3

Social Emotional Optimization Algorithm with Random Emotional Selection Strategy

Zhihua Cui[1,2], Yuechun Xu[1] and Jianchao Zeng[1]
[1]*Complex System and Computational Intelligence Laboratory,*
Taiyuan University of Science and Technology,
[2]*State Key Laboratory of Novel Software Techchnology,*
Nanjing University,
China

1. Introduction

With the industrial and scientific developments, many new optimization problems are needed to be solved. Several of them are complex, multi-modal, high dimensional, non-differential problems. Therefore, some new optimization techniques have been designed, such as genetic algorithm, simulated annealing algorithm, Tabu search, etc. However, due to the large linkage and correlation among different variables, these algorithms are easily trapped to a local optimum and failed to obtain the reasonable solution.

Swarm intelligence (SI) is a recent research topic which mimics the animal social behaviors. Up to now, many new swarm intelligent algorithms have been proposed, such as group search optimizer[1], artificial physics optimization[2], firefly algorithm[3] and ant colony optimizer (ACO)[4]. All of them are inspired by different animal group systems. Generally, they are decentralized, self-organized systems, and a population of individuals are used to interacting locally. Each individual maintains several simple rules, and emergence of "intelligent" global behaviour are used to mimic the optimization tasks. The most famous one is particle swarm optimization.

Particle swarm optimization (PSO) [5-8] is a population-based, self-adaptive search optimization method motivated by the observation of simplified animal social behaviors such as fish schooling, bird flocking, etc. It is becoming very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution. In a PSO system, multiple candidate solutions coexist and collaborate simultaneously. Each solution called a "particle", flies in the problem search space looking for the optimal position to land. A particle, as time passes through its quest, adjusts its position according to its own "experience" as well as the experience of neighboring particles. Tracking and memorizing the best position encountered build particle's experience. For that reason, PSO possesses a memory (i.e. every particle remembers the best position it reached during the past). PSO system combines local search method (through self experience) with global search methods (through neighboring experience), attempting to balance exploration and exploitation.

Human society is a complex group which is more effective than other animal groups. Therefore, if one algorithm mimics the human society, the effectiveness maybe more robust than other swarm intelligent algorithms which are inspired by other animal groups. With this manner, social emotional optimization algorithm (SEOA) was proposed by Zhihua Cui et al. in 2010[9-13]

In SEOA methodology, each individual represents one person, while all points in the problem space constructs the status society. In this virtual world, all individuals aim to seek the higher social status. Therefore, they will communicate through cooperation and competition to increase personal status, while the one with highest score will win and output as the final solution. In the experiments, social emotional optimization algorithm (SEOA) has a remarkable superior performance in terms of accuracy and convergence speed [9-13].

In this chapter, we proposed a novel improved social emotional optimization algorithm with random emotional selection strategy to evaluate the performance of this algorithm on 5 benchmark functions in comparison with standard SEOA and other swarm intelligent algorithms.

The rest of this paper is organized as follows: The standard version of social emotional optimization algorithm is presented in section 2, while the modification is listed in section 3. Simulation resutls are listed in section 4.

2. Standard social emotional optimization algorithm

In this paper, we only consider the following unconstrained problem:

$$min\, f(\vec{x}) \quad x \in [L,U]^D \subseteq R^D$$

In human society, all people do their work hardly to increase their social status. To obtain this object, people will try their bests to find the path so that more social wealthes can be rewarded. Inspired by this phenomenon, Cui et al. proposed a new population-based swarm methodology, social emotional optimization algorithm, in which each individual simulates a virtual person whose decision is guided by his emotion. In social emotional optimization algorithm methodology, each individual represents a virtual person, in each generation, he will select his behavior according to the corresponding emotion index. After the behavior is done, a status value is feedback from the society to confirm whether this behavior is right or not. If this choice is right, the emotion index of himself will increase, and vice versa.

In the first step, all individuals's emotion indexes are set to 1, with this value, they will choice the following behaviour:

$$\vec{x}_j(1) = \vec{x}_j(0) \oplus \text{Manner}_1 \tag{1}$$

where $\vec{x}_j(1)$ represents the social position of j's individual in the initialization period, the corresponding fitness value is denoted as the society status. Symbo $\oplus$ means the operation, in this paper, we only take it as addition operation +. Since the emotion index of j is 1, the movement phase $Manner_1$ is defined by:

$$Manner_1 = -k_1 \cdot rand_1 \cdot \sum_{w=1}^{L} (\vec{x}_w(0) - \vec{x}_j(0)) \tag{2}$$

where k_1 is a parameter used to control the emotion changing size, $rand_1$ is one random number sampled with uniform distribution from interval (0,1). The worst L individuals are selected to provide a reminder for individual j to avoid the wrong behaviour. In the initialization period, there is a little emotion affection, therefore, in this period, there is a little good experiences can be referred, so, $Manner_1$ simulates the affection by the wrong experiences.

