IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 20, 2020, accepted July 1, 2020, date of publication July 13, 2020, date of current version July 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008692

Beetle Colony Optimization Algorithm and Its Application

HENG ZHANG[1], ZIMING LI[2], XIANGYUAN JIANG[1], XIAOJING MA[1], JIYANG CHEN[1], SHUAI LI[3], (Senior Member, IEEE), YIZHONG LUAN[1], ZHENYI LV[4], AND SILE MA[1]

[1]Institute of Marine Science and Technology, Shandong University, Qingdao 266237, China
[2]College of Computer Science, State University of New York at Albany, Albany, NY 12222, USA
[3]College of Engineering, Swansea University, Swansea SA1 8EN, U.K.
[4]Shenzhen Keweitai Enterprise Development Company Ltd., Shenzhen 518040, China

Corresponding authors: Zhenyi Lv (lzy139029@keweitai.com) and Sile Ma (masile@sdu.edu.cn)

This work was supported in part by the International Science and Technology Cooperation Program of China under Grant 2014DFR70730.

ABSTRACT Massive data sets and complex scheduling processes have high-dimensional and non-convex features bringing challenges on various applications. With deep insight into the bio-heuristic opinion, we propose a novel Beetle Colony Optimization (BCO) being able to adapt NP-hard issues to meet growing application demands. Two important mechanisms are introduced into the proposed BCO algorithm. The first one is Beetle Antennae Search (BAS), which is a mechanism of random search along the gradient direction but not use gradient information at all. The second one is swarm intelligence, which is a collective mechanism of decentralized and self-organized agents. Both of them have reached a performance balance to elevate the proposed algorithm to maintain a wide search horizon and high search efficiency. Finally, our algorithm is applied to traveling salesman problem, and quadratic assignment problem and possesses excellent performance, which also shows that the algorithm has good applicability from the side. The effectiveness of the algorithm is also substantiated by comparing the results with the original ant colony optimization (ACO) algorithm in 3D simulation model experimental path planning.

INDEX TERMS Single-agent random search, swarm-intelligence optimization, path planning, bio-heuristic algorithm.

I. INTRODUCTION

In recent years, unmanned aerial vehicles (UAV) have developed rapidly and the market has been expanding. The range of its applications are from climate monitoring to water area inspection, power grid inspection to forest inspection, fire detection to disaster detection, and express delivery to disaster relief. It can be seen that UAV are playing an increasingly important role in various complex and dangerous tasks. Path planning of UAV is an important preliminary step in UAV flight mission which can be fulfilled by finding the optimum solution for an optimization problem [1]. Path planning is one of the most important problems to be explored in UAV for finding an optimal path between source and destination [2], which also has attracted more and more attentions of researchers and developed rapidly due to its wide applications [3].

Intelligent algorithms play an very important role in solving path planning problems. Methods of path planning for UAV can generally be classified into three types: node-based methods, sampling-based methods and bio-heuristic algorithms. Node-based path planning methods mainly include breadth-first search [4], depth-first search [5], Dijkstra algorithm [6], A* algorithm [7], lifelong planning algorithm (LPA) [8], and theta star [9]. Breadth-first search and depth-first search are the most general node-based search algorithms, and Dijkstra algorithm and A* algorithm all use idea of breadth-first search to solve the problem. Song *et al.* proposed an improved A* algorithm applied to the bpringer USV and a new path smoothing process with three path smoothers to improve the performance of the generated route [10]. Sampling-based algorithms mainly include probabilistic road maps (PRM) [11], rapidly exploring random tree (RRT) [12], RRT* [13], artificial potential field (APF) [14] and so on. Galceran *et al.* finished a survey on coverage path planning for robotics. Coverage path planning (CPP) is the task of determining a path that passes over all points of an area or volume of interest while avoiding obstacles [15]. These types of algorithms usually need to pre-process the

The associate editor coordinating the review of this manuscript and approving it for publication was Dongxiao Yu.

map, grid or sample the map,and then randomly search for paths. Bio-heuristic algorithms include simulated annealing algorithm [16], genetic algorithm (GA) [17], particle swarm optimization algorithm (PSO) [18], ant colony optimization (ACO) [19] algorithm, artificial fish swarm algorithm (AFSA) [20] and so on. These heuristic algorithms are generated from the characteristics of biological creatures in nature, such as biological foraging behavior, group behavior,biological evolution,which had been used widely for different applications [21]. These types of path planning algorithms can search for surroundings under heuristic search strategies and go in the most promising direction.

The heuristic algorithms had been proved that can evidently improve the efficiency of path planning [22] and obtain global approximate result by feature of bio-heuristic behavior. This type of method does not require constructing complex mathematical models to find a globally feasible solution for path planning. Kroumov *et al.* proposed a highly efficient potential field-based 3D path planning technique by simulated annealing neural network for mobile robots and several simulation results prove the effectiveness of the proposed algorithm [23]. Xu *et al.* proposed an improved artificial bee colony(ABC) optimization algorithm for path planning of UCAV in various combat field environments and series of experimental comparison results show effectiveness and robustness of improved ABC optimization algorithm [24]. Duan *et al.* proposed a new hybrid meta-heuristic ACO and differential evolution(DE) algorithm for UCAV three-dimension path planning, and DE is applied to optimize the pheromone trail of the improved ACO model during the process of pheromone updating [25]. Roberge *et al.* proposed using GA and PSO to cope with complex problem and compute feasible and quasi-optimal trajectories for fixed wing UAVs in a complex 3D environment and considering the dynamic properties of the vehicle [26]. Wu *et al.* proposed a novel fallback beetle antenna search algorithm for planning planning of mobile robots by introducing a fallback mechanism in the traditional beetle antenna search algorithm [27]. Li *et al.* proposed a three dimensional path planning method for effective and engineering-oriented path planning of low altitude penetration of UAV based on improved genetic algorithm [28]. Panda *et al.* finished a comprehensive review of path planning algorithm for autonomous underwater vehicles [29]. Wu *et al.* proposed a novel path palnner named obstacle avoidance beetle antennae search (OABAS) algorithm applied to the global path planning of unmanned aerial vehicles (UAVs) [22]. Ojha *et al.* analysed the performance of ACO algorithm for continuous function optimization [30]. Pal *et al.* proposed an optimization algorithm based on combination of ant colony optimization and particle swarm optimization and the new optimization algorithm makes complete use of parameter of both algorithms [31]. Tan *et al.* proposed a swarm optimization algorithm called normative fish swarm algorithm (NFSA) to obtain effective global optimum at superior convergence speed [32]. Liu *et al.* proposed a dynamic adaptive firefly algorithm to improve the convergence rate and solution precision and to avoid the premature algorithm trapping at the local extreme with global-oriented moving mechanism and dynamically adjusting the step size and attractiveness [33]. Jiang *et al.* proposed a novel bio-heuristic optimization strategy called beetle antennae search(BAS) algorithm inspired by the searching behavior of longhorn beetles [34]. Wang *et al.* proposed a new kind of nature-inspired metaheuristic algorithm, called monarch butterfly optimization (MBO) by simplifying and idealizing the migration of monarch butterflies [35]. Wang *et al.* proposed a new meta-heuristic algorithm called beetle swarm optimization (BSO) algorithm by enhancing the performance of swarm optimization through beetle foraging principles [36]. Yu *et al.* introduced a novel intelligent optimization algorithm based artificial bee colony algorithm (ABC), called self-adaptive artificial bee colony algorithm [37]. Michalis *et al.* finished a survey of swarm intelligence for dynamic optimization: algorithms and applications [38].

