201	11.52	201
gamma	17.13	19.97	20
coef0	88.25	27.79	67.1
degree	7	8	3

CPSO, i.e., AR-CPSO and RBH-CPSO, are developed to enhance the convergent ability. For the three proposed algorithms, the local search space is enlarged significantly by the corresponding clonal operations, which accelerate the local search greatly whilst we not only keep the velocity of the original particles but also keep the diversity of the swarm for global search. Furthermore, an application to spam detection by a SVM classifier, optimized by the proposed PSO algorithms, is completely conducted to achieve a promising result, which implies that the proposed PSO algorithms will find themselves helpful in many real-world applications in future.

ACKNOWLEDGMENT

This work was supported by the National High Technology Research and Development Program of China (863 Program), with grants No.2007AA01Z453, and also supported by the National Natural Science Foundation of China under grant No.60875080 and No.60673020. The author would like to appreciate Prof. Yuhui Shi andanonymous reviewers for their constructive comments and suggestions which help to improve the quality of this article highly. The author also thanks my students in my CIL at Peking University, Dr. Junqi Zhang, Mr. Kun Liu, Mr. Guangchen Ruan and Mr. Zhongmin Xiao for their efforts in conducting all experiments.

REFERENCES

Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G., & Spyropoulos, C. D. (2000). An evaluation of naive bayesian anti-spam filtering. Paper presented at the European Confer*ence on Machine Learning (ECML'00).*

*Androutsopoulo*s, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000). An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Developm*ent in Information Retrieval (pp. 160-167).*

*Bergh, F. V. D., & Engelbrecht, A. P. (2004). A co-operative approach to p*article swarm optimization. IEEE Transactions on Evolutionary Computation, 8, 225–239. doi:10.1109/TEVC.2004.826069

Bezerra, G. B., Barra, T. V., et al. (2006). An immunological filter for spam. Paper presented at the International Conference on Art*ificial Immune Systems (ICARIS'06*).

Castro, L. N. D. (2002). Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation, 6, 239–251. doi:10.1109/TEVC.2002.101153*9*

Castro, L. N. D., & Timmis, J. I. (2003). Art*i*ficial immune system as a novel soft computing paradigm. Soft Computing Journal, 7(8), 526–544.

Chang, C.-C., & Lin, C.-J. (2001). LIBSVM: a Library for Support Vector Machines. Retrieved from http://www.csie.ntu.edu.tw/~cjlin/libsvm

Chen, J., Qin, Z., Liu, Y., & Lu, J. (2005). Particle swarm optimization with local search. In Proceedings of the IEEE International Conference on Neural Networks and Brains (pp. 481-484).

Clark, J., Koprinska, I., & Poon, J. (2003). A neural network based approach to automated e-mail classification. In Proceedings of the IEEE International Conference on Web Intelligence (WI'03) (pp. 702-705).

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1), 58–73. doi:10.1109/4235.985692

Dasgupta, D., & Attoh-Okine, N. (1997). Immunity-based systems: A survey. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics.

Drucker, H., Burges, C. J. C., Kauffman, L., Smola, A., & Vapnik, V. N. (1997). Support vector regression machines. [NIPS]. Advances in Neural Information Processing Systems, 9, 155–161.

Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 10, 1048–1054. doi:10.1109/72.788645

Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In Proceedings of the 6th International Symposium on Mcro Machine Human Science (pp. 39-43).

El-Abd, M., & Kamel, M. S. (2006). A hierarchal cooperative particle swarm optimizer. In Proceedings of the Swarm Intelligence Symposium (pp. 43-47).

Ge, H. W., Lu, Y. H., Zhou, Y., Guo, X. C., & Liang, Y. C. (2005). A particle swarm optimization-based algorithm for job-shop scheduling problem. International Journal of Computational Methods, 2(3), 419–430. doi:10.1142/S0219876205000569

Ismael, A. I. F., Pereira, A. I. P. N., & Fernandes, E. M. G. P. (2005). Particle swarm and simulated annealing for multi-global optimization. WSEAS Transactions on Information Science and Applications, 2, 534–539.

Kennedy, J. (1997). The particle swarm: Social adaption of knowledge. In Proceedings of the IEEE International Conference on Evolutionary Computation.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks (pp. 1942-1948).

Kennedy, J., & Mendes, R. (2002). Population structure and particle swarm performance. In Proceedings of the IEEE Congress on Evolutionary Computation (pp. 1671-1676).

Koprinska, I., Poon, J., Clark, J., & Chan, J. (2007). Learning to classify e-mail. Information Science, ▪▪▪, 2167–2187. doi:10.1016/j.ins.2006.12.005

Leiba, B., & Borenstein, N. (2004). A multifaceted approach to spam reduction. In Proceedings of the First Conference on Email and AntiSpam (CEAS'04).

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 10, 281–296. doi:10.1109/TEVC.2005.857610

Liu, Y., Qin, Z., & Shi, Z. (2004). Hybrid particle swarm optimizer with line search. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (pp. 3751-3755).

NASA. (n.d.). Black hole. From http://imagine.gsfc.nasa.gov/docs/science/knowl2/blackholes.html

Noel, M. M., & Jannett, T. C. (2004). Simulation of a new hybrid particle swarm optimization algorithm. In Proceedings of the thirty-sixth Southeastern Symposium, Mumbai, India (pp. 150-153).

