From the Social Learning Theory to a Social Learning Algorithm for Global Optimization

Yue-Jiao Gong and Jun Zhang*

Department of Computer Science
Sun Yat-sen University
Guangzhou, China
{gongyuejiao, junzhanghk}@gmail.com

Yun Li

School of Engineering, University of Glasgow
Glasgow G12 8LT, Scotland, UK
yun.li@glasgow.ac.uk

Abstract—**Traditionally, the Evolutionary Computation (EC) paradigm is inspired by Darwinian evolution or the swarm intelligence of animals. Bandura's Social Learning Theory pointed out that the social learning behavior of humans indicates a high level of intelligence in nature. We found that such intelligence of human society can be implemented by numerical computing and be utilized in computational algorithms for solving optimization problems. In this paper, we design a novel and generic optimization approach that mimics the social learning process of humans. Emulating the observational learning and reinforcement behaviors, a virtual society deployed in the algorithm seeks the strongest behavioral patterns with the best outcome. This corresponds to searching for the best solution in solving optimization problems. Experimental studies in this paper showed the appealing search behavior of this human intelligence-inspired approach, which can reach the global optimum even in ill conditions. The effectiveness and high efficiency of the proposed algorithm has further been verified by comparing to some representative EC algorithms and variants on a set of benchmarks.**

Keywords—Global optimization; evolutionary computation; swarm intelligence; social learning theory; observational learning

I. INTRODUCTION

In the past few decades, bio-inspired evolutionary computation (EC) has attracted significant attention. Evolutionary algorithms (EAs), which draw inspiration from natural evolution, are widely accepted as powerful global optimizers for solving various complex problems. The well-known branches of EAs include genetic algorithms (GAs), genetic programing (GP), evolutionary programming (EP), evolutionary strategy (ES), and differential evolution (DE). Meanwhile, by simulating the social behavior of animals, swarm intelligence (SI) algorithms such as ant colony optimization (ACO), particle swarm optimization (PSO), artificial bee colony (ABC), and bacterial foraging algorithm (BFA) have proved their powerfulness and become popular.

Different from the literature works, this paper develops a social learning algorithm (SLA) that mimics the social learning process of humans in the society. SLA is based on Bandura's *Social Learning Theory* [1]. It is a widely accepted theory in the fields of psychology and praxeology that human society makes progress via observational learning. The observational learning of society members involves four procedures: attention, retention, reproduction, and motivation. These procedures are embraced by the proposed SLA as its basic operators for solving optimization problems. This background makes the algorithm conceptual simple to follow. To the best of our knowledge, there is no reported work related to using the Social Learning Theory for developing global optimization algorithm. Our considerations of adopting such a method are as follows. First, compared to the natural evolution process that improves the fitness of individuals via the evolution of genotypes, the self-improvement through learning is more direct and rapid [2]. In this sense, the developed algorithm has potential to improve the efficiency of the algorithms mimicking natural evolution. Second, compared to the interaction and learning behaviors in animal groups, the social learning process of human beings exhibits a higher level of intelligence. By emulating human learning behaviors, it may be possible to arrive at more effective optimizers than existing swarm intelligence algorithms.

When come to the implementations of the proposed algorithm, it is to be noticed that the operation of SLA is based on numerical calculation, which is lightweight and consistent with the principle of Computational Intelligence (CI). In the experiments of this paper, the proposed algorithm is tested on 14 benchmark instances and compared to five different EC algorithms and variants including GA [3], PSO [4], DE [5], CLPSO [6], and SaDE [7]. Experimental results verify the powerfulness of SLA in terms of local exploitation, global exploration, and search efficiency. At last, as the algorithm developed in this paper is just a prototype (or bare bones), there is still room for the further development of this algorithm, which can attract attention of researchers in this domain.

