Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 6469721, 12 pages
http://dx.doi.org/10.1155/2016/6469721

Research Article

Parameter Selection for Ant Colony Algorithm Based on Bacterial Foraging Algorithm

Peng Li and Hua Zhu

School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China

Correspondence should be addressed to Hua Zhu; zhuhua83591917@163.com

Received 8 October 2016; Revised 28 November 2016; Accepted 6 December 2016

Academic Editor: Francisco Valero

The optimal performance of the ant colony algorithm (ACA) mainly depends on suitable parameters; therefore, parameter selection for ACA is important. We propose a parameter selection method for ACA based on the bacterial foraging algorithm (BFA), considering the effects of coupling between different parameters. Firstly, parameters for ACA are mapped into a multidimensional space, using a chemotactic operator to ensure that each parameter group approaches the optimal value, speeding up the convergence for each parameter set. Secondly, the operation speed for optimizing the entire parameter set is accelerated using a reproduction operator. Finally, the elimination-dispersal operator is used to strengthen the global optimization of the parameters, which avoids falling into a local optimal solution. In order to validate the effectiveness of this method, the results were compared with those using a genetic algorithm (GA) and a particle swarm optimization (PSO), and simulations were conducted using different grid maps for robot path planning. The results indicated that parameter selection for ACA based on BFA was the superior method, able to determine the best parameter combination rapidly, accurately, and effectively.

1. Introduction

In the 1990s, Dorigo et al. [1] were inspired by the foraging behavior of ants and proposed an ant colony algorithm (ACA) that had the characteristics of strong robustness, a high degree of parallelism, and positive feedback. The ACA demonstrates high effectiveness and superiority in global optimization and in solving the traveling salesman [2, 3], shop scheduling [4–7], and robot path planning [8–10] problems.

However, the optimal performance and efficiency of the ACA are closely related to the chosen parameters that include the information heuristic factor α, the expectation heuristic factor β, the pheromone evaporation factor ρ, the pheromone strength Q, and the number of ants M [11]. Parameter selection differs for different types of optimization problems. Moreover, even for the same type of optimization problem, parameters may change due to the different scales of the problem. Traditional parameter selection methods for the ACA include the empirical selection method, the trial and error method, and the orthogonal experimental design method [12]. The empirical selection method requires a priori knowledge of the problem in question, and the method requires the need to conduct multiple experiments, consuming considerable time and effort, rendering it difficult to obtain the optimum parameters. Hei and Du [13] applied the trial and error method, which required a large number of digital simulations to obtain the parameters, and the authors determined that it was difficult to guarantee that the obtained parameters were optimal. Gan and Li [14] studied parameter selection for ACA based on the orthogonal experimental design method, which ignores coupling between the parameters. This method is computationally intensive, which is a disadvantage. Traditional parameter selection methods for the ACA are mainly based on a large number of replicated experiments, which is inefficient and time consuming. Moreover, traditional parameter selection methods ignore the coupling between the parameters, making it difficult to obtain the optimal parameters.

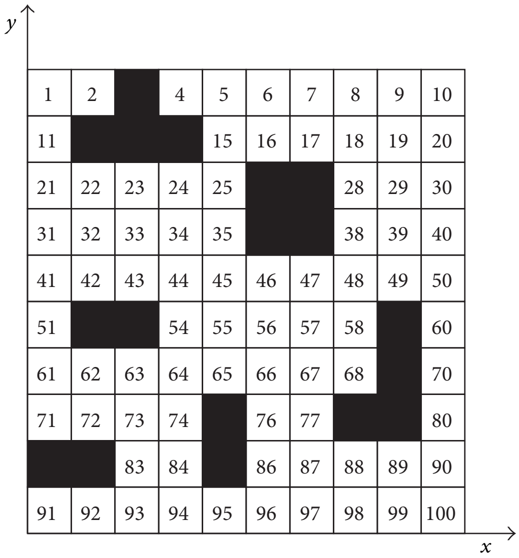
In the ACA, the parameters are coupled with each other, and it is difficult to achieve the best performance for the algorithm by adjusting only a single parameter. Therefore, it is common to adjust multiple parameters simultaneously to determine the optimal parameter combination. Recently, scholars who have considered the parameter selection in the ant colony algorithm as an optimization problem have applied intelligent algorithms to solve the parameter selection problem in the ACA. Feng [15] proposed a parameter selection method based on a GA. Although this method is effective with regard to global optimization, it had the disadvantage of slow convergence speed. Li et al. [16] applied a PSO, which can avoid subjectivity in parameter selection and obtain optimum parameters rapidly. In spite of that, this method is prone to premature convergence and falling into a local optimal solution. Although GA and PSO are suitable for obtaining ACA parameters quickly and accurately, due to the disadvantages of the methods, the parameter selection can be improved upon. Therefore, a suitable intelligent algorithm that can optimize globally is required to obtain the optimal parameters for the ACA.

The bacterial foraging algorithm (BFA) [17] has a fast convergence speed and searches objects in parallel. The algorithm includes a chemotaxis operator, a reproduction operator, and an elimination-dispersal operator. In addition, the BFA is able to jump out of the local optimal solution. Therefore, the BFA has shown good adaptability for solving the problems of job shop scheduling [18], robot path planning [19], image processing [20], and high dimensional optimization [21]. In this study, ACA parameters are mapped into a multidimensional space, and a chemotactic operator is used to enable each group of parameters to approach the optimal value and speed up the convergence of each set of parameters. A reproduction operator is used to accelerate the optimization for the entire set of parameters. Lastly, the elimination-dispersal operator is utilized to strengthen the global optimization ability of the parameters to avoid falling into a local optimal solution. The BFA is commonly applied to solve the problem of parameter selection for ACA and is able to take into account the coupling between parameters, which allows for adjusting the parameters automatically with little prior knowledge. Therefore, it is appropriate to apply the BFA to obtain the ACA parameters.

2. Model and Parametric Analysis of the Ant Colony Algorithm

The ACA described in this paper will be applied to robot path planning. Generally, environmental modeling is an important aspect of path planning.

2.1. Environmental Modeling. By the direct encoding format for the data, this study uses the grid method [22] for environmental modeling. XY is described as a convex polygon, an area of limited motion that contains several static obstacles in a two-dimensional plane. A Cartesian coordinate system is established for XY. Figure 1 shows a grid sequence diagram with a size of 10×10, and the black grids in the figure represent obstacles. v is an arbitrary grid within XY, V is the sum of all grids, and, therefore, $v \in V$. $v(x, y)$ are the coordinates of a grid, where its center point coordinates are x and y. Suppose that $N = \{1, 2, \ldots, n\}$ is a set of grid sequence number, and this study regards a grid i $(i \in N)$ as $v_i(x_i, y_i)$.



1	2	■	4	5	6	7	8	9	10
11	■	■	■	15	16	17	18	19	20
21	22	23	24	25	■	■	28	29	30
31	32	33	34	35	■	■	38	39	40
41	42	43	44	45	46	47	48	49	50
51	■	■	54	55	56	57	58	■	60
61	62	63	64	65	66	67	68	■	70
71	72	73	74	■	76	77	■	■	80
■	■	83	84	■	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Figure 1: Grid sequence diagram.

2.2. Model of Ant Colony Algorithm

2.2.1. Rules of Path Selection. $p_{ij}^{m}(t)$ is defined as the transition probability that ant m walks forward from v_i to v_j, which can be expressed as follows [23]:

$$p_{ij}^{m}(t) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}(t)\right]^{\beta}}{\sum_{v_m \in V_{\text{allowed}}^{m}} \left[\tau_{im}(t)\right]^{\alpha} \cdot \left[\eta_{im}(t)\right]^{\beta}}, & \text{if } v_j \in V_{\text{allowed}}^{m}, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where V_{allowed}^{m} is a set of the next feasible grid when ant m arrives at v_i; $\tau_{ij}(t)$ is the residual pheromones between v_i and v_j at t time; α represents the information elicitation factor, which shows the relative importance of $\tau_{ij}(t)$; $\eta_{ij}(t)$ is the expectation heuristic function between v_i and v_j at t time, which can be defined as the reciprocal of the distance d_{ij} between v_i and v_j, namely, $\eta_{ij}(t) = 1/d_{ij}$; β is the expected heuristic factor, which shows the relative importance of $\eta_{ij}(t)$.