In t generation, if individual j does not obtain one better society status value than previous value, the j's emotion index is decreased as follows:

$$BI_j(t+1) = BI_j(t) - \Delta \tag{3}$$

where Δ is a predefined value, and set to 0.05, this value is coming from experimental tests. If individual j is rewarded a new status value which is the best one among all previous iterations, the emotion index is reset to 1.0:

$$BI_j(t+1) = 1.0 \tag{4}$$

Remark: According to Eq.(3), $BI_j(t+1)$ is no less than 0.0, in other words, if $BI_j(t+1) < 1.0$, then $BI_j(t+1) = 0.0$.

In order to simulate the behavior of human, three kinds of manners are designed, and the next behavior is changed according to the following three cases:

If $BI_j(t+1) < TH_1$

$$\vec{x}_j(t+1) = \vec{x}_j(t) \oplus Manner_2 \tag{5}$$

If $TH_1 \le BI_j(t+1) < TH_2$

$$\vec{x}_j(t+1) = \vec{x}_j(t) \oplus Manner_3 \tag{6}$$

Otherwise

$$\vec{x}_j(t+1) = \vec{x}_j(t) \oplus Manner_4 \tag{7}$$

Parameters TH_1 and TH_2 are two thresholds aiming to restrict the different behavior manner. For Case1, because the emotion index is too small, individual j prefers to simulate others successful experiences. Therefore, the symbol $Manner_2$ is updated with:

$$\begin{aligned} Manner_2 &= k_3 \cdot rand_3 \cdot (\vec{X}_{j,best}(t) - \vec{x}_j(t)) \\ &+ k_2 \cdot rand_2 \cdot (\vec{Status}_{best}(t) - \vec{x}_j(t)) \end{aligned} \tag{8}$$

where $\vec{Status}_{best}(t)$ represents the best society status position obtained from all people previously. In other words, it is:

$$\vec{Status}_{best}(t) = \arg\ \min\{f(\vec{x}_w(h) \mid 1 \le h \le t)\} \tag{9}$$

With the similar method, $Manner_2$ is defined:

$$\begin{aligned} Manner_3 &= k_3 \cdot rand_3 \cdot (\vec{X}_{j,best}(t) - \vec{x}_j(t)) \\ &+ k_2 \cdot rand_2 \cdot (\vec{Status}_{best}(t) - \vec{x}_j(t)) \\ &- k_1 \cdot rand_1 \cdot \sum_{w=1}^{L} (\vec{x}_w(0) - \vec{x}_j(0)) \end{aligned} \tag{10}$$

where $\vec{X}_{j,best}(t)$ denotes the best status value obtained by individual j previously, and is defined by

$$\vec{X}_{j,best}(t) = \arg\ \min\{f(\vec{x}_j(h) \mid 1 \le h \le t)\} \tag{11}$$

For $Manner_4$, we have

$$\begin{aligned} Manner_4 &= k_3 \cdot rand_3 \cdot (\vec{X}_{j,best}(t) - \vec{x}_j(t)) \\ &- k_1 \cdot rand_1 \cdot \sum_{w=1}^{L} (\vec{x}_w(0) - \vec{x}_j(0)) \end{aligned} \tag{12}$$

$Manner_2$, $Manner_3$ and $Manner_4$ refer to three different emotional cases. In the first case, one individual's movement is protective, aiming to preserve his achievements (good experiences) in $Manner_2$ due to the still mind. With the increased emotion, more rewards are expected, so in $Manner_3$, a temporized manner in which the dangerous avoidance is considered by individual to increase the society status. Furthermore, when the emotional is larger than one threshold, it simulates the individual is in surged mind, in this manner, he lost the some good capabilities, and will not listen to the views of others, $Manner_4$ is designed to simulate this phenomenon.

To enhance the global capability, a mutation strategy, similarly with evolutionary computation, is introduced to enhance the ability escaping from the local optima, more details of this mutation operator is the same as Cai XJ[14], please refer to corresponding reference. The detail of social emotion optimization are listed as follows:

Step 1. Initializing all individuals respectively, the initial position of individuals randomly in problem space.
Step 2. Computing the fitness value of each individual according to the objective function.
Step 3. For individual j, determining the value $\vec{X}_{j,best}(0)$.
Step 4. For all population, determining the value $\vec{Status}_{best}(0)$.
Step 5. Determining the emotional index according to Eq.(5)-(7) in which three emotion cases are determined for each individual.
Step 6. Determining the decision with Eq. (8)-(12), respectively.
Step 7. Making mutation operation.
Step 8. If the criteria is satisfied, output the best solution; otherwise, goto step 3.

3. Random emotional selection strategy

To mimic the individual decision mechanism, emotion index $BI_j(t)$ is employed to simulate the personal decision mechanism. However, because of the determined emotional selection strategy, some stochastic aspects are omitted. To provide a more precisely simulation, we replace the determined emotional selection strategy in the standard SEOA with three different random manners to mimic the human emotional changes.

3.1 Gauss distribution

Gauss distribution is a general distribution, and in WIKIPEDIA is defined as "normalis a continuous probability distribution that is often used as a first approximation to describe real-valued random variables that tend to cluster around a single mean value. The graph of the associated probability density function is "bell"-shaped, and is known as the Gaussian function or bell curve" [15] (see Fig.1):

$$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where parameter μ is called the mean, σ^2 is the variance. The standard normal Gauss distribution is one special case with $\mu=0$ and $\sigma^2=1$.