The rest of this paper is organized as follows: Section II revisit the Social Learning Theory; Section III formulate the proposed SLA; then, Section IV shows the behavior analysis; experiments and comparisons are conducted in Section V; and conclusions are drawn in Section VI with future work highlighted.

II. REVISIT OF SOCIAL LEARNING THEORY

Social learning theory describes how people learn in a social context [1][8][9]. Adjustment of behaviors can take place either from direct experience or by observing other

people. Although learning by direct experience of rewarding and punishing consequences is effective, the cost could be high. For example, it is dangerous for a people to learn crossing a road, driving a car, or cooking a meal through trials-and-errors. Most of the behavioral patterns that people display are deliberately or inadvertently learnt by observing and imitating the actions of others. Hence, a good model can act as a better teacher than the consequences of unguided actions in most cases [1].

The observational learning starts with an attention process. In a social group, members with interesting qualities are likely to receive more attention than the others. At the same time, informative function determines which characteristics of the models capture attention and which will be ignored. Then, people remember the details of their exemplary behavior with a retention process and practice to reproduce the behavior with a reproduction process. However, even if the attention-retention-reproduction process is finished, the person will not engage in the behavior without motivation. In the motivation process, the learnt behavior, which previously remained unexpressed, will take action when incentives are provided.

Reinforcement plays an important role in observational learning for it distinguishes learning from simply imitating the others. In social learning theory, behavior is regulated by external reinforcement, vicarious reinforcement, and self-reinforcement, among which vicarious reinforcement has a crucial role [1]. Vicarious reinforcement is defined as the adaptation in the behavior of observers when they notice the response consequences of the models. Generally, it includes vicarious positive reinforcement (that observers display an increase in the behavior when they see models get positive consequences), and vicarious punishment (that negative consequences prevent observers from behaving similarly like the models). By the effect of vicarious reinforcement, observers may perform even better than the models.

Moreover, observational learning is the primary source of innovation in a social group owing to the following reason. Observers will neither concentrate on a single model, nor absorb all characteristics of the preferred model, but they abstract common features of diverse models to form a behavior rule or combine different attributes of the models to develop distinct personalities. The more diverse the models are, the more likely the observers exhibit creative, innovative behavioral patterns. For example, an artist creates unique art via combining, discarding, and recombining the characteristics of other works. On the contrary, for an isolated small social group, the possible combinations of behavioral patterns are limited. After a long period of observational learning, the members in the group have a tendency to behave in similar ways and have trouble in producing innovation.

III. THE PROPOSED SOCIAL LEARNING ALGORITHM

In any social group, people adapt to the environment or seek acceptance from the society via observing models with influence. The effects of the observational learning include not only imitation, but also reinforcement and innovation. As a result, the entire social group makes progress in overcoming challenges or pursuing goals with time passing by. In this
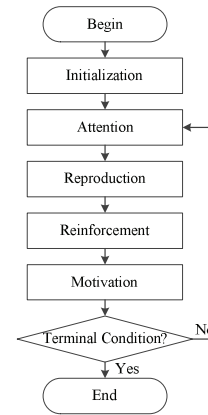

Fig. 1   Flowchart of the Social Learning Algorithm (SLA).

sense, the social learning process of humans indicates a high-order form of intelligence in nature. Based on this observation, we develop a novel optimization technique, termed the Social Learning Algorithm (SLA), which emulates the social intelligence of humans in computers.

A. Methodology

Similar to the other population-based optimization techniques, SLA maintains a population of individuals, specifically, a social group of people. Each member i in the group is assigned with a behavioral pattern vector $x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,D}]$ and a consequence score S_i denoting the response from the environment. Note that x_i is multi-dimensional, which consists of the member's behavioral patterns in different aspects. For example, when cooking steak, the behavioral patterns involve the use of frying pan, the cut of steak, salt timing, and frying time, etc., while the consequence score is the taste of the steak.