2.2.2. Rules of Pheromone Updating. Ants will leave pheromones in their path, and the pheromones will evaporate over time to avoid the masking of heuristic information due to excessive residual pheromones. Supposing that ρ $(0 \leq \rho \leq 1)$ is the pheromone evaporation coefficient, the rules of pheromone updating between v_i and v_j at $t + \Delta t$ time can be expressed as follows [23]:

$$\begin{gathered} \tau_{ij}(t + \Delta t) = (1 - \rho) \cdot \tau(t) + \Delta\tau_{ij}(t), \\ \Delta\tau_{ij}(t) = \sum_{m=1}^{M} \Delta\tau_{ij}^{m}(t), \end{gathered} \quad (2)$$

where $\Delta\tau_{ij}^{m}(t)$ is the pheromone that ant m leaves between v_i and v_j, and this paper applies the Ant-Cycle model [1] proposed by Dorigo et al., which can be expressed as follows:

$$\Delta\tau_{ij}^{m}(t)=\begin{cases}\dfrac{Q}{L_m}, & v_i, v_j \in \mathrm{Path}_m,\\ 0, & \text{otherwise},\end{cases} \tag{3}$$

where Q is the pheromone intensity; L_m is the total length of path traversed by ant m; Path_m is the set of grids traversed by ant m.

2.3. Parametric Analyses of the Ant Colony Algorithm. The parameters for ACA are extremely important to the performance of the algorithm, and appropriate parameters can enhance the global search ability of the algorithm and improve the convergence speed considerably [24]. From the above formulas, it is evident that the parameters with a significant impact on the path selection of the ant m are α, β, ρ, and Q. We will elaborate further on the relationship between the four parameters and the performance of the ACA.

2.3.1. Information Elicitation Factor α. The information elicitation factor α, which represents the relative importance of the pheromone, reflects the importance of the accumulation of the pheromone with regard to the ants' path selection. If α is very large, the ants tend to choose the same path that the preceding ants have chosen, resulting in stronger cooperation among the ants. Although the convergence speed of ACA will be accelerated, it is likely for the algorithm to fall into the local optimal solution and reduce the global search ability. Conversely, if α is too small, the convergence speed of the ACA is slowed down, regardless of the fact that the global search ability of the algorithm can be improved.

2.3.2. Expected Heuristic Factor β. The expected heuristic factor β, which represents the relative importance of the visibility, reflects the importance of the heuristic information with regard to the ants' path selection. If the value is very large, the probability of a state transition is close to that of a greedy algorithm. If β is too small, the heuristic information has virtually no effect on path selection, which leads ACA to fall into stagnation or a local optimum.

2.3.3. Pheromone Evaporation Coefficient ρ. The pheromone evaporation coefficient ρ, which represents the degree of pheromone evaporation, reflects the degree of mutual influence among ants. Generally, the value of ρ is $[0,1]$, which prevents the infinite accumulation of pheromone effectively. If ρ is too small, the global search ability of ACA will be reduced. Otherwise, if ρ is too large, it will improve the global search ability of ACA; however, the convergence speed will be slow.

2.3.4. Pheromone Intensity Q. The pheromone intensity Q, which represents the total pheromone, affects the convergence speed of the ACA to a certain extent. If Q is large, the pheromone concentration will be highly concentrated, which leads the algorithm to fall into a local optimum. Furthermore, a small Q results in a slow optimization speed.

3. Parameters Selection of ACA Based on BFA

In 2002, Passino put forward a bacterial foraging algorithm [17], based on the foraging behavior of *Escherichia coli* in the human intestinal tract. This algorithm includes a chemotaxis operator, a reproduction operator, and an elimination-dispersal operator. The algorithm is able to search in parallel and jump out of the local minima easily. In this paper, we will transfer any set of parameters $(\alpha_i, \beta_i, \rho_i, Q_i)$ to four-dimensional arrays, which are seen as an individual bacteria p_i, where $p_i = (\alpha_i, \beta_i, \rho_i, Q_i)^T$. P is a bacterial population, and $P = \{p_1, p_2, \ldots, p_n\}$. Because the application of the algorithms discussed in this paper is aimed at the problem of robot path planning in a grid map environment, the objective of this study is the selection of optimal parameters for the ACA. path_i is the path for a robot under the individual bacteria p_i, and $\mathrm{path}_i \in \mathrm{PATH}$. Therefore, the fitness function of BFA can be expressed as follows:

$$f(p_i)=\mathrm{Length}(\mathrm{path}_i), \tag{4}$$

where $\mathrm{Length}(*)$ is adopted to calculate the length of a path.

3.1. Chemotaxis Operator T_c^B. The chemotaxis operator represents the core of the BFA, and it determines changes in the location of the bacteria that are searching for food. Therefore, the chemotaxis operator plays a decisive role in finding a food source for the bacteria, which has an important influence on the performance and convergence of the algorithm. $P(m)$ is the bacterial population, and $P(m) = [p_1(m)\ \ p_2(m)\ \ \cdots\ \ p_n(m)]^T$. And the chemotaxis operator can be expressed as follows:

$$\begin{aligned}P_c^B(m)&=T_c^B(P(m))\\&=\left[p_1^{Bc}(m)\ \ p_2^{Bc}(m)\ \ \cdots\ \ p_n^{Bc}(m)\right]^T\end{aligned} \tag{5}$$

$\forall i \in [1,n]$, $p_i^{Bc}(m) = T_c^B(p_i(m))$.

Supposing that $p_i(m) = p_i(m,j,k,l)$ and $p_i^{Bc}(m) = p_i(m,j+1,k,l)$, this paper adjusts the position of a bacteria in accordance with the formula (6), which can be expressed as follows [25]:

$$p_i(m,j+1,k,l)=p_i(m,j,k,l)+\mathrm{Step}\times\varphi(i), \tag{6}$$

$$\varphi(i)=\frac{p_i(m,j,k,l)-p_{\mathrm{rand}}(m,j,k,l)}{\mathrm{sqrt}\left((p_i(m,j,k,l)-p_{\mathrm{rand}}(m,j,k,l))^T\times(p_i(m,j,k,l)-p_{\mathrm{rand}}(m,j,k,l))\right)}, \tag{7}$$

where $p_i(m,j,k,l)$ represents the current location of the bacterial individual i; j is the number of the chemotaxis operator; k is the number of the reproduction operator; l is the number of the elimination-dispersal operator; Step represents the step that the bacteria moves forward each time; $\varphi(i)$ is the direction of random tumbling; $p_{\text{rand}}(m,j,k,l)$ represents a random position in the neighborhood of the current individual.

In the chemotaxis operator, the movement patterns of the bacteria include flipping and moving. The bacteria move forward in any direction with a unit step, which is defined as the flipping operator. After $p_i(m)$ executes the flipping operator, namely, $p_i^{Bc}(m) = T_c^B(p_i(m))$, and the fitness value of $p_i(m)$ is not improved, namely, $f(p_i^{Bc}(m)) > f(p_i(m))$, $p_i(m)$ jumps out of the loop. If the fitness value of $p_i(m)$ is improved, namely, $f(p_i^{Bc}(m)) < f(p_i(m))$, $p_i(m)$ keeps moving in the same direction until the fitness value cannot be improved further, or the algorithm achieves the maximum number of flipping, which is defined as the moving operator.