3.2 Cauchy distribution

Cauchy distribution is also called Lorentz distribution, Lorentz(ian) function, or Breit–Wigner distribution. The probability density function of Cauchy distribution is

$$f(x,x_0,\gamma)=\frac{1}{\pi}\cdot\frac{\gamma}{(x-x_0)^2+\gamma^2}$$

where x_0 is the location parameter, specifying the location of the peak of the distribution, and γ is the scale parameter which specifies the half-width at half-maximum. The special case when $x_0 = 0$ and $\gamma = 1$ is called the standard Cauchy distribution with the probability density function

$$f(x) = \frac{1}{\pi(1+x^2)}$$

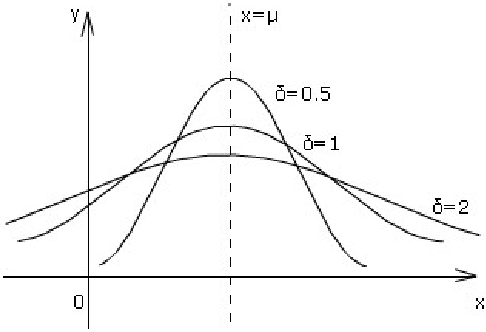


Fig. 1. Illustration of Probability Density Function for Gauss Distribution

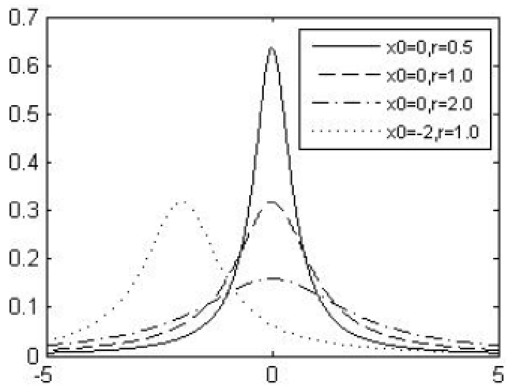


Fig. 2. Illustration of Probability Density Function for Cauchy Distribution

3.3 Levy distribution

In the past few years, there are more and more evidence from a variety of experimental, theoretical and field studies that many animals employ a movement strategy approximated by Levy flight when they are searching for resources. For example, wandering Albatross were observed to adopt Levy flight to adapted stochastically to their prey field[16]. Levy flight patterns have also been found in a laboratory-scale study of starved fruit flies. In a recent study by Sims[17], marine predators adopted Levy flights to pursuit Levy-like fractal distributions of prey density. In [18], the authors concluded that ``Levy flights may be a universal strategy applicable across spatial scales ranging from less than a meter, ..., to several kilometers, and adopted by swimming, walking, and airborne organisms". Shaped by natural selection, the Levy flights searching strategies of all living animals should be regarded as optimal strategies to some degree[19]. Therefore, it would be interesting to incorporate Levy flight into the SEOA algorithm to improve the performance.

Indeed, several studies have already incorporated Levy flight into heuristic search algorithms. In [20], the authors proposed a novel evolutionary programming with mutations based on the Levy probability distribution. In order to improve a swarm intelligence algorithm, Particle Swarm Optimizer, in [21], a novel velocity threshold automation strategy was proposed by incorporated with Levy probability distribution. In a different study of PSO algorithm[22], the particle movement characterized by a Gaussian probability distribution was replaced by particle motion with a Levy flight. A mutation operator based on the Levy probability distribution was also introduced to the Extremal Optimization (EO) algorithm[23].

Levy flights comprise sequences of straight-line movements with random orientations. Levy flights are considered to be 'scale-free' since the straight-line movements have no characteristic scale. The distribution of the straight-line movement lengths, L has a power-law tail:

$$P(L) \propto L^{-\mu}$$

where $1 < \mu < 3$.

The sum of the a set $\{L_i\}$ converge to the Levy distribution, which has the following probability density:

$$D_{\alpha,\gamma}(L) = \frac{1}{\pi}\int_0^{+\infty} e^{-\gamma q^{\alpha}} \cos(qL) dq$$

where α and γ are two parameters that control the sharpness of the graph and the scale unit of the distribution, respectively. The two satisfy $1 < \alpha < 2$ and $\gamma > 0$. For $\alpha \to 1$, the distribution becomes Cauchy distribution and for $\alpha \to 2$, the distribution becomes Gaussian distribution. Without losing generality, we set the scaling factor $\gamma = 1$.

Since, the analytic form of the Levy distribution is unknown for general α , in order to generate Levy random number, we adopted a fast algorithm presented in [24]. Firstly, Two independent random variables x and y from Gaussian distribution are used to perform a nonlinear transformation

$$v = \frac{x}{|y|^{\frac{1}{\alpha}}}$$

Then the random variable z :

$$z = \gamma^{\frac{1}{\alpha}} w$$

now in the Levy distribution is generated using the following nonlinear transformation

$$w = \{ |K(\alpha) - 1| e^{\frac{-v}{C(\alpha)}} + 1 \} \cdot v$$

where the values of parameters $K(\alpha)$ and $C(\alpha)$ are given in[24].

In each iteration, different random number $BI_j(t)$ is generated for different individual with Gauss distribution, Cauchy distribution and Levy fligh, then choices the different rules for different conditions according to Eq.(5)-(7).

