

A Global Optimization Based on Physicomimetics Framework

Li-Ping Xie

1 College of Electrical and Information Engineering,
Lanzhou University of Technology, Lanzhou 730050

2 Complex System and Computational Intelligence
Laboratory, Taiyuan University of Science and
Technology, Taiyuan, Shanxi, P.R. China, 030024

jiangzhou2007@sohu.com

Jian-Chao Zeng

Complex System and Computational Intelligence
Laboratory, Taiyuan University of Science and
Technology, Taiyuan, Shanxi, P.R. China, 030024

zengjianchao@263.net

ABSTRACT

Based on physicomimetics framework, this paper presents a global optimization algorithm inspired by physics, which is a stochastic population-based algorithm. In the approach, each physical individual has a position and velocity which move through the feasible region of global optimization problem under the influence of gravity. The virtual mass of each individual corresponds to a user-defined function of the value of an objective function to be optimized. An attraction-repulsion rule is constructed among individuals and utilized to move individuals towards the optimality. Experimental simulations show that the algorithm is effective.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, Search – *heuristic methods, simulated annealing.*

G.1.6 [Numerical Analysis]: Optimization – *global optimization.*

General Terms

Algorithms, Theory.

Keywords

Physicomimetics; global optimization; virtual force; Simulated Annealing; Newton's Second law

1. INTRODUCTION

Global optimization problems are ubiquitous in many practical applications, e.g., in advanced engineering design, process control, biotechnology, data analysis, financial planning, risk management and so on. Some problems involve nonlinear functions of many variables which are irregular or high dimensional. These functions are difficult to optimize by traditional deterministic optimization techniques. Heuristic strategies have become effective approaches, which search the

feasible region of the objective functions in various intelligent ways. Commonly used strategies include Simulated Annealing (SA) [1], evolutionary methods [2] and swarm intelligence algorithms [3]. SA algorithm is based upon the physical analogy of cooling crystal structures that spontaneously attempt to arrive at some stable (globally or locally minimal potential energy) equilibrium. Evolutionary methods and swarm intelligence algorithms utilize the concept of "population" and "evolution". Evolutionary methods heuristically "mimic" biological evolution: namely, the process of natural selection and the "survival of the fittest" principle. Swarm intelligence algorithms are inspired by animal grouping behaviors in natural, like Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) [4]. PSO uses a swarm of candidate particles that fly over the fitness landscape looking for optima. The particles are driven by two forces which attract them to the best location encountered both by any other members of the swarm and by themselves.

In recent years, some scholars introduced a new school of population-based search algorithms motivated by natural physical rules. They are Electromagnetism-like (EM) method [5] and Central Force Optimization (CFO) [6]. The two heuristic methods construct virtual forces among individuals and attraction-repulsion rules for looking for optima. However, they have some differences. First, they adopt different physical laws. EM method uses Coulomb's law. Each sample solution is treated as a charged particle that is released to a space. All points attract or repel each others. The charge of each point is inversely proportional to its objective function value. Each point moves in the direction of the combination force exerted on it via other points and with the step length randomly in feasible region. CFO adopts Newton's gravity law and searches the decision space by a group of probes satellites flown into the solar system. Most of the probes will cluster near the planet with the strongest gravity after a long enough time. This is analogous to finding the global maximum of the objective function. The masses of probes are directly proportional to their objective function values. The probes attract each other and their motions comply with gravitational kinematics. Second, they have different swarm initialization. EM procedure starts with randomly sample points from the feasible domain. However, CFO requires the initial distribution of probes with uniform along each coordinate axis. Finally, EM is a stochastic search algorithm and CFO is a deterministic algorithm.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GEC'09, June 12-14, 2009, Shanghai, China.

Copyright 2009 ACM 978-1-60558-326-6/09/06... \$5.00.

Similarly, we presented a population-based stochastic algorithm based on Physicomimetics framework. Physicomimetics framework [7, 8] is inspired by natural physical forces. The virtual physical forces drive a swarm robotics system to a desired configuration or state. It has been applied to the distributed control of swarms of robots, such as robots formation [9, 10, 11], obstacle avoidance tasks [12] and coverage tasks [13].

