Robot Path Planning Using Bacterial Foraging Algorithm

Wei Liu1, Ben Niu3, Hanning Chen2,*, and Yunlong Zhu2

1 College of Information and Technology, Jilin Normal University, Siping, 136000, China
2 Laboratory of Information Service and Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
3 College of Management, Shenzhen University, Shenzhen 518060, China

The goal of the robot path planning problem is to determine an optimal collision-free path for a mobile robot between a start and a target point in an environment surrounded by obstacles. Optimal collision-free trajectory planning for mobile robot is always a major issue in robotics due to the necessity for the robots' course of movement. In recent years, as the emergence of another member of the swarm intelligence family—bacterial foraging optimization (BFO), the bacterial foraging strategy has attracted a great deal of interests. In this work, the path planning problem is approached by the mobile robot that mimics the foraging strategy of BFO algorithm. The objective is to minimize the path length and the number of turns without colliding with an obstacle. In the simulation studies, two test scenarios of static environment with different obstacle distribution are adopted to evaluate the performance of the proposed method. Simulation results show that our method is able to generate a collision-free path in complex environment.

Keywords: Robot Path Planning, Bacterial Foraging Behaviors, Swarm Intelligence.

1. INTRODUCTION

The goal of robot system is to do tasks at a cost as low as possible. Thus, optimal path planning for robot manipulators is always a hot spot in research fields of robotics.1 The goal of robot path planning is to find an optimal, collision-free trajectory between two points in a working environment composed of many obstacles. That is, the path planning approach can be applied in static, dynamic or both environments, depending on the mode in which the environment is known and given in advance. The optimality of the path is usually measured by the traveling time and penalty for obstacle avoidance of the mobile robot.

The motile bacteria such as E. coli and Salmonella propel themselves by rotating its flagella2–4 (see Fig. 1). To move forward, the flagella counterclockwise rotate and the organism “swims” (or “run”). While a clockwise rotation of the flagellum causes the bacterium to randomly “tumble” itself in a new direction and swim again. Alternation between “swim” and “tumble” enable the bacterium search for nutrients in random directions. Swimming is more frequent as the bacterium approaches a nutrient gradient. Tumbling, hence direction changes, is more frequent as the bacterium moves away from some food to search for more. Basically, bacterial chemotaxis is a complex combination of swimming and tumbling that keeps bacteria in places of higher concentrations of nutrients. It is can also be considered as the optimization process of the exploitation of known resources, and costly exploration for new, potentially more valuable resources.

In recent years, a few models have been developed to mimic bacterial foraging behavior and have been applied for solving some practical problems.5–7 Among them, Bacterial Foraging Optimization (BFO) is a simple but powerful optimization tool that mimics the foraging behavior of E. coli bacteria.8 Until now, BFO has been applied successfully to some engineering problems, such as optimal control,9 harmonic estimation,10 transmission loss reduction11 and machine learning.12

In this context, the bacterial activities, including food seeking, prey selection and avoidance of noxious substances, may be used in the robot system, based in the forage strategy, as proposed by Passino.4 In this work, we provide some initial insights into this potential by using the BFO based search model to creating optimal collision-free trajectories for mobile robot, which focusing on minimizing the path length and the number of turns without colliding with an obstacle.
In the experiments, two cases study of static environment with obstacles are presented and evaluated. Simulation results show the adaptation of the bacterial robot in different environments in the planned trajectories.

The rest of the paper is organized as follows. Section 2 gives a review of the original BFO algorithm. Section 3 describes the detailed design scheme for mobile robot path planning based on BFO. Then the simulation results of robot navigation on two test scenarios are discussed in Section 4. Finally, conclusions are drawn in Section 5.

2. THE BACTERIAL FORAGING OPTIMIZATION

Bacterial foraging optimization algorithm is inspired by an activity called “chemotaxis” exhibited by bacterial foraging behaviors.

2.1. BFO Model

The original bacterial foraging optimization (BFO) algorithm is one of the state-of-the-art evolutionary algorithms,13-15 which consists of three principal mechanisms, namely chemotaxis, reproduction, and elimination-dispersal. We briefly describe each of these processes as follows:

2.1.1. Chemotaxis

In the original BFO, a unit walk with random direction represents a “tumble” and a unit walk with the same direction in the last step indicates a “run.” Suppose \( \theta'(j, k, l) \) represents the bacterium at \( j \)th chemotactic, \( k \)th reproductive, and \( l \)th elimination-dispersal step. \( C(i) \) is the chemotactic step size during each run or tumble (run-length unit). Then in each computational chemotactic step, the movement of the \( i \)th bacterium can be represented as

\[
\theta'(j + 1, k, l) = \theta'(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^2(i) \Delta(i)}}
\]

where \( \Delta(i) \) is the direction vector of the \( j \)th chemotactic step. When the bacterial movement is run, \( \Delta(i) \) is the same with the last chemotactic step; otherwise, \( \Delta(i) \) is a random vector whose elements lie in \([-1, 1]\).

With the activity of run or tumble taken at each step of the chemotaxis process, a step fitness, denoted as \( J(i, j, k, l) \), will be evaluated.