4. Simulation

To testify the performance of proposed variant SEOA with random emotional selection strategy, five typical unconstraint numerical benchmark functions are chosen, and compared with standard particle swarm optimization (SPSO), modified particle swarm optimization with time-varying accelerator coefficients (TVAC)[25] and the standard version of SEOA (more details about the test suits can be found in [26]). To provide a more clearly insight, SEOA combined with Gauss distribution, Cauchy distribution and Levy distribution are denoted with SEOA-GD, SEOA-CD and SEOA-LD, respectively.

Sphere Model:

$$f_1(x) = \sum_{j=1}^{n} x_j^2$$

where $|x_j| \leq 100.0$, and

$$f_1(x^*) = f_1(0,0,...,0) = 0.0$$

Rosenbrock Function:

$$f_2(x) = \sum_{j=1}^{n-1} [100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2]$$

where $| \, |x_j| \leq 30.0$, and

$$f_2(x^*) = f_1(0,0,...,0) = 0.0$$

Schwefel 2.26:

$$f_3(x) = -\sum_{i=1}^{n} \left(x_i \sin(\sqrt{|x_i|}) \right),$$

where $|x_j| \leq 500.0$, and

$$f_3(x^*) = f_3(420.9687, ..., 420.9687) = -418.98n$$

Rastrigin:

$$f_4(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$$

where $|x_j| \le 5.12$, and

$$f_4(x^*) = f_4(0.0, ..., 0.0) = 0.0$$

Penalized Function2:

$$f_5(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{j=1}^{n-1} (x_j - 1)^2 \cdot [1 + \sin^2(3\pi x_{j+1})] + (x_n - 1)^2$$
$$\cdot [1 + \sin^2(3\pi x_n)]\} + \sum_{j=1}^{n} u(x_j, 5, 100, 4)$$

where $|x_j| \le 50.0$, and

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a, \\ 0, -a \le x_i \le a, \\ k(-x_i - a)^m, x_i < -a. \end{cases}$$

$$y_i = 1 + \frac{1}{4}(x_i + 1)$$

$$f_5(x^*) = f_5(1, ..., 1) = 0.0$$

The inertia weight w is decreased linearly from 0.9 to 0.4 for SPSO and TVAC, accelerator coefficients c_1 and c_2 are both set to 2.0 for SPSO, as well as in TVAC, c_1 decreases from 2.5 to 0.5, while c_2 increases from 0.5 to 2.5. Total individuals are 100, and the velocity threshold v_{max} is set to the upper bound of the domain. The dimensionality is 30, 50, 100, 150, 200, 250 and 300. In each experiment, the simulation run 30 times, while each time the largest iteration is 50 times dimension, e.g. the largest iteration is 1500 for dimension 30. For SEOA, all parameters are used the same as Cui et al[9].

1. Comparison with SEOA-GD, SEOA-CD and SEOC-LD

From the Tab.1, we can find the SEOA-GD is the best algorithm for all 5 benchmarks especially for high-dimension cases. This phenomenon implies that SEOA-GD is the best choice between three different random variants.

2. Comparison with SPSO, TVAC and SEOA

In Tab.2, SEOA-GD is superior to other three algorithm in all benchmarks especially for multi-modal functions.

Based on the above analysis, we can draw the following conclusion:

SEOA-GD is the most stable and effective among three random variants, and is superior to other optimization algorithms significantly, e.g. SPSO, TVAC and SEOA. It is especially suit for high-dimensional cases.

Dimension	Algorithm	Mean Value	Standard Deviation
30	SEOA-GD	6.4355e-034	2.6069e-033
	SEOA-LD	2.4887e-019	1.3127e-018
	SEOA-CD	3.8304e-034	8.9763e-034
50	SEOA-GD	7.1686e-031	3.8036e-030
	SEOA-LD	2.5210e-016	7.5977e-016
	SEOA-CD	3.1894e-032	1.2666e-031
100	SEOA-GD	1.0111e-032	2.3768e-032
	SEOA-LD	3.8490e-013	1.1092e-012
	SEOA-CD	2.4269e-030	1.3091e-029
150	SEOA-GD	6.8757e-032	2.8083e-031
	SEOA-LD	5.7554e-012	3.1401e-011
	SEOA-CD	1.9043e-032	5.0848e-032
200	SEOA-GD	3.1075e-032	5.3236e-032
	SEOA-LD	1.1350e-009	4.4368e-009
	SEOA-CD	2.7272e-031	8.6026e-031
250	SEOA-GD	7.1304e-031	2.7851e-030
	SEOA-LD	9.0872e-010	1.9692e-009
	SEOA-CD	1.0602e-029	5.4445e-029
300	SEOA-GD	1.2563e-029	6.7502e-029
	SEOA-LD	3.3374e-009	6.4169e-009
	SEOA-CD	1.2338e-027	6.7241e-027