2. MOTIVATION

Physicomimetics (or Artificial Physics, AP) is presented by William M. Spear, Diana F. Spear and Rodney Heil, etc. The authors focus on robotic behaviors and make robots execute complex tasks. They map the problems to physical phenomena. Given a set of initial conditions and some desired global behavior, it is necessary to define what sensors, effectors, and force laws are required for the desired global behaviour to emerge. For example, robots formations are mapped to solid crystalline formations, obstacle avoidance tasks are mapped to liquids maneuvering around obstacles while retaining connectivity. The two phenomena are formed using a similar force law, which has attractive and repulsive components. Coverage tasks are mapped to gases diffusedness, which behaviors are created using purely repulsive forces. In basic AP framework, robots are treated as physical particles. Each particle has a position, mass, velocity and momentum. Particles move in response to the virtual total forces that are exerted on them by other particles. The system acts as a molecular dynamics ($F = ma$) simulation.

The initial application of robots formations based on artificial physics required that a swarm of micro-air vehicles (MAVs) self-organize into a hexagonal lattice, creating a distributed sensing grid with spacing R between MAVs [14]. To map this into a force law, each robot repels other robots that are closer than R , while attracting robots that are further than in distance. A force law is defined by $F = Gm_i m_j / r^p$, where $F < F_{\max}$ is the magnitude of the force between two particles i and j , and r is the distance between the two particles. The variable p is a user-defined power, which ranges from -5.0 to 5.0. The “gravitational constant” G is set at initialization. The force is repulsive if $r < R$ and attractive if $r > R$. The attraction force and repulsion force can be balance if $r = R$.

Similarly, we map global optimization problems to the robots formations model based on artificial physics. The candidate solutions of global optimization problem are treated as physical individuals that are released in a landscape. The population of individuals is driven by physical virtual forces to look for optima. We establish an attraction-repulsion mechanism to move a population of individuals towards optimality. In basic AP framework, the attractive force between two particles is only determined by the distance between them, since each particle is point-mass. In our approach, the virtual masses of individuals are not constants, they are variables and change with their objective function values, which we are trying to optimize. When searching the minimum value of objective function, each individual’s mass is inversely proportional to its objective function value. The better the objective function value, the bigger the mass, then the higher the magnitude of attraction. In this way, individuals that have better objective function values possess bigger masses and higher attractive forces, which make other individuals move towards the

better search region. After calculating the mass of each individual, we use them to gain a velocity vector for each particle to move in next iterations. The change of velocity is controlled by the total force on the individual. The total force is calculated by adding vectorially the forces from each of the other individuals calculated separately. Newton’s second law supplies reference to the motion of individuals.

The above method is called Artificial Physics Optimization (APO) method. We introduce APO framework in next section. In addition, we adopt simulated annealing method as a local search procedure and apply it to the framework.

3. ARTIFICIAL PHYSICS OPTIMIZATION

In this paper, we study a class of nonlinear optimization problems with bounded variables in the following form:

$$\min \{f(X) : X \in \Omega \subset R^n\}, f : \Omega \subset R^n \rightarrow R^l \quad (1)$$

where $\Omega := \{X \in \Omega \mid x_k^{\min} \leq x_k \leq x_k^{\max}, k = 1, \dots, n\}$.

We deal with the function of the form (1), with the following parameters given:

n : dimension of the problem.

x_k^{\max} : upper bound in the k th dimension.

x_k^{\min} : lower bound in the k th dimension.

$f(x)$: pointer to the function that is minimized.

In this section, we introduce the general framework of APO and give its pseudo-code.

3.1 General Framework for APO

APO algorithm consists of four phases. They are the initialization of the algorithm, calculation of the total force exerted on each

```

ALGORITHM 1. APO( $m, MAXITER$ )
 $m$  : number of samples
 $MAXITER$ : maximum number of search iterations
1: Initialize()
2: iteration  $\leftarrow$  1
3: while iteration  $<$   $MAXITER$  do
4:    $F \leftarrow \text{CalcForce}()$ 
5:    $\text{Move}(F, \text{iteration})$ 
6:   for  $i = 1$  to  $m$  do
7:     Calculate  $f(X_i)$ 
8:   end for
9:    $X_{\text{best}} \leftarrow \arg \min \{f(X_i), \forall i\}$ 
10:  Localsearch ()
11:  iteration  $\leftarrow$  iteration + 1
12: end while

```

Figure 1. APO framework.

individual, movement along the direction of the force and application of neighborhood search to exploit the local minima. APO general framework is given by the following Algorithm 1 shown by Figure 1.

