

# A New Meta-Heuristic Technique for Engineering Design Optimization: Seven-Spot Ladybird Algorithm

Peng Wang, Zhouquan Zhu, Shuai Huang

School of Marine Science and Technology  
Northwestern Polytechnical University

Xi'an, China

wangpeng305@nwpu.edu.cn

**Abstract**—Increasing attention is being paid to solving engineering design optimization problems which are generally nonlinear and constrained. In this paper, the Seven-spot Ladybird Optimization (SLO) algorithm is developed to obtain the solution of such optimization problems. This paper presents the basic concepts and main steps of the SLO and demonstrates its efficiency. The performance of the SLO is compared with some popular meta-heuristic algorithms by using four different dimensional classical benchmark functions. The simulation results indicate that the proposed algorithm is able to escape from a local minimum and is efficient for multimodal function optimization. Moreover, three constrained engineering problems are solved by using SLO and numerical results show the efficiency of this proposed method in practical computations.

**Keywords**—seven-spot ladybird optimization; meta-heuristic algorithm; benchmark function; constrained mechanical design optimization

## I. INTRODUCTION

In general, engineering optimization problems are divided into unconstrained optimization problems and constrained optimization problems. In an unconstrained optimization problem with  $D$  dimensions, we need to find the optimal parameter vector  $X$  with the purpose of minimizing the objective function  $f(X)$ . The optimization problem is complicated because the objective function is non-linear or there exists numerous local optima [1].

Unlike unconstrained optimization problems, tasks with constraints are usually found in practical applications such as shape optimization, structural design, location and allocation problems [2]. Typically, these problems are difficult to solve. Thus, it is of great significance to solve constrained optimization problems effectively and efficiently. An  $n$ -dimensional constrained optimization problem with  $m$  constraints can be described as follows:

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && g_j(x) \leq 0, j = 1, 2, \dots, q \\ & && h_j(x) = 0, j = q+1, q+2, \dots, m \\ & && l_i \leq x_i \leq u_i, i = 1, 2, \dots, n \end{aligned} \quad (1)$$

Due to the complexity and unpredictability of large-scale optimization problems, especially for constrained

optimization problems, a general deterministic solution is hard to find within a reasonable amount of computation time. In recent years, several evolutionary algorithms which are capable of finding the high-precision near-optimal solutions to the large-scale optimization problems have been proposed. Some of the optimization algorithms that are in wide use these days are Genetic Algorithm (GA) [3], Simulated Annealing [4], Tabu Search [5], Covariance Matrix Adaptation Evolution Strategies [6], Teaching–Learning–Based Optimization [7], Immune Algorithm [8], Differential Evolution [9], Particle Swarm Optimization (PSO) [10], Ant-Colony Optimization [11], Artificial Bee Colony (ABC) algorithm [12] and some hybrid algorithms [13]. According to the famous No-Free-Lunch theorem for optimization [14], all non-repeating search algorithms have the same mean performance when averaged uniformly over all possible objective functions. But it does not exclude the possibility that some certain algorithms will obtain better results for some certain objective functions. So the requirement to develop new optimization algorithms increasingly continues.

This paper introduces a novel biologically inspired meta-heuristic algorithm called seven-spot ladybird optimization (SLO). SLO is inspired by the foraging behavior of a seven-spot ladybird. In this paper, the performance of the SLO is compared with that of GA, PSO, and ABC by using four different dimensional classical benchmark functions and three constrained engineering problems are solved by using SLO. The results show that SLO has the ability to find the best solution with a comparatively small population size and is suitable for solving optimization problems with lower dimensions. The remainder of this paper is organized as follows: Section II presents the foraging behavior of the seven-spot ladybird; Section III describes the SLO and the steps in detail; Section IV discusses the experiments and the results; and Section V draws the conclusions.

## II. SEVEN-SPOT LADYBIRD IN THE NATURE

The seven-spot ladybird, also known as *Coccinella septempunctata*, has attracted the interest of a growing number of professional entomologists because of its ecological effectiveness and social behaviors. Recent studies have shown that seven-spot ladybirds are more social than we believe them to be [15]. Like most insects, pheromone is the paramount way they communicate and share information with others. Some chemical ecologies of the seven-spot

ladybirds, with special attention to semiochemicals involved in social communication and foraging behaviors, have been reviewed in [16].