Oda, T., & White, T. (2003). Increasing the accuracy of a spam-detecting artificial immune system. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC'03) (Vol. 1, pp. 390-396).

Parsopoulos, K. E., Plagianakos, V. P., Magoulas, G. D., & Vrahatis, M. N. (2001). Stretching technique for obtaining global minimizers through particle swarm optimization. In Proceedings of the Workshop on Particle Swarm Optimization (pp. 22-29).

Parsopoulos, K. E., & Vrahatis, M. N. (2004). Upso-a united particle swarm optimization scheme. In Lecture Series on Computational Sciences (pp. 868-873).

Peram, T., Veeramachaneni, K., & Mohan, C. K. (2003). Fitness-distance-ratio based particle swarm optimization. In Proceedings of the Swarm Intelligence Symposium (pp. 174-181).

Pereira, A. I. P. N., & Fernandes, E. M. G. P. (2005). A new algorithm to identify all global maximizers based on simulated annealing. In Proceedings of the 6th World Congresses of Structural and Multidisciplinary Optimization.

Rigoutsos, I., & Huynh, T. (2004). Chung-kwei: A pattern-discovery-based system for the automatic identification of unsolicited e-mail messages(spam). In Proceedings of the first Conference on Email and AntiSpam (CEAS'04).

Ruan, G. C., & Tan, Y. (2007). Intelligent detection approaches for spam. In Proceedings of Third International Conference on Natural Computation (ICNC'07).

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A bayesian approach to filtering junk e-mail. In Proceedings of the AAAI Workshop on Learning for Text Categorization (pp. 55-62).

Secker, A., Freitas, A. A., & Timmis, J. (2003). AISEC: An artificial immune system for email classification. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC'031) (Vol. 1, pp. 131-139).

Shi, Y. H., & Eberhart, R. (1998). A modified particle swarm optimizer. In Proceedings of the IEEE World Congress on Computational Intelligence (pp. 69-73).

Sirisanyalak, B., & Sornil, O. (2007). An artificial immune-based spam detection system. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC'07).

Stuart, I., Cha, S.-H., & Tappert, C. (2004). A neural network classifier for junk e-mail. In Document Analysis Systems VI (LNCS 3163, pp. 442-450).

Tan, Y. (2006). Multiple-point bit mutation method of detector generation for snsd model. In Advances in Neural Networks - ISNN 2006 (LNCS 3973, pp. 340-345).

Tan, Y., Deng, C., & Ruan, G. C. (2009). Concentration based feature construction approach for spam detection. In Proceedings of the International Joint Conference of Neural Networks (IJCNN'09) (pp. 3088-3093).

Tan, Y., & Wang, J. (2004). A support vector network with hybrid kernel and minimal vapnik-chervonenkis dimension. IEEE Transactions on Knowledge and Data Engineering, 26, 385–395.

Tan, Y., & Xiao, Z. M. (2007). Clonal particle swarm optimization and its applications. In Proceedings of the IEEE Congress on Evolutionary Computation (pp. 2303-2309).

Trelea, I. C. (2003). The particle swarm optimization algorithm: Convergence analysis and parameter selection. Information Processing Letters, ▪▪▪, 317–325. doi:10.1016/S0020-0190(02)00447-7

Vapnik, V. (1995). The Nature of Statistical Learning Theory. New York: Springer-Verlag.

Vaz, A. F., & Fernandes, E. M. G. P. (2005). Particle swarm algorithms for multi-local optimization. Paper presented at the Congresso de Estatistica e Investigacao Operacional da Galiza e Norte de Portugal.

Wu, M.-W., Huang, Y., Lu, S.-K., Chen, I.-Y., & Kuo, S.-Y. (2005). A multi-faceted approach towards spam-resistible mail. In Proceedings of the IEEE Pacific Rim International Symposium on Dependable Computing (pp. 208-218).

Zhang, J. Q., Liu, K., Tan, Y., & He, X. G. (2008). Random black hole particle swarm optimization and its application. In Proceedings of the IEEE International Conference on Neural Networks for Signal Processing (ICNNSP2008) (pp. 359-365).

Zhang, J. Q., Xiao, Z. M., Tan, Y., & He, X. G. (2008). Hybrid particle swarm optimizer with advance and retreat strategy and clonal mechanism for global numerical optimization. In Proceedings of the IEEE World Congress on Computational Intelligence (WCCI'2008—IEEE CEC'2008) (pp. 2059-2066).

Zhou, Y., & Tan, Y. (2009). GPU-based parallel particle swarm optimization. In Proceedings of the 2009 IEEE Congress on Evolutionary Computation (pp. 1493-1500).

ENDNOTE

[1] The PU1 corpus and Ling corpus may be downloaded from http://www.iit.demokritos.gr/skel/ iconfig/

Ying Tan is a professor at Key Laboratory of Machine Perception (MOE), Peking University, and Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, China. He received his BSc in electronic engineering from Electronic Engineering Institute, China, MSc in electronic engineering from Xidian University, China, PhD in signal and information processing from Southeast University, China. His primary research interests are in computational intelligence, artificial immune system, swarm intelligence, intelligent information processing, machine learning algorithms and their applications in computer security.