2.1.2. Swarming

A simple cell-to-cell communication mechanism is also mimicked in the classical BFO model: as each bacterium moves, it releases attractant to signal other bacteria to swarm towards it; meanwhile, each bacterium releases repellent to warn other bacteria to keep a safe distance from it. BFO simulates this mechanism by representing the combined cell-to-cell attraction and repelling effect as:

\[
J_{cc}(\theta'(j, k, l), (j, k, l)) = \sum_{i=1}^{s} J_{i cc}(\theta', \theta) = \sum_{i=1}^{s} \left[ -d_{\text{attract}} \exp \left( -\omega_{\text{attract}} \sum_{m=1}^{D} (\theta'_{m} - \theta_{m})^2 \right) \right] + \sum_{i=1}^{s} \left[ -h_{\text{repellant}} \exp \left( -\omega_{\text{repellant}} \sum_{m=1}^{D} (\theta'_{m} - \theta_{m})^2 \right) \right] \quad (2)
\]

where \( J_{cc}(\theta', \theta) \) is the objective function value to be added to the actual objective function (to be minimized) to present a time varying objective function, \( S \) is the total number of bacteria, \( D \) is the number of variables to be optimized, and \( d_{\text{attract}}, \omega_{\text{attract}}, h_{\text{repellant}}, \omega_{\text{repellant}} \) are different coefficients that should be chosen properly.

2.1.3. Reproduction

The health status of each bacterium is calculated as the sum of the step fitness during its life, i.e., \( \sum_{i=1}^{N_c} J(i, j, k, l) \), where \( N_c \) is the maximum step in a chemotaxis process. All bacteria are sorted in reverse order according to health status. In the reproduction step, only the first half of population survives and a surviving bacterium splits into two identical ones, which are then placed in the same locations. Thus, the population of bacteria keeps constant.

**Fig. 1.** Bacterial foraging behaviors.
2.1.4. Elimination and Dispersal

The chemotaxis provides a basis for local search, and the reproduction process speeds up the convergence which has been simulated by the classical BFO. While to a large extent, only chemotaxis and reproduction are not enough for global optima searching. Since bacteria may get stuck around the initial positions or local optima, it is possible for the diversity of BFO to change either gradually or suddenly to eliminate the accidents of being trapped into the local optima. In BFO, the dispersion event happens after a certain number of reproduction processes. Then some bacteria are chosen, according to a preset probability, to be killed and moved to another position within the environment.

2.2. Step-by-Step Algorithm

In what follows we briefly outline the original BFO algorithm step by step:

[Step 1] Initialize parameters n, S, N, N, N_r, N_c, P_c, C(i) (i = 1, 2, ..., S), θ. Where,

n: Dimension of the search space,
S: The number of bacterium,
N: Chemotactic steps,
N: Swim steps,
N_r: Reproductive steps,
N_c: Elimination and dispersal steps,
P_c: Probability of elimination,
C(i): The run-length unit. (i.e., the chemotactic step size during each run or tumble).


[Substep a] For i = 1, 2, ..., S, take a chemotactic step for bacterium i as follows.

[Substep b] Compute fitness function, J(i, j, k, l).

[Substep c] Let J_u = J(i, j, k, l) to save this value since we may find better value via a run.

[Substep d] Tumble: Generate a random vector Δ(i) ∈ R^n with each element Δ_u(i), m = 1, 2, ..., S, a random number on [-1, 1].

[Substep e] Move: Let the ith bacterium swim using Eq. (1). This results in a step of size C(i) in the direction of the tumble for bacterium i.

[Substep f] Compute J(i, j + 1, k, l) with θ(j+1, k, l).

[Substep g] Swim:

(i) Let m = 0 (counter for swim length).
(ii) While m < N (if have not climbed down too long).

(a) If J(i, j + 1, k, l) ≤ J_u, let J_u = J(i, j + 1, k, l).
(b) else let m = N.

[Substep h] Go to next bacterium (i+1): if i ≠ S go to (b) to process the next bacterium.

[Step 5] If j < N, go to step 3. In this case, continue chemotaxis since the life of the bacteria is not over.

[Step 6] Reproduction:

[Substep a] For the given k and l, and for each i = 1, 2, ..., S, let

\[ J_{healh} = \sum_{j=1}^{N_c+1} J(i, j, k, l) \]

be the health of the bacteria. Sort bacterium in order of ascending values (J_{healh}).

[Substep b] The S_r bacterium with the highest J_{healh} values die and the other S_c bacterium with the best values split and the copies that are made are placed at the same location as their parent.

[Step 7] If k < N go to step 2. In this case the number of specified reproduction steps is not reached and start the next generation in the chemotactic loop.

[Step 8] Elimination-dispersal: For i = 1, 2, ..., S, with probability p_c, eliminate and disperse each bacterium, which results in keeping the number of bacteria in the population constant. To do this, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain. If l < N then go to step 2; otherwise end.