(a) Sphere Model

Dimension	Algorithm	Mean Value	Standard Deviation
30	SEOA-GD	1.9254e+001	3.1878e+001
	SEOA-LD	5.7495e+001	5.5242e+001
	SEOA-CD	1.7432e+001	3.6001e+001
50	SEOA-GD	1.2247e+001	2.4146e+001
	SEOA-LD	1.0847e+002	7.3577e+001
	SEOA-CD	3.1019e+001	4.6183e+001
100	SEOA-GD	3.3119e+001	5.7253e+001
	SEOA-LD	2.6886e+002	1.1566e+002
	SEOA-CD	3.4328e+001	5.7243e+001
150	SEOA-GD	3.2798e+001	5.0613e+001
	SEOA-LD	3.7234e+002	9.1565e+001
	SEOA-CD	5.6862e+001	8.7306e+001

200	SEOA-GD	7.4345e+001	6.7799e+001
	SEOA-LD	3.6658e+002	8.1035e+001
	SEOA-CD	9.5224e+001	1.2905e+002
250	SEOA-GD	7.9152e+001	1.7714e+002
	SEOA-LD	4.1573e+002	1.0684e+002
	SEOA-CD	7.0330e+001	9.5850e+001
300	SEOA-GD	7.8918e+001	1.0940e+002
	SEOA-LD	7.2125e+002	1.6142e+002
	SEOA-CD	9.2294e+001	1.7148e+002

(b) Rosenbrock

Dimension	Algorithm	Mean Value	Standard Deviation
30	SEOA-GD	-1.0935e+004	3.1474e+002
	SEOA-LD	-1.0485e+004	3.7499e+002
	SEOA-CD	-1.0846e+004	3.4926e+002
50	SEOA-GD	-1.8013e+004	4.3216e+002
	SEOA-LD	-1.7623e+004	5.6499e+002
	SEOA-CD	-1.7997e+004	4.5048e+002
100	SEOA-GD	-3.6064e+004	5.8230e+002
	SEOA-LD	-3.3434e+004	1.3006e+003
	SEOA-CD	-5.4032e+004	5.6218e+002
150	SEOA-GD	-5.3692e+004	6.5254e+002
	SEOA-LD	-4.5623e+004	2.5695e+003
	SEOA-CD	-5.4032e+004	5.6218e+002
200	SEOA-GD	-7.1830e+004	7.4485e+002
	SEOA-LD	-6.2516e+004	2.4362e+003
	SEOA-CD	-7.1926e+001	8.0021e+002
250	SEOA-GD	-9.0088e+004	1.0428e+003
	SEOA-LD	-7.3541e+004	4.0967e+003
	SEOA-CD	-8.9629e+004	8.8930e+002
300	SEOA-GD	-1.0853e+005	2.0551e+003
	SEOA-LD	-8.5244e+004	3.7267e+003
	SEOA-CD	-1.0788e+005	1.1546e+003

(c) Schwefel 2.26

Dimension	Algorithm	Mean Value	Standard Deviation
30	SEOA-GD	5.6381e-001	7.6996e-001
	SEOA-LD	1.1343e+001	5.1179e+000
	SEOA-CD	6.9647e-001	1.0170e+000
50	SEOA-GD	1.0945e+000	1.1787e+000
	SEOA-LD	3.5087e+001	1.3085e+001
	SEOA-CD	9.9496e-001	9.0513e-001

100	SEOA-GD	1.9927e+000	1.3044e+000
	SEOA-LD	6.7273e+001	1.9863e+001
	SEOA-CD	1.8904e+000	1.3156e+000
150	SEOA-GD	2.9849e+000	1.6317e+000
	SEOA-LD	1.6024e+002	3.0511e+001
	SEOA-CD	2.1557e+000	1.2823e+000
200	SEOA-GD	3.2502e+000	2.1216e+000
	SEOA-LD	2.1515e+002	4.3832e+001
	SEOA-CD	3.7145e+000	1.7709e+000
250	SEOA-GD	5.2733e+000	2.1884e+000
	SEOA-LD	2.4853e+002	4.8847e+001
	SEOA-CD	5.0743e+000	1.4861e+000
300	SEOA-GD	5.6049e+000	2.4578e+000
	SEOA-LD	4.4945e+002	8.3985e+001
	SEOA-CD	5.7376e+000	2.2881e+000

(d) Rastrigin

Dimension	Algorithm	Mean Value	Standard Deviation
30	SEOA-GD	6.7596e-020	3.7024e-019
	SEOA-LD	3.6502e-028	1.1039e-027
	SEOA-CD	3.5767e-032	5.3917e-032
50	SEOA-GD	2.8538e-022	1.5631e-021
	SEOA-LD	1.1715e-027	3.6895e-027
	SEOA-CD	4.3395e-026	2.3769e-025
100	SEOA-GD	3.7192e-030	1.7204e-029
	SEOA-LD	1.0191e-017	5.1895e-017
	SEOA-CD	6.6188e-021	3.6152e-020
150	SEOA-GD	2.0858e-030	5.0533e-030
	SEOA-LD	5.8928e-025	2.3415e-024
	SEOA-CD	3.0817e-019	1.6879e-018
200	SEOA-GD	2.9720e-026	1.5923e-025
	SEOA-LD	4.4726e-020	1.9939e-019
	SEOA-CD	1.4251e-030	2.9692e-030
250	SEOA-GD	6.7744e-024	3.7100e-023
	SEOA-LD	7.7143e-025	1.0616e-024
	SEOA-CD	3.0722e-023	1.6827e-022
300	SEOA-GD	2.7092e-030	4.7730e-030
	SEOA-LD	4.4726e-020	1.9939e-019
	SEOA-CD	1.6320e-026	7.5692e-026