For a given optimization problem, x_i corresponds to a candidate solution in the problem space, i.e., a D-dimensional numerical vector where D stands for the number of variables. The score of the behavior vector is evaluated as $S_i = f(x_i)$, where f denotes the objective of the problem. In the initialization, all the behavior vectors are randomly generated in the problem space with the consequence scores evaluated by f. Then, the members perform observational learning in the society to improve their behavioral pattern vectors in order to receive higher and higher scores. In this way, better and better solutions are found, until the global optimum is achieved.

The optimization process of SLA is based on a simplified social learning model. As illustrated in Fig. 1, after the initialization, SLA performs an iteration process in which the members conduct 'attention', 'reproduction', 'reinforcement', and 'motivation' operators repeatedly. The four operators are similar to those in the process of observational learning in social learning theory. Attention captures model members and their attractive attributes according to the scores; reproduction builds new behavior vectors for all members by imitation; reinforcement further improves the learnt behavior with positive reward or negative punishment; and motivation activates the new behavior vectors with incentives. Here the
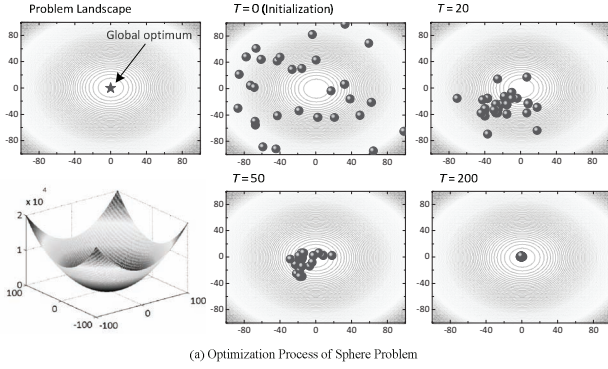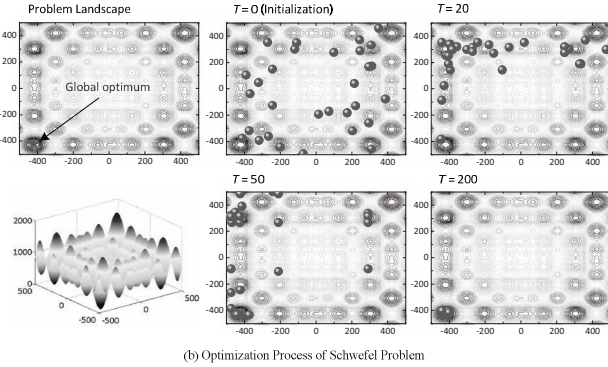
Fig. 2　Search behavior of SLA in unimodal and multimodal landscapes.

Fig. 3　Search behavior of SLA in unimodal and multimodal landscapes (ill conditioned).

retention process is omitted because it is easy for computers to store ('remember') the models and exemplary attributes. In the following, these four basic operators of SLA are defined in detail.

B. Attention Operator

The operator determines whose and which characteristics capture attention in the social group. In SLA, for simplicity, half of the members in the group that currently with the highest scores are selected as models. Therefore, the entire society is divided equally into two segments, *US* (upper society) and *LS* (lower society), such that members in *US* have better consequence scores than members in *LS*. Then, for each dimension of the behavioral pattern vectors, significance test is used to determine whether the values of model members on this dimension are significantly different from that of non-model members. If the difference is significant, the models' values on the dimension capture attention, which are named 'attractive attributes'. Otherwise, they are ignored. More specifically, we mark the dimension with symbol '>>' if the attractive attributes are significantly larger than the others, or with '<<' if the attributes are significantly smaller than the others. In other cases, the dimensions are marked with '≈'.