The latest researches have shown that the environmental levels of seven-spot ladybirds can be classified into prey, patches, and habitats (Fig. 1) [17]. This provides a framework for discussing the foraging behaviors of seven-spot ladybirds. In Fig. 1, movement between prey within aggregates of aphids is referred to as intensive search which is slow and sinuous. Movement between aggregates within a patch is referred to as extensive search which is relatively linear and fast. Movement between patches is called dispersal and movement from patches to hibernation is called migration.

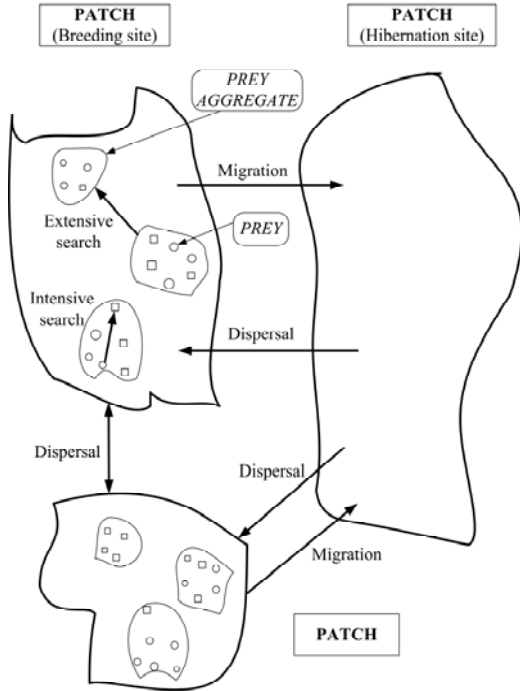


Figure 1. Diagram illustrating how a ladybird might perceive its environment and forage for resources

Seven-spot ladybirds locate their prey via extensive search and then switch to intensive search after feeding. While searching for its prey, a seven-spot ladybird holds its antennae parallel to its searching substratum and its maxillary palpi perpendicular to the substratum. The ladybird vibrates its maxillary palpi and turns its head from side to side. The sideward vibration can increase the area wherein the prey may be located.

How seven-spot ladybirds decide when to leave a patch for another, also known as dispersal, remains unclear. Several authors suggested that beetles decide to leave when the capture rate falls below a critical value or when the time since the last aphid was captured exceeds a certain threshold [18].

### III. SEVEN-SPOT LADYBIRD OPTIMIZATION ALGORITHM

This section describes the proposed seven-spot ladybird optimization (SLO) algorithm. The main steps are given below:

#### Step 1: Dividing Patches

Assume that the search space (environment) is a  $D$ -dimensional space. The  $i$ -th dimensional space is divided into  $n_i$  subspaces, and the whole dimensional space is divided into  $n = \prod n_i$  subspaces (patches).

#### Step 2: Initializing Population

Suppose that each seven-spot ladybird is regarded as a point in a  $D$ -dimensional patch. The  $i$ -th ladybird is represented as  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , where  $X_i$  is a search vector to the optimized problem.

If  $m$  is the number of seven-spot ladybirds initialized with random positions in a patch, then the population size of the seven-spot ladybirds is  $N, N = m \times n$ .

#### Step 3: Calculating fitness

For each particle, evaluate the optimization fitness in a  $D$ -dimensional patch.

#### Step 4: Choosing the best ladybird

Compare the current position of each ladybird with ladybird's previous best position, which is called *sbest* (self best). If current value is better, then *sbest* = current value.

Compare the current best position of all the ladybirds in a patch with previous best position in a patch, which is called *lbest* (local best). If current value is better, then *lbest* = current value.

Compare the current best position of all the ladybirds in the population with previous best position in the population, which is called *gbest* (global best). If current value is better, then *gbest* = current value.

#### Step 5: Dispersal

In the SLO, if a position does not improve in a predetermined number of cycles, then a new position is produced in the patch where *gbest* exists, replacing the abandoned position. The new position is produced near the *gbest* to share the information of the best ladybird in the whole particle. The value of the predetermined number of cycles (*limit*) is an important control parameter in the SLO.