BCO algorithm combined single-agent search strategy of beetle antennae with swarm optimization algorithms. Compared with ACO and BAS algorithm, it has powerful search range and outstanding search speed. BCO algorithm will introduce the single-agent search mechanism into swarm-intelligence optimization and implements a delicate balancing strategy between the two. Finally, BCO algorithm proposed in this paper has a positive significance for solving many NP-hard problems, especially the path planning problem of UAV. The main contributions of our algorithm are listed as follows,

(a) The single-agent random search mechanism is introduced into the swarm-intelligence optimization to solve problem.
(b) The advantages and characteristics of the swarm-intelligence optimization algorithm and the search mechanism of a single-agent implement a delicate balancing strategy between the two.
(c) A novel heuristic optimization algorithm, termed beetle colony optimization algorithm (BCO), is proposed and it can solve traveling salesman problem(TSP), quadratic assignment problem(QAP) and be applied to path planning for UAV in 3D space.

The next organization of this paper is as follows. Design of algorithm is introduced in Section II. In Section III, we introduce validation of BCO with typical benchmarks. In Section IV, we introduce the validation and application of BCO for path planning of UAV and show the simulation results and compare the performance of BCO with ACO and BAS algorithm in 3D path planning. In Section V, we summarize the full paper.

II. DESIGN AND ANALYSIS OF ALGORITHM

In this section, the description of BCO algorithm is introduced, including single-agent random search, swarm-intelligence optimization synthesization and the analysis of BCO algorithm.

A. SINGLE-AGENT RANDOM SEARCH

In this paper, a beetle is as single-agent for random search. When the beetle forages, it does not know where the destination is, but forages according to the strength of the food smell in the air through two antennae on the head. Therefore, when the strength of the food smell felt by the two antennae is different, the beetle will move to the side of high strength of the food smell, so back and forth until the target point. The mathematical model of single-agent random search is as follows.

First, the fitness function of our algorithm is $f(\boldsymbol{x}^t)$, where $f(\boldsymbol{x}^t)$ is the fitness value when the position of the beetle is $\boldsymbol{x}^t$ at iteration t. The next position of the beetle is updated by the following equation.

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \delta \overrightarrow{s}\, sign(f(\boldsymbol{x}_{right}) - f(\boldsymbol{x}_{left})) \tag{1}$$

where $\overrightarrow{s}$ indicates the random search direction of the beetle and will be introduced in next subsection, and the $\boldsymbol{x}_{right}$ and $\boldsymbol{x}_{left}$ indicate the position of right and left antenna of beetle at iteration $\boldsymbol{t}\ \mathcal{C}\ \mathbf{1}$. Where $f(\boldsymbol{x}_{right})$ and $f(\boldsymbol{x}_{left})$ are separately the value of the fitness of the two antennae. The $sign()$ is a symbolic function,

$$sign(x) = \begin{cases} 1, & x > 0; \\ 0, & x = 0; \\ \text{-1}, & x < 0. \end{cases} \tag{2}$$

Then we calculate the position of the left and right antennae of beetle on the basis of the $\overrightarrow{s}$ by

$$\begin{aligned} \boldsymbol{x}_{right} &= \boldsymbol{x}^t + \delta^{t+1}\overrightarrow{s} \\ \boldsymbol{x}_{left} &= \boldsymbol{x}^t - \delta^{t+1}\overrightarrow{s} \end{aligned} \tag{3}$$

where

$$d^{t+1} = \delta^{t+1}/c \tag{4}$$

where δ^{t+1} is the size of the antenna of the beetle which can be decided by the size of step $\boldsymbol{d}^{t+1}$ at iteration $bmt + 1$.

$$\delta^{t+1} = \delta^t \eta \tag{5}$$

where η is a constant, it indicates the decay speed of the step when our algorithm begins its iterations, typically between 0 and 1. The c is a constant, the ratio of the step to the size of the beetle, typically 1.

B. SWARM-INTELLIGENCE SYNTHESIZATION

In our algorithm, we select the ant colony optimization algorithm (ACO) as our swarm-intelligence optimization method. ACO is a population based bio-heuristic optimization algorithm inspired by the foraging behavior of ants. In computer science and operations research, ACO is a probabilistic technique for solving computational problems which can be modeled to finding goo d paths through graphs. The BCO algorithm is proposed in this paper by combining the advantages of ACO algorithm with the random search of single-agent of beetle. So we briefly introduced how ACO algorithm works. When a colony of ants is confronted with the choice of reaching their food via two different routes of which one is much shorter than the other, their choice is entirely random. However, those who use the shorter route reach the food faster and therefore go back and forth more often between the anthill and the food. It is observed from the study that the ACO is sensitive towards the parameter evaporation rate. A proper choice may improve its performance significantly and a comprehensive comparison between ACO and other metaheuristic suggested that the performance of the improvised ACO surpasses its counterparts [30]. In the ant colony optimization algorithm, each ant needs to construct a solution to move through the graph. To select the next edge in its tour, an ant will consider the length of each edge available from its current position, as well as the corresponding pheromone level. At each step of the algorithm, each ant moves from a state $\boldsymbol{x}^t$ to state $\boldsymbol{x}^{t+1}$. Thus, each an ant computes a set of feasible expansions to its current state in each iteration, and moves to one of these in probability. In the BCO, we resorted to a random direction that is one of the direction of the largest pheromone level with certain covariance instead of probability mechanism of ACO. The search direction of the BCO algorithm is

$$\overrightarrow{s} = \frac{rand(\overrightarrow{\tau_{max}}, \sigma)}{||rand(\overrightarrow{\tau_{max}}, \sigma)||_2} \tag{6}$$