3. BFO BASED ROBOT PATH PLANNING

Navigation in mobile robot is a methodology that allows guiding a robot to accomplish a mission through an environment with obstacles in a good and safe way. The two basic tasks involved in navigation are the environment perception, and path planning. Generally global planning methods complemented with local methods are used for indoor missions since the environments are known or partially known; for outdoor applications, local planning methods are more suitable, becoming global planning methods a complement because of the scant information of the environment.

Literature is rich in boarding to solve mobile robots trajectory planning in presence of static or dynamic obstacles. One of the most popular planning methods is the artificial potential fields. However, this method gives only one trajectory solution that may not be the smaller trajectory in a static environment. Recently, the interest in using evolutionary algorithms and swarm intelligence for robot path planning is increasing. Up to now, the genetic algorithm and particle swarm optimization methods are used in mobile robot trajectory planning, and generally when the environment description is given. In some literatures, the genetic algorithm and particle swarm optimizer has been used for robot path planning.

In this paper, based on the BFO model, we consider the utilization of biomimicry of bacterial chemotaxis mechanism to develop a bio-inspired path planning strategy for mobile robot. In the proposed model, a bacterial robot that mimics the behavior of bacteria is able to determine an optimal collision-free path between a start and a target.
point in an environment surrounded by obstacles. That is, the bacterial “tumble” and “run” strategy in BFO model enable the robot explore the environment and finally locate the target point without colliding any obstacles.

In the navigation process, the location of bacterial robot can be evaluated as a multi-objective cost function:

$$f(x) = w_1 f_g(x) + w_2 f_o(x)$$  \hspace{1cm} (4)

where $x$ represents the coordinate of the robot in the working environment, $t$ is the time step, $f_g$ is the goal function that represents the distance between the current robot position and the target point, $f_o$ is the obstacle function that represents the distance between the current robot position and the obstacles nearby, $w_1$ and $w_2$ are the weight parameter that specifies the relative importance of achieving obstacle avoidance and reaching the goal. The overall operating process of robot path planning based on BFO can be described as in Figure 2.

**4. SIMULATION RESULTS**

In this experiment, the proposed planning method is evaluated against an ideal 2-dimension square working area.

The 2D Sphere function with the global minimum (i.e., the goal point) at $[25, 25]$ is used for the goal function that is given by:

$$f_g(x, y) = (x - 25)^2 + (y - 25)^2, x, y \in [0, 30]$$  \hspace{1cm} (5)

Figures 3(a) and (b) show the landscape and contour plot of the Sphere goal function respectively. That is, at each time step, the bacterial chemotaxis driven the robot moved to go down the surface toward the goal, while maybe run into an obstacle.

In this work, we take Gaussian function of unity height to represent an obstacle. Then the obstacle function can be formulated as a composition of a number of Gaussian functions and can be formulated as:

$$f_o(x, y) = \max_{i=1}^{n} \exp^{-0.8 \times ((x-x_i)^2+(y-y_i)^2)}$$  \hspace{1cm} (6)

where $n$ is the number of the obstacles in the working environment and $(x_i, y_i)$ is the position of the $i$th obstacle. It should be note that the use of the maximum of all the Gaussian functions ensures that each obstacle position is represented independent of the others.
Since the different locations of obstacles can result in a different degree of environmental complexity, two obstacle functions, namely \( f_{o1} \) and \( f_{o2} \) with two different distributions each has six obstacles, are evaluated in our experiment respectively. In Figures 4 and 5 we show the landscape of \( f_{o1} \) and \( f_{o2} \) respectively. Figures 6(a) and (b) show the contour plots of the final working environments combining the goal and obstacle function of two test cases.
respectively, along with the initial position and goal position. Here we set \( w_1 = 1 \) and \( w_2 = 0.0001 \).

For the tested case 1 with six obstacles, the optimal trajectory and fitness convergence obtained by the proposed planning strategy are illustrated in Figures 7(a) and (b) respectively. We can observe from the results that the bacterial chemotaxis mechanism enable the robot avoid the obstacles but tries to stay on course to the goal.

The simulation results on tested case 2 are illustrated in Figure 8. Although the environment is more complex in this case, the robot using bacterial foraging strategy is still able to find the target point without colliding into any obstacles. The Figures 7(b) and 8(b) shows that the robust convergences of path planning for both cases are obtained after about 400 generations.

5. CONCLUSIONS

In this paper, we consider the utilization of biomimicry of bacterial foraging strategy to develop an adaptive control strategy for mobile robot, and proposed a bacterial foraging approach for robot path planning. In the proposed model, robot that mimics the behavior of bacteria is able to determine an optimal collision-free path between a start and a target point in the environment surrounded by obstacles. In the simulation studies, two test scenarios of static environment with different number obstacles are adopted to evaluate the performance of the proposed method. Simulation results show that the robot which reflects the bacterial foraging behavior can adapt to complex environments in the planned trajectories with both satisfactory accuracy and stability.

Acknowledgments: This work was supported in part by the Natural Science Foundation of China under Grant Nos. 61105067, 71001072, 71271140 and 61174164, and the Strategic Cooperation Project of Foshan and Chinese Academy of Sciences under Grant No. 2011BY100383.

References


Received: 26 June 2012. Accepted: 25 August 2012.