If the abandoned position is  $X_i$  and  $j \in (1, 2, \dots, D)$ , then the seven-spot ladybird discovers a new position  $X_i'$  as follows:

$$x_{i,j}' = x_{gbest,j} + \phi w \quad (2)$$

where  $w$  is the neighborhood space of *gbest* and  $\phi$  is a random number between  $[-1, 1]$ .

#### Step 6: Updating positions

The position of a ladybird is updated associated with its previous movement. If a ladybird has done extensive search, then the position of the ladybird is changed as follows:

$$V_i(t) = c * r_1 * (S_i(t) - X_i(t)) + \varepsilon_1 \quad (3)$$

$$X_i(t+1) = X_i(t) + V_i(t), |V_i(t)| \leq V_{\max} \quad (4)$$

After intensive search, a ladybird switches to extensive search. The position is updated according to the following equations:

$$V_i(t) = c * r_2 * (L_i(t) - X_i(t)) + \varepsilon_2 \quad (5)$$

$$X_i(t+1) = X_i(t) + V_i(t), |V_i(t)| \leq V_{\max} \quad (6)$$

In (3) and (5),  $r_1$  and  $r_2$  are two random numbers uniformly distributed from 0 to 1 and the positive constant  $c$  is used for adjusting the search step and search direction in each iteration. In (4) and (6), the velocities of the ladybirds in each dimension are limited to the maximum velocity  $V_{max}$ , which decides the search precision of the ladybirds in a solution space. If  $V_{max}$  is too high, then the ladybirds will possibly fly over the optimal solution. However, if the  $V_{max}$  is too low, then the ladybirds will fall into the local search space and have no method to carry on with the global search. Typically,  $V_{max}$  is set as follows:

$$V_{max} = 0.2(ub - lb) \quad (7)$$

where  $ub$  and  $lb$  are the upper and lower bounds of each patch, respectively.

From equations above, we can see that the velocity updating rule is composed of three parts. The first part, known as **intensive search**, is inspired by the slow and sinuous movements of ladybirds. The second part, known as **extensive search**, is derived from the relatively linear and fast movement behavior of ladybirds. The third part imitates the **sideward vibration** of ladybirds to increase the search area where the potential solution may exist. The parameter  $\varepsilon_1$  and  $\varepsilon_2$  are usually set as relatively small random numbers.

#### Step 7: Inspecting termination condition

If the termination condition is satisfied, i.e., the SLO has achieved the maximum iteration number, then the SLO is terminated; otherwise, it returns to Step 3.

### IV. EXPERIMENTS AND RESULTS

#### A. Unconstrained numerical optimization problems

In order to verify the efficiency and analyze the performance of Seven-spot Ladybird Optimization algorithm, four typical benchmark test functions are given as following:

##### Griewank Function

$$f_1(X) = \frac{1}{4000} \left( \sum_{i=1}^D (x_i^2) \right) - \left( \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) \right) + 1, \quad (8)$$

$$x_i \in [-600, 600]$$

The value of Griewank function is 0 with global optimum (0, 0, ..., 0).

##### Rastrigin Function

$$f_2(X) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10), x_i \in [-15, 15] \quad (9)$$

The value of Rastrigin function is also 0 with global optimum (0, 0, ..., 0).

##### Rosenbrock Function

$$f_3(X) = \sum_{i=1}^D 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2, x_i \in [-15, 15] \quad (10)$$

The value of Rosenbrock function is 0 with global optimum (1, 1, ..., 1).

##### Ackley Function

$$f_4(X) = 20 + e - 20e^{-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}} - e^{\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)}, \quad (11)$$

$$x_i \in [-32.786, 32.786]$$

The value of Ackley function is 0 with global optimum (0, 0, ..., 0).

Common control parameters of the algorithms are population size and number of maximum generation. In the experiments, the population size was 50, and maximum generations were 750, 1000 and 1500 for Dimensions 5, 10 and 30, respectively. Other parameters employed for GA, PSO and ABC are consistent with [19, 20]. In SLO, every dimension is divided into 2 equal parts, so there are totally  $2^D$  patches. In each patch, the initial population of ladybirds is set to 20. The parameter *limit* is 100 and  $w$  is 1. The parameter  $c$  in (3) and (5) decreases linearly from 10 to 2. The sideward vibration  $\varepsilon_1$  is  $\text{Rand} * 10e-4$  and  $\varepsilon_2$  is  $\text{Rand} * 10e-8$ .