3.2. Reproduction Operator T_r^B. The reproduction operator ensures the improvement in the performance of the bacterial population, which encourages the population to move towards the optimal direction. The reproduction operator is conducive to achieving the global optimal, which can be expressed as follows:

$$\begin{aligned} P_r^B(m) &= T_r^B\left(P_c^B(m)\right) \\ &= \left[p_1^{Br}(m) \;\; p_2^{Br}(m) \;\; \cdots \;\; p_n^{Br}(m)\right]^T. \end{aligned} \tag{8}$$

Supposing that $P_c^B(m)_{\text{value}}$ is the fitness value of the bacterial population after performing the chemotaxis operator, which can be expressed as follows:

$$P_c^B(m)_{\text{value}} = f\left(P_c^B(m)\right) = \text{Length}\left(\text{PATH}_c^B(m)\right). \tag{9}$$

With the standard of $P_c^B(m)_{\text{value}}$, this paper gets $P_c^B(m)_{\text{bad}}$ that is half of the bacterial population with a bad fitness value and $P_c^B(m)_{\text{good}}$ that represents half of the bacterial population with a good fitness value. Therefore, $P_c^B(m) = P_c^B(m)_{\text{bad}} \cap P_c^B(m)_{\text{good}}$. Then, this paper replaces $P_c^B(m)_{\text{bad}}$ with $P_c^B(m)_{\text{good}}$ to get $P_r^B(m)$, and $P_r^B(m) = \left[P_c^B(m)_{\text{good}} \;\; P_c^B(m)_{\text{good}}\right]$.

3.3. Elimination-Dispersal Operator T_e^B. The elimination-dispersal operator generates a new individual with a certain probability, which has a different location compared to the dead individual. The operator has a promoting effect on the algorithm, because the new individual may be closer to the food source, which is more conducive to jumping out of the local optimal solution and finding the global optimal solution. The elimination-dispersal operator can be expressed as follows:

$$\begin{aligned} P_e^B(m) &= T_e^B\left(P_r^B(m)\right) \\ &= \left[p_1^{Be}(m) \;\; p_2^{Be}(m) \;\; \cdots \;\; p_n^{Be}(m)\right]^T \end{aligned} \tag{10}$$

$\forall i \in [1,n]$, $p_i^{Be}(m) = T_e^B(p_i^{Br}(m))$. In this paper, some bacteria individuals pass away in a certain probability. Supposing that p_e is the probability of death for the individual. If rand $< p_e$, $p_i^{Br}(m)$ passes away, and $p_i^{Be}(m) = p_{\text{rand}}(m)$, where $p_{\text{rand}}(m)$ is a new bacterial individual generated randomly.

3.4. Steps of the Algorithm. See Algorithm 1.

4. Experimental Analysis

In order to analyze the application performance of the parameter selection for the ACA based on the BFA, we ran a simulation for robot path planning in a grid map environment.

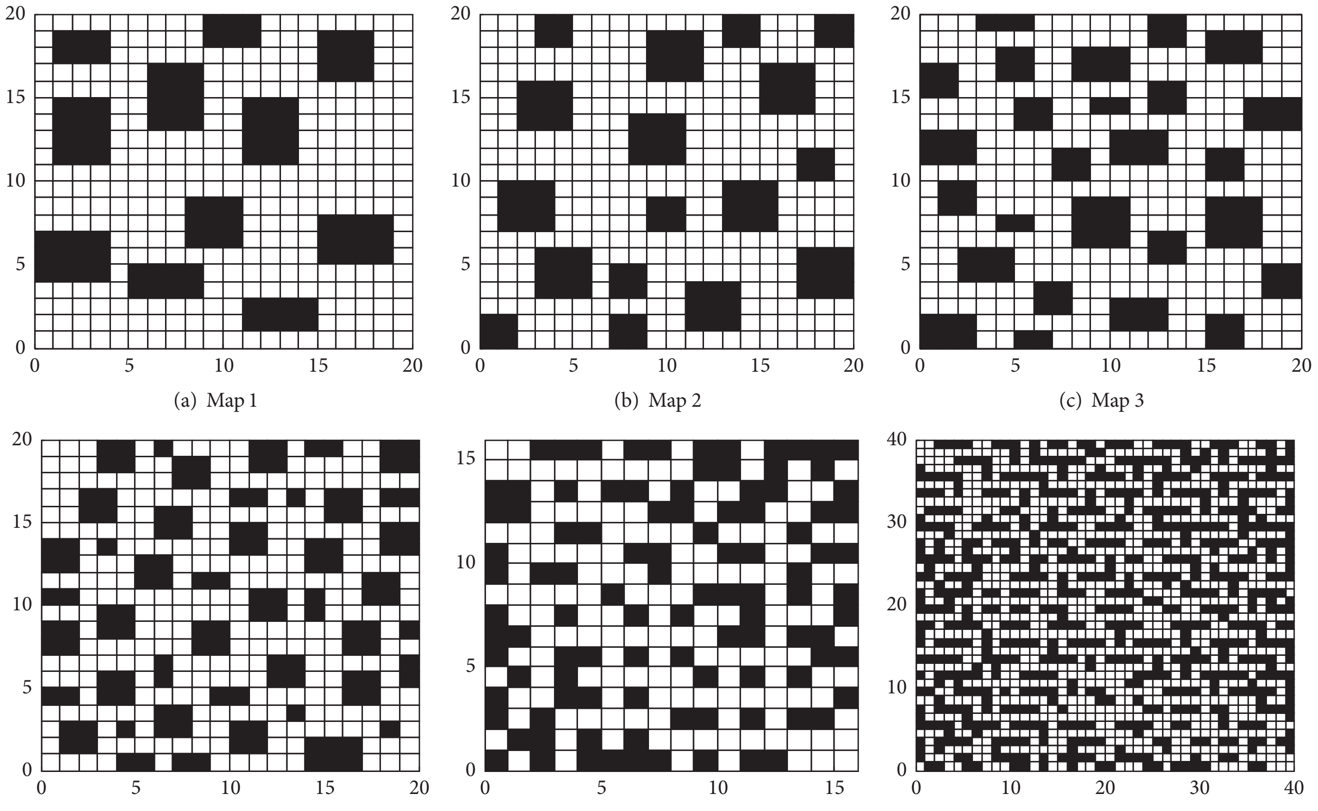
4.1. Establishing Grid Maps. Figure 2 shows six grid maps, where the white grid represents an accessible area and the black grid is a barrier. The ACA based on the parameters obtained using the BFA is used to determine the robot's optimal path in the map without running into obstacles. The upper left corner grid is the starting point, and the lower right corner grid is the end point. Beginning at the starting point, the robot has the potential to move into 8 different directions, including the front, rear, left, right, left front, left rear, right front, and right rear. Excluding the grids with obstacles or already gone, the robot moves one grid at a time. Details of the grid maps are shown in Table 1, including the size, the number of grids, the number of obstacles, and the coverage rate. Maps 1–4 are of the same size, but the complexity is different for each map. Although Map 5 is small, its complexity is high. Map 6 is not only large, but also highly complex. In general, the larger and the more complex the grid map is, the more difficult robot path planning becomes.

4.2. Simulation. To analyze further the application performance of the parameters obtained by using the BFA for the ACA, we conducted the tests for the different grid maps using MATLAB and compared the results with the GA and the PSO. The following parameters were used: parameter initialization of ACA: $m = 40$; parameter initialization of GA: size of population: 20, probability of selection: 0.08, probability of crossover: 0.3, probability of mutation: 0.1, maximum generation: 32; parameter initialization of PSO: number of particles: 20, inertia weight: 0.5, parameter of speed adjusting: 1.9 and 0.8; parameter initialization of BFA: $P = 20$, $N_c = 4$, $N_s = 4$, Step $= 1$, $N_r = 4$, $N_e = 2$, and $P_e = 0.1$. Prior to using the different intelligent algorithms to determine ACA parameter selection, we initialized α, β, ρ, and Q: $\alpha \in [0,10]$, $\beta \in [0,20]$, $\rho \in [0,1]$, and $Q \in [50,150]$.