In a formal definition, for each dimension $d = 1, 2, …, D$ of the society, conduct students' *t*-test to compare the values of *US* and *LS* on this dimension and record the calculated statistical *t*-value as $t(d)$. (Noted that other significance test methods such as some rank-based nonparametric tests could also be used, *t*-test is considered as an example here.) Define an attention threshold *AT* as the absolute value of the *t*-value
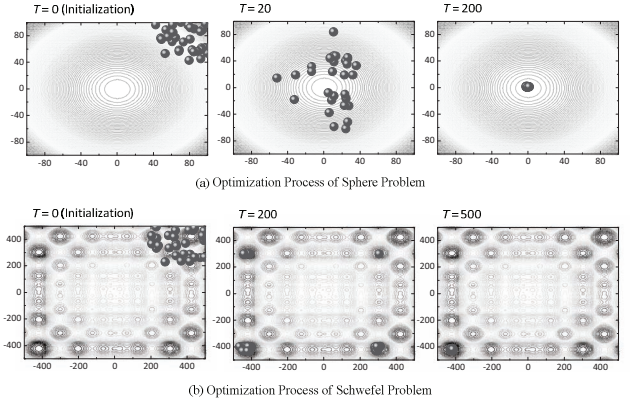
on a random dimension, i.e., $AT = |\ t\ (r)\ |$, where r is a randomly selected dimension index. Then, mark each dimension of the society with an attention symbol $\Gamma(d)$ as

$$\Gamma(d) = \begin{cases} '\gg', & \text{if} & t(d) \geq AT \\ '\approx', & \text{if} & -AT < t(d) < AT \\ '\ll', & \text{if} & t(d) \leq -AT \end{cases} \quad (1)$$

C. Reproduction Operator

After attention, the reproduction process is conducted, by which each member i (in the entire society) generates a new behavior vector $\boldsymbol{x_i}' = [x_{i,1}', x_{i,2}', …, x_{i,D}']$. For each dimension d that the models capture attention, $\boldsymbol{x_i}'$ imitates (copies) the attractive attribute of a random model, i.e., if $\Gamma(d) =$ '>>' or '<<' then we have $x_{i,d}' = x_{ru_i,d}$, where ru_i is a random index of members in *US*. As discussed in the social learning theory, combining different attributes of several models can result in innovative behavioral patterns. For the other dimensions marked with '≈', the attributes of models are not so attractive. In these cases, $\boldsymbol{x_i}'$ maintains the values of $\boldsymbol{x_i}$ or explores the entire society according to

$$x_{i,d}' = \begin{cases} x_{re_i,d}, & \text{if } r_1 < PI \\ rand(l_d, u_d), & \text{otherwise if } r_2 < PR \\ x_{i,d}, & \text{otherwise} \end{cases} \quad (2)$$

Here re_i is the index of a member randomly selected from the entire society; l_d and u_d are the lower and upper bounds of the variable on dimension d; *PI* stands for the probability of imitation; *PR* stands for the probability of randomization; r_1 and r_2 are random numbers distributed in [0, 1].

In this way, reproduction exploits good models' strength and explores the dimensions without models.

D. Reinforcement Operator

Members in the society not only imitate the attractive attributes, but also make reinforcement on them. This includes positive reinforcement that observers further increase the learnt value and negative punishment by which the values are

decreased. In SLA, positive reinforcement and negative punishment are performed on dimensions marked with '>>' and '<<' respectively as

$$x_{i,d}' = \begin{cases} x_{i,d}'(\text{i.e., } x_{ru_i,d}) + \Delta_{i,d}, & \text{if } \Gamma(d) = '\gg' \\ x_{i,d}'(\text{i.e., } x_{ru_i,d}) - \Delta_{i,d}, & \text{if } \Gamma(d) = '\ll' \end{cases} \quad (3)$$

The step length of reinforcement $\Delta_{i,d}$ can be defined in various ways. Here we use a random differential reinforcement $\Delta_{i,d} = rand(0,1) \times | (x_{ru_i,d} - x_{i,d}) |$, in which the step length ranges from 0 to the differential value between the model and the observer.