Each of the experiments was repeated 30 times with different random seeds, and the mean, best and standard values produced by different algorithms are presented in Table I-IV.

TABLE I. RESULTS OF THE GRIEWANK FUNCTION

Algorithm	Dimension	Mean	Best	SD
SLO	5	1.29E-01	7.40E-03	9.12E-02
	10	3.10E-01	2.46E-02	5.29E-01
	30	1.47E+00	6.20E-03	1.79E+00
GA	5	1.22E+01	1.22E+01	1.23E-10
	10	1.46E+01	6.56E+00	4.71E+00
	30	1.41E-02	1.91E-10	2.47E-02
PSO	5	2.37E-02	7.40E-03	1.24E-02
	10	7.69E-02	2.70E-02	3.36E-02
	30	1.32E-02	0.00E+00	1.49E-02
ABC	5	6.88E-04	0.00E+00	2.10E-03
	10	2.80E-03	0.00E+00	4.60E-03
	30	6.35E-04	0.00E+00	3.50E-03

TABLE II. RESULTS OF THE RASTRIGIN FUNCTION

Algorithm	Dimension	Mean	Best	SD
SLO	5	0.00E+00	0.00E+00	0.00E+00
	10	2.28E+01	9.98E-01	1.47E+01
	30	3.61E+02	2.31E+02	5.90E+01
GA	5	6.30E-01	2.07E-09	6.12E-01
	10	7.96E-01	3.33E-08	8.43E-01
	30	3.08E+00	2.51E-06	2.37E+00
PSO	5	6.63E-02	0.00E+00	2.52E-01
	10	1.73E+00	0.00E+00	1.17E+00
	30	2.90E+01	1.79E+01	8.56E+00
ABC	5	0.00E+00	0.00E+00	0.00E+00
	10	0.00E+00	0.00E+00	0.00E+00
	30	0.00E+00	0.00E+00	0.00E+00

TABLE III. RESULTS OF THE ROSENBRICK FUNCTION

Algorithm	Dimension	Mean	Best	SD
SLO	5	1.33E+00	1.62E-08	8.78E-01
	10	1.69E+01	6.70E+00	2.36E+01
	30	1.61E+04	2.56E+03	9.16E+03
GA	5	5.08E-02	9.20E-03	2.01E-02
	10	7.06E-01	1.06E-01	5.61E-01
	30	2.34E+01	7.50E-02	2.23E+01
PSO	5	2.20E-01	3.61E-04	3.95E-01
	10	2.61E+00	3.98E-02	1.38E+00
	30	4.11E+01	1.12E+01	2.78E+01
ABC	5	5.13E-02	4.90E-03	5.67E-02
	10	5.22E-02	1.90E-03	5.11E-02
	30	6.07E-02	3.64E-04	7.53E-02

TABLE IV. RESULTS OF THE ACKLEY FUNCTION

Algorithm	Dimension	Mean	Best	SD
SLO	5	0.00E+00	0.00E+00	0.00E+00
	10	9.34E-02	0.00E+00	3.61E-01
	30	9.27E-01	0.00E+00	2.33E+00
GA	5	1.91E-05	8.74E-07	1.47E-05
	10	2.89E-05	4.12E-06	1.22E-05
	30	7.65E-05	5.40E-05	1.22E-05
PSO	5	0.00E+00	0.00E+00	0.00E+00
	10	0.00E+00	0.00E+00	0.00E+00
	30	1.71E-08	1.59E-09	2.11E-08
ABC	5	0.00E+00	0.00E+00	0.00E+00
	10	0.00E+00	0.00E+00	0.00E+00
	30	0.00E+00	0.00E+00	0.00E+00

According to the best function values obtained using the different algorithms with  $D=5$ , the SLO can find the global optimization solution with values close to the theoretical solution and has the same search ability as PSO. This indicates the proposed SLO algorithm has the ability to find the best solution with a comparatively small population size. Based on the mean results of all experiments, the proposed SLO has better performance than GA for Griewank function and Ackley function. However, when dimension is 30, the result of SLO is no better than that of PSO and ABC. Considering the No Free Lunch Theorem, if we compare two searching algorithms with all possible functions, the performance of any two algorithms will be, on average, the same. In this paper, the proposed SLO is suitable for solving optimization problems with lower dimensions.