Formula (4) was used as the fitness function for BFA, which was used to compare the parameter performance. In order to ensure the rigor of the experiment, formula (4) was also used as the fitness function of GA and PSO. Length$(*)$ was adopted to calculate the length of the route path$_i$ of the bacterial individual p_i. Therefore, ACA was used for path planning to obtain the different route path under different map environments for the GA, PSO, and BFA. Due to the randomness of ACA, a fixed mathematical formula for ACA could not be used for path planning. As long as

```
Begin
    Initialize: the bacterial population $P$; times of chemokines $N_c$; maximum steps of chemokines $N_s$; step Step; times of reproduction $N_r$; times of elimination-dispersal $N_e$; probability of elimination-dispersal $P_e$; maximum generations $G_{\max}$.
    $m = 1$;
    While $m <= G_{\max}$
        For $j = 1$: $N_e$
            For $k = 1$: $N_r$
                For $l = 1$: $N_c$
                    While $i <= n$
                        $P_c^B(m) = T_c^B(P(m)) = \left[p_1^{Bc}(m) \quad p_2^{Bc}(m) \quad \cdots \quad p_n^{Bc}(m)\right]^T$;
                        $i = i + 1$;
                    End
                End
                $P_r^B(m) = T_r^B(P_c^B(m)) = \left[p_1^{Br}(m) \quad p_2^{Br}(m) \quad \cdots \quad p_n^{Br}(m)\right]^T$;
            End
            If rand < $P_e$
                $P_e^B(m) = T_e^B(P_r^B(m)) = \left[p_1^{Be}(m) \quad p_2^{Be}(m) \quad \cdots \quad p_n^{Be}(m)\right]^T$;
            End
        End
        $m = m + 1$;
    End
End
```

Algorithm 1



(a) Map 1

(b) Map 2

(c) Map 3

(d) Map 4

(e) Map 5

(f) Map 6

Figure 2: Six grid maps.

Table 1: Details of grid maps.

	Size	Number of grids	Number of obstacles	Coverage rate
Map 1	20 × 20	400	106	26.5%
Map 2	20 × 20	400	113	28.25%
Map 3	20 × 20	400	123	30.75%
Map 4	20 × 20	400	133	33.25%
Map 5	16 × 16	256	103	40.23%
Map 6	40 × 40	1600	663	41.44%

Table 2: Performance comparison of GA, PSO, and BFA.

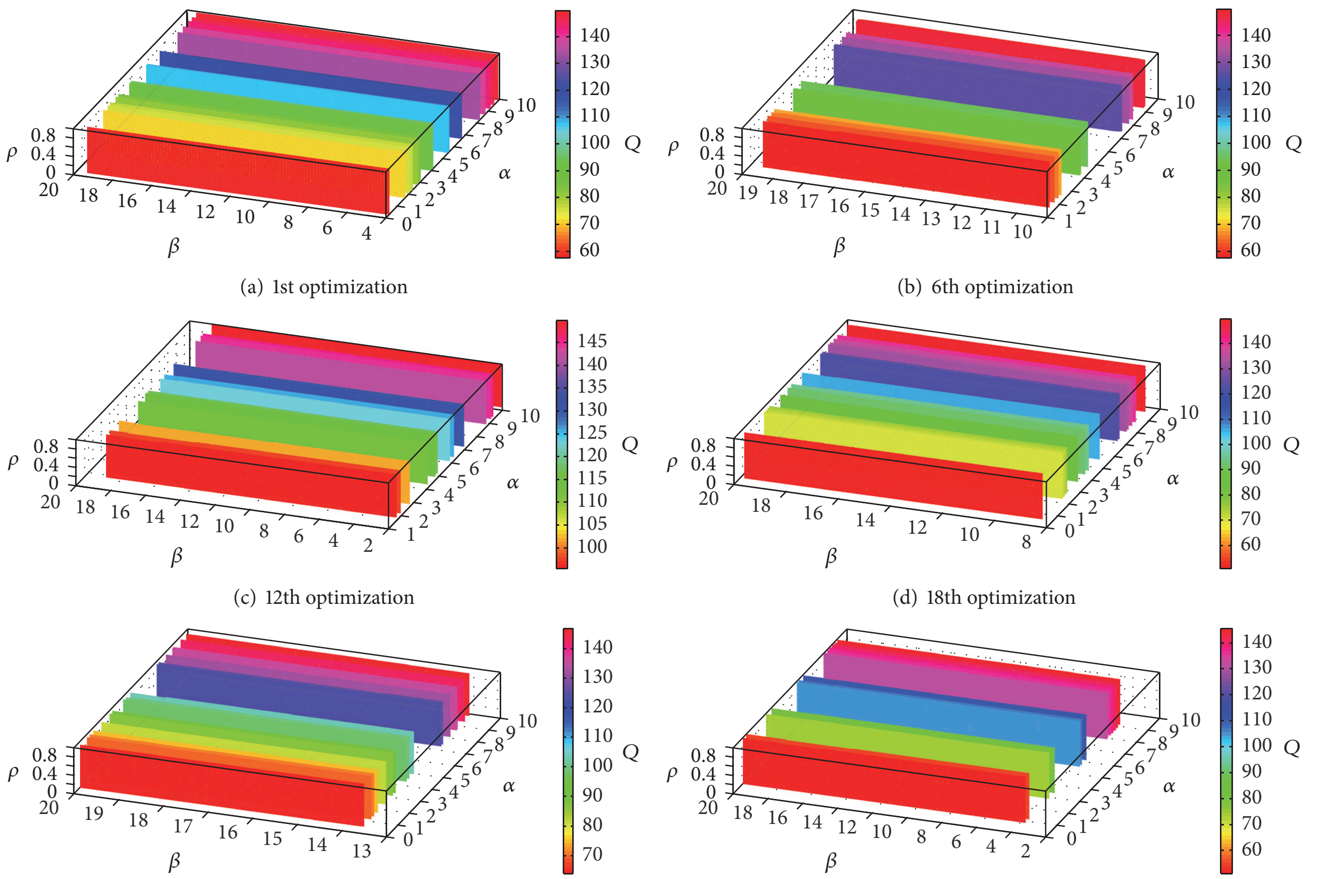
	Optimization times	Number of fitness function calls	Simulation time/s	Minimum distance	Average distance
GA	10	274	142.1787	78.1838	101.9677
	20	532	298.3970	75.4975	101.1249
	30	788	449.1179	72.6690	101.1712
PSO	10	220	118.1516	76.7696	107.8844
	20	420	217.0521	75.4975	107.5190
	30	620	334.9856	72.9411	106.6085
BFA	10	505	249.2963	72.6690	96.1448
	20	1018	526.5445	71.2548	91.8834
	30	1570	810.1920	71.0122	91.1293

Table 3: Test results of GA, PSO, and BFA.

		Map 1	Map 2	Map 3	Map 4	Map 5	Map 6
GA	Maximum distance	82.6690	80.6690	90.0833	85.4975	56.5269	169.9239
	Minimum distance	28.6274	28.6274	28.6427	28.6427	22.3848	72.6690
	Average distance	34.9960	34.7249	36.0491	35.4845	24.8728	101.5615
	Standard deviation	2.1426	2.1856	3.1071	2.3975	1.4661	3.0223
PSO	Maximum distance	45.4558	47.1127	41.1127	43.7990	28.7279	191.5807
	Minimum distance	28.0416	28.0416	28.0416	28.6427	22.3848	74.4264
	Average distance	32.2744	31.4533	33.2238	32.7348	24.2252	100.0113
	Standard deviation	0.4886	0.5327	0.3746	0.5969	0.1976	6.2984
BFA	Maximum distance	39.7990	38.8701	39.4558	39.7990	29.2132	119.3970
	Minimum distance	28.0416	28.0416	28.0416	28.0416	22.3848	69.0122
	Average distance	30.9162	31.2931	31.5497	31.6602	23.3873	88.9554
	Standard deviation	0.7087	0.6806	0.5672	0.6367	0.7376	2.7994

GA, PSO, and BFA call the fitness function, these algorithms have to use ACA for path planning, which increases the complexity and simulation time for the algorithms, but cannot be avoided. For example, for Map 6, Table 2 shows the performance comparison of GA, PSO, and BFA. It is evident that the fitness function is invoked multiple times for either algorithm in Table 2. The chemotaxis operator and reproduction operator of the BFA are based on the fitness value of individual bacteria for the evolution of the evaluation criteria. As a result, BFA had the highest number for invoking the fitness function, resulting in the longest simulation time for BFA. However, this study was mainly concerned with path planning based on global maps, which did not require rapid real-time performance. Therefore, it was acceptable to use the BFA for ACA parameter selection. In Table 2, the performance for parameter selection for GA and PSO for 30 optimizations was inferior compared to the performance of BFA for 10 optimizations, which demonstrated that BFA had fast convergence speed and superior performance.