(e) Penalized 2

Table 1. Comparison results between SEOA-GD, SEOA-CD and SEOA-LD

Dimension	Algorithm	Mean Value	Standard Deviation
30	SPSO	1.1470e-009	1.9467e-009
	TVAC	4.1626e-030	1.2140e-029
	SEOA	2.9026e-010	2.4315e-010
	SEOA-GD	6.4355e-034	2.6069e-033
50	SPSO	1.6997e-007	2.2555e-007
	TVAC	1.0330e-012	3.7216e-012
	SEOA	3.1551e-010	2.0241e-010
	SEOA-GD	7.1686e-031	3.8036e-030
100	SPSO	3.0806e-004	3.6143e-004
	TVAC	1.4014e-004	3.0563e-004
	SEOA	1.4301e-009	7.0576e-010
	SEOA-GD	1.0111e-032	2.3768e-032
150	SPSO	1.4216e-002	8.3837e-003
	TVAC	3.9445e-001	1.7831e+000
	SEOA	3.3950e-000	1.4518e-009
	SEOA-GD	6.8757e-032	2.8083e-031
200	SPSO	1.5234e-001	1.1698e-001
	TVAC	2.1585e-001	4.1999e-001
	SEOA	7.2473e-009	3.1493e-009
	SEOA-GD	3.1075e-032	5.3236e-032
250	SPSO	1.0056e+000	1.0318e+000
	TVAC	8.1591e-001	3.8409e+000
	SEOA	1.4723e-008	5.4435e-009
	SEOA-GD	7.1304e-031	2.7851e-030
300	SPSO	1.0370e+ 001	2.2117e+001
	TVAC	3.1681e+000	1.2412e+001
	SEOA	2.0420e-008	6.4868e-009
	SEOA-GD	1.2563e-029	6.7502e-029

(a) Sphere Model

Dimension	Algorithm	Mean Value	Standard Deviation
30	SPSO	5.6170e+001	4.3585e+001
	TVAC	3.3589e+001	4.1940e+001
	SEOA	4.7660e+001	2.8463e+001
	SEOA-GD	1.9254e+001	3.1878e+001
50	SPSO	1.1034e+002	3.7489e+001
	TVAC	7.8126e+001	3.2497e+001
	SEOA	8.7322e+001	7.4671e+001
	SEOA-GD	1.2247e+001	2.4146e+001
100	SPSO	4.1064e+002	1.0585e+002
	TVAC	2.8517e+002	9.8129e+001
	SEOA	1.3473e+002	5.4088e+001
	SEOA-GD	3.3119e+001	5.7253e+001

150	SPSO	8.9132e+002	1.6561e+002
	TVAC	1.6561e+002	6.4228e+001
	SEOA	2.2609e+002	9.6817e+001
	SEOA-GD	3.2798e+001	5.0613e+001
200	SPSO	2.9071e+003	5.4259e+002
	TVAC	8.0076e+002	2.0605e+002
	SEOA	2.9250e+002	9.2157e+001
	SEOA-GD	7.4345e+001	6.7799e+001
250	SPSO	7.4767e+003	3.2586e+003
	TVAC	1.3062e+003	3.7554e+002
	SEOA	3.4268e+002	9.0459e+001
	SEOA-GD	7.9152e+001	1.7714e+002
300	SPSO	2.3308e+004	1.9727e+004
	TVAC	1.4921e+003	3.4572e+002
	SEOA	3.8998e+002	5.1099e+001
	SEOA-GD	7.8918e+001	1.0940e+002

(b) Rosenbrock

Dimension	Algorithm	Mean Value	Standard Deviation
30	SPSO	-6.2762e+003	1.1354e+003
	TVAC	-6.7672e+003	5.7051e+002
	SEOA	-1.0716e+004	4.0081e+002
	SEOA-GD	-1.0935e+004	3.1474e+002
50	SPSO	-1.0091e+004	1.3208e+003
	TVAC	-9.7578e+003	9.6392e+002
	SEOA	-1.7065e+004	6.9162e+002
	SEOA-GD	-1.8013e+004	4.3216e+002
100	SPSO	-1.8148e+004	2.2012e+003
	TVAC	-1.7944e+004	1.5061e+003
	SEOA	-3.2066e+004	8.9215e+002
	SEOA-GD	-3.6064e+004	5.8230e+002
150	SPSO	-2.5037e+004	4.7553e+003
	TVAC	-2.7863e+004	1.6351e+003
	SEOA	-4.5814e+004	1.3892e+003
	SEOA-GD	-5.3692e+004	6.5254e+002
200	SPSO	-3.3757e+004	3.4616e+003
	TVAC	-4.0171e+004	4.3596e+003
	SEOA	-5.9469e+004	1.6065e+003
	SEOA-GD	-7.1830e+004	7.4485e+002
250	SPSO	-3.9984e+004	4.7100e+003
	TVAC	-4.7338e+004	3.7545e+003
	SEOA	-7.3460e+004	1.5177e+003
	SEOA-GD	-9.0088e+004	1.0428e+003