### B. Constrained mechanical design optimization problems

Generally, constrained optimization problems are more complicated due to the additional requirement that a set of constraints must be satisfied. Many different approaches have been proposed to handle these constraints. The penalty function method is the most popular technique due to its simple principle. In these experiments, the penalty function method [21] is employed to handle three classical constrained engineering problems: pressure container, spring and gearbox.

#### Problem 1: pressure container

The first example is to minimize the cost of a pressure container. The  $x_1, x_2, x_3, x_4$  represent the design parameters  $T_H, T_S, R, L$  respectively as shown in Fig. 2.  $T_H$  and  $T_S$  are the integral multiples of the thicknesses of rolled steel plates, which is 0.0625 inch.

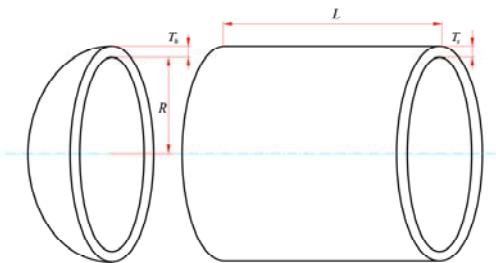


Figure 2. The pressure container problem

$$\begin{aligned}
 &\text{minimize} && f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + \\
 & && 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
 &\text{subject to} && g_1(X) : -x_1 + 0.0193x_3 \leq 0 \\
 & && g_2(X) : -x_2 + 0.00954 \leq 0 \\
 & && g_3(X) : -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
 & && g_4(X) : x_4 - 240 \leq 0
 \end{aligned} \tag{12}$$

where  $X=(x_1, x_2, x_3, x_4)^T$ . The ranges of the design parameters are  $0 \leq x_1, x_2 \leq 99, 10 \leq x_3, x_4 \leq 200$ .

#### Problem 2: spring problem

The spring problem is to minimize the weight of the spring as shown in Fig. 3. The design parameters,  $d(x_1)$  and  $D(x_2)$ , represent the diameter of line wire and the spring. The third parameter is the number of the force coils,  $N(x_3)$ .

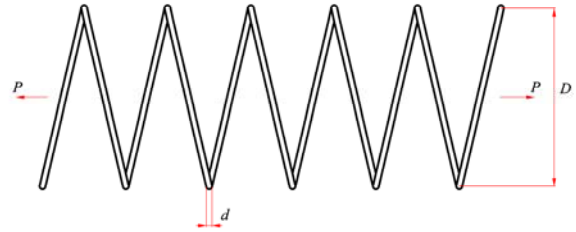


Figure 3. The spring problem

The problem is formulated as:

$$\begin{aligned}
 &\text{minimize} && f(X) = (N+2)Dd^2 \\
 &\text{subject to} && g_1(X) : 1 - \frac{D^3N}{71785d^4} \leq 0 \\
 & && g_2(X) : \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \leq 0 \\
 & && g_3(X) : 1 - \frac{140.45d}{D^2N} \leq 0 \\
 & && g_4(X) : \frac{D+d}{1.5} - 1 \leq 0
 \end{aligned} \tag{13}$$

where  $X=(d, D, N)^T$ ,  $0.05 \leq d \leq 2.0$ ,  $0.25 \leq D \leq 1.3$ ,  $2.0 \leq N \leq 15.0$ .

#### Problem 3: gearbox

The gearbox design problem is the minimization of the total weights of the gearbox. The  $x_1, x_2, \dots, x_7$  represent the design variables as shown in Fig. 4.