The test results for the use of GA, PSO, and BFA are shown in Table 3. The maximum, minimum, and average values were the best for BFA, indicating that the BFA algorithm was the preferred method for determining the combination of optimal parameters. The chemotactic operator that adjusts the parameters in an adaptive manner ensures that each group of parameters approaches the optimal value in order to speed up the convergence. The reproduction operator accelerates the speed of optimization for the entire set of parameters. Therefore, the parameters obtained using BFA were the most appropriate to achieve the shortest path for the robot using ACA. The standard deviation for BFA was slightly larger than for PSO due to the elimination-dispersal operator, which introduced new parameter combinations. The new parameter combinations with unknown performance increased the



(a) 1st optimization

(b) 6th optimization

(c) 12th optimization

(d) 18th optimization

(e) 24th optimization

(f) 32nd optimization

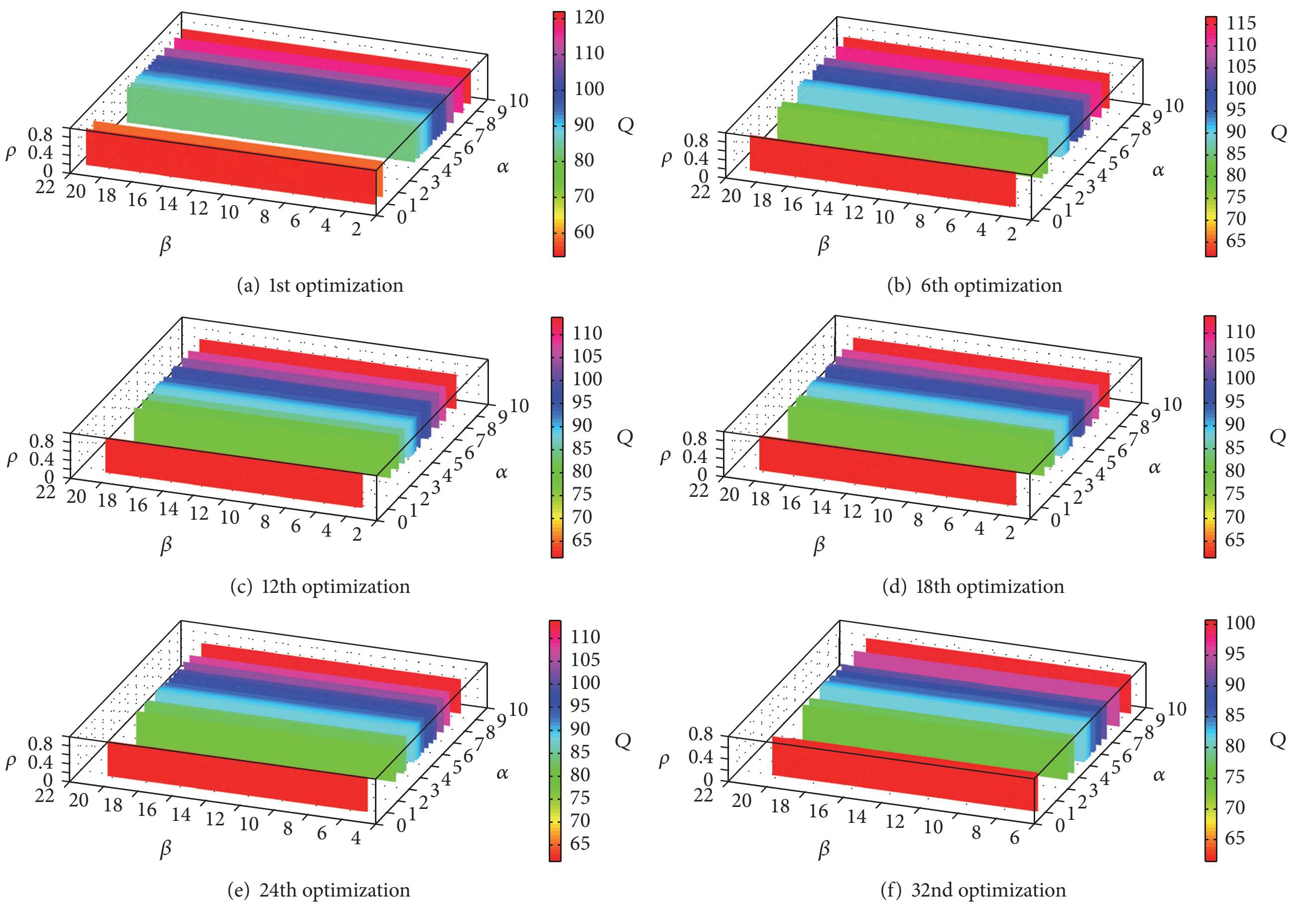
Figure 3: Distribution of parameters obtained by GA.

diversity of the parameter combinations, which improved the global optimization and avoided the algorithm to fall into a local optima. Furthermore, the initial path of the robot was relatively long. The test results shown in Map 1–5 demonstrate that the performance based on GA was the worst, that of BFA was the best, and that of PSO was close to BFA. The test results shown in Map 6 indicate that BFA performed best, demonstrating that BFA was most suitable for parameter optimization in a complex environment. Therefore, the test results proved that parameter selection for ACA based on BFA was highly effective and superior to the other algorithms.

The initial number of parameter combinations was 20 for each algorithm. Using Map 6 as an example, Figures 3, 4, and 5 show the distribution of parameters under different optimal generation for the three algorithms. In Figure 3, the distribution of parameters underwent a process of aggregating diverging and aggregating. Therefore, it was difficult for the parameters to approach the optimal value continuously. The distribution of parameters obtained using PSO is illustrated in Figure 4, and it shows that the position and velocity of a particle are updated constantly based on the population information and individual experience in order to obtain the optimal parameters. The distribution of the 20-parameter combinations exhibited a gradual trend to aggregate. The parameters aggregated quickly from the 1st optimization to the 12th optimization. However, after the 18th optimization, the aggregating trend was almost unchanged, indicating that PSO fell into a local optimal solution and underwent premature convergence. Figure 5 depicts the parameter distribution obtained using BFA and shows that the 20 parameter combinations aggregate continuously due to the chemotactic operator and reproduction operator. These results demonstrated the effectiveness and superiority of parameter selection for ACA using BFA.

Figure 6 shows the evolution curves of the distances using the three algorithms in the environment of Map 6. The average evolution curves indicate that the convergence speed was highest for BFA. The average evolution curves for GA and PSO exhibited relatively large fluctuations. The convergence speed of GA and PSO was slow, indicating that GA and PSO fell into a local optimal solution. The optimal evolution curves show that BFA had the strongest optimization ability. Moreover, the optimization process for BFA was the most stable. The optimization processes for GA and PSO had large fluctuations.