where $\overrightarrow{\tau}$ is the next movement direction of each individual of swarm-intelligence optimization algorithms, and τ represents the pheromone level of ants in ant colony optimization(ACO) algorithm. So $\overrightarrow{\tau_{max}}$ indicates the direction of the largest pheromone of each ants. The $rand(.)$ is a function that generates a random direction with $\overrightarrow{\tau_{max}}$ as the average value and the σ as the covariance, and then we normalize it as search direction of BCO algorithm. We believe that the fast search performance of the search strategy of single-agent of beetle will produce greater effect in ACO algorithm, and finally both of them have achieved a good balance to obtain better performance. In addition, the pheromone updating rule of BCO algorithm is the same as the ACO. Pheromone levels are usually updated when all ants have completed a solution or a step, increasing or decreasing the pheromone level corresponding to moves that were part of "good" or "bad" solutions, respectively. A global pheromone updating rule is

$$\tau_{\boldsymbol{x}^t\boldsymbol{x}^{t+1}} = (1-\rho)\tau_{\boldsymbol{x}^t\boldsymbol{x}^{t+1}} + \sum_k \Delta\tau^k_{\boldsymbol{x}^t\boldsymbol{x}^{t+1}} \tag{7}$$

where $\tau_{\boldsymbol{x}^t\boldsymbol{x}^{t+1}}$ is the value of pheromone deposited for a state transition $\boldsymbol{x}^t\boldsymbol{x}^{t+1}$, ρ is the pheromone evaporation coefficient and $\Delta\tau^k_{\boldsymbol{x}^t\boldsymbol{x}^{t+1}}$ is the value of pheromone deposited by kth ant, typically given for a TSP problem by

$$\Delta\tau^k_{\boldsymbol{x}^t\boldsymbol{x}^{t+1}} = \begin{cases} Q/L_k, & \text{curve } \boldsymbol{x}^t\boldsymbol{x}^{t+1} \text{in } k\text{th ant tour;} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

where L_k is the cost of the kth ant's tour (typically length) and Q is a constant.

Algorithm 1 Beetle Colony Optimization Algorithm(BCO)

Input: Path point $\boldsymbol{x}_{start}$ and $\boldsymbol{x}_{end}$.
Output: The best path ***Path***, the best fitness f_{best}.
Establish an objective function $f(\boldsymbol{x}^t)$, where variable $\boldsymbol{x}^t = [x_1, \cdots, x_k]^{\text{T}}$, initialize the parameters $\boldsymbol{x}_0, d_0, \delta_0, \eta, c, \tau_0, \sigma_0$;
for $t = 1{:}p$ **do**
  while $\boldsymbol{x}^{t+1} = \boldsymbol{x}_{end}$ **do**
    Calculate the largest pheromone of each ant τ_{max};
    Generate search direction $\overrightarrow{s}$ according to Equation (6);
    Calculate the position of right and left antenna of beetle $\boldsymbol{x}_{right}$ and $\boldsymbol{x}_{left}$ according to Equation (3);
    Calculate the value of the fitness of the two antennae of beetle $f(\boldsymbol{x}_{right})$ and $f(\boldsymbol{x}_{left})$;
    Update the position of the beetle $\boldsymbol{x}^{t+1}$ according to Equation (1);
  Save $\boldsymbol{path}_{\text{t}}$ and fitness $f(\boldsymbol{path}_{\text{t}})$ after t th iteration;
  if $f_{best} >= f(\boldsymbol{path}_t)$ **then**
    $f_{\text{best}} = f(\boldsymbol{path}_{\text{t}})$;
    $\boldsymbol{path}_{\text{best}} = \boldsymbol{path}_{\text{t}}$;
  Update $\boldsymbol{d}_t$ and δ_t according to Equations (4) and (5);
  Update pheromone τ according to Equation (7) and Equation (8);
$\boldsymbol{Path} = \boldsymbol{path}_{\text{best}}$;

In order to explain the BCO algorithm more clearly, we have detailed the steps of the BCO algorithm in Algorithm 1. The explanation of variables are as follows: $\boldsymbol{d}_0$ represents the initial exploration step size, and δ_0 represents the initial length of antennae. The optimal fitness value of the cost function is initialized to infinity, σ_0 represents initial covariance, and τ_0 indicates initial pheromone of ant colony. $\boldsymbol{x}_0$ indicates initial position of beetle. When $f(\boldsymbol{x})$ satisfies the convergence condition and the optimal fitness value $f(\boldsymbol{x}_{best})$ is saved.

C. THE ANALYSIS OF BCO ALGORITHM

In this section, the discussion on the computational complexity and limitations of the BCO algorithm are completed. The paper use the classic big oh notation to show the computation complexity of the BCO, ignoring constants such as the number of iterations and the number of groups of the BCO, so we only need to discuss the time complexity of one iteration. Assuming that the scale of the problem input is n, the scale of the problem refers to the number of cities and the dimension of a feasible solution is n in the traveling salesman problem, so a feasible solution requires n steps to complete. In each step, except for the constant term, the time complexity of taking the largest pheromone is a linear time complexity $O(n)$ and the worst time complexity of taking the random direction is also linear time complexity $O(n)$, so the time complexity of searching one step in a feasible solution is $O(n)$. In summary, the time complexity to complete a feasible solution is $O(n^2)$, so the time complexity of the BCO algorithm is $O(n^2)$. Moreover, it is very necessary and meaningful to discuss the applicability and the limitations of the BCO algorithm. The discussion is as follows. The BCO is an intelligent optimization algorithm, a global random search algorithm, which can randomly sample the feasible solutions of the optimization problem, and can rely on a powerful random search mechanism to seek a better feasible solution to the optimization objective function. Intelligent optimization algorithm can jump out of the local extremum to explore the global approximate solution by the biological heuristic behavior mechanism of algorithm. It does not require derivation of the objective function and only needs to calculate the value of the objective function, so its application is broader. However, the characteristic of its random search is also the shortcoming of the intelligent optimization algorithm, and the BCO is also similar. Because of the feature of the uncertainty of the algorithm, the BCO cannot guarantee that the two feasible solutions can keep the same fitness value, nor can it assess the degree of deviation of the feasible solution from the optimal solution.