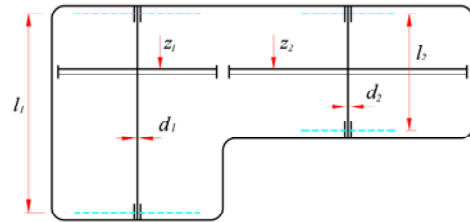


Figure 4. The gearbox problem

$$\begin{aligned}
&\text{minimize} \quad f(X) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - \\
&\quad 43.0934) - 1.508x_1(x_6^2 + x_7^2) + \\
&\quad 7.4777(x_6^3 + x_7^3) \\
&\text{subject to} \quad g_1(X): \frac{27}{x_1x_2^2x_3^3} - 1 \leq 0 \quad g_2(X): \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0 \\
&\quad g_3(X): \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0 \quad g_4(X): \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0 \\
&\quad g_5(X): \frac{((\frac{745x_4}{x_2x_3})^2 + 16.9 \times 10^6)^{1/2}}{110.0x_6^3} - 1 \leq 0 \\
&\quad g_6(X): \frac{((\frac{745x_5}{x_2x_3})^2 + 157.5 \times 10^6)^{1/2}}{85.0x_7^3} - 1 \leq 0 \\
&\quad g_7(X): \frac{x_2x_3}{40} - 1 \leq 0 \quad g_8(X): \frac{5x_2}{x_1} - 1 \leq 0 \\
&\quad g_9(X): \frac{x_1}{12x_2} - 1 \leq 0 \quad g_{10}(X): \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0 \\
&\quad g_{11}(X): \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0
\end{aligned} \tag{14}$$

where  $2.6 \leq x_1 \leq 3.6$ ,  $0.7 \leq x_2 \leq 0.8$ ,  $17 \leq x_3 \leq 28$ ,  $7.3 \leq x_4 \leq 8.3$ ,  $7.8 \leq x_5 \leq 8.3$ ,  $2.9 \leq x_6 \leq 3.9$ ,  $5.0 \leq x_7 \leq 5.5$ .

In the experiments, results of the SLO are compared with the Society and Civilization Algorithm (SCA), the Evolution Strategy ( $(\mu+\lambda)$ -ES) and the ABC algorithms [12]. The values of the algorithm-specific control parameters are given in [12]. For SLO, the parameters are set the same as those described in section A. We conducted 30 independent experiments with random seeds for each problem. The best values were recorded to show the ability of an algorithm to find the optimal solution. Table V presents the reference optimal values given in [12] and the best values obtained using SLO. Parameter values of the best solutions to the three optimization problems are given in Table VI, VII and VIII.

TABLE V. BEST RESULTS OF THE SLO, SCA, EVOLUTION STRATEGY AND ABC ALGORITHMS

Problem	SLO	SCA	$(\mu+\lambda)$ -ES	ABC
Pressure container	6106.8097	6171.0000	6059.7143	6059.7143
Spring problem	0.0127	0.0127	0.0127	0.0127
Gearbox	2897.4658	2994.7442	2996.3481	2997.0584

TABLE VI. PARAMETER VALUES FOR PRESSURE CONTAINER PROBLEM

	SLO	SCA	$(\mu+\lambda)$ -ES	ABC
$x_1$	0.8757	0.8125	0.8125	0.8125
$x_2$	0.5	0.4375	0.4375	0.4375
$x_3$	45.3754	41.9768	42.0984	42.0984
$x_4$	139.8608	182.2845	176.6366	176.6366
$f(x)$	6106.8097	6171.0000	6059.7143	6059.7143

TABLE VII. PARAMETER VALUES FOR SPRING PROBLEM

	SLO	SCA	$(\mu+\lambda)$ -ES	ABC
$x_1$	0.0505	0.0521	0.0528	0.0517
$x_2$	0.3293	0.3682	0.3849	0.3582
$x_3$	13.1136	10.6484	9.8077	11.2038
$f(x)$	0.0127	0.0127	0.0127	0.0127

TABLE VIII. PARAMETER VALUES FOR GEARBOX

	SLO	SCA	$(\mu+\lambda)$ -ES	ABC
$x_1$	3.6000	3.5000	3.4999	3.4999
$x_2$	0.7500	0.7000	0.6999	0.7
$x_3$	17	17	17	17
$x_4$	7.8000	7.3276	7.3000	7.3000
$x_5$	8.3000	7.7153	7.8000	7.8000
$x_6$	3.4	3.3503	3.3502	3.3502
$x_7$	5.0	5.2867	5.2867	5.2878
$f(x)$	2897.4658	2994.7442	2996.3481	2997.0584

According to the results in Table V, for pressure container problem, the optimal solutions are generated by  $(\mu+\lambda)$ -ES and ABC algorithms. On spring problem, the SLO, SCA,  $(\mu+\lambda)$ -ES and ABC algorithms deliver equal performances. As for the gearbox problem with seven parameters, SLO algorithm is more successful. In general, the performances of these four algorithms are roughly the same. But considering the population size, the proposed SLO algorithm is more efficient.