3.2 Initialization

We consider a population of m individuals in an n -dimensional feasible space. The position and the velocity of individual i can be shown as $X_i(t) = (x_{i,1}(t), x_{i,2}(t), \dots, x_{i,n}(t))$ and $V_i(t) = (v_{i,1}(t), v_{i,2}(t), \dots, v_{i,n}(t))$, respectively. $x_{i,k}$ and $v_{i,k}$ represent the position and velocity of individual i in k th dimension, respectively. $F_{ij,k}$ represents the force exerted on individual i via individual j in k th dimension. The initialization of APO is given as in Algorithm 2, which is shown by Figure 2. In initialization state, m individuals are randomly sampled from the feasible domain of the problem (1), which is an n dimensional hyper-cube. The values of $v_{i,k}$ and $F_{ij,k}$ are all zero. After an individual is sample from the space, the objective function value for the individual is calculated using the function pointer $f(x)$ (Algorithm 2, line 10). The procedure ends with m individuals identified, and the individual that has the best function value is elected and stored in X_{best} .

ALGORITHM 2. Initialize()

```

1: for  $i = 1$  to  $m$  do
2:   for  $k = 1$  to  $n$  do
3:     for  $j = 1$  to  $m$  do
4:        $F_{ij,k} \leftarrow 0$ 
5:     end for
6:      $\lambda \leftarrow U(0, 1)$ 
7:      $x_{i,k} \leftarrow x_k^{\min} + \lambda(x_k^{\max} - x_k^{\min})$ 
8:      $v_{i,k} \leftarrow 0$ 
9:   end for
10: Calculate  $f(X_i)$ 
11: end for
12:  $X_{best} \leftarrow \arg \min \{f(X_i), \forall i\}$ 

```

Figure 2. Initialize function of APO.

3.3 Local Search

Local search is used to carefully exploit the neighborhood of individuals. We adopt SA method as local search algorithm. The method is a well-known stochastic search method, which uses a single point to guide the search. In the method, the parameters T_0 , T_f and p represent the original, terminate temperature and the neighborhood size of the current best individual, respectively. Temperature descends by the parameter γ . These are key parameters of SA algorithm. In this paper, SA algorithm is only applied to local search, which differs from the method for global optimization problem. When dealing with global optimization problem, γ is restrict with $\gamma \in [0.95, 0.99]$, T_0 should be set enough high to make the sample point adequately search the whole feasible domain. While dealing with local searching, we set

$\gamma \in [0.3, 0.7]$ and a relative lower T_0 . Of course, local search can be applied to all individuals in the population. But that will make the number of evaluations drastically increases for the problems with high dimensionality. To reduce the number of function evaluations, we applied the local search procedure to the current best individual only. The choice balances the accuracy of the results and the number of evaluations.

ALGORITHM3. Localsearch()

```

1: for  $k = 1$  to  $n$  do
2:    $Z_k \leftarrow X_{best,k}$ 
3:    $R_k^{\min} \leftarrow X_{best,k} - p$ 
4:    $R_k^{\max} \leftarrow X_{best,k} + p$ 
5: end for
6:  $f(Z) \leftarrow f(X_{best})$ 
7:  $T_k \leftarrow T_0$ 
8: while  $T_k > T_f$  do
9:   for  $i = 1$  to  $LSITER$  do
10:    for  $k = 1$  to  $n$  do
11:       $\lambda \leftarrow U(0, 1)$ 
12:       $y_k \leftarrow (X_{best,k}^{\min} - p) + \lambda * 2p$ 
13:      if  $Y_k < R_k^{\min}$  then
14:         $Y_k \leftarrow R_k^{\min}$ 
15:      end if
16:      if  $Y_k > R_k^{\max}$  then
17:         $Y_k \leftarrow R_k^{\max}$ 
18:      end if
19:    end for
20:    Calculate  $f(Y)$ 
21:     $d \leftarrow f(Y) - f(Z)$ 
22:     $\xi \leftarrow U(0, 1)$ 
23:    if  $d < 0$  or  $\exp(d / T_k) < \xi$  then
24:      for  $k = 1$  to  $n$  do
25:         $Z_k \leftarrow Y_k$ 
26:      end for
27:       $f(Z) \leftarrow f(Y)$ 
28:    end if
29:  end for
30:   $T_k \leftarrow \gamma T_k$ 
31: end while
32: if  $f(Z) < f(X_{best})$  then
33:   for  $k = 1$  to  $n$  do
34:      $X_{best,k} \leftarrow Z_k$ 
35:   end for
36:    $f(X_{best}) \leftarrow f(Z)$ 
37: end if

```

Figure 3. Local search of APO.

The local search of APO can be shown by Figure 3. The procedure iterates as follows: First, The original temperature is set and the search domain is restricted. The current best individual $best$ is assigned to a temporary individual z to store the initial information. The neighborhood size of individual $best$ is calculated according to the parameter p (Algorithm 3, line 3-4).