300	SPSO	-4.6205e+004	6.0073e+003
	TVAC	-5.6873e+004	3.5130e+003
	SEOA	-8.6998e+004	2.1240e+003
	SEOA-GD	-1.0853e+005	2.0551e+003

(c) Schwefel 2.26

Dimension	Algorithm	Mean Value	Standard Deviation
30	SPSO	1.7961e+001	4.2277e+000
	TVAC	1.5472e+001	4.2024e+000
	SEOA	1.8453e+001	5.6818e+000
	SEOA-GD	5.6381e-001	7.6996e-001
50	SPSO	3.9959e+001	7.9259e+000
	TVAC	3.8007e+001	7.0472e+000
	SEOA	3.8381e+001	9.6150e+000
	SEOA-GD	1.0945e+000	1.1787e+000
100	SPSO	9.3680e+001	9.9635e+000
	TVAC	8.4479e+001	9.4569e+000
	SEOA	8.0958e+001	1.1226e+001
	SEOA-GD	1.9927e+000	1.3044e+000
150	SPSO	1.5354e+002	1.2171e+001
	TVAC	1.3693e+002	2.0096e+001
	SEOA	1.3112e+002	1.5819e+001
	SEOA-GD	2.9849e+000	1.6317e+000 1.631749589612318e+000
200	SPSO	2.2828e+002	1.1196e+001
	TVAC	1.9920e+002	2.8291e+001
	SEOA	1.6894e+002	1.8414e+001
	SEOA-GD	3.2502e+000	2.1216e+000
250	SPSO	2.8965e+002	2.8708e+001
	TVAC	2.4617e+002	2.4220e+001
	SEOA	2.3165e+002	2.6751e+001
	SEOA-GD	5.2733e+000	2.1884e+000
300	SPSO	3.5450e+002	1.9825e+001
	TVAC	2.7094e+002	3.7640e+001
	SEOA	2.8284e+002	2.6353e+001
	SEOA-GD	5.6049e+000	2.4578e+000

(d) Rastrigin

Dimension	Algorithm	Mean Value	Standard Deviation
30	SPSO	5.4944e-004	2.4568e-003
	TVAC	9.3610e-027	4.1753e-026
	SEOA	9.7047e-012	5.7057e-012
	SEOA-GD	6.7596e-020	3.7024e-019

50	SPSO	6.4280e-003	1.0769e-002
	TVAC	4.9271e-002	2.0249e-001
	SEOA	2.5386e-011	4.0780e-011
	SEOA-GD	2.8538e-022	1.5631e-021
100	SPSO	3.8087e+001	1.8223e+001
	TVAC	3.7776e-001	6.1358e-001
	SEOA	2.6187e-010	5.3124e-010
	SEOA-GD	3.7192e-030	1.7204e-029
150	SPSO	1.6545e+002	5.5689e+001
	TVAC	1.2655e+000	1.4557e+000
	SEOA	1.8553e-009	2.9614e-009
	SEOA-GD	2.0858e-030	5.0533e-030
200	SPSO	1.8030e+003	2.8233e+003
	TVAC	3.7344e+000	2.6830e+000
	SEOA	2.9760e-006	1.2540e-005
	SEOA-GD	2.9720e-026	1.5923e-025
250	SPSO	6.7455e+003	9.5734e+003
	TVAC	2.8991e+000	1.3026e+000
	SEOA	1.8303e-007	1.5719e-007
	SEOA-GD	6.7744e-024	3.7100e-023
300	SPSO	3.2779e+004	4.4432e+004
	TVAC	3.7344e+000	2.6830e+000
	SEOA	2.9760e-006	1.2540e-005
	SEOA-GD	2.7092e-030	4.7730e-030

(e) Penalized 2

Table 2. Comparison results between SEOA-GD and SPSO, TVAC, SEOA

5. Conclusion

In standard version of social emotional optimization algorithm, all individuals' decision are influenced by one constant emotion selection strategy. However, this strategy may provide a wrong search selection due to some randomness omitted. Therefore, to further improve the performance, three different random emotional selection strategies are added. Simulation results show SEOA with Gauss distribution is more effective. Future research topics includes the application of SEOA to the other problems.

6. Acknowledgement

This paper were supported by the Key Project of Chinese Ministry of Education under Grant No.209021 and National Natural Science Foundation of China under Grant 61003053.

7. References

He S, Wu QH and Saunders JR. (2006) Group search optimizer an optimization algorithm inspired by animal searching behavior. *IEEE International Conference on Evolutionary Computation*, pp.973–990.

Xie LP, Tan Y, Zeng JC and Cui ZH. (2010) Artificial physics optimization: a brief survey. *International Journal of Bio-inspired Computation*, 2(5),291-302.

Yang XS. (2010) Firefly algorithm, stochastic test functions and design optimization. *International Journal of Bio-inspired Computation*, 2(2),78-84.

Laalaoui Y and Drias H. (2010) ACO approach with learning for preemptive scheduling of real-time tasks.*International Journal of Bio-inspired Computation*, 2(6),383-394.

Abraham S, Sanyal S and Sanglikar M. (2010) Particle swarm optimisation based Diophantine equation solver,*International Journal of Bio-inspired Computation*, 2(2),100-114.