By the vicarious reinforcement, there is a big chance for observers to overtake their models. This mechanism brings a competition effect among the social members since the previous models that are overtaken by the others would strive to win back their leading positions. The ultimate effect we obtain here is the progress (or evolution) of the entire society.

E. Motivation Operator

In social learning system, the motivation process determines whether the learnt behavior will be expressed by people. In SLA, after reproduction and reinforcement, the leant behavior vector x_i' of each member is evaluated. Only when x_i' obtains a higher score than the current behavior x_i, the member updates his behavior vector, i.e., replace x_i with x_i'. This step is similar to the selection operation in traditional evolutionary algorithms.

$$x_i = \begin{cases} x_i', & \text{if } f(x_i') \text{ betterthan } f(x_i) \\ x_i, & \text{otherwise} \end{cases} \quad (4)$$

SLA performs the above four operators iteratively until the terminal condition is satisfied. At the end of optimization, the best behavior vector with the highest score is output as the result of the problem. The terminal condition could be a maximum number of iterations or a predefined acceptable value of the result. For public use, the source code of SLA is available to download on the Internet. (http://www.ai.sysu.edu.cn/GYJ/SLA/c_code/).

IV. BEHAVIORAL ANALYSIS

The behavioral analysis of SLA has two effects. On one hand, it shows how SLA obtains the global optimal solution in solving an optimization problem without prior information. On the other, as SLA itself can be considered as a simulation of the natural social learning system, the optimization process of SLA in some ways confirms or reveals the effect of humans' observational learning in a social group.

We use two typical problems, a unimodal Sphere problem and a multimodal Schwefel problem [10], to investigate the search behavior of SLA. The landscapes of the two problems are depicted in Fig. 2. It should be emphasized that SLA is in the absence of any prior information of the problem landscapes to be optimized, neither does it require the problem to satisfy some mathematical properties such as continuity and differentiability. In the initialization, all members' behavior vectors are distributed uniformly in the problem space. Afterwards, SLA conducts its problem-independent operation (except the evaluation of consequence scores) to search for the optimum.

As shown in Fig. 2(a), in searching the Sphere problem space containing a single minimum, within the iterations of SLA, the behavior vectors gradually descend to the minimum. This demonstrates the local exploitation ability of SLA. Further, the global exploration ability of SLA can be verified by testing on the Schwefel problem. As shown in Fig. 2(b), the landscape of the Schwefel problem involves a number of local optima, among which some deep local optima look similar to but locate far from the global optimum (which we call the deceptive valleys). Traditionally, it is easy for an optimization algorithm to get trapped into these deceptive valleys. Considering SLA, it can be observed in Fig. 2(b) that, in the early stage of the optimization, the algorithm conducts full-on exploration in the problem space. Although the members are distributed around some deceptive optima at the 20th and 50th iteration, they can finally converge to the global optimum in 200 iterations. Afterwards, SLA performs local exploitation as solving unimodal problems.

In addition, to investigate the search behavior of SLA in ill conditions, we use worst-case initialization instead of the uniform initialization. As shown in Fig. 3, members are initialized in a small domain far away from the global optimum. Fig. 3(a) reveals an interesting phenomenon: even

TABLE I.　BENCHMARK FUNCTIONS

Function	Domain	ε
$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0.01
$f_2(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|$	$[-10, 10]^D$	0.01
$f_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^D$	100
$f_4(x) = \max_i(|x_i|, 1 \le i \le D)$	$[-100, 100]^D$	10
$f_5(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100, 100]^D$	0.01
$f_6(x) = \sum_{i=1}^{D} ix^4 + random[0,1)$	$[-1.28, 1.28]^D$	0.1
$f_7(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$	100
$f_8(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{x_i})$	$[-500, 500]^D$	2569.5
$f_9(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	20
$f_{10}(x) = -20\exp(-0.2\sqrt{1/D\sum_{i=1}^{D} x_i^2}) - \exp(1/D\sum_{i=1}^{D} \cos 2\pi x_i) + 20 + e$	$[-32, 32]^D$	0.01
$f_{11}(x) = 1/4000\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(x_i/\sqrt{i}) + 1$	$[-600, 600]^D$	0.01
$f_{12}(x) = \frac{\pi}{D}\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, a, k, m)$	$[-50, 50]^D$	0.01
$f_{13}(x) = \frac{1}{10}\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, a, k, m)$	$[-50, 50]^D$	0.01
$f_{14}(x) = \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i) + 10]$	$[-5.12, 5.12]^D$	20