$[R_k^{\min}, R_k^{\max}]$ is utilized to restrict the local search domain. Next, local search is executed in the temperature T_k . Individual y is randomly sampled from the neighborhood of individual $best$. It is restricted in $[R_k^{\min}, R_k^{\max}]$ (Algorithm 3, line 13-18). If individual y observes a better individual or its Boltzmann equation value is bigger than a random parameter ξ within $LSITER$ iterations, individual z is replaced by individual y . Then, temperature descends and the procedure of local search in the isothermal condition is repeated until T_k is lower than the terminate temperature T_f . Finally, if the objective function value of individual z is better than that of individual $best$, individual $best$ is updated.

3.4 Calculation of Total Force

According to the gravity law, the force exerted on an individual via another individual is inversely proportional to the distance between the individuals and directly proportional to the product of their masses. In each iteration, the mass of each individual is calculated according to their objective function value. It is not constant and changes from iteration to iteration. The mass of each individual is evaluated as equation (2). m_i is the mass of individual i . It can be easily seen that the mass of each individual determines the individual's power of attraction or repulsion. We compute individual's mass with the exponential function, which can ensure the mass is a positive number. The value of individual's mass is restricted to $(e^{-1}, 1)$, excluding the best individual's mass. The magnitude of the best individual's mass is K , K is a positive constant and $K \geq 1$.

$$m_i = \begin{cases} K & \text{if } i = best \\ \frac{f(x_{best}) - f(x_i)}{e^{\max\{f(x_i) - f(x_{best}), i=1, \dots, m\}}} & \text{if } i \neq best \end{cases} \quad (2)$$

The gravity principle only shows the magnitude but no direction of the force. We define the direction of the force by comparing two individuals' fitness values. The force can be calculated with equation (3). The gravity constant $G = 1$. The individual with a better fitness attracts the other one. Contrarily, the individual with a worse fitness repels the other one. $F_{i,k}$ is the total force exerted on the individual i via other individuals in k th dimension, shown by equation (4). The total force is calculated by adding vectorially the forces from each of the other individuals calculated separately. Figure 4 gives the algorithm of calculating total force. Since the individual $best$ has the minimum objective function value, it acts as an absolute individual of attraction and attracts all other individuals in the population. The individual $best$ can not be attracted or repelled by other individuals (Algorithm 4, lines 14-22).

When individual i approximates to individual $best$, the distance between them is close to zero. Then the force on individual i by individual $best$ is infinite according to equation (3). Thus, individual i can not carefully exploit the neighborhood of

individual $best$. So, we restrict the magnitude of the total force exerted on each individual with $F_{i,k} \in [F_k^{\min}, F_k^{\max}]$, shown by Algorithm 4, lines 24-29.

$$F_{ij,k} = \begin{cases} (x_{j,k} - x_{i,k}) \frac{Gm_i m_j}{(x_{i,k} - x_{j,k})^2} & \text{if } f(X_j) < f(X_i) \\ (x_{i,k} - x_{j,k}) \frac{Gm_i m_j}{(x_{i,k} - x_{j,k})^2} & \text{if } f(X_j) \geq f(X_i) \end{cases}, \forall i \neq j \quad (3)$$

$$F_{i,k} = \sum_{\substack{j=1 \\ j \neq i}}^m F_{ij,k} \quad \forall i \neq best \quad (4)$$

ALGORITHM 3. CalcForce():F

```

1: for  $i = 1$  to  $m$  do
2:   for  $k = 1$  to  $n$  do
3:      $F_{i,k} \leftarrow 0$ 
4:   end for
5:   if  $i = best$  then
6:      $m_i = K$ 
7:   else
8:      $m_i = e^{\frac{f(x_{best}) - f(x_i)}{\max\{f(x_i) - f(x_{best}), i=1, \dots, m\}}}$ 
9:   end if
10: end for
11: for  $i = 1$  to  $m$  do
12:   for  $k = 1$  to  $n$  do
13:     for  $j = 1$  to  $m$  do
14:       if  $i \neq best$  then
15:         if  $f(X_j) < f(X_i)$  then
16:            $F_{i,k} \leftarrow F_{i,k} + (x_{j,k} - x_{i,k}) \frac{Gm_i m_j}{(x_{i,k} - x_{j,k})^2}$ 
17:         else
18:            $F_{i,k} \leftarrow F_{i,k} + (x_{i,k} - x_{j,k}) \frac{Gm_i m_j}{(x_{i,k} - x_{j,k})^2}$ 
19:         end if
20:       else
21:          $F_{i,k} \leftarrow 0$ 
22:       end if
23:     end for
24:     if  $F_{i,k} > F_k^{\max}$  then
25:        $F_{i,k} \leftarrow F_k^{\max}$ 
26:     end if
27:     if  $F_{i,k} < F_k^{\min}$  then
28:        $F_{i,k} \leftarrow F_k^{\min}$ 
29:     end if
30:   end for
31: end for

```

Figure 4. Calculation total force function of APO.