The parameters of ACA for the shortest path are illustrated in Table 4, where the shortest path represents the minimum value shown in Table 3. The results indicated that



(a) 1st optimization

(b) 6th optimization

(c) 12th optimization

(d) 18th optimization

(e) 24th optimization

(f) 32nd optimization

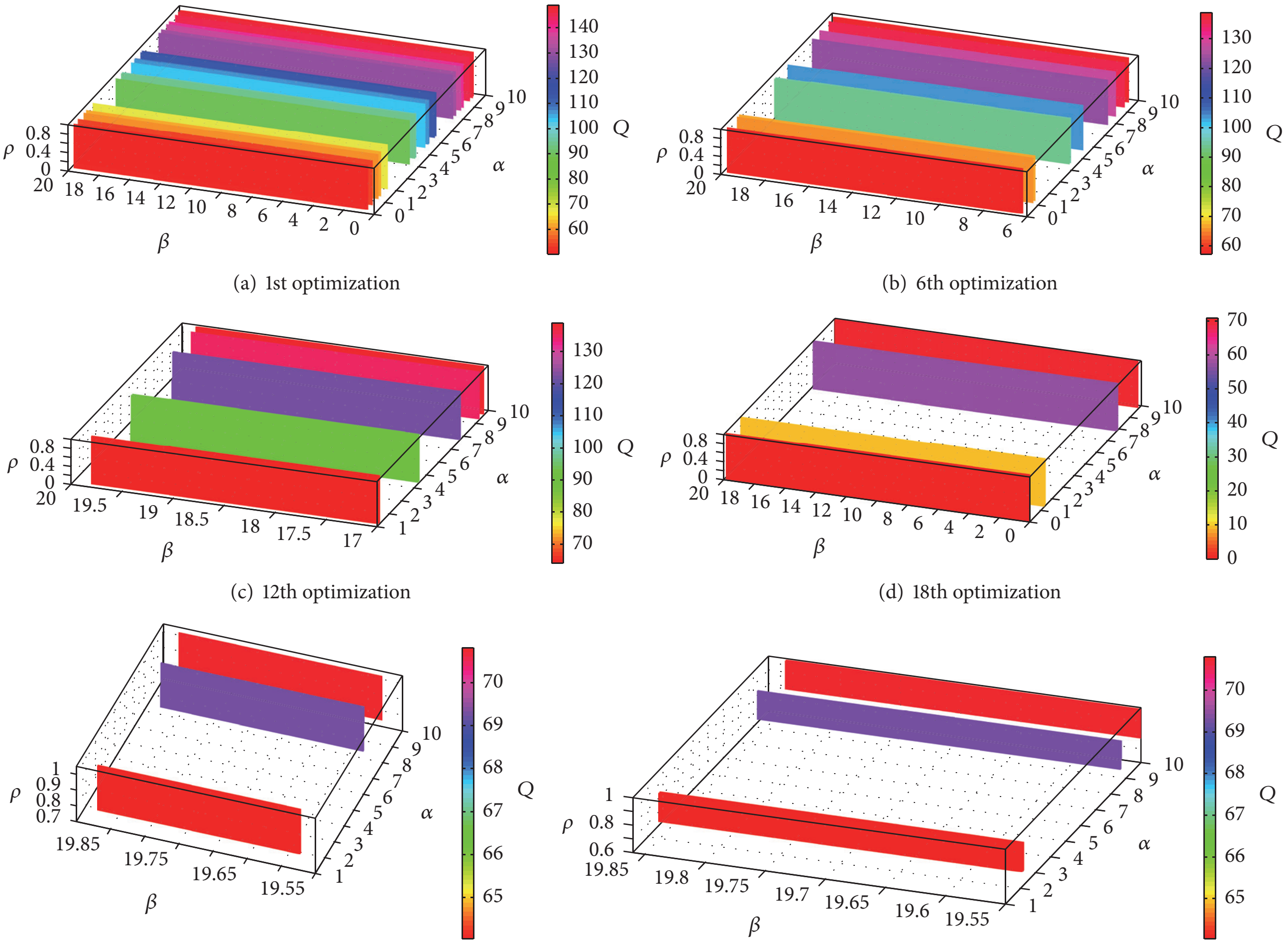
Figure 4: Distribution of parameters obtained by PSO.

Table 4: Parameters of ACA in the shortest path.

		Map 1	Map 2	Map 3	Map 4	Map 5	Map 6
GA	α	2.1260	9.2126	8.1890	2.2835	7.1654	8.7402
	β	20	19.451	17.2549	18.9804	12.8627	18.6667
	ρ	1	0.8	04667	0.6667	0.4667	0.8000
	Q	54.3988	116.8622	78.1525	122.5318	51.7595	146.6764
PSO	α	9.1656	7.2622	6.2604	4.4029	1.7695	5.0572
	β	19.0578	22.7214	14.6075	16.6394	15.0183	18.1021
	ρ	0.4479	0.7653	0.5728	0.4764	0.3806	0.5898
	Q	94.8379	133.7824	91.8428	129.7067	132.0365	66.2206
BFA	α	7.7052	10.0753	1.1790	4.8531	9.6174	1.3500
	β	19.9652	18.8933	18.5790	19.3417	19.2342	19.5401
	ρ	0.2760	0.0199	0.5992	0.3716	0.1962	0.8699
	Q	51.6965	132.6316	100.8766	139.1583	147.0755	70.7550

the shortest distance obtained by the different algorithms was the same for the same grid map. Nevertheless, the corresponding parameter combinations were different, which proved that the ACA parameters were coupled with each other. Therefore, to find the relationship between the parameters and the performance of ACA, some researchers use the control variable method and change a single parameter for an experiment, which is considered not rigorous.

The performance of the parameter combinations in Table 4 was determined using the fitness function. The fitness function performs path planning for the current grid map, which means that the performance is based on chance and