TABLE II.   COMPARISONS OF THE SOLUTION ACCURACY OBTAINED BY THE SIX ALGORITHMS

f_1 - f_7		f_1	f_2	f_3	f_4	f_5	f_6	f_7
SLA	Mean	9.88E-110	**5.91E-82**	1.97E-02	3.56E-03	**0**	5.05E-03	1.05E-01
	Std	2.85E-109	**9.61E-82**	3.17E-02	2.82E-03	**0**	1.52E-03	1.86E-01
GA	Mean	5.23E-01	2.77E-01	7.32E+02	2.46E+00	6.67E-02	1.84E-02	1.41E+02
	Std	2.22E-01	4.61E-01	3.30E+02	9.73E-01	2.54E-01	5.35E-03	6.98E+01
PSO	Mean	3.82E-61	7.46E-40	1.43E-02	1.63E-01	**0**	5.78E-03	2.24E+01
	Std	7.88E-61	2.71E-39	1.73E-02	1.08E-01	**0**	1.65E-03	2.14E+01
DE/rand/1/bin	Mean	3.49E-36	9.38E-18	1.57E-05	3.62E-01	**0**	4.49E-03	**1.37E-02**
	Std	5.92E-36	8.41E-18	1.30E-05	9.86E-01	**0**	1.54E-03	**2.83E-02**
CLPSO	Mean	1.73E-29	9.80E-18	1.06E+02	2.17E+00	**0**	3.70E-03	6.94E-01
	Std	1.02E-29	3.29E-18	4.06E+01	3.08E-01	**0**	9.00E-04	9.49E-01
SaDE	Mean	**1.36E-146**	6.23E-82	**1.51E-22**	**1.65E-21**	**0**	**1.91E-03**	7.97E-01
	Std	**5.97E-146**	1.05E-81	**3.07E-22**	**8.05E-21**	**0**	**7.12E-04**	1.63E+00
f_8 - f_{14}		f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
SLA	Mean	**3.82E-04**	**0**	**4.14E-15**	2.79E-03	2.16E-32	1.86E-31	**0**
	Std	**0**	**0**	**0**	6.89E-03	1.60E-32	2.20E-31	**0**
GA	Mean	1.52E+00	2.88E-01	2.27E-01	4.87E-01	2.40E-03	2.96E-02	**0**
	Std	6.24E-01	1.15E-01	5.88E-02	1.35E-01	3.01E-03	1.25E-02	**0**
PSO	Mean	2.30E+03	2.42E+01	9.83E-15	2.10E-02	6.91E-03	7.32E-04	1.37E+01
	Std	3.95E+02	6.71E+00	1.77E-15	2.04E-02	2.63E-02	2.79E-03	6.55E+00
DE/rand/1/bin	Mean	4.53E+03	1.39E+02	**4.14E-15**	**0**	**1.57E-32**	**1.35E-32**	1.06E+02
	Std	9.31E+02	1.88E+01	**0**	**0**	**2.78E-48**	**5.57E-48**	2.36E+01
CLPSO	Mean	2.37E+01	3.65E-08	1.42E-14	2.33E-12	1.72E-30	2.14E-29	**0**
	Std	4.82E+01	1.99E-07	1.89E-15	4.45E-12	1.73E-30	2.40E-29	**0**
SaDE	Mean	**3.82E-04**	**0**	1.28E-01	1.68E-03	4.15E-03	4.39E-04	**0**
	Std	**0**	**0**	3.61E-01	4.25E-03	2.07E-02	2.20E-03	**0**