3.5 Individual Motion

Each individual is driven by the total force exerted on it. If the motion of each individual complies with Newton's second law completely, then APO method becomes a determine algorithm, which cause that individuals cannot visit some regions along the total force vector. In order to ensure that individuals have a nonzero probability to move the unvisited regions along the direction, a random variable λ is introduced in the velocity equation of each individual. λ is assumed to be uniformly

distributed between 0 and 1. APO has become a stochastic algorithm. The evolutionary equations of each individual's velocity and position are shown as equations (5) and (6) in k th dimension, respectively.

$$v_{i,k}(t+1) = wv_{i,k}(t) + \lambda F_{i,k} / m_i \quad (5)$$

$$x_{i,k}(t+1) = x_{i,k}(t) + v_{i,k}(t+1) \quad (6)$$

In equation (5), w is an inertia weight and linearly decrease with the increased iteration. It is shown

as $w = 0.9 - \frac{\text{iteration}}{\text{MAXITER}} \times 0.5$. When w is bigger, the algorithm

has a capability of global exploration. When w is smaller, the algorithm has a capability of local exploitation. Hence, with the ceaselessly increased iteration, w is constantly lessen and the algorithm has a good capability of global exploration in early search and a good capability of local exploitation in the final stage of the search. The movement of each individual is restricted in the feasible domain which is shown by Figure 5 (Algorithm 5, line 8-13).

ALGORITHM 5. Move(F , iteration)

```

1: for  $i = 1$  to  $m$  do
2:   for  $k = 1$  to  $n$  do
3:     if  $i \neq \text{best}$  then
4:        $w \leftarrow 0.9 - \frac{\text{iteration}}{\text{MAXITER}} \times 0.5$ 
5:        $\lambda \leftarrow U(0, 1)$ 
6:        $v_{i,k} \leftarrow wv_{i,k} + \lambda F_{i,k} / m_i$ 
7:        $x_{i,k} \leftarrow x_{i,k} + v_{i,k}$ 
8:       if  $x_{i,k} > x_k^{\max}$  then
9:          $x_{i,k} \leftarrow x_k^{\max}$ 
10:      end if
11:      if  $x_{i,k} < x_k^{\min}$  then
12:         $x_{i,k} \leftarrow x_k^{\min}$ 
13:      end if
14:    end if
15:  end for
16: end for

```

Figure 5. Individual movement function of APO.

4. EXPERIMENT

In order to analyze the convergence property and the capability of global exploration of APO algorithms, EM algorithms, PSO and an improved PSO algorithm (AEPsO1) were selected to make

comparison experiments. In these experiments, thirteen global optimization problems were used to test the performance of APO algorithms. These functions and their parameters are shown in Table 1. The former seven test functions are selected from literature [5] and the latter six test functions are taken from literature [15].

Table 1. Parameters for Test functions

Function	n	m	Search range	Known optimum	MAX-ITER
Complex	2	10	$[-2, 2]^n$	0.0	50
Davis	2	20	$[-100, 100]^n$	0.0	50
Himmel-blau	2	10	$[-6, 6]^n$	0.0	50
Kearfott	4	10	$[-3, 10]^n$	0.0	50
Sine Envelope	2	20	$[-100, 100]^n$	0.0	75
Stenger	2	10	$[-1, 4]^n$	0.0	75
Griewank	2	30	$[-100, 100]^n$	0.0	100
Tablet	30	20	$[-100, 100]^n$	0.0	6000
Quadric	30	20	$[-100, 100]^n$	0.0	6000
Rosenbrock	30	20	$[-50, 50]^n$	0.0	6000
Griewank	30	20	$[-300, 300]^n$	0.0	6000
Rastrigin	30	20	$[5.12, 5.12]^n$	0.0	6000
Schaffer's f_7	30	20	$[-100, 100]^n$	0.0	6000