## V. CONCLUSIONS

This paper investigated the foraging behaviors of seven-spot ladybirds and proposed a novel biologically inspired meta-heuristic algorithm called SLO. The performance of SLO algorithm was compared with those of GA, PSO and ABC optimization algorithms by using four numerical benchmark functions. The simulated results show that SLO has the ability to find the best solution with a comparatively small population size and is suitable for solving optimization problems with lower dimensions. Moreover, the proposed SLO algorithm has been applied to solve three constrained engineering problems. The above experiments indicate that the proposed SLO algorithm is an effective method for handling constraints in practical computations. However, it should be noted that the proposed algorithm is not as efficient as ABC. According to the No Free Lunch Theorem [14], "Any elevated performance over one class of problems is offset by performance over another class." Future studies will focus on improving the SLO.

## ACKNOWLEDGMENT

This research was supported by the National Natural Science Foundation of China (Grant No. 51375389).

## REFERENCES

- [1] M. Nasir, S. Das, D. Maity, *et al.*, "A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization," *Inform. Sciences*, vol. 209, pp. 16-36, November 2012.
- [2] H. Lu, and W. Chen, "Self-adaptive velocity particle swarm optimization for solving constrained optimization problems," *J. Global Optim.*, vol. 41, pp. 427-445, July 2008.
- [3] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Michigan: Addison-Wesley Professional, 1989.
- [4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, May 1983.
- [5] F. Glover, and M. Laguna, *Tabu search* vol. 1. Boston, MA: Kluwer Academic Publishers, 1997.
- [6] N. Hansen, and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of IEEE International Conference on Evolutionary Computation 1996*, Nagoya, Japan, pp. 312-317, 1996.

- [7] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput. Aided. Design.*, vol. 43, pp. 303-315, March 2011.
- [8] A. R. Yildiz, "Hybrid taguchi-harmony search algorithm for solving engineering optimization problems," *Int. J. Ind. Eng.-Theory*, vol. 15, pp. 286-293, 2008.
- [9] J. Zhang, and A. C. Sanderson, "Jade: Adaptive differential evolution with optional external archive," *IEEE T. Evolut. Comput.*, vol. 13, pp. 945-958, October 2009.
- [10] R. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, Nagoya, Japan, pp. 39-43, 1995.
- [11] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the cec'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, pp. 617-644, December 2009.
- [12] B. Akay, and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *J. Intell. Manuf.*, vol. 23, pp. 1001-1014, August 2012.
- [13] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Appl. Soft. Comput.*, vol. 10, pp. 629-640, March 2010.
- [14] D. H. Wolpert, and W. G. Macready, "No free lunch theorems for optimization," *IEEE T. Evolut. Comput.*, vol. 1, pp. 67-82, April 1997.
- [15] A. F. G. Dixon, *Insect predator-prey dynamics: Ladybird beetles and biological control*. London: Cambridge University Press, 2000.
- [16] J. Pettersson, V. Ninkovic, R. Glinwood, M. A. Birkett, and J. A. Pickett, "Foraging in a complex environment-semiochemicals support searching behaviour of the seven spot ladybird," *Eur. J. Entomol.*, vol. 102, p. 365, 2005.
- [17] A. Ferran, and A. F. G. Dixon, "Foraging behaviour of ladybird larvae (coleoptera: Coccinellidae)," *Eur. J. Entomol.*, vol. 90, pp. 383-402, 1993.
- [18] P. Kindlmann, and A. F. G. Dixon, "Optimal foraging in ladybird beetles (coleoptera: Coccinellidae) and its consequences for their use in biological control," *Eur. J. Entomol.*, vol. 90, pp. 443-450, 1993.
- [19] S. Dipti, and T. H. Seow, "Particle swarm inspired evolutionary algorithm (ps-ea) for multiobjective optimization problems," in *The 2003 Congress on Evolutionary Computation*, Canberra, ACT, Australia, pp. 2292-2297, 2003.
- [20] D. Karaboga, and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm," *J. Global Optim.*, vol. 39, pp. 459-471, November 2007.
- [21] S. Kitayama, M. Arakawa, and K. Yamazaki, "Penalty function approach for the mixed discrete nonlinear problems by particle swarm optimization," *Struct. Multidiscip. Optim.*, vol. 32, pp. 191-202, September 2006.