though the initial society is short of diversity, after some iterations of social learning, the diversity of the society can be increased. Then, by the motivation operation, inferior patterns are discarded and the entire group converges to the minimum. In Fig. 3(b), although the entire society is generated near the deceptive local optimum farthest from the global optimum by the worst-case initialization, it is not trapped in the local optimum. At the 200th iteration, after an exploration process, members successfully find four promising valleys in the landscape and conduct local exploitation on them to determine which one is the best. Finally, at the 500th iteration, members successfully converge to the global optimum. The above behavioral analysis shows the powerfulness of SLA in optimization, which possesses both strong global exploration and local exploitation abilities even with an ill-conditioned initial environment.

From the perspective of psychology, SLA emulates the social learning system so that it in some ways reveals the outcomes of observational learning in a social society. By observing and learning through influential models, members in the society do not simply copy the actions of others, but they make further improvements on the learnt behavioral patterns, resulting in producing new actions better than any existing ones. In this way, the society makes innovation and progress constantly, whose intelligence can be greater than the simple collection of all personal intelligence in the group. By utilizing such social intelligence, people can defeat adverse circumstances and achieve unprecedented success in the history.

V.   EXPERIMENTAL STUDIES

A. Experimental Setup

In the experiments, we use 14 continuous benchmark functions with different characteristics to test the performance of SLA [10]-[12]. As shown in Table I, f_1 to f_4 are unimodal functions, f_5 is a step function, f_6 is a noisy quartic function, and f_7 to f_{14} are multimodal functions. The dimensionality of these tested functions is set to 30.

The proposed algorithm is compared with classical EC algorithms for continuous optimization, including GA, PSO, DE/rand/1/bin, and two representative variants of PSO and DE: CLPSO [6] and SaDE [7]. In SLA, PI and PR are set to 0.7 and 0.2 respectively. The parameters of the three classical EC algorithms are set to the values that are commonly used in the literature. In GA, the probability of crossover and mutation are set as $PC = 0.7$ and $PM = 0.07$. In PSO, the inertia weight ω linearly decreases from 0.9 to 0.4 while the accelerating coefficients are set as $c_1 = c_2 = 2.0$. In DE/rand/1/bin, the scalar factor F and crossover rate CR are set to 0.5 and 0.9 respectively. Besides, the parameter settings of CLPSO and SaDE are based on that suggested in [6] and [7]. For a fair comparison, all algorithms are allowed to conduct 300,000 function evaluations. Each algorithm runs 30 times independently with the statistical result recorded.

Func	GA	PSO	DE	CLPSO	SADE	SLA
f_1	N/A	0.815	0.168	0.235	0.177	**0.128**
f_2	N/A	0.803	0.236	0.258	0.183	**0.108**
f_3	N/A	1.563	**0.454**	1.818	0.533	1.214
f_4	0.120	0.704	0.116	0.613	**0.045**	0.291
f_5	0.947	1.040	0.183	0.221	0.151	**0.138**
f_6	0.096	0.527	0.076	0.122	**0.021**	0.133
f_7	0.681	0.943	**0.138**	0.314	0.144	0.244
f_8	0.027	2.219	1.897	0.135	0.159	**0.026**
f_9	**0.110**	1.289	1.550	0.529	0.228	0.233
f_{10}	N/A	1.149	0.268	0.399	1.278	**0.173**
f_{11}	N/A	1.230	0.286	0.482	0.611	**0.184**
f_{12}	1.381	1.791	0.446	0.536	0.677	**0.351**
f_{13}	2.688	2.037	**0.495**	0.633	0.694	0.540
f_{14}	**0.039**	2.057	2.072	0.364	0.191	0.214

VI. CONCLUSIONS

Inspired by the human social learning process, we have developed a novel EC algorithm: the SLA. SLA absorbs a high form of intelligence in nature, the social intelligence of humans, to seek the global best solution. Experimental results have demonstrated the effectiveness and efficiency of this new algorithm, which has in turn also verified through computer simulations the outcomes of the social learning behavior in human society.