Table 2. Parameters of local search for Test functions

Function	p	T_0	T_f	γ	LSITER
Complex	0.01	100	1.0e-10	0.5	10
Davis	1	100	1.0e-10	0.5	10
Himmelblau	0.05	100	1.0e-10	0.5	10
Kearfott	0.05	100	1.0e-10	0.3	10
Sine Envelope	1	100	1.0e-10	0.3	10
Stenger	0.05	100	1.0e-10	0.5	10
Griewank	1	100	1.0e-10	0.5	10
Tablet	1	100	1.0e-10	0.7	100
Quadric	1	100	1.0e-10	0.6	100
Rosenbrock	0.5	100	1.0e-10	0.5	100
Griewank	1	100	1.0e-10	0.3	100
Rastrigin	0.05	100	1.0e-10	0.3	100
Schaffer's f_7	1	100	1.0e-10	0.3	100

We studied two versions of APO, which differ in the local search procedure. One is called APO1, which is without the local search procedure. The other is named APO2, which is with the local search procedure to the best individual. In APO2 algorithm, Simulated Annealing method is applied to local search. The parameters of local search for the thirteen functions are shown in Table 2.

4.1 Low-dimension Functions Test

The former seven test functions in Table 1 are well-known global optimization problems, which are moderately difficult and difficult problems [16]. In APO2 algorithm, local search is applied to all of iterations in APO framework for the seven functions. We compare the results for these functions tested by APO1 and APO2 algorithms with those tested by two different versions of EM methods, which is shown by Table 3. In this paper, EM without local search procedure is called EM1 method, EM2 method is EM with local search procedure applied to the current best point.

The results in Table 3 show that APO algorithms perform better than EM algorithms. When we examine APO algorithms carefully, we can find that the determination of a direction via total force vector resembles the statistical evaluation with the gradient of the objective function. Meanwhile, APO algorithms are more exact than EM algorithms on statistically evaluating the gradient of the objective function. Since the mass of the best individual is set

enough big corresponding to the size of the feasible region in APO algorithms. Thus the best individual's force is big enough to lead most of population towards it. But in EM algorithms, the mass of the best individual is set 1 at any time. Obviously, both the best solution and the average best solution with APOs are better than those with EMs. In addition, APO1 have a better capability of global exploration than EM1. The solutions for these functions with APO1 are all in the vicinity of their global optima, while those with EM1 are almost in the neighborhood of their global optima apart from Davis function. The results show that even if we do not use the local procedure, the average function values are still good. In most of the problems, APO1 is able to approximate the optimum.

Nevertheless, APO1 and EM1 have a poor capability of local exploitation. APO2 and EM2 are APO1 and EM1 combined with local search methods, respectively. In these hybrid methods, local search strategies are embedded inside heuristics methods in order to guide them especially in the vicinity of local minima, and overcome their slow convergence especially in the final stage of the search. Table 3 also shows that the results of APO2 and EM2 much better than APO1 and EM1. In addition, the results with APO2 show that the solution for the functions Davis and Griewank are less satisfactory than those for the other functions. Davis and Griewank are highly irregular in the vicinity of the global optima. APO algorithms are likely trapped in one of local minima of the two functions.

Table 3. Performance comparison between two APOs and two EMs for low-dimension function

Function	EM1		EM2		APO1			APO2		
	Best f(x)	Avg f(x)	Best f(x)	Avg f(x)	Best f(x)	Avg f(x)	Deviation	Best f(x)	Avg f(x)	Deviation
Complex	0.0158	0.0175	0.0000	0.0000	0.0000	0.0000	1.75e-05	0.0000	0.0000	6.37e-10
Davis	1.5641	1.6157	0.2356	0.4538	0.0950	0.3409	2.76e-02	0.0061	0.3129	3.39e-02
Himmelblau	0.0759	0.0934	0.0000	0.0001	0.0000	0.0015	4.65e-04	0.0000	0.0000	3.28e-09
Kearfott	0.0000	0.0008	0.0000	0.0000	0.0000	0.0472	2.70e-02	0.0000	0.0000	2.91e-07
Sine Envelope	0.0400	0.0744	0.0097	0.0352	0.0000	0.0000	6.0e-06	0.0000	0.0000	2.78e-06
Stenger	0.0019	0.0020	0.0000	0.0000	0.0000	0.0000	1.43e-05	0.0000	0.0000	4.14e-10
Griewank	0.0032	0.0896	0.0000	0.0000	0.0000	0.0087	1.57e-03	0.0000	0.0069	1.45e-03

4.2 High-dimension Functions Test

In Table 1, the latter six test functions are Benchmark optimization problems, which often used to data testing. The functions Tablet, Quadric and Rosenbrock are unimodal functions. Rosenbrock is a classical complex optimization problem. Its optimum is in a smooth and steep valley. Little information is supported to optimization algorithms by itself. It is nearly impossible to find its optimum. So Rosenbrock is often utilized to evaluate the executing efficiency of optimization algorithms [17]. The functions Griewank, Rastrigin and Schaffer's f_7 are well-known multimodal nonlinear functions. They have a wide search space and lots of local optima, which are difficult to optimize in Genetic Algorithm.