Yuan DL and Chen Q. (2010) Particle swarm optimisation algorithm with forgetting character. *International Journal of Bio-inspired Computation*, 2(1),59-64.

Lu JG, Zhang L, Yang H and Du J. (2010) Improved strategy of particle swarm optimisation algorithm for reactive power optimization. *International Journal of Bio-inspired Computation*, 2(1),27-33.

Upendar J, Singh GK and Gupta CP. (2010) A particle swarm optimisation based technique of harmonic elimination and voltage control in pulse-width modulated inverters. *International Journal of Bio-inspired Computation*, 2(1),18-26.

Cui ZH and Cai XJ. (2010) Using social cognitive optimization algorithm to solve nonlinear equations. *Proceedings of 9th IEEE International Conference on Cognitive Informatics (ICCI 2010)*, pp.199-203.

Chen YJ, Cui ZH and Zeng JH. (2010) Structural optimization of lennard-jones clusters by hybrid social cognitive optimization algorithm. *Proceedings of 9th IEEE International Conference on Cognitive Informatics (ICCI 2010)*, pp.204-208

Cui ZH, Shi ZZ and Zeng JC. (2010) Using social emotional optimization algorithm to direct orbits of chaotic systems, *Proceedings of 2010 International Conference on Computational Aspects of Social Networks (CASoN2010)*, pp.389-395.

Wei ZH, Cui ZH and Zeng JC (2010) Social cognitive optimization algorithm with reactive power optimization of power system, *Proceedings of 2010 International Conference on Computational Aspects of Social Networks (CASoN2010)*, pp.11-14.

Xu YC, Cui ZH and Zeng JC (2010) Social emotional optimization algorithm for nonlinear constrained optimization problems, *Proceedings of 1st International Conference on Swarm, Evolutionary and Memetic Computing (SEMCCO2010)*, pp.583-590.

Cai XJ, Cui ZH, Zeng JC and Tan Y. (2008) Particle swarm optimization with self-adjusting cognitive selection strategy, *International Journal of Innovative Computing, Information and Control*. 4(4): 943-952.

http://en.wikipedia.org/wiki/Normal_distribution

G.M.Viswanathan, S.V.Buldyrev, S.Havlin, M.G.daLuz, E.Raposo and H.E.Stanley. (1999) Optimizing the success of random searches.*Nature*.401(911-914) .

D.W.Simsand. (2008) Scaling laws of marine predator search behavior. *Nature*.451:1098–1102.

A.M.Reynolds and C.J.Rhodes. (2009) The levy flight paradigm: random search patterns and mechanisms. *Ecology*.90(4):877 – 887.

G.A.Parkerand and J.MaynardSmith. (1990) Optimality theory in evolutionary biology. *Nature*.348(1):27 – 33.

C.Y.Lee and X.Yao. (2004) Evolutionary programming using mutations based on the levy probability distribution.*IEEE Transactions on Evolutionary Computation*.8(1):1 – 13.

X.Cai, J.Zeng, Z.H.Cui and Y.Tan. (2007) Particle swarm optimization using Levy probability distribution. *Proceedings of the 2nd International Symposium on Intelligence Computation and Application*, 353 – 361, Wuhan, China.

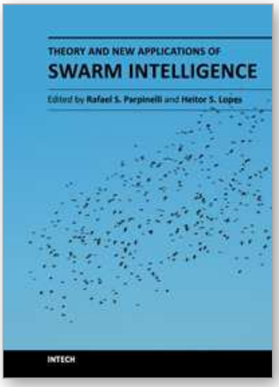
T.J.Richer and T.M.Blackwell. (2006) The levy particle swarm. *Proceedings of IEEE Congress on Evolutionary Computation*, 808 – 815.

M.R.Chen,Y.Z.Lu and G.Yang. (2006) Population-based extremal optimization with adaptive levy mutation for constrained optimization. *Proceedings of 2006 International Conference on Computational Intelligence and Security*, pp.258-261.November 3-6;Guangzhou,China.

R.N.Mantegna.(1994) Fast,accurate algorithm for numerical simulation of Levy stable stochastic processes. *Physical Review E*.49(5):4677 – 4683.

Ratnaweera A, Halgamuge SK and Watson HC. (2004) Self-organizing hierarchical particle swarm opitmizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240-255.

Yao X, Liu Y and Lin GM. (1999) Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2).82-102.



Theory and New Applications of Swarm Intelligence
Edited by Dr. Rafael Parpinelli

ISBN 978-953-51-0364-6
Hard cover, 194 pages
Publisher InTech
Published online 16, March, 2012
Published in print edition March, 2012

The field of research that studies the emergent collective intelligence of self-organized and decentralized simple agents is referred to as Swarm Intelligence. It is based on social behavior that can be observed in nature, such as flocks of birds, fish schools and bee hives, where a number of individuals with limited capabilities are able to come to intelligent solutions for complex problems. The computer science community have already learned about the importance of emergent behaviors for complex problem solving. Hence, this book presents some recent advances on Swarm Intelligence, specially on new swarm-based optimization methods and hybrid algorithms for several applications. The content of this book allows the reader to know more both theoretical and technical aspects and applications of Swarm Intelligence.