As a new optimization algorithm, SLA is worth further study. First, there exists room for performance enhancement. The adjustment or adaptation of parameters, fine-tuning of operators, hybridization with other techniques, and parallel and distributed implementations, etc., are all possible ways to further improve the performance of the SLA. Second, the promising results encourage a wide range of real-world applications of the algorithm. Although SLA is generic and a-posteriori, embedding with prior knowledge can help improve its efficiency in solving domain-specific problems. The methodological and performance characteristics of SLA are likely to set a new trend in nature-inspired computing.

B. Results and Comparisons

Table II reports the numerical results achieved by the five algorithms. It can be observed that, for solving unimodal functions $f_1 - f_6$, SLA exhibits promising performance. The proposed algorithm obtained higher solution accuracy than GA, PSO, DE/rand/1/bin, and CLPSO on most functions, which shows its fast convergence and good local exploitation ability. However, its performance on unimodal functions is not as good as that of SaDE. Generally speaking, SaDE performs the best on $f_1 - f_6$.

Considering the multimodal functions, SLA achieves the best results on f_8, f_9, f_{10}, and f_{14}. Moreover, it has very promising performance in optimizing f_{12} and f_{13}. These results verify that the proposed algorithm possesses good global exploration ability, which is not easy to get trapped in local optimum. In these cases, SaDE performs worse than SLA. Besides, although DE/rand/1/bin obtains the best solution on $f_{10} - f_{13}$, it is vulnerable to premature convergence in optimizing the Schwefel problem f_8, Rastrigin problem f_9, and Noncontinuous Rastrigin problem f_{14}. Generally speaking, the proposed SLA algorithm exhibits the best global search ability among the six algorithms when solving multimodal problems.

In addition, the search efficiency of these six algorithms is compared in Table III, where their average computing time (in seconds) used to obtain the predefined error thresholds is tabulated. Note that all algorithms are coded in C, run on a machine using Intel Xeon CPU E5405, 2.00GHz/2GB of RAM. It can be observed that SLA costs the least amount of computing time in optimizing 7 out of the 14 benchmark functions. Meanwhile, for the other 7 functions, SLA is still among the fastest solvers. In this way, we can conclude that our proposed algorithm is not only very effective but also efficient for solving optimization problems.

REFERENCES

[1] A. Bandura, *Social Learning Theory*, Oxford, U.K, 1977.

[2] M. Fahle and S. Edelman, "Long-term learning in vernier acuity: Effects of stimulus orientation, range and of feedback," *Vision Res.*, vol. 33, no. 3, pp. 397-412, 1993.

[3] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *Comput.*, vol. 27, no. 6, pp. 17-26, 1994.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. 1995 IEEE Int. Conf. Neural Netw.*, vol. 4, pp. 1942-1948, 1995.

[5] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341-359, 1997.

[6] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281-295, 2006.

[7] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398-417, 2009.

[8] A. Nicolle, M. Symmonds, and R. J. Dolan, "Optimistic biases in observational learning of value," *Cognition.*, vol. 119, no. 3, pp. 394-402, 2011.

[9] J. R. Brauer and C. R. Tittle, "Social learning theory and human reinforcement," *Sociological Spectr.*, vol. 32, pp. 157-177, 2012.

[10] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82-102, 1999.

[11] Y.-J. Gong, J. Zhang, H.S.-H. Chung, *et al*, "An efficient resource allocation scheme using particle swarm optimization," *IEEE Trans. Evolut. Comput.*, vol. 16, no. 6, pp. 801-816, 2012.

[12] T. Blackwell, "A study of collapse in bare bones particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 354-372, 2012.