Table 4 shows that the results for the six Benchmark functions with the four algorithms in the same conditions. Each algorithm independently runs 50 times, 6000 iterations every time. From Table 4, we can see that the solutions for the functions Tablet and Quadric with APO1 are in the neighborhood of optima. And the solutions for Quadric with APO1 are better than those with PSO. These show that APO1 have a good capability of global exploration. However, these solutions are not as good as those with AEPsO1. Because of a poor capability of local exploitation, APO1 is slow convergence especially in the final stage of the search. Hence, in APO2 algorithm, local search runs only in the last 300 iterations of APO framework for the six functions. From the comparison of results of APO1 and APO2, APO2 is a practical remedy to overcome the drawback of slow convergence. The

solutions for the function Rosenbrock have a bigger deviation than those for Tablet and Quadric with the four algorithms. Because the four stochastic algorithms are difficult to find the correct directions in optimizing Rosenbrock function. APO1 and APO2 find the directions faster than PSO and AEPsO1. And the solutions with APO1 and APO2 are better than those of PSO and AEPsO1. These show that APO algorithms have a better stability and faster convergence than PSO and AEPsO1 for the function Rosenbrock. In multimodal Benchmark Problems, the mean and deviation data for the function Griewank and Schaffer's f7 with APO1 and APO2 are obviously better than those with PSO and AEPsO1. However, APO1 and APO2 can not find the global optima limited iterations. They are likely trapped in one of these

local minima. APO1, APO2 and PSO are trapped in one of local minima of Rastrigin function in the final stage of the search. However, AEPsO1 can escape from local minima recur to escape operator.

The solutions of all test functions with APO2 are better than those with APO1, they are almost in the vicinity of their global optima except Rastrigin function. That indicates that APO1 algorithms have a good capability of global exploration but a poor capability of local exploitation, which drawback can be overcome by APO2. The deviation data for all functions with APO1 and APO2 are relatively smaller than those with PSO and AEPsO1, which shows that APO algorithms are robust.

Table 4. Performance comparison between two APOs and two PSOs for high-dimension Benchmark function

Function	Algorithm	Min	Median	Mean	Deviation	Max
Tablet	PSO	8.2e-22	1.3e-20	1.5e-19	7.8e-19	5.6e-18
	AEPsO1	4.1e-115	9.7e-109	1.4e-101	9.1e-101	6.4e-100
	APO1	0.203	0.376	0.461	2.18e-02	1.115
	APO2	2.53e-03	3.45e-03	4.76e-03	1.63e-04	7.21e-03
Quadric	PSO	0.772	70.45	1.0e+2	1.0e+2	4.8e+02
	AEPsO1	1.4e-04	1.3e-03	2.1e-03	2.3e-03	9.0e-03
	APO1	0.137	0.177	0.188	3.9e-03	0.284
	APO2	1.64e-05	2.34e-05	2.19e-05	3.14e-07	2.53e-05
Rosenbrock	PSO	4.873	77.24	93.71	1.5e+02	1.0e+03
	AEPsO1	9.4e-03	22.17	40.13	41.78	1.5e+02
	APO1	22.01	24.62	28.30	9.01	2.5e+02
	APO2	1.57e-03	2.29e-03	8.53	5.59	8.72
Griewank	PSO	6.7e-11	1.96e-02	2.6e-02	2.8e-02	0.108
	AEPsO1	0(20%)	7.4e-03	1.2e-02	1.3e-02	6.1e-02
	APO1	3.79e-03	6.76e-03	1.14e-02	1.55e-03	5.26e-02
	APO2	8.69e-09	1.33e-08	1.00e-02	1.57e-03	4.41e-02
Rastrigin	PSO	30.84	54.72	55.18	12.31	81.59
	AEPsO1	1.2e-13	0.995	1.234	1.577	7.960
	APO1	21.52	55.33	59.55	2.46	1.0e+02
	APO2	26.8638	28.853	45.9073	1.62	68.6519
Schaffer's f ₇	PSO	2.4e+02	2.9e+02	2.9e+02	30.91	3.5e+02
	AEPsO1	54.83	1.5e+02	1.4e+02	41.63	2.4e+02
	APO1	0.728	3.618	4.577	0.485	13.92
	APO2	0.143	2.812	4.146	0.496	17.378

5. CONCLUSION AND FURTHER RESEARCH

In this paper, we have introduced Artificial Physics Optimization as a nature inspired heuristic based on AP method. The APO

framework supplies a new method for global optimization. Preliminary analysis suggested that APO is an effective stochastic population-based search algorithm.