III. VALIDATION OF BCO ALGORITHM WITH TYPICAL BENCHMARKS

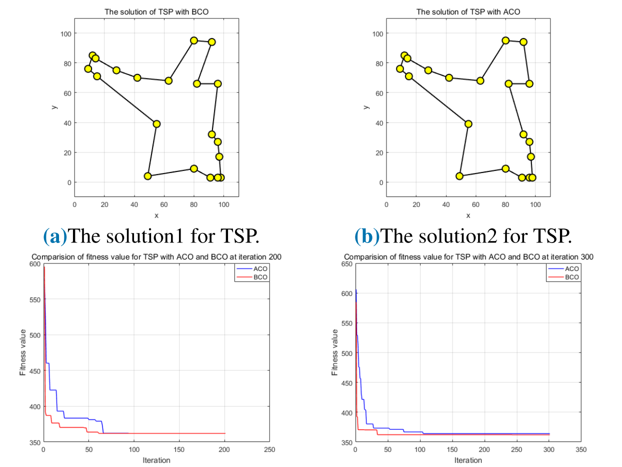
The proposed algorithm in this paper also is applied to many classic problems, such as the traveling salesman problem [39] and quadratic assignment problem [40]. In response to these classic problems, various feasible solutions have been proposed. In this section, in order to verify the effectiveness of BCO algorithm, we finished the experiments, using BCO algorithm to solve the classic traveling salesman problem and quadratic assignment problem in the Matlab R2016a environment and an Intel computer with a processing core i9.

A. TRAVELING SALESMAN PROBLEM (TSP) WITH BCO ALGORITHM

Traveling salesman problem is one of the famous problems in the field of mathematics. If a traveling businessman wants to visit n cities, he must choose a route to visit. The restriction of the chose route is that each city can only visit once, and finally the traveling businessman must return to the first city. The taken route is that the cost of distance is the minimum value among all feasible routes. In this section, ACO and BCO algorithm are tested for traveling salesman problem simultaneously. The final result is shown by Fig.1. The feasible solutions for route are shown by Fig.1a and Fig.1b. The traveling salesman problem has 20 cities to visit, so the scale of the problem input is 20 in BCO. The maximum number of iteration of each algorithm is 300, and the number of groups of ACO and BCO algorithm is 40. The single-agent step of BCO is set to 2, and the search random covariance σ is set to 19. The attenuation coefficients η is set to 1 and c is set to 1. The trend of fitness value of feasible solution with ACO

TABLE 1. Comparison of optimization results from ACO and BCO.

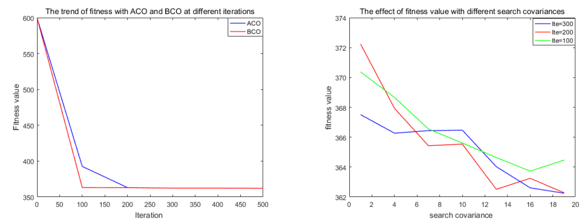
name	iteration	ave_fitness	fitness ratio	ave_time(s)	time ratio
ACO	100	392.59	100%	7.01	100%
ACO	300	362.27	92.28%	20.61	294.01%
BCO	300	362.26	92.27%	16.40	233.95%



(a)The solution1 for TSP. (b)The solution2 for TSP.

(c)The trend of fitness at iteration 200. (d)The trend of fitness at iteration 300.

FIGURE 1. The comparison of *ACO* and *BCO* algorithm in traveling salesman problem, the fitness values are 362.27 and 362.26 respectively, and the corresponding actual running time are 20.16*s*, and 16.40*s* with 300 iterations in the Matlab R2016a environment.



(a)The trend fitness value. (b)The fitness value.

FIGURE 2. The trend of fitness value of feasible solution with ACO and BCO on different iterations for TSP; The effect of fitness value for TSP on different search covariances.

and BCO algorithm for TSP is shown by Fig.1c and Fig.1d. From the experimental results, two algorithms can almost search for the best route with the minimum fitness value, but the corresponding actual running time of BCO algorithm is shorter than ACO algorithm at the same iteration. In essence, the BCO algorithm is a swarm-intelligence optimization algorithm, which draws on the pheromone mechanism of the ACO algorithm, so it can also search for the minimum fitness value like ACO algorithm. Although both the BCO and ACO algorithms are swarm optimization algorithms, each individual in the BCO algorithm uses a method of single-agent random search unlike ACO. So the search speed of the BCO algorithm is faster than ACO algorithm when the number of groups is the same at the same scale of the problem input. It can be analyzed from the experimental results of TSP problem that BCO algorithm is an effective swarm-intelligence optimization algorithm. In summary, BCO algorithm combines the advantages of swarm-intelligence optimization method and single-agent random search method and implements a delicate balancing strategy between the two. So the BCO algorithm can finish a faster decay for the solution of traveling salesman problem.

Then, experiments of ACO and BCO algorithm on different iteration are completed and data of fitness value and running time are recorded 10 times for each experiment in this paper, as shown in Table.2. After averaging for each iteration, we recorded statistical average of fitness value and running time to Table.1. The comparison of the data in the Table.1 shows that the BCO algorithm have a significant improvement in running time than ACO algorithm and they both can reach minimum fitness value for the best route. The trend of fitness value of two algorithms under different iteration is shown in Fig.2a. As shown in the Fig.2a, ACO algorithm basically can search the minimum fitness value after 200 iteration times but BCO algorithm basically can search the minimum fitness value after 100 iteration times. It further indicates that BCO algorithm possesses a rapid search speed for feasible solution of TSP. We believe single-agent random search plays an important role in BCO algorithm, and it makes BCO algorithm have extremely efficient computing ability. Moreover, the BCO algorithm has two main parameters, search step size and search random covariance. When the BCO algorithm is used to solve the TSP, the search step size doesn't work and it is set to a constant 2. Therefore, the discussion and analysis on search random covariance for performance of algorithm are meaningful and important. This paper carried out experiment to analyze the effect of feasible solution on different search random covariance in TSP and the experimental result is shown by Fig.2b. In the Fig.2b, it can be seen that different search random covariance has a different effect on feasible solution and the fitness value of solution is minimum when the search random covariance σ is set to 19. The fitness value of TSP is expressed in Euclidean distance. In conclusion, the value of search random covariance usually must be greater than 1, otherwise the algorithm cannot converge. The maximum value of search random covariance usually is set to the scale of the problem input. When search random covariance is closer to the scale of the problem input, the convergence performance of the algorithm is the better and the feasible solution is closer to the optimal solution.

