

## Research Article

# Seven-Spot Ladybird Optimization: A Novel and Efficient Metaheuristic Algorithm for Numerical Optimization

**Peng Wang, Zhouquan Zhu, and Shuai Huang**

*School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Peng Wang; wangpeng305@nwpu.edu.cn

Received 9 September 2013; Accepted 17 October 2013

Academic Editors: T. Chen and J. Yang

Copyright © 2013 Peng Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a novel biologically inspired metaheuristic algorithm called seven-spot ladybird optimization (SLO). The SLO is inspired by recent discoveries on the foraging behavior of a seven-spot ladybird. In this paper, the performance of the SLO is compared with that of the genetic algorithm, particle swarm optimization, and artificial bee colony algorithms by using five numerical benchmark functions with multimodality. 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.

## 1. Introduction

In recent years, heuristic algorithms have gained popularity because of their ability to find near-optimal solutions to problems unsolved by analytical methods within reasonable computation time due to the multimodality or high dimensionality of their objective functions [1]. Heuristic algorithms are usually developed to solve a specific problem. There is also a class of heuristic algorithms which can be used to solve a large class of problems either directly or with minor modifications, hence the name meta-heuristic algorithms [2].

Researchers continue to develop many meta-heuristic algorithms. Some of the most successful meta-heuristic algorithms include genetic algorithm (GA) [3], ant colony optimization [4], particle swarm optimization (PSO) [5], and artificial bee colony (ABC) [6]. Some of the recently proposed meta-heuristic algorithms include cuckoo search [7], monkey search [8], firefly algorithm [9], grenade explosion method [10], cat swarm optimization [11], and the artificial chemical reaction optimization algorithm [12]. The majority of these algorithms are biologically inspired; that is, they mimic nature for problem solving.

Meta-heuristic algorithms are widely used in different fields and problems, such as manufacturing [13], services [14], scheduling [15], transportation [16], health [17], sports

[18], geology [19], and astronomy [20]. A single meta-heuristic algorithm that can solve all optimization problems of different types and structures does not exist, and, thus, new meta-heuristic optimization algorithms are continuously developed [21].

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. 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, such as GA, PSO, and ABC, by using five different dimensional classical benchmark functions, as given in [6, 22]. The simulated results show that the SLO has the ability to get out of a local minimum and is efficient for some multivariable, multimodal function optimizations.

In general, all the metaheuristic algorithms have something in common in the sense that they are population-based search methods. These methods move from a set of points (population) to another set of points in a single iteration with likely improvement using a combination of deterministic or probabilistic rules. The most remarkable difference of these metaheuristic algorithms lies in the updating rules. The GA is inspired by the principles of genetics and evolution and



FIGURE 1: Seven-spot ladybird.

mimics the reproduction behavior observed in biological populations. The GA employs the principal of “survival of the fittest” in its search process to select and generate individuals that are adapted to their environment. In PSO, instead of using genetic operators, each particle adjusts its “flying” according to its own flying experience and its companions’ flying experience [23]. ABC uses minimal model that mimics the foraging behavior of bees comprising of employed bees, onlooker bees and scouts [24]. The bees aim at discovering places of food sources with high amount of nectar (good fitness values). Differently, in our paper, SLO attempts to simulate the foraging behavior of a seven-spot ladybird which is rarely researched in the field of metaheuristic algorithm. The SLO algorithm consists of three essential components: dividing patches, searching food, and dispersal. Dividing patches increases the search efficiency and dispersal progressively reduces the search space. The search strategy in our algorithm is classified into extensive search and intensive search. Extensive search overcomes the weakness of local search and intensive search increases the possibility of achieving latent best solution. All the ideas are inspired by recent discoveries on the foraging behavior of a seven-spot ladybird which are quite different from other metaheuristic algorithms.

The rest of this paper is organized as follows. Section 2 presents the foraging behavior of the seven-spot ladybird. Section 3 describes the SLO and the steps in detail. Section 4 discusses the experiments and the results. Section 5 draws the conclusions.

## 2. Seven-Spot Ladybird Foraging Behaviors

The seven-spot ladybird (Figure 1), *Coccinella septempunctata*, is a common, easily recognizable insect that has attracted a considerable amount of interest from professional entomologists.

Recent studies have shown that seven-spot ladybirds are more social than we believe them to be [25–28]. Seven-spot ladybirds use different kinds of *pheromones* at different stages of its life history, such as eggs, larvae, pupa, and adult stages (Figure 2). Some chemical ecologies of the seven-spot ladybirds, with special attention to semiochemicals involved

in *social communication* and *foraging behaviors*, have been reviewed in [29].

Seven-spot ladybirds are effective predators of aphids and other homopteran pests, and, thus, their foraging behaviors have been extensively studied [30–35]. Some scholars classified the *environmental levels* of seven-spot ladybirds into prey, patches, and habitats (Figure 3) [33–35], providing a framework for discussing the foraging behaviors of seven-spot ladybirds.

In Figure 3, 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*.

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 [36–38].

## 3. Proposed Seven-Spot Ladybird Optimization (SLO) Algorithm

This section describes the proposed seven-spot ladybird optimization (SLO) algorithm, which simulates the foraging behavior of seven-spot ladybirds to solve multidimensional and multimodal optimization problems. The main steps of the SLO are as follows.

*Step 1* (dividing patches). Suppose 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 treated 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 latent solution to the optimized question.

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). The current fitness evaluation of each ladybird was compared with the fitness value of its best historical position (sbest). If the current value is better than the previous one, then set sbest value is equal to