Of course, many improvements should be done to APO method. Firstly, how the parameters of APO algorithm should be chosen

undoubtedly is a key question. For example, the magnitude of F_{\max} and F_{\min} has a great influence to the performance of APO algorithm. Because when individual i closes to the best individual, the force on individual i by the best individual will be very big according to the force equation (2). It is impossible that individual i could visit the vicinity of the best individual. We will modify the force equation to enhance the capability of local exploitation of APO itself in further research. Second, APO is a new method and lacks a deep theoretical foundation. The convergence properties of APO will be studied. Finally, APO can be entrapped in local minima sometimes. Swarm diversity of APO algorithm will be considered in our further research.

6. REFERENCES

- [1] Inger, L. 1994. Simulated annealing: practice versus theory. *Journal of Mathematical Computation Modeling*. 18, 29-57.
- [2] Michalewicz, Z. 1994. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Press, Berlin.
- [3] Zeng Jianchao, Jie Jing, Cui Zhihua. 2004. *Particle Swarm Optimization*. Science and Technology Press, Beijing. 3-8.
- [4] Kennedy, J. and Eberhart, R.C. Particle Swarm Optimization. *Proc. IEEE Int'l. Conf. on Neural Networks*, IV, 1942-1948. Piscataway, NJ: IEEE Service Center.
- [5] Birbil, S. I. and Fang, S.-C. 2003. An Electromagnetism-like Mechanism for Global Optimization. *Journal of Global Optimization*. 25, 3, 263-282.
- [6] Richard A. Formato, JD, PhD. 2008. Central Force Optimization: A New Nature Inspired Computational Framework for Multidimensional Search and Optimization. *Nature Inspired Cooperative Strategies for Optimization (NICSO)*. 129, 221-238.
- [7] Spears D F, Kerr W, et al. 2005. An Overview of Physicomimetics. *Lecture Notes in Computer Science-State of the Art Series*. 3324, 84-97.
- [8] Spears W M, Spears W M, Spears D F, Hamann J, et al. 2004. Distributed, Physics-Based Control of Swarms of Vehicles. *Autonomous Robots*. 17, 2-3, 137-162.
- [9] Spears D F, Spears W M. 2003. Analysis of a Phase Transition in a Physics-Based Multiagent System. *Lecture Notes in Computer Science*. 2699, 193-207.
- [10] Spears W M, Heil R, Spears D F, Zarzhitsky D. 2004. Physicomimetics for Mobile Robot Formations. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-04)*. New York, USA. 3, 1528-1529.
- [11] Spears W M, Heil, R. Zarzhitsky, D. 2005. Artificial Physics for Mobile Robot Formations. *Systems, Man and Cybernetics*. 3, 2287-2292.
- [12] Spears D F, Kerr W, Spears W F. 2006. Physics-Based Robots Swarms For Coverage Problems. *International Journal on Intelligent Control and Systems*. 11, 3, 11-23.
- [13] Kerr W, Spears D F, Spears W M, et al. 2005. Two Formal Gas Models for Multi-agent Sweeping and Obstacle Avoidance. *Lecture Notes in Artificial Intelligence*. 3228, 111-130.
- [14] Spears W M, Spears D F. 1999. Using Artificial Physics to Control Agents. In: *IEEE International Conference on Information, Intelligence, and Systems*. Washington, DC. 281-288.
- [15] He Ran, Wang Yongji, Wang Qing et al. 2005. An Improved Particle Swarm Optimization Based on Self-Adaptive Escape Velocity. *Journal of Software*. 16, 12, 2036-2044.
- [16] Torn, A., Ali, M. M. and Viitanen, S. 1999. Stochastic global optimization: Problem classes and solution techniques. *Journal of Global Optimization*. 14, 437-447.
- [17] Ratnaweera A, Halgamuge SK, Watson HC. 2004. Self-Organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. On Evolutionary Computation*. 8, 3, 240-255.