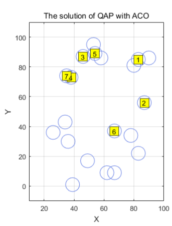
TABLE 2. **Comparison details between ACO and BCO.**

Ite	$BestFitness$		$ElapsedTime(s)$	
	ACO	BCO	ACO	BCO
100	363.1002	362.2803	7.03	5.58
	362.0380	362.2803	7.05	5.59
	365.1002	362.0380	7.04	5.53
	364.0380	362.2803	7.01	5.54
	369.8244	369.6776	6.98	5.54
	368.6596	362.0380	7.03	5.57
	362.0380	362.0380	6.80	5.55
	364.0380	362.0380	6.99	5.58
	368.4315	362.2803	7.06	5.60
	368.6596	363.8970	7.07	5.60
200	368.2077	362.0380	14.03	11.14
	362.0380	362.0380	13.76	10.97
	362.0380	362.2803	13.77	10.96
	362.0380	362.0380	13.83	11.04
	364.9591	362.0380	13.81	10.91
	362.0380	362.2803	13.85	11.04
	362.0380	368.2077	13.83	11.01
	363.1002	362.2083	13.87	10.99
	362.0380	364.1392	13.86	11.01
	362.0380	362.0380	13.83	11.03
300	362.0380	362.0380	20.61	16.41
	364.1392	362.0380	20.56	16.19
	362.0380	362.2803	20.63	16.43
	362.0380	362.0380	20.78	16.41
	362.0380	364.0380	20.56	16.40
	362.2803	362.0380	20.53	16.37
	362.0380	362.0380	20.65	16.61
	362.0380	362.0380	20.64	16.42
	362.0380	362.2803	20.60	16.34
	362.0380	362.0380	20.58	16.37

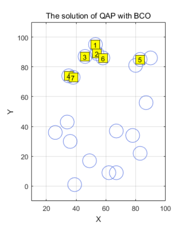
B. QUADRATIC ASSIGNMENT PROBLEM (QAP) WITH BCO ALGORITHM

In this part, ACO and BCO algorithm are tested for quadratic assignment problem simultaneously. The final result is shown by Fig.3. The quadratic assignment problem refers to assigning a set of devices to a set of nodes. Given the distance between nodes and the flow of information between devices and devices, an assignment method is sought to make the sum of the product of information flow and distance minimal. The quadratic assignment problem has 7 devices and 20 nodes, so the scale of the input of QAP is 20 in this paper. The maximum number of iterations of ACO and BCO algorithm for quadratic assignment problem are 600, and the number of groups of both ACO and BCO algorithm is 50. The reasonable step size of BCO is set to 2 and the search random covariance σ is set to 10. The attenuation coefficients η is set to 1 and c is set to 1. The Fig.3a represents a feasible solution of the ACO algorithm and Fig.3b represents a feasible solution of the BCO algorithm for QAP. The trend of fitness value of feasible solution for QAP with ACO and BCO algorithm are shown by Fig.3c and Fig.3d. From the experimental results, both two algorithms can search the feasible solution for quadratic assignment problem, but the fitness value of feasible solution with BCO algorithm is smaller than ACO algorithm throughout the search process. The fitness value is the sum of the product of information flow and distance in QAP. It can be seen from Fig.3c and Fig.3d that BCO algorithm has a significant improvement for feasible solution of QAP on the basis of ACO algorithm. Essentially, single-agent random search plays an important role on the search of feasible solution of QAP in BCO algorithm, and it makes BCO algorithm can sample more feasible solution on the basis of the pheromone mechanism. So BCO algorithm can finish a broader search for the feasible solution. From the results, It can be analyzed that BCO algorithm is an effective swarm-intelligence optimization algorithm and BCO algorithm is superior than ACO algorithm almost locally and globally at the same iteration. We believe single-agent random search mechanism improves the performances of swarm-intelligence optimization. This allows BCO algorithm can jump out of local fitness value and seek global fitness value. In summary, BCO algorithm combines the advantages of swarm-intelligence optimization method and single-agent random search method and implements a delicate balancing strategy between the two. So the BCO algorithm can finish a boarder search for the solution of less cost in quadratic assignment problem.

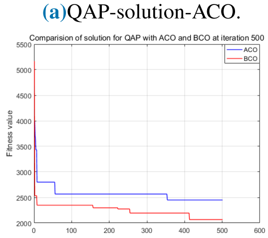
Then, this paper conducted experiments with ACO and BCO algorithm on different iteration and data of fitness value are recorded 10 times for each iteration, as shown in Table.4. After averaging for each experiment, we recorded statistical average of fitness value to Table.3. The comparison of the data in the table show that the BCO algorithm have a significant improvement than ACO algorithm in final fitness value throughout the search process. The fitness value trend of two algorithms under different iteration is shown



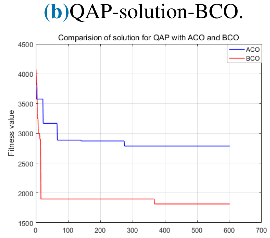
(a)QAP-solution-ACO.



(b)QAP-solution-BCO.



(c)The trend of fitness at iteration 500.

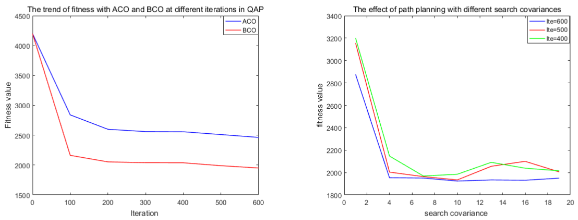


(d)The trend of fitness at iteration 600.

FIGURE 3. **The comparison of *ACO*, *BCO* algorithm for quadratic assignment problem, the final fitness values are 2462.87 and 1950.82 respectively at iteration 600.**

TABLE 3. **Comparison of optimization results between ACO and BCO.**

name	iteration	ave_fitness	fitness ratio
ACO	400	2557.29	100%
ACO	600	2462.87	96.31%
BCO	600	1950.82	76.28%



(a)The trend of fitness value. (b)The fitness value.

FIGURE 4. The trend of fitness value for QAP with ACO and BCO on different iterations; The effect of fitness value for QAP on different search covariances.

TABLE 4. Comparison details between ACO and BCO.

Ite	400		500		600	
	ACO	BCO	ACO	BCO	ACO	BCO
Fitness	2622.6	1947.9	2672.0	1978.9	2666.0	1862.0
	2696.1	2034.7	2573.5	1868.5	2217.0	1905.6
	2559.5	1950.8	2546.7	1981.1	2466.3	1955.0
	2879.4	2067.1	2531.1	2037.5	2360.8	2122.2
	2281.8	1928.8	2348.9	2095.9	2283.1	1947.2
	2770.9	2234.6	2635.3	2322.3	2580.0	2247.6
	2315.4	2210.2	2522.9	1837.7	2038.1	1815.5
	2558.6	1860.6	2438.9	2100.6	2812.9	1896.2
	2294.3	2225.6	2321.3	1799.9	2741.0	1997.7
	2594.3	1921.3	2514.7	1873.4	2463.5	1759.2

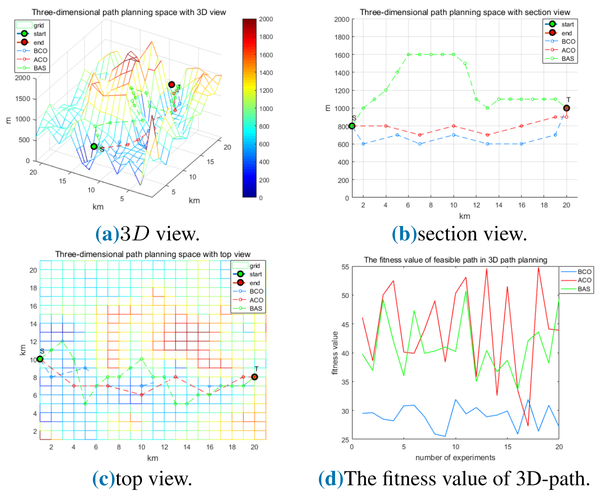
in Fig.4a. As shown in the Fig.4a, both ACO and BCO algorithm basically search the global approximate solution after 200 iteration, while BCO algorithm has a faster decay to reach smaller fitness value for the assignment solution. This shows that BCO algorithm have extremely efficient computing ability. It further indicates that BCO algorithm possesses a broad search space for feasible solution of QAP. In addition, when the BCO algorithm is used to solve the QAP, the search step size doesn't work and it is taken as a constant 2. Therefore, the discussion and analysis on search random variance for performance of algorithm are essential. This paper carried out experiment to analyze the effect of feasible solution on different search random covariance in QAP and the experimental result is shown by Fig.4b. In the Fig.4b, it can be seen that different search random covariance has a different effect on feasible solution and the fitness value of feasible solution is global approximate solution when the search random variance σ is bigger than 4. In conclusion, the value of search random covariance usually must be greater than 1, otherwise the algorithm cannot converge. The maximum value of search random variance usually is set to the scale of the problem input. When search random variance is closer to the size of the scale of the problem, the convergence performance of BCO algorithm is the better and the feasible solution is closer to the global optimal solution.

IV. THE VALIDATION AND APPLICATION OF BCO FOR PATH PLANNING

In this section, we will apply BCO algorithm for path planning of unmanned aerial vehicles(UAV). Compared with ACO and BAS algorithm, BCO algorithm has advantages of a wide search horizon and high search efficiency, which resolves the high computational load of heuristic algorithms and the real-time path planning of UAV to a certain extent. This section includes two branches, the validation of BCO algorithm for path planning in 3D simulation and the application of BCO in 3D real environment.

A. THE VALIDATION OF BCO FOR PATH PLANNING IN 3D SIMULATION SITUATION

In this part, the 3D simulation situation is modeled in the Matlab R2016a environment and BCO, ACO and BAS algorithm for path planning are tested at the same time in 3D simulation situation. The simulation environment is a three-dimensional grid map and Cartesian coordinate system is established in three-dimensional grid map. The distance scale of a grid in the horizontal direction is $1km$ and the distance scale of a grid in the vertical direction is $100m$. The size of the three-dimensional grid map is $20 * 20 * 20$. The final result is shown by Fig.5. The green point is the starting point and its coordinate is $(1, 10, 8)$ in 3D simulation grid map. The red point is the end and its coordinate is $(20, 8, 10)$ in 3D simulation grid map. Euclidean distance between start point and end point is 19.21. The number of iteration of BCO, ACO and BAS algorithm are 2, 20, 500 respectively. The population sizes of the three algorithm are all 40. The fitness value of path is expressed by Euclidean distance. The initial step size is set to 3 and search random covariance σ is set to 3 in BCO algorithm. The attenuation coefficients η is 0.998 and c is set to 1. The Fig.5a shows three feasible paths with three algorithms in 3D simulation grid map by three-dimensional view. The Fig.5b and Fig.5c show the feasible solution of path planning with three algorithm in 3D simulation grid map by Y-axis section view and top vertical view respectively. The blue path represents the feasible solution with BCO algorithm. The red path represents the feasible solution with



(a)$3D$ view. (b)section view.
(c)top view. (d)The fitness value of 3D-path.

FIGURE 5. The comparison of *BCO*, *ACO* and *BAS* algorithm in *3D* path planning. The average fitness value of path are 28.90, 44.04 and 41.05 respectively; The average running time are 0.31*s*, 2.05*s* and 0.49*s* separately.

TABLE 5. Comparison of feasible path with three algorithm.

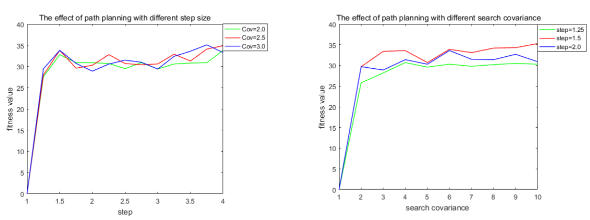
name	ave_fitness	fitness ratio	ave_time(s)
ACO	44.04	220.2%	2.0s
BAS	41.05	205.3	0.49s
BCO	28.90	144.5%	0.31s

ACO algorithm and the green path represents the feasible solution with BAS algorithm. The Fig.5d represents the fitness value of the feasible path with three algorithm. The best path of the experiment is along the right side edge of the canyon and the fitness value, distance of the path is minimal. We repeated the experiment 20 times with three algorithms simultaneously and recorded the data to Table.5 after averaging 20 calculations. The average fitness value of BCO, ACO and BAS algorithm are 28.90, 44.04 and 41.05 respectively. The average running time of BCO, ACO and BAS algorithm are 0.31s, 2.05s and 0.49s respectively. From the experimental results, three algorithms all can search a feasible solution for path planning of 3D. It can evidently be seen that BCO algorithm is superior to the ACO and BAS algorithms in terms of final fitness value and actual running time. Compared with ACO algorithm, higher search efficiency of BCO algorithm can satisfy the real-time path planning of unmanned aerial vehicles, almost under 0.30s and the feasible solution of the BCO algorithm is obviously better than ACO. Compared with BAS algorithm, the feasible solution of the BCO algorithm is obviously better than BAS. Undoubtedly, BAS algorithm has a faster search performance and higher efficiency than ACO algorithm. Single-agent search strategy makes the BCO algorithm has a significant improvement in running time for path planning. A delicate balancing strategy of swarm-intelligence optimization and single-agent random search makes BCO algorithm can finish a broader search for path of less cost. In conclusion, the BCO algorithm can be applied for path planning of unmanned aerial vehicles and it has faster search performance and higher efficiency than ACO and BAS algorithm.

In the research and solution of the three-dimensional path planning problem, this paper conducted experimental analysis and discussion on the BCO algorithm's main parameter, search step size and random covariance. As shown in Fig6a, we test a different search step size 20 times when search random covariance σ are 2.0, 2.5 and 3.0 separately. The range of the search step size is from 1.0 to 4.0. It can be seen that different search step size has different effect on the fitness value of path planning and the average distance of all feasible paths is minimal when the step size is set to 2.0. No feasible path when the step size is 1.0. Meanwhile, as shown in Fig 6b, we test a different search random covariance 20 times when search step size are 1.25, 1.5 and 2.0 separately. The range of the search random covariance is from 1 to 10. It can be seen that different search covariance has different effect on the fitness value of path planning and the average distance of all feasible paths is minimal when the search random covariance is set to 3. No feasible path When the search covariance is 1. In conclusion, the search step size must be greater than 1, otherwise the algorithm cannot converge. When the search step size is too large, the algorithm's convergence performance will seriously deteriorate, so you can take a large step size at the beginning of the algorithm. The appropriate initial step size generally depends on the specific problem. In the three-dimensional grid map, the maximum number of grids for each step of a single-agent search is 2. So the maximum search step size is usually twice the maximum number of grids for each step of a single-agent search. The value of search random variance usually must be greater than 1, otherwise the algorithm cannot converge. The maximum value of search random variance usually is set to the scale of the problem input. In 3D simulation situation, the scale of the problem input is 20. We believe that single-agent random search and swarm-intelligence optimization reached a balance when the search step size is set to 2 and the search random covariance is set to 3.

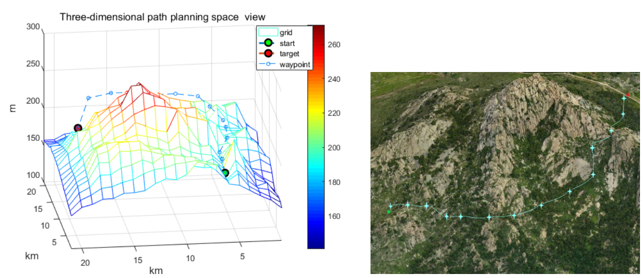
B. THE APPLICATION OF BCO FOR 3D PATH PLANNING

In order to verify the effectiveness of the algorithm for 3D path planning, we conducted experiments in real 3D image maps. First, we extracted the real-world 3D image data and imported it into MATLAB 2016a for modeling and path planning experiments with BCO algorithm. We tested 20 times and record it. We use the generated offline path as the drone's track in the real world. The final result is shown by Fig7b and Fig7a. The Fig7b is the path for our real 3D image maps with BCO algorithm and Fig7a is the track of UAV with path of BCO algorithm in real 3D image maps. The experiment



(a)The effect of path planning with different step size.
(b)The effect of path planning with different search covariance.

FIGURE 6. The effect of feasible solution with different BCO algorithm's main parameter.



(a)3D view.
(b)the track of UAV in real-world.

FIGURE 7. Experiment validation with BCO for 3D path planning.

show that the BCO algorithm can be fully applied to path planning of UAV and the algorithm has a fast search speed and real-time characteristics. Compared with other heuristic algorithms, our algorithm has advantages of a wide search horizon and high search efficiency, which can solve the high computational load of heuristic algorithms and the real-time path planning of UAV.

V. CONCLUSION

This paper has proposed a novel heuristic swarm optimization algorithm, called BCO algorithm. Two important mechanisms are introduced into BCO algorithm. The first one is single-agent random search of a beetle, which is a mechanism of random search along the gradient direction but not use gradient information at all. The second one is swarm intelligence optimization, which is a collective mechanism of decentralized and self-organized agents. Meanwhile, BCO is applied to solve traveling salesman problem and quadratic assignment problem, and demonstrates excellent performance and high efficiency, which also show that BCO algorithm has good applicability. The validation and effectiveness of BCO algorithm is also substantiated by comparing feasible solution of path planning with ACO and BAS algorithm in three-dimensional grid map.

REFERENCES

[1] S. Ghambari, J. Lepagnot, L. Jourdan, and L. Idoumghar, "A comparative study of meta-heuristic algorithms for solving UAV path planning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 174–181.

[2] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, pp. 270–299, Jan. 2020.

[3] L. Zhang, Q. Zhang, B. Du, X. Huang, Y. Y. Tang, and D. Tao, "Simultaneous spectral-spatial feature selection and extraction for hyperspectral images," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 16–28, Jan. 2018.

[4] A. Bundy and W. Lincoln, "Breadth-first search," in *Catalogue of Artificial Intelligence Tools*. Berlin, Germany: Springer, 1984, p. 13.

[5] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, Jun. 1972.

[6] H. I. Kang, B. Lee, and K. Kim, "Path planning algorithm using the particle swarm optimization and the improved Dijkstra algorithm," in *Proc. IEEE Pacific-Asia Workshop Comput. Intell. Ind. Appl.*, Dec. 2008, pp. 1002–1004.

[7] J. Yao, C. Lin, X. Xie, A. J. Wang, and C.-C. Hung, "Path planning for virtual human motion using improved A* star algorithm," in *Proc. 7th Int. Conf. Inf. Technol., New Generat.*, Apr. 2010, pp. 1154–1158.

[8] J. Carsten, D. Ferguson, and A. Stentz, "3D field D: Improved path planning and replanning in three dimensions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3381–3386.

[9] L. De Filippis, G. Guglieri, and F. Quagliotti, "Path planning strategies for UAVS in 3D environments," *J. Intell. Robotic Syst.*, vol. 65, nos. 1–4, pp. 247–264, Jan. 2012.

[10] R. Song, Y. Liu, and R. Bucknall, "Smoothed A* algorithm for practical unmanned surface vehicle path planning," *Appl. Ocean Res.*, vol. 83, pp. 9–20, Feb. 2019.

[11] F. Yan, Y.-S. Liu, and J.-Z. Xiao, "Path planning in complex 3D environments using a probabilistic roadmap method," *Int. J. Autom. Comput.*, vol. 10, no. 6, pp. 525–533, Dec. 2013.

[12] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," *Annu. Res. Rep.*, pp. 1–11, 1998.

[13] B. D. Tenezaca, C. Canchignia, W. Aguilar, and D. Mendoza, "Implementation of dubin curves-based RRT* using an aerial image for the determination of obstacles and path planning to avoid them during displacement of the mobile robot," in *Proc. Multidisciplinary Int. Conf. Res. Appl. Defense Secur.*, vol. 152, 2020, pp. 205–215.

[14] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Auto. Robots*, vol. 13, no. 3, pp. 207–222, 2002.

[15] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, no. 1, pp. 122–128, Jan. 1986.

[18] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Boston, MA, USA: Springer, 2011, pp. 760–766.

[19] M. Dorigo and D. C. Gianni, "Ant colony optimization: A new meta-heuristic," in *Proc. Congr. Evol. Comput.-CEC*, vol. 2, Jul. 1999, pp. 1470–1477.

[20] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, "Artificial fish swarm algorithm: A survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artif. Intell. Rev.*, vol. 42, no. 4, pp. 965–997, Dec. 2014.

[21] Q. Wu, Z. Ma, G. Xu, S. Li, and D. Chen, "A novel neural network classifier using beetle antennae search algorithm for pattern classification," *IEEE Access*, vol. 7, pp. 64686–64696, 2019.

[22] Q. Wu, X. Shen, Y. Jin, Z. Chen, S. Li, A. H. Khan, and D. Chen, "Intelligent beetle antennae search for UAV sensing and avoidance of obstacles," *Sensors*, vol. 19, no. 8, p. 1758, Apr. 2019.

[23] V. Kroumov, J. Yu, and K. Shibayama, "3D path planning for mobile robots using simulated annealing neural network," *Int. J. Innov. Comput. Inf. Control.*, vol. 6, no. 7, pp. 2885–2899, 2010.

[24] C. Xu, H. Duan, and F. Liu, "Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning," *Aerosp. Sci. Technol.*, vol. 14, no. 8, pp. 535–541, Dec. 2010.

[25] H. Duan, Y. Yu, X. Zhang, and S. Shao, "Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm," *Simul. Model. Pract. Theory*, vol. 18, no. 8, pp. 1104–1115, Sep. 2010.

[26] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[27] Q. Wu, H. Lin, Y. Jin, Z. Chen, S. Li, and D. Chen, "A new fallback beetle antennae search algorithm for path planning of mobile robots with collision-free capability," *Soft Comput.*, vol. 24, no. 3, pp. 2369–2380, Feb. 2020.

[28] X. Li, R. X. Wei, J. Zhou, X. S. Li, and C. Zhang, "A three dimensional path planning for unmanned air vehicle based on improved genetic algorithm," *J. Northwestern Polytechnical Univ.*, vol. 28, no. 3, pp. 343–348, 2010.

[29] M. Panda, B. Das, B. Subudhi, and B. B. Pati, "A comprehensive review of path planning algorithms for autonomous underwater vehicles," *Int. J. Autom. Comput.*, vol. 17, no. 3, pp. 321–352, Jun. 2020.

[30] V. K. Ojha, A. Abraham, and V. Snasel, "ACO for continuous function optimization: A performance analysis," in *Proc. 14th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2014, pp. 145–150.

[31] D. Pal, P. Verma, D. Gautam, and P. Indait, "Improved optimization technique using hybrid ACO-PSO," in *Proc. 2nd Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Oct. 2016, pp. 277–282.

[32] W.-H. Tan and J. Mohamad-Saleh, "Normative fish swarm algorithm (NFSA) for optimization," *Soft Comput.*, vol. 24, no. 3, pp. 2083–2099, Feb. 2020.

[33] J. Liu, Y. Mao, X. Liu, and Y. Li, "A dynamic adaptive firefly algorithm with globally orientation," *Math. Comput. Simul.*, vol. 174, pp. 76–101, Aug. 2020.

[34] X. Jiang and S. Li, "BAS: Beetle antennae search algorithm for optimization problems," 2017, *arXiv:1710.10724*. [Online]. Available: http://arxiv.org/abs/1710.10724

[35] G. G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019.

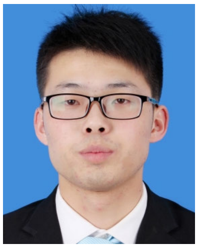
[36] T. Wang and L. Yang, "Beetle swarm optimization algorithm:Theory and application," 2018, *arXiv:1808.00206*. [Online]. Available: http://arxiv.org/abs/1808.00206

[37] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Comput.*, vol. 22, no. 9, pp. 2935–2952, May 2018.

[38] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.

[39] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *Transp. Sci.*, vol. 52, no. 4, pp. 965–981, Aug. 2018.

[40] R. Poveda and J. Gómez, "Solving the quadratic assignment problem (QAP) through a fine-grained parallel genetic algorithm implemented on GPUs," in *Proc. Int. Conf. Comput. Collective Intell. (ICCCI)*, vol. 11056, 2018, pp. 145–154.



HENG ZHANG received the B.E. degree in automation from Shandong University, Jinan, China, in 2018. He is currently pursuing the M.S. degree in control science and engineering from Shandong University, Qingdao, China.

ZIMING LI is currently pursuing the B.E. degree in computer science with the State University of New York at Albany, Albany, NY, USA.

XIANGYUAN JIANG received the B.E. degree in biomedical engineering and the Ph.D. degree in control science and control engineering from Shandong University, China, in 2006 and 2012, respectively. He is currently an Associate Professor with the Institute of Marine Science and Technology, Shandong University, China. His current research interests include adaptive control of source seeking and distributed estimation and control.

XIAOJING MA received the B.E. and M.S. degrees from the School of Control Science and Engineering, Shandong University, China, where she is currently pursuing the Ph.D. degree.

JIYANG CHEN received the B.E. and M.S. degrees from the School of Control Science and Engineering, Shandong University, China, in 2013 and 2017, respectively. He is currently working as an Assistant Researcher at Shandong University.

SHUAI LI (Senior Member, IEEE) received the B.E. degree in precision mechanical engineering from the Hefei University of Technology, Hefei, China, in 2005, the M.E. degree in automatic control engineering from the University of Science and Technology of China, Hefei, in 2008, and the Ph.D. degree in electrical and computer engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2014. He is currently an Associate Professor (Reader) at Swansea University, U.K., leading the Robotic Lab, conducting research on robot manipulation and impedance control, multi-robot coordination, distributed control, intelligent optimization and control, and legged robots. He is the General Co-Chair of 2018 International Conference on Advanced Robotics and Intelligent Control and the Founding Editor-in-Chief of *International Journal of Robotics and Control*.

YIZHONG LUAN received the B.E. degree in power system and automation from the China University of Mining and Technology, China, in 1997, and the M.S. degree in control engineering from Shandong University, in 2006. He is currently an Associate Research Fellow with the Institute of Marine Science and Technology, Shandong University, China. His current research interests include industrial process control and computer intelligent detection and control.

ZHENYI LV received the B.E. degree in radio technology from Shandong University, in 2006, China, in 1982. He is currently a member of the Academic Committee of the Forest Fire Prevention Engineering Technology Research Center of the Forestry Bureau, China. His current research interests include unmanned aerial vehicle and forest fire prevention.

SILE MA received the B.E. degree in automation engineering from Beijing Technology and Business University, China, in 1985, and the Ph.D. degree in control science and control engineering from Shandong University, in 2014. He was an Associate Professor with the Department of Automation, Shandong Industrial University, China. He is currently a Professor at the Institute of Marine Science and Technology, Shandong University, China. His current research interests include industrial process control and distributed control for industrial manipulators.