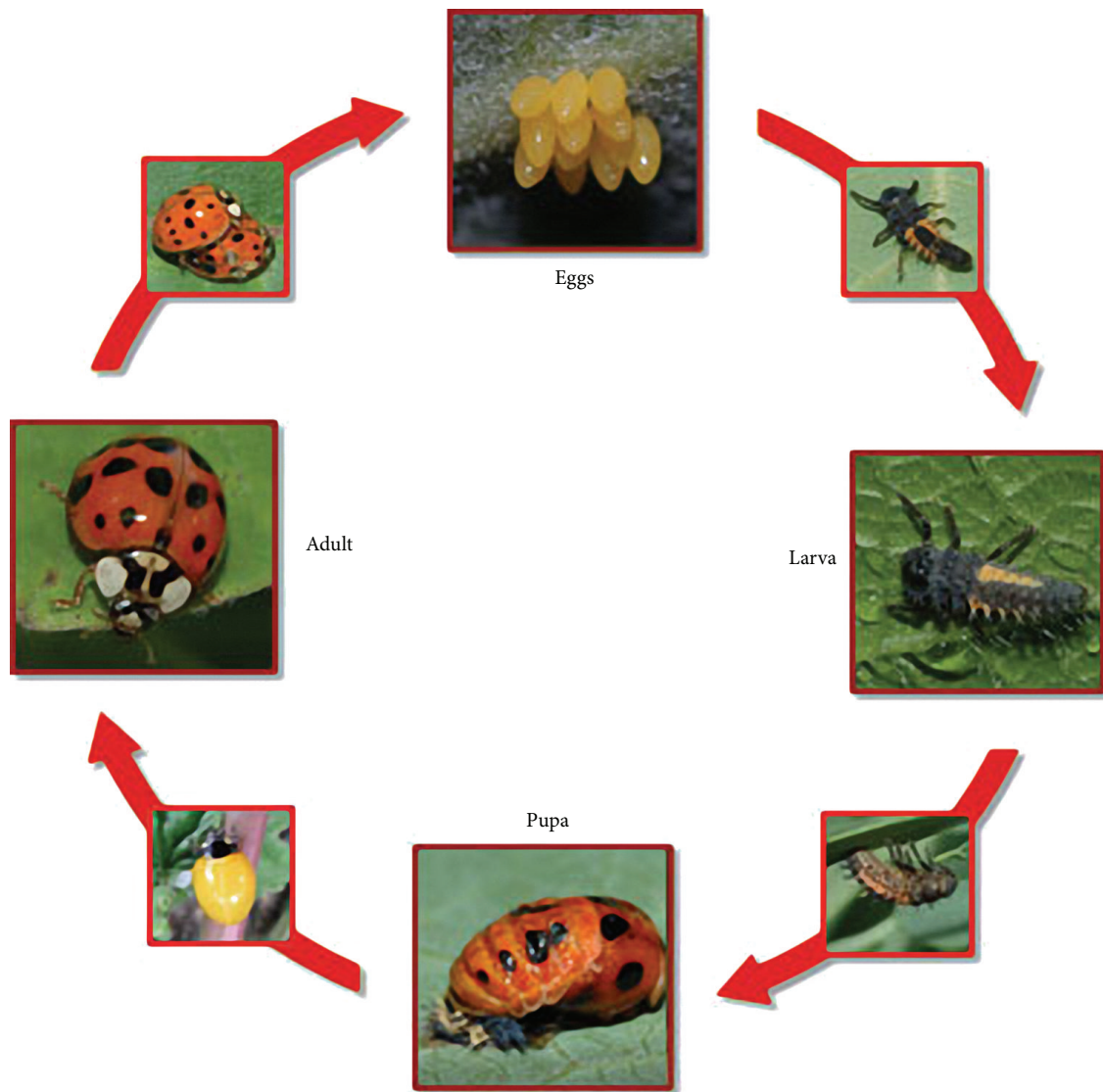


FIGURE 2: The life history of the seven-spot ladybird.

the current value, and the *sbest* position is equal to the current position.

The current best fitness evaluation of all the ladybirds in a patch was compared with the fitness value of their previous best position (*lbest*). If current value is better than the previous one, then set *lbest* value equal to the current value, and the *lbest* position equal to the current position.

The current best fitness evaluation of all the ladybirds in the population was compared with the fitness value of their previous best position (*gbest*). If the current value is better than the previous one, then set *gbest* value equal to the current value, and the *gbest* position equal to the current position.

**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:

$$x'_{i,j} = x_{gbest,j} + \phi w, \quad (1)$$

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:

$$V_i(t) = c * r_1 * (S_i(t) - X_i(t)) + \epsilon_1, \quad (2)$$

$$X_i(t+1) = X_i(t) + V_i(t), \quad |V_i(t)| \leq V_{\max}. \quad (3)$$

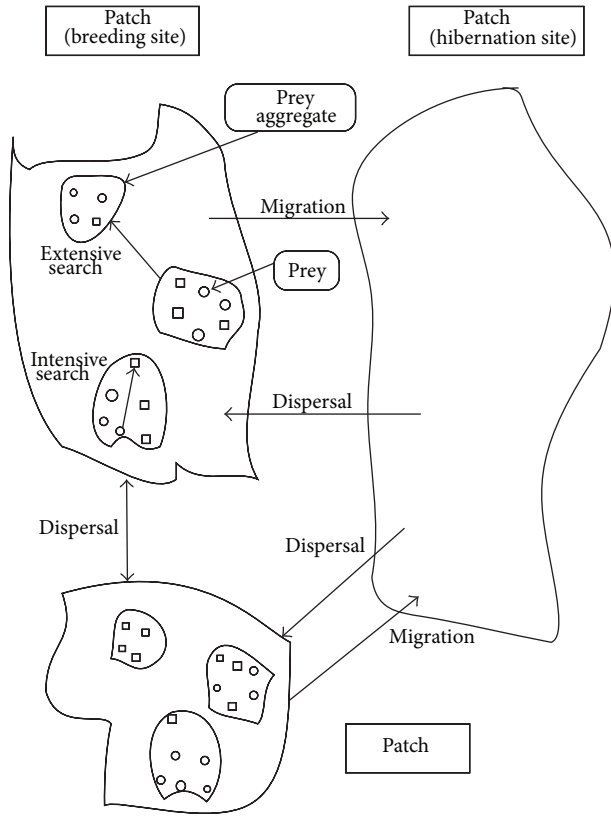


FIGURE 3: Diagram illustrating how a ladybird might perceive its environment and forage for resources.

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

$$V_i(t) = c * r_2 * (L_i(t) - X_i(t)) + \varepsilon_2, \quad (4)$$

$$X_i(t+1) = X_i(t) + V_i(t), \quad |V_i(t)| \leq V_{\max}. \quad (5)$$

In (2) and (4),  $r_1$  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 (3) and (5), 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:

$$V_{\max} = 0.2 (ub - lb), \quad (6)$$

where  $ub$  and  $lb$  are the upper and lower bounds of each patch, respectively. Equation (6) came from [39]. We adopt it here to clamp the particles' velocities on each dimension.

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, that is, the SLO has achieved the maximum iteration number, then the SLO is terminated; otherwise, it returns to *Step 3*.

## 4. Experiments

**4.1. Benchmark Functions.** In the field of heuristic computation, it is common to compare different algorithms using a set of test functions. However, the effectiveness of an algorithm against another algorithm cannot be measured by the number of problems that it solves better [40]. In this way, we have made a previous study of the functions to be optimized for constructing a test set with fewer functions and a better selection. We used five classical benchmark functions to compare the performance of the proposed SLO with those of GA, PSO, and ABC. This set is adequate to include different kinds of problems such as unimodal, multimodal, regular, irregular, separable, nonseparable, and multidimensional. Mathematical descriptions of the benchmark functions were obtained from [6, 22].

The first function is the Griewank function whose value is 0 at its global minimum  $(0, 0, \dots, 0)$  (7). Initialization range for the function is  $[-600, 600]$ . The Griewank function has a product term that introduces interdependence among its variables. The aim is to overcome the failure of the techniques that optimize each variable independently. The optima of the Griewank function are regularly distributed. Since the number of local optima increases with the dimensionality, this function is strongly multimodal. The multimodality disappears for sufficiently high dimensionalities ( $n > 30$ ) and makes the problem unimodal. Consider

$$f_1(\vec{x}) = \frac{1}{4000} \left( \sum_{i=1}^D (x_i^2) \right) - \left( \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) \right) + 1. \quad (7)$$

The second function is the Rastrigin function whose value is 0 at its global minimum  $(0, 0, \dots, 0)$  (8). Initialization range for the function is  $[-15, 15]$ . The Rastrigin function is based on the Sphere function with the addition of cosine modulation to produce many local minima, making it multimodal. The locations of the minima are regularly distributed. The difficult part about finding optimal solutions to the Rastrigin function is that an optimization algorithm is easily trapped in a local optimum on its way towards the global optimum. Consider

$$f_2(\vec{x}) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10). \quad (8)$$

The third function is the Rosenbrock function whose value is 0 at its global minimum  $(1, 1, \dots, 1)$  (9). Initialization



range for the function is  $[-15, 15]$ . The global optimum is inside a long, narrow, parabolic-shaped flat valley. Since it is difficult to converge to the global optimum, the variables are strongly dependent, and the gradients generally do not point towards the optimum, this problem is repeatedly used to test the performance of the optimization algorithms. Consider

$$f_3(\vec{x}) = \sum_{i=1}^D 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2. \quad (9)$$

The fourth function is the Ackley function whose value is 0 at its global minimum  $(0, 0, \dots, 0)$  (10). Initialization range for the function is  $[-32.768, 32.768]$ . The Ackley function has an exponential term that covers its surface with numerous local minima, making its complexity moderated. An algorithm that only uses the gradient steepest descent will be trapped in the local optima, but any search strategy that analyzes a wider region will be able to cross the valley among the optima and achieve better results. A search strategy must combine the exploratory and exploitative components efficiently to obtain good results for the Ackley function. Consider

$$f_4(\vec{x}) = 20 + e - 20e^{(-0.2\sqrt{(1/D)\sum_{i=1}^D x_i^2})} - e^{(1/D)\sum_{i=1}^D \cos(2\pi x_i)}. \quad (10)$$

The fifth function is the Schwefel function whose value is 0 at its global minimum  $(420.9867, 420.9867, \dots, 420.9867)$  (11). Initialization range for the function is  $[-500, 500]$ . The surface of the Schwefel function is composed of a large number of peaks and valleys. The Schwefel function has a second best minimum far from the global minimum where many search algorithms are trapped. Moreover, the global minimum is near the bounds of the domain. Consider

$$f_5(\vec{x}) = D * 418.9829 + \sum_{i=1}^D -x_i \sin\left(\sqrt{|x_i|}\right). \quad (11)$$

**4.2. Settings for Algorithms.** The common control parameters for the algorithms include population size and number of maximum generation. In the experiments, maximum generations were 750, 1000, and 1500 for Dimensions 5, 10, and 30, respectively, and the population size was 50. Other control parameters of the algorithms and the schemes used in [6], including the control parameter values employed for GA, PSO, and ABC are presented below.

**4.2.1. GA Settings.** The settings for the used GA scheme presented in [6] are as follows: single point uniform crossover with rate of 0.95, random selection mechanism, Gaussian mutation with rate of 0.1, and linear ranking fitness function. A child chromosome is added to the population by using the child production scheme.

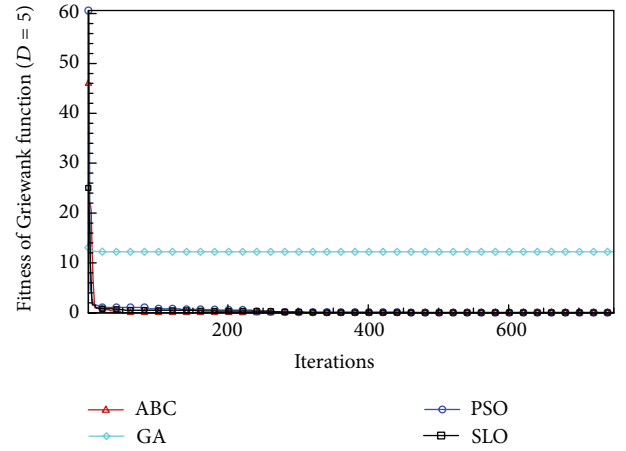


FIGURE 4: Convergence characteristics of the Griewank function with  $D = 5$ .

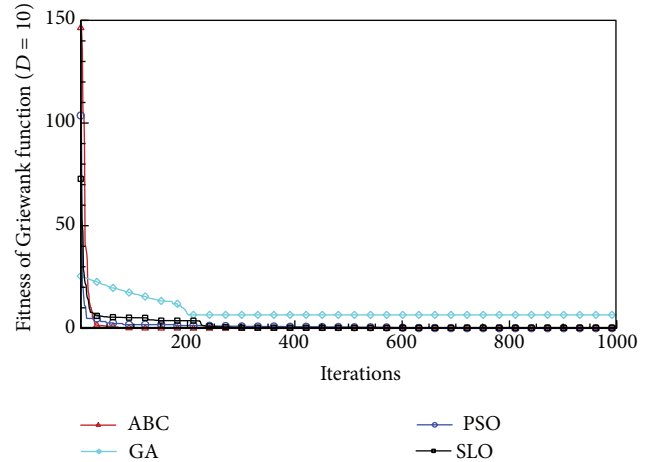


FIGURE 5: Convergence characteristics of the Griewank function with  $D = 10$ .

**4.2.2. PSO Settings.** PSO equations can be expressed as follows:

$$\begin{aligned} \vec{v}(t+1) &= w\vec{v}(t) + \phi_1 \text{rand}(0, 1) (\vec{p}(t) - \vec{x}(t)) \\ &\quad + \phi_2 \text{rand}(0, 1) (\vec{g}(t) - \vec{x}(t)), \\ \vec{x}(t+1) &= \vec{x}(t) + \vec{v}(t+1), \end{aligned} \quad (12)$$

where  $w$  is the additional inertia weight that varies from 0.9 to 0.4 linearly with the iterations. The learning factors,  $\phi_1$  and  $\phi_2$ , are set to 2. The upper and lower bounds for  $v$ ,  $(v_{\min}, v_{\max})$  are set as the maximum upper and lower bounds of  $x$ ; that is,  $(v_{\min}, v_{\max}) = (x_{\min}, x_{\max})$ . If the sum of accelerations would cause the velocity on that dimension  $v(t+1)$  to exceed  $v_{\min}$  or  $v_{\max}$ , then the velocity on that dimension  $v(t+1)$  will be limited to  $v_{\min}$  or  $v_{\max}$ , respectively [6].

TABLE 1: Results of the Griewank function.

Algorithm	Dimension	Mean	Best	SD
SLO	5	$1.2870E - 01$	$7.4000E - 03$	$9.1200E - 02$
	10	$3.1000E - 01$	$2.4600E - 02$	$5.2890E - 01$
	30	$1.4705E + 00$	$6.2000E - 03$	$1.7894E + 00$
GA	5	$1.2240E + 01$	$1.2240E + 01$	$1.2267E - 10$
	10	$1.4568E + 01$	$6.5624E + 00$	$4.7149E + 00$
	30	$1.4100E - 02$	$1.9119E - 10$	$2.4700E - 02$
PSO	5	$2.3700E - 02$	$7.4000E - 03$	$1.2400E - 02$
	10	$7.6900E - 02$	$2.7000E - 02$	$3.3600E - 02$
	30	$1.3200E - 02$	$1.1102E - 15$	$1.4900E - 02$
ABC	5	$6.8808E - 04$	$0.0000E + 00$	$2.1000E - 03$
	10	$2.8000E - 03$	$1.1102E - 16$	$4.6000E - 03$
	30	$6.3515E - 04$	$0.0000E + 00$	$3.5000E - 03$

TABLE 2: Results of the Rastrigin function.

Algorithm	Dimension	Mean	Best	SD
SLO	5	$0.0000E + 00$	$0.0000E + 00$	$0.0000E + 00$
	10	$2.2781E + 01$	$9.9760E - 01$	$1.4721E + 01$
	30	$3.6123E + 02$	$2.3147E + 02$	$5.9041E + 01$
GA	5	$6.3010E - 01$	$2.0653E - 09$	$6.1180E - 01$
	10	$7.9600E - 01$	$3.3258E - 08$	$8.4260E - 01$
	30	$3.0844E + 00$	$2.5108E - 06$	$2.3711E + 00$
PSO	5	$6.6300E - 02$	$0.0000E + 00$	$2.5240E - 01$
	10	$1.7267E + 00$	$0.0000E + 00$	$1.1662E + 00$
	30	$2.9001E + 01$	$1.7927E + 01$	$8.5603E + 00$
ABC	5	$0.0000E + 00$	$0.0000E + 00$	$0.0000E + 00$
	10	$0.0000E + 00$	$0.0000E + 00$	$0.0000E + 00$
	30	$1.0658E - 15$	$0.0000E + 00$	$4.2397E - 15$

TABLE 3: Results of the Rosenbrock function.

Algorithm	Dimension	Mean	Best	SD
SLO	5	$1.3325E + 00$	$1.6163E - 08$	$8.7800E - 01$
	10	$1.6880E + 01$	$6.6976E + 00$	$2.3646E + 01$
	30	$1.6123E + 04$	$2.5639E + 03$	$9.1648E + 03$
GA	5	$5.0800E - 02$	$9.2000E - 03$	$2.0100E - 02$
	10	$7.0620E - 01$	$1.0590E - 01$	$5.6060E - 01$
	30	$2.3368E + 01$	$7.5000E - 02$	$2.2251E + 01$
PSO	5	$2.2040E - 01$	$3.6080E - 04$	$3.9500E - 01$
	10	$2.6053E + 00$	$3.9800E - 02$	$1.3755E + 00$
	30	$4.1055E + 01$	$1.1177E + 01$	$2.7793E + 01$
ABC	5	$5.1300E - 02$	$4.9000E - 03$	$5.6700E - 02$
	10	$5.2200E - 02$	$1.9000E - 03$	$5.1100E - 02$
	30	$6.0700E - 02$	$3.6401E - 04$	$7.5300E - 02$

**4.2.3. ABC Settings.** The control parameters of the ABC algorithm are as follows: the maximum number of cycles is equal to the maximum number of generation and the colony size is equal to the population size, that is, 50, as presented in [6]. The percentage of onlooker bees was 50% of the colony, the employed bees were 50% of the colony, and one bee was selected as the scout bee. The increase in the number of scouts

encourages the exploration because the increase in onlookers for a food source increases exploitation.

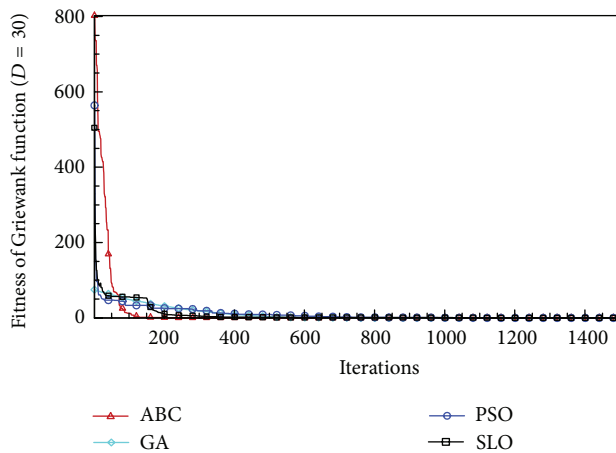
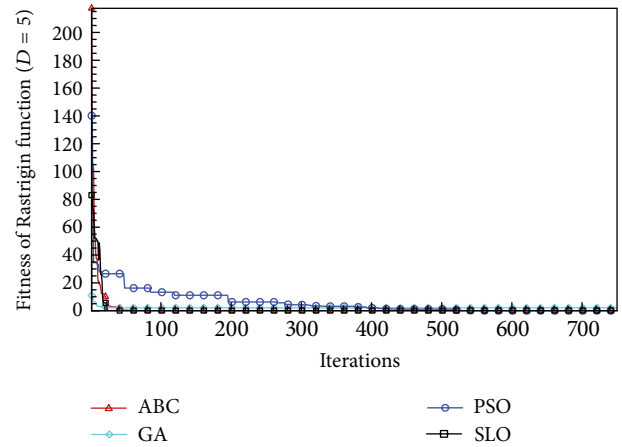
**4.2.4. SLO Settings.** In SLO, each dimension is divided into two equal parts, and thus,  $2^D$  patches are generated. In each patch, the initial population of ladybirds is set to 20. The

TABLE 4: Results of the Ackley function.

Algorithm	Dimension	Mean	Best	SD
SLO	5	$-8.8818E-16$	$-8.8818E-16$	$0.0000E+00$
	10	$9.3400E-02$	$2.6645E-15$	$3.6120E-01$
	30	$9.2670E-01$	$6.2172E-15$	$2.3272E+00$
GA	5	$1.9147E-05$	$8.7429E-07$	$1.4740E-05$
	10	$2.8850E-05$	$4.1240E-06$	$1.2182E-05$
	30	$7.6502E-05$	$5.3960E-05$	$1.2228E-05$
PSO	5	$2.0724E-15$	$-8.8818E-16$	$1.3467E-15$
	10	$3.6119E-15$	$2.6645E-15$	$1.5979E-15$
	30	$1.7064E-08$	$1.5857E-09$	$2.1073E-08$
ABC	5	$2.6645E-15$	$2.6645E-15$	$0.0000E+00$
	10	$6.9278E-15$	$2.6645E-15$	$2.1681E-15$
	30	$4.4154E-13$	$1.6964E-13$	$3.0035E-13$

TABLE 5: Results of the Schwefel function.

Algorithm	Dimension	Mean	Best	SD
SLO	5	$3.2505E+02$	$6.3638E-05$	$1.7285E+02$
	10	$1.2493E+03$	$7.5042E+02$	$2.7913E+02$
	30	$5.8700E+03$	$4.7145E+03$	$7.6059E+02$
GA	5	$2.0752E+03$	$2.0752E+03$	$5.2257E-11$
	10	$4.1504E+03$	$4.1504E+03$	$9.4875E-11$
	30	$1.2451E+04$	$1.2451E+04$	$7.3221E-10$
PSO	5	$2.9610E+02$	$6.3638E-05$	$1.3104E+02$
	10	$5.6719E+02$	$2.3688E+02$	$1.6709E+02$
	30	$2.8564E+03$	$1.5989E+03$	$4.0442E+02$
ABC	5	$6.3638E-05$	$6.3638E-05$	$1.9653E-14$
	10	$1.2728E-04$	$1.2728E-04$	$5.2425E-14$
	30	$3.1000E-03$	$3.8183E-04$	$1.4200E-02$

FIGURE 6: Convergence characteristics of the Griewank function with  $D = 30$ .FIGURE 7: Convergence characteristics of the Rastrigin function with  $D = 5$ .

parameter  $limit$  is 100 and  $w$  is 1; that is, after 100 cycles of search, if a position in a patch cannot be improved, then it will be abandoned and a new position will be produced in

the neighborhood of  $g_{best}$ . The parameter  $c$  in (2) and (4) decreases linearly from 10 to 2. The sideward vibration  $\epsilon_1$  is  $Rand * 10e-4$  and  $\epsilon_2$  is  $Rand * 10e-8$ .

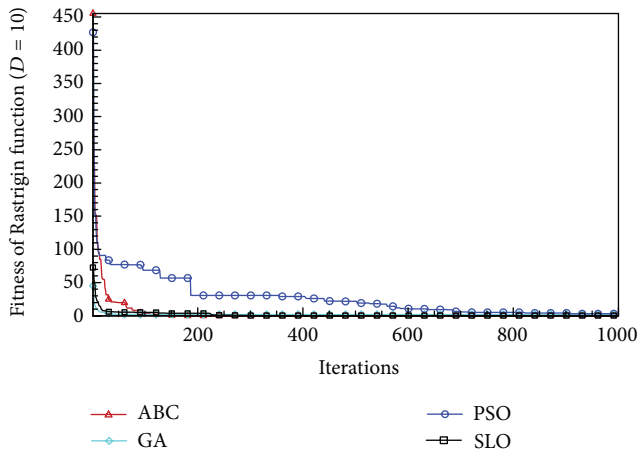


FIGURE 8: Convergence characteristics of the Rastrigin function with  $D = 10$ .

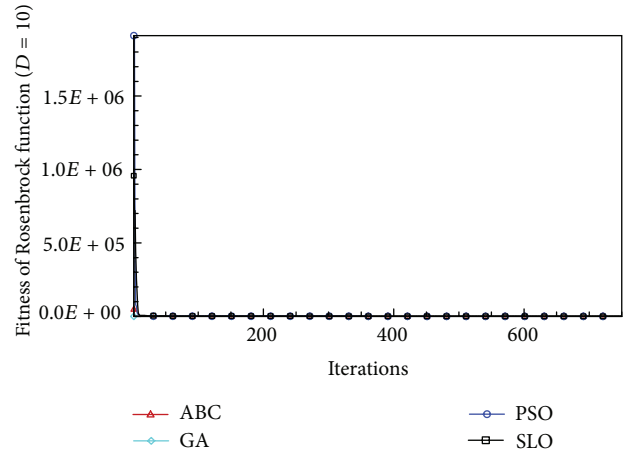


FIGURE 11: Convergence characteristics of the Rosenbrock function with  $D = 10$ .

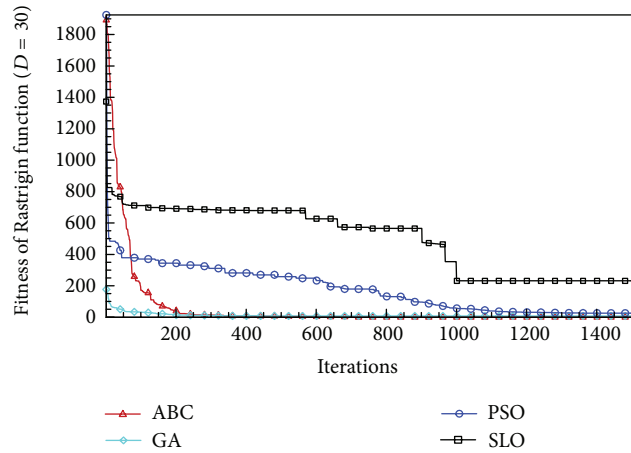


FIGURE 9: Convergence characteristics of the Rastrigin function with  $D = 30$ .

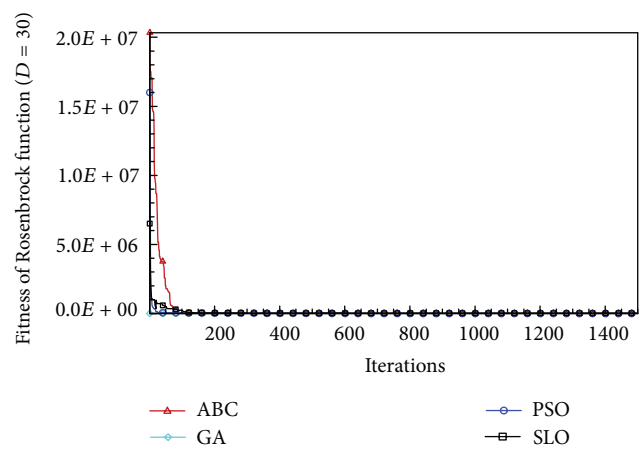


FIGURE 12: Convergence characteristics of the Rosenbrock function with  $D = 30$ .

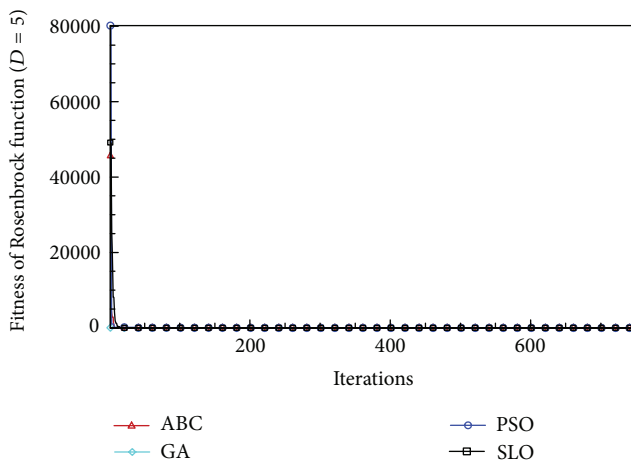


FIGURE 10: Convergence characteristics of the Rosenbrock function with  $D = 5$ .

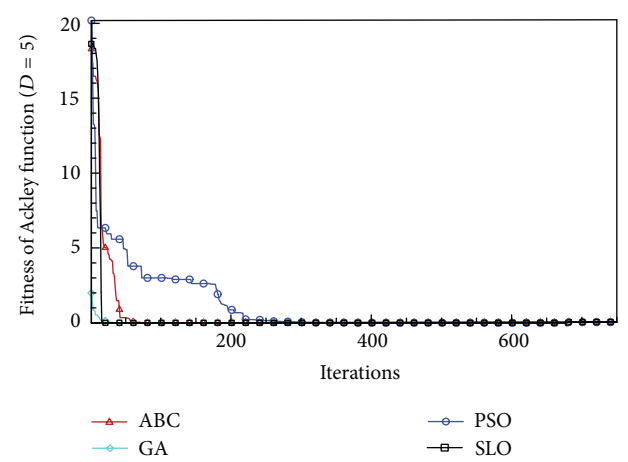


FIGURE 13: Convergence characteristics of the Ackley function with  $D = 5$ .



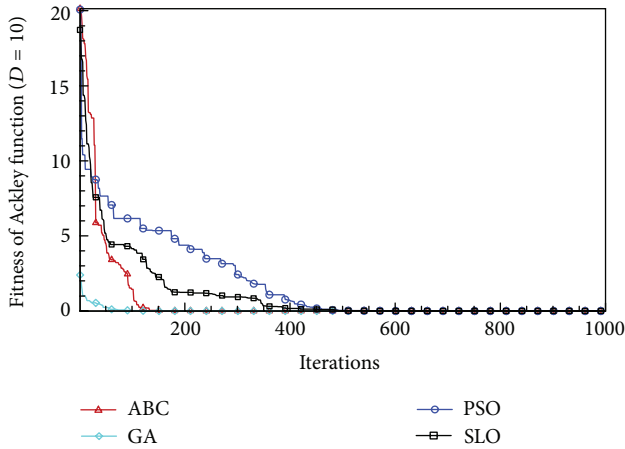


FIGURE 14: Convergence characteristics of the Ackley function with  $D = 10$ .

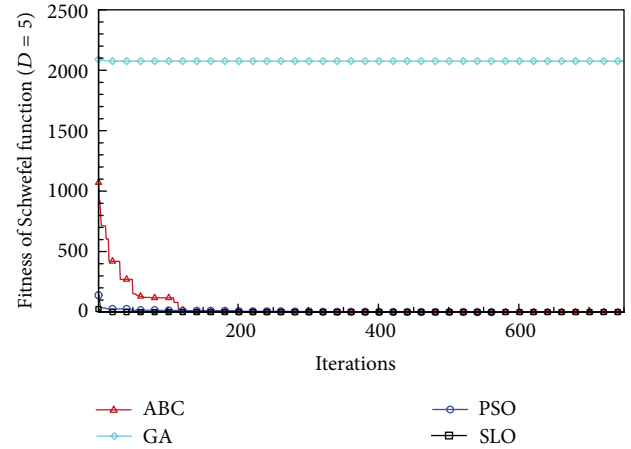


FIGURE 16: Convergence characteristics of the Schwefel function with  $D = 5$ .

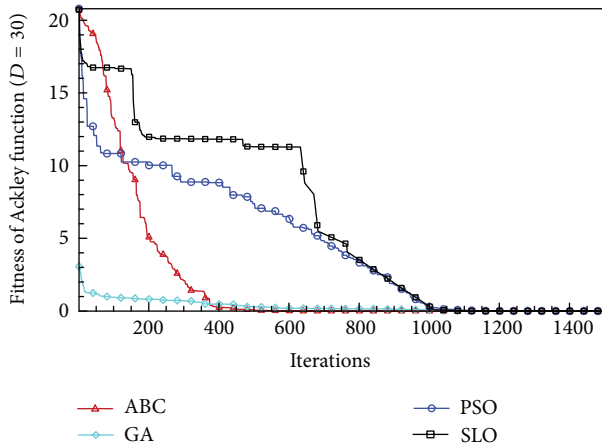


FIGURE 15: Convergence characteristics of the Ackley function with  $D = 30$ .

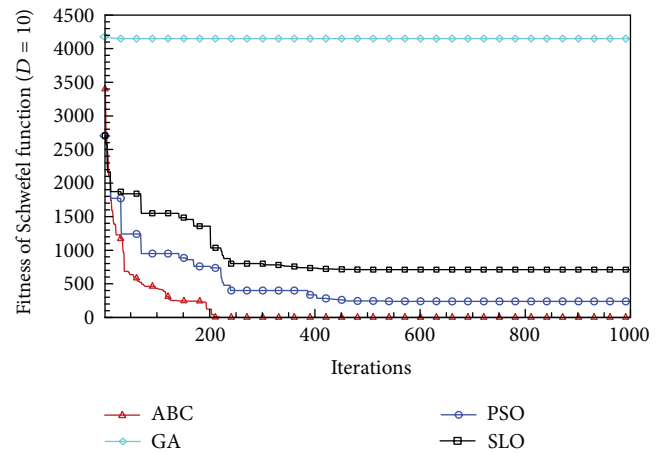


FIGURE 17: Convergence characteristics of the Schwefel function with  $D = 10$ .

**4.3. Results and Discussion.** In this paper, all the experiments were repeated 30 times with different random seeds. The best and mean function values of the solutions found using the algorithms for different dimensions were recorded. Tables 1, 2, 3, 4, and 5 present the mean, best, and standard deviations of the function values obtained using SLO, GA, PSO, and ABC with  $D = 5$ ,  $D = 10$  and  $D = 30$ . Figures 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, and 18 show the convergence characteristics in terms of the fitness value of each algorithm for each test function.

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. Many literatures have pointed out that larger population size and large number of generations increase the likelihood of obtaining a global optimum solution. Thus, the performance of PSO with a swarm size of 50 is better than that with a

swarm size of 20. According to the experiments in our paper, SLO with a small population of 20 was able to find the global optimization solution with values close to the theoretical solution. 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, Ackley function, and Schwefel function. However, when dimension is 30, the result of SLO is no better than that of PSO and ABC. Comparing the convergence graphs, SLO converged faster and performed better compared with GA.

From the results, we can see that the SLO does not obtain better result along with the growing dimensions. Considering the No Free Lunch Theorem [41], if we compare two searching algorithms with all possible functions, the performance of any two algorithms will be, on average, the same. As a result, when an algorithm is evaluated, we must look for the kind of problems where its performance is good, in order to characterize the type of problems for which the algorithm is

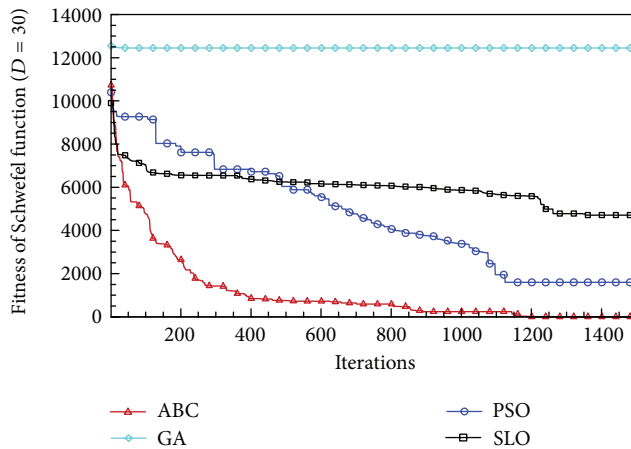


FIGURE 18: Convergence characteristics of the Schwefel function with  $D = 30$ .

suitable [42]. In this paper, the proposed SLO is suitable for solving optimization problems with lower dimensions.

## 5. Conclusion

This paper investigated the foraging behaviors of seven-spot ladybirds and proposed a novel biologically inspired meta-heuristic algorithm called SLO. The SLO, GA, PSO, and ABC algorithms were tested on five numerical benchmark functions with multimodality to validate the performance of SLO. The simulated results show that SLO has the ability to find the best solution and is suitable for solving optimization problems with lower dimensions. In this paper, the ABC algorithm outperformed all other algorithms, but according to the No Free Lunch Theorem [41], “any elevated performance over one class of problems is offset by performance over another class.” Future studies will focus on improving the SLO.

## Acknowledgments

The authors are grateful to the editor and the anonymous referees for their insightful and constructive comments and suggestions, which have been very helpful for improving this paper. This research was supported by the National Natural Science Foundation of China (Grant no. 51375389) and the National High Technology Research and Development Program of China (863 Program) no. 2011AA09A104.

## References

- [1] D. T. Pham and D. Karaboga, *Intelligent Optimisation Techniques*, Springer, New York, NY, USA, 2000.
- [2] F. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*, Kluwer Academic, Boston, Mass, USA, 2003.
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Lansing, Mich, USA, 1975.
- [4] M. Dorigo, V. Maniezzo, and A. Coloni, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.
- [5] R. C. Eberhart and J. Kennedy, “New optimizer using particle swarm theory,” in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
- [6] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [7] X. S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, Coimbatore, India, December 2009.
- [8] A. Mucherino and O. Seref, “Monkey search: a novel meta-heuristic search for global optimization,” in *Proceedings of the Conference on Data Mining, Systems Analysis, and Optimization in Biomedicine*, pp. 162–173, Gainesville, Fla, USA, March 2007.
- [9] X. S. Yang, “Firefly algorithms for multimodal optimization,” *Stochastic Algorithms: Foundations and Applications*, vol. 5792, pp. 169–178, 2009.
- [10] A. Ahrari and A. A. Atai, “Grenade explosion method—a novel tool for optimization of multimodal functions,” *Applied Soft Computing Journal*, vol. 10, no. 4, pp. 1132–1140, 2010.
- [11] S. C. Chu, P. W. Tsai, and J. S. Pan, “Cat swarm optimization,” in *PRICAI 2006: Trends in Artificial Intelligence*, vol. 4099 of *Lecture Notes in Computer Science*, pp. 854–858, Springer, Guilin, China, 2006.
- [12] B. Alatas, “Acroa: artificial chemical reaction optimization algorithm for global optimization,” *Expert Systems with Applications*, vol. 38, no. 10, pp. 13170–13180, 2011.
- [13] M. Kapanoglu and W. A. Miller, “An evolutionary algorithm-based decision support system for managing flexible manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 6, pp. 529–539, 2004.
- [14] N. Mansour, H. Tabbara, and T. Dana, “A genetic algorithm approach for regrouping service sites,” *Computers and Operations Research*, vol. 31, no. 8, pp. 1317–1333, 2004.
- [15] Z. Lian, X. Gu, and B. Jiao, “A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan,” *Applied Mathematics and Computation*, vol. 175, no. 1, pp. 773–785, 2006.
- [16] L. Barcos, V. Rodríguez, M. J. Álvarez, and F. Robusté, “Routing design for less-than-truckload motor carriers using ant colony optimization,” *Transportation Research E*, vol. 46, no. 3, pp. 367–383, 2010.
- [17] G. N. Ramos, Y. Hatakeyama, F. Dong, and K. Hirota, “Hyper-box clustering with ant colony optimization (HACO) method and its application to medical risk profile recognition,” *Applied Soft Computing Journal*, vol. 9, no. 2, pp. 632–640, 2009.
- [18] J. P. Hamiez and J. K. Hao, “Using solution properties within an enumerative search to solve a sports league scheduling problem,” *Discrete Applied Mathematics*, vol. 156, no. 10, pp. 1683–1693, 2008.
- [19] M. Tamer Ayvaz, “Application of harmony search algorithm to the solution of groundwater management models,” *Advances in Water Resources*, vol. 32, no. 6, pp. 916–924, 2009.
- [20] P. Charbonneau, “Genetic algorithms in astronomy and astrophysics,” *Astrophysical Journal*, vol. 101, no. 2, pp. 309–334, 1995.
- [21] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “Gsa: a gravitational search algorithm,” *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

- [22] D. Srinivasan and T. H. Seow, "Particle swarm inspired evolutionary algorithm (ps-ea) for multiobjective optimization problems," in *Proceedings of the Congress on Evolutionary Computation*, pp. 2292–2297, Canberra, Australia, 2003.
- [23] R. Hassan, B. Cohanin, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pp. 1–13, Austin, Tex, USA, April 2005.
- [24] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1001–1014, 2012.
- [25] I. Hodek, G. Iperti, and M. Hodkova, "Long-distance flights in Coccinellidae (Coleoptera)," *The European Journal of Entomology*, vol. 90, no. 4, pp. 403–414, 1993.
- [26] A. F. G. Dixon, *Insect Predator-Prey Dynamics: Ladybird Beetles and Biological Control*, Cambridge University Press, New York, NY, USA, 2000.
- [27] J. L. Hemptinne, M. Gaudin, A. F. G. Dixon, and G. Lognay, "Social feeding in ladybird beetles: adaptive significance and mechanism," *Chemoecology*, vol. 10, no. 3, pp. 149–152, 2000.
- [28] J. L. Hemptinne, G. Lognay, C. Gauthier, and A. F. G. Dixon, "Role of surface chemical signals in egg cannibalism and intraguild predation in ladybirds (Coleoptera: Coccinellidae)," *Chemoecology*, vol. 10, no. 3, pp. 123–128, 2000.
- [29] 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," *The European Journal of Entomology*, vol. 102, no. 3, pp. 365–370, 2005.
- [30] V. Ninkovic, S. Al Abassi, and J. Pettersson, "The influence of aphid-induced plant volatiles on ladybird beetle searching behavior," *Biological Control*, vol. 21, no. 2, pp. 191–195, 2001.
- [31] N. Suzuki and T. Ide, "The foraging behaviors of larvae of the ladybird beetle, *Coccinella septempunctata* L., (Coleoptera: Coccinellidae) towards ant-tended and non-ant-tended aphids," *Ecological Research*, vol. 23, no. 2, pp. 371–378, 2008.
- [32] A. Vantaux, O. Roux, A. Magro, and J. Orivel, "Evolutionary perspectives on myrmecophily in ladybirds," *Psyche*, vol. 2012, Article ID 591570, 7 pages, 2012.
- [33] M. P. Hassell and T. R. E. Southwood, "Foraging strategies of insects," *Annual Review of Ecology and Systematics*, vol. 9, pp. 75–98, 1978.
- [34] I. Hodek, S. Chakrabarti, and M. Rejmanek, "The effect of prey density on food intake by adult *Cheilomenes sulphurea* [Col.: Coccinellidae]," *BioControl*, vol. 29, no. 2, pp. 179–184, 1984.
- [35] A. Ferran and A. F. G. Dixon, "Foraging behaviour of ladybird larvae (Coleoptera: Coccinellidae)," *The European Journal of Entomology*, vol. 90, no. 4, pp. 383–402, 1993.
- [36] J. L. Hemptinne, A. F. G. Dixon, and J. Coffin, "Attack strategy of ladybird beetles (Coccinellidae): factors shaping their numerical response," *Oecologia*, vol. 90, no. 2, pp. 238–245, 1992.
- [37] P. Kindlmann and A. F. G. Dixon, "Optimal foraging in ladybird beetles (Coleoptera: Coccinellidae) and its consequences for their use in biological control," *European Journal of Entomology*, vol. 90, no. 4, pp. 443–450, 1993.
- [38] N. Minoretti and W. W. Weisser, "The impact of individual ladybirds (*Coccinella septempunctata*, Coleoptera: Coccinellidae) on aphid colonies," *The European Journal of Entomology*, vol. 97, no. 4, pp. 475–479, 2000.
- [39] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 81–86, Seoul, Republic of Korea, May 2001.
- [40] D. Ortiz-Boyer, C. Hervás-Martínez, and N. García-Pedrajas, "CIXL2: a crossover operator for evolutionary algorithms based on population features," *Journal of Artificial Intelligence Research*, vol. 24, pp. 1–48, 2005.
- [41] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [42] F. Valdez and P. Melin, "Comparative study of particle swarm optimization and genetic algorithms for complex mathematical functions," *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 2, no. 1, pp. 43–51, 2008.