(a) 1st optimization

(b) 6th optimization

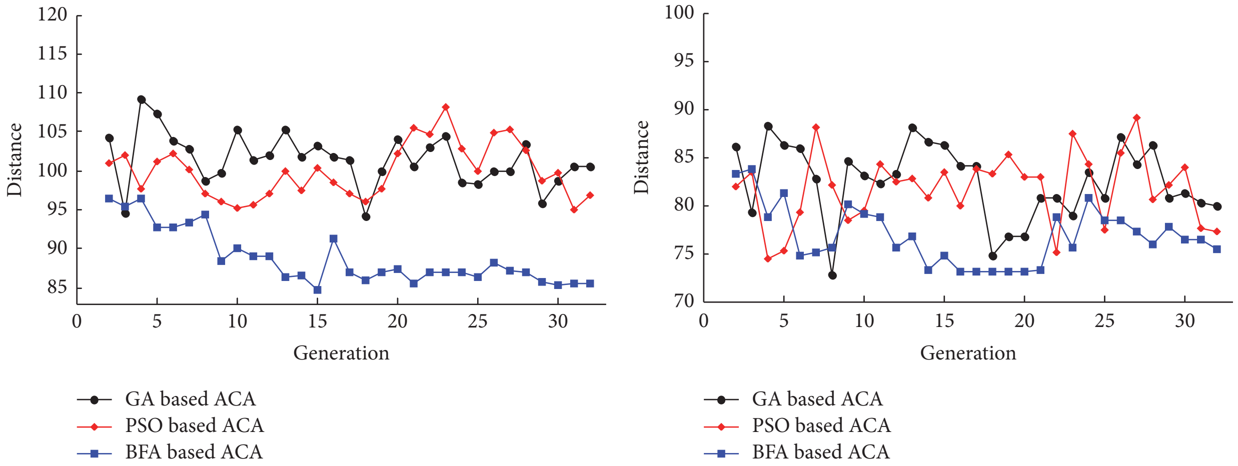
(c) 12th optimization

(d) 18th optimization

(e) 24th optimization

(f) 32nd optimization

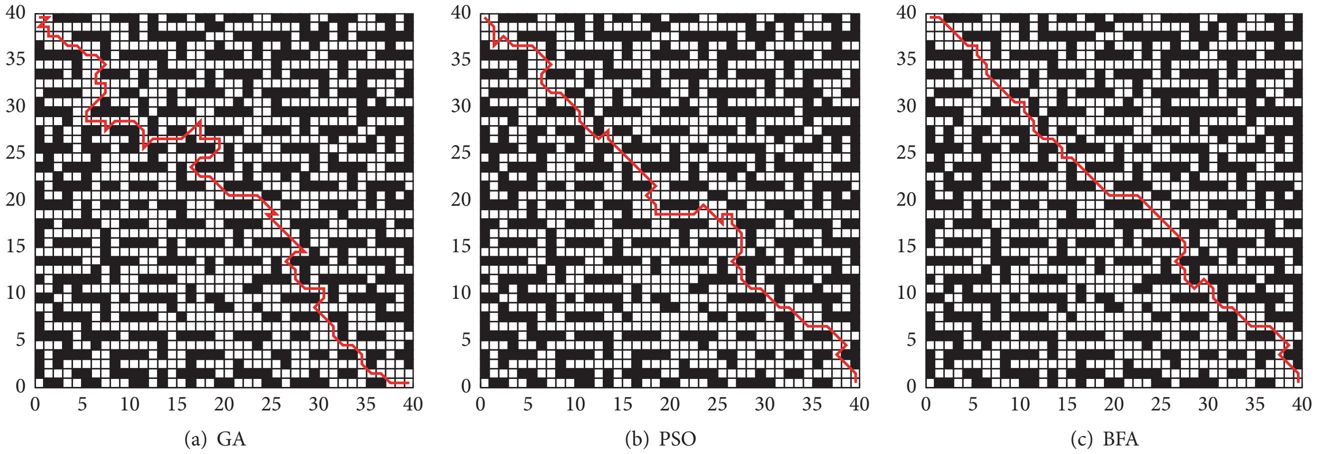
Figure 5: Distribution of parameters obtained by BFA.



(a) The average solution

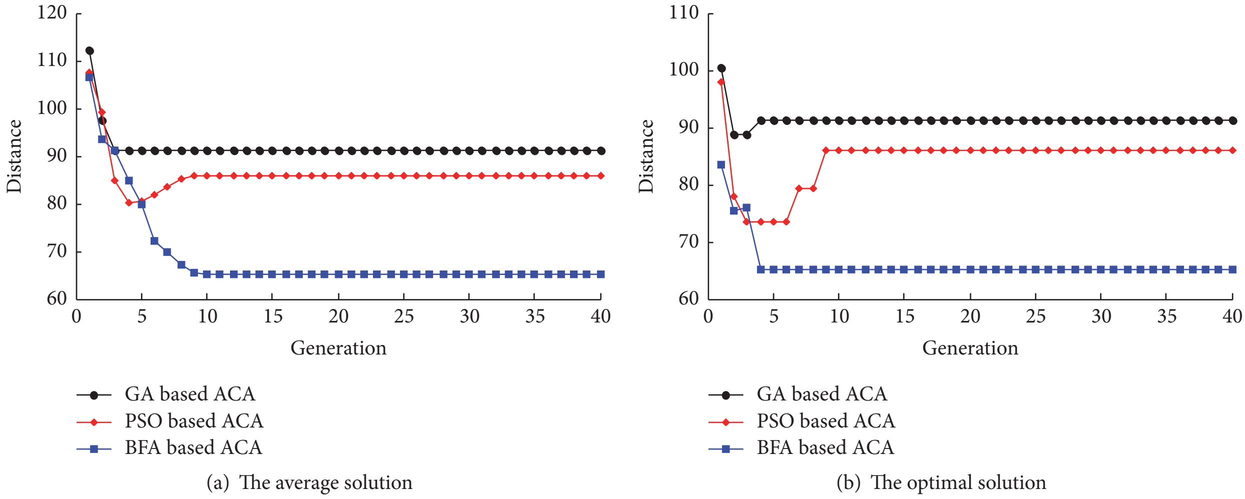
(b) The optimal solution

Figure 6: Evolution curves of the distance by the three algorithms.



(a) GA (b) PSO (c) BFA

Figure 7: Optimal paths under different parameter combinations.



(a) The average solution (b) The optimal solution

Figure 8: Evolution curves of the distance under different parameter combinations.

Table 5: Test results under different parameter combinations.

	GA ($\alpha = 8.7402$, $\beta = 18.666$, $\rho = 0.8000$, $Q = 146.6764$)	PSO ($\alpha = 5.0572$, $\beta = 18.1021$, $\rho = 0.5898$, $Q = 66.2206$)	BFA ($\alpha = 1.3500$, $\beta = 19.5401$, $\rho = 0.8699$, $Q = 70.7550$)
Maximum distance	139.6396	128.2254	129.7817
Minimum distance	88.669	73.4975	65.2548
Average distance	91.41	85.5504	66.7729
Standard deviation	2.1076	3.033	6.1966

randomness. To validate further the performance of the parameter combinations, tests were conducted based on ACA with different parameter combinations and using Map 6 for robot path planning. It was assumed that the number of ants was 40 and the cycle time was 40.

The results of this test are shown in Table 5. Based on the maximum, minimum, and average values, it was evident that the use of BFA resulted in the best performance regarding parameter combination. The initial path selection of the ants was poor. As the cycle increased, the path became shorter. The ants found the best route based on the parameters obtained by BFA, and the path was much shorter than the initial path, resulting in a large standard deviation. To some extent, the large standard deviation proved the effectiveness and superiority of BFA used for parameters selection for ACA.

The optimal paths using different parameter combinations are shown in Figure 7. The use of ACA based on the parameters obtained using BFA, resulted in the shortest path, indicating that BFA was able to obtain the most appropriate parameters. Figure 8 illustrates the evolution curves of the distance under different parameter combinations, and it is evident that the parameters obtained by GA or PSO led to ACA falling into the local optimal solution and to an apparent premature convergence. However, using ACA based on parameters obtained by BFA resulted in the optimal path. The test results in this paper showed that the parameters had a large impact on the performance of ACA. Choosing suitable parameters enabled ACA to achieve an optimal performance. However, unsuitable parameters resulted in ACA falling into a local optima. The test results showed that the use of BFA could determine optimal parameters for ACA to achieve the best performance.

5. Conclusion

Optimal performance of ACA mainly depended on suitable parameters. However, the parameters were coupled with each other, and the potential number of parameters was large, presenting a considerable challenge for ACA parameter selection. Therefore, multiple parameter selection for ACA based on BFA was proposed. The parameters for ACA were mapped into a multidimensional space, and the optimal parameters were obtained automatically using BFA, demonstrating the superiority of the method. Moreover, the four parameters α, β, ρ, and Q could be obtained simultaneously. The parameter optimization process took into account the coupling between the parameters, which ensured the rationality of the parameters. The test results showed that the parameters obtained using BFA for ACA were the most suitable. Compared with GA and PSO, BFA had a better performance for parameter selection of ACA with regard to convergence speed, optimization capability, and stability, which demonstrated the effectiveness and superiority of parameter selection for ACA based on BFA.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The project is supported by the National High Technology Research and Development Program of China (863 Program) (Grant no. 2012AA041504) and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[1] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[2] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997.

[3] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[4] K.-L. Huang and C.-J. Liao, "Ant colony optimization combined with taboo search for the job shop scheduling problem," *Computers and Operations Research*, vol. 35, no. 4, pp. 1030–1046, 2008.

[5] C. Blum and M. Sampels, "An ant colony optimization algorithm for shop scheduling problems," *Journal of Mathematical Modelling and Algorithms*, vol. 3, no. 3, pp. 285–308, 2004.

[6] Q. Xu, J. Mao, and Z. Jin, "Simulated annealing-based ant colony algorithm for tugboat scheduling optimization," *Mathematical Problems in Engineering*, vol. 2012, Article ID 246978, 22 pages, 2012.

[7] Y. Jiang, Z. Xu, X. Xu, Z. Liao, and Y. Luo, "A schedule optimization model on multirunway based on ant colony algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 368208, 11 pages, 2014.

[8] H. Mei, Y. Tian, and L. Zu, "A hybrid ant colony optimization algorithm for path planning of robot in dynamic environment," *International Journal of Information Technology*, vol. 12, no. 3, pp. 78–88, 2006.

[9] G.-Z. Tan, H. He, and A. Sloman, "Ant colony system algorithm for real-time globally optimal path planning of mobile robots," *Acta Automatica Sinica*, vol. 33, no. 3, pp. 279–285, 2007.

[10] J. Zhao, D. Cheng, and C. Hao, "An improved ant colony algorithm for solving the path planning problem of the omnidirectional mobile vehicle," *Mathematical Problems in Engineering*, vol. 2016, Article ID 7672839, 10 pages, 2016.

[11] Y. Dai, L. Liu, and S. Wang, "Parameter optimization of ant colony algorithm based on particle swarm optimization," in *Proceedings of the International Conference*, pp. 1266–1269, niversity of Electronic Science and Technology of China, China, December 2007.

[12] X. S. Wang, J. Pan, and Y. H. Cheng, "Parameter selection for ant colony algorithms based on graph knowledge transfer," *Control and Decision*, vol. 26, no. 12, pp. 1855–1860, 2011.

[13] Y. Hei and P. Du, "Optimal choice of the parameters of ant colony algorithm," *Journal of Convergence Information Technology*, vol. 6, no. 9, pp. 96–104, 2011.

[14] Y. Gan and S. Li, "Studies on parameters configuration for microhabitat ant colony optimization algorithm," *Manufacturing Automation*, vol. 33, no. 3, pp. 66–69, 2011.

[15] Y. Feng, "Parameters optimization research of ant colony optimization based on genetic algorithm," *Journal of Guiyang College Natural Sciences*, vol. 9, no. 4, pp. 25–27, 2014.

[16] Q. Li, C. Zhang, P. Chen, and Y.-X. Yin, "Improved ant colony optimization algorithm based on particle swarm optimization," *Control and Decision*, vol. 28, no. 6, pp. 873–883, 2013.

[17] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[18] F. Zhao, Y. Liu, Z. Shao, X. Jiang, C. Zhang, and J. Wang, "A chaotic local search based bacterial foraging algorithm and its application to a permutation flow-shop scheduling problem," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 9, pp. 962–981, 2016.

[19] X.-D. Liang, L.-Y. Li, J.-G. Wu, and H.-N. Chen, "Mobile robot path planning based on adaptive bacterial foraging algorithm," *Journal of Central South University*, vol. 20, no. 12, pp. 3391–3400, 2013.

[20] O. P. Verma, M. Hanmandlu, A. K. Sultania, and A. S. Parihar, "A novel fuzzy system for edge detection in noisy image using bacterial foraging," *Multidimensional Systems and Signal Processing*, vol. 24, no. 1, pp. 181–198, 2013.

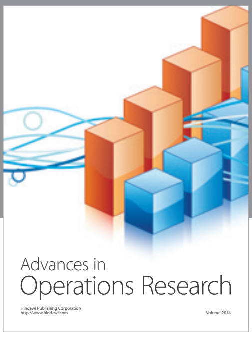
[21] S. Dasgupta, A. Biswas, S. Das, B. K. Panigrahi, and A. Abraham, "A micro-bacterial foraging algorithm for high-dimensional optimization," in *2009 IEEE Congress on Evolutionary Computation (CEC)*, pp. 785–792, Trondheim, Norway, May 2009.

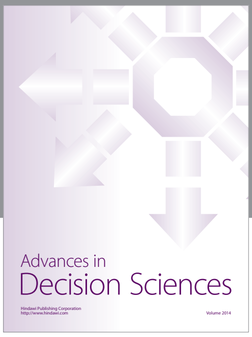
[22] P. Li and H. Zhu, "Immune optimization of path planning for coal mine rescue robot," *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, vol. 11, no. 1, pp. 91–98, 2016.

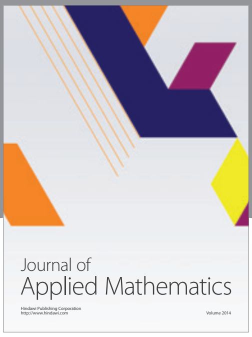
[23] Q. Yang, L. Fang, and X. Duan, "RMACO: a randomly matched parallel ant colony optimization," *World Wide Web*, vol. 19, no. 6, pp. 1009–1022, 2016.

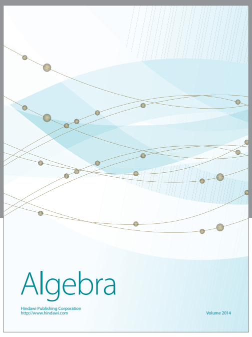
[24] L. Liu, Y. Dai, and L. Wang, "Ant colony algorithm parameters optimization," *Computer Engineering*, vol. 34, no. 11, pp. 208–210, 2008.

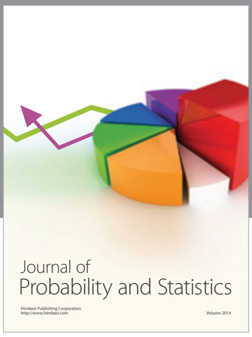
[25] C. Yang, J. Ji, J. Liu, J. Liu, and B. Yin, "Structural learning of Bayesian networks by bacterial foraging optimization," *International Journal of Approximate Reasoning*, vol. 69, pp. 147–167, 2016.


Advances in
Operations Research

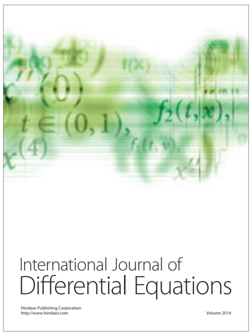

Advances in
Decision Sciences


Journal of
Applied Mathematics

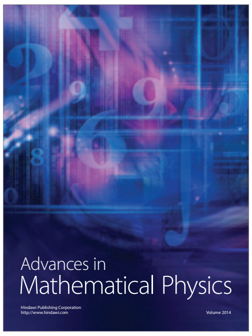

Algebra

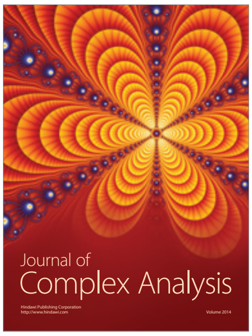

Journal of
Probability and Statistics

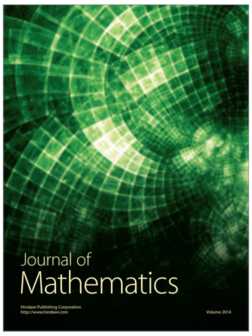
The Scientific
World Journal

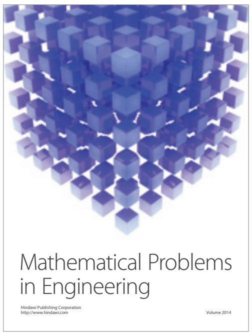

International Journal of
Differential Equations

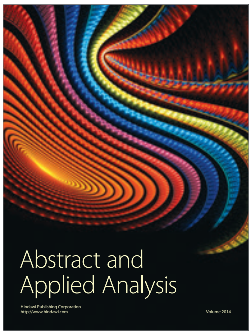
International Journal of
Combinatorics

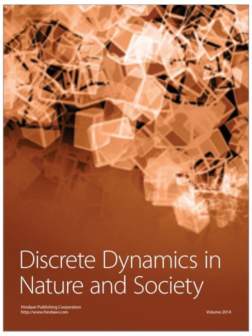

Advances in
Mathematical Physics

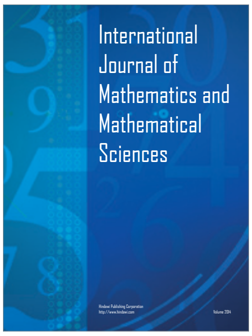

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

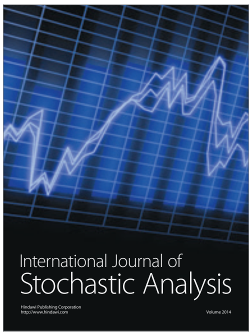

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization