

PATH PLANNING FOR MOBILE ROBOTS IN DYNAMIC ENVIRONMENT USING IMPROVED BACTERIAL FORAGING ALGORITHM

*Yisti Vita Via¹, Henni Endah Wahanani² and Salamun Rohman Nudin³

^{1,2}Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur, Indonesia;

³Faculty of Engineering, Universitas Negeri Surabaya, Indonesia

*Corresponding Author, Received: 12 Dec. 2017, Revised: 28 March 2018, Accepted: 28 April 2018

ABSTRACT: The topic of navigation is one of the focused problems in the correlation of autonomous mobile robots. It is necessary to have a methodology to accomplish a collision path of navigation from an initial position (state) to a target position (goal). This paper presents a new method for path planning to determine a feasible, optimal, and safe path. Based on Bacterial Foraging algorithm, we propose an improved algorithm for solving the one-line path planning of mobile robots by choosing the proper objective function for the target, obstacles, and robots collision avoidance. This algorithm finds a path towards the target and avoiding the obstacles using particles, which are randomly distributed, on a determined area around a robot. The criterion on which it selects the best article is the distance to the target. By the proposed approach, the next positions of each robot are designed.

Keywords: Mobile Robot, Path Planning, Bacterial Foraging Algorithm, Nature Inspired Algorithm

1. INTRODUCTION

An autonomous mobile robot can locate itself and determine a path to a specific target in the environment including obstacles using localization and orientation information through the sensors. The mobile robot is expected to be used in higher, deeper, and dangerous environment where human is difficult or cannot imagine reaching, such as military reconnaissance, emergency rescuing, aerospace exploration, and underground detection [1]. For this, it is inevitable that as one of planning research in robotics, the path planning area for mobile robot grew significantly over time after Lozano-Perez and Wesley [2].

The topic of navigation is one of the focused problems in the correlation of autonomous mobile robots. In order to take it from one position to another one without the assistance of an operator, it is necessary to have a methodology that decides their own actions to accomplish a set of navigation and operation goals. Navigation consists of two essential components known as localization and path planning. The first one refers to the ability to determine the accurate position at any moment relative to the search space according to the environment perceptions gathered by sensors. Whereas path planning consists in designing a collision path from an initial position (state) to a target position (goal) to optimize it with respect to some criteria such as distance, time, cost, and energy.

Path planning is an essential aspect to design a

fast and efficient procedure for navigation. It can be divided into static and dynamic environment according to the atomicity and availability of knowledge of the information about the environment. In the static environment, all the obstacles would be static. While in a dynamic environment, obstacles can be static and dynamic with varying speeds. The path planning is again divided into two sections global and local path planning. Global path planning requires the environment to be completely known and terrain should be static. On the other hand, local path planning means that path planning is done while the robot is moving, in other words, the algorithm is capable of producing a new path in response to environmental changes [3]. Classification of path planning is given in Figure 1.

The methodologies for path planning problems are categorized into classical and heuristic approaches. The classical methods have dominated this research area in the past. However, since the environment becomes complex and dynamic, research with heuristic methods has been developed so that to overcome the drawbacks of classical methods [4].

Many studies have used representative heuristic methods such as meta-heuristic in order to efficiently generate a feasible solution even in a complex environment. Martínez-Alfaro and Gómez-García applied fuzzy control and simulated annealing (SA) to accomplish the obstacle avoidance for a mobile robot [5]. Genetic algorithm (GA) is used in the papers where the

fitness is evaluated with respect to the path length or sum of angles and where genetic operators are conducted for the evolution [6]. Huang and Tsai generated an initial feasible path using PSO and GA together [7]. Several other studies [8][9][10] used artificial bee colony (ABC) and ant colony optimization (ACO).

There are several hybrid approaches which employ the advantages of classical methods. Liang and Xu [11] incorporated several meta-heuristics and the Dijkstra algorithm after node generation via free space modeling. Kala, Shukla, and Tiwari [12] presented the A* algorithm and multi-neuron heuristic search to identify whether the path could be further decomposed.

Meta-heuristic approaches have been attracting considerable research interest in recent years like Genetic Algorithm (GA), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Bacterial Foraging Algorithm (BFA). BFA is a relatively new method, which provides efficiency to find the destination in a relatively short time. By using this approach, a feasible path can avoid all known obstacles, finds and follows the determined path accurately, and also reduces the computational complexity, free from getting stuck in the local minima and no need to tuning algorithm, which offers availability. For these reasons, this paper presents a path improvement method inspired by BFA in order to generate the enhanced final path with respect to terms of time and path's feasibility.

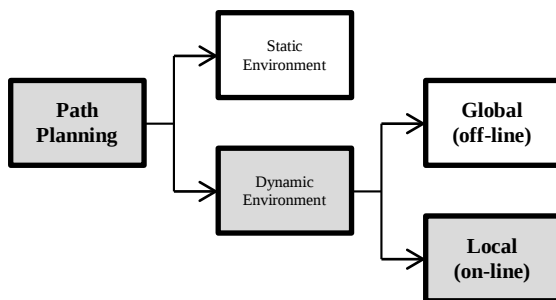


Fig.1 Classification of Path Planning

2. BACTERIAL FORAGING ALGORITHM

The Bacterial Foraging Algorithm (BFA) was first presented by Pasino in 2002 [13]. It was inspired from biological and physical living behavior of *E. coli* bacteria. The BFA is modeled after the group foraging strategy of a swarm of *Escherichia coli* bacteria, which search for nutrients so as to maximize energy obtained per unit time. Individual bacterium communicates with others by sending signals. The classical BFA consists of four main steps, namely, chemo taxis, swarming, reproduction, and elimination and dispersal [14].

2.1 Chemotactic

This process simulates the movement of an *E. coli* bacterium in search of food. It can be defined in two different ways with regard to its environment. It said to be “swimming” if it moves for a period of time in the same direction or “tumbling” if moving in altogether different direction. The rotation of flagella in each bacterium decides whether it should go for swimming or for tumbling in the entire lifetime of the bacterium. Figure 2 shows the bacterium movement [15].

Suppose $\theta(j,k,l)$ is i -th bacterium at j -th chemo-tactic, k -th reproductive and l -th elimination dispersal step. $C(i)$ is the size of the step taken in the random direction specified by the tumble. Then in computational chemotaxis, the movement of the bacterium may be represented by [16]:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) = \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

where Δ (Delta) is not unitary, and it is normalized before use it $((\Delta(i)/\sqrt{\Delta^T(i)\Delta(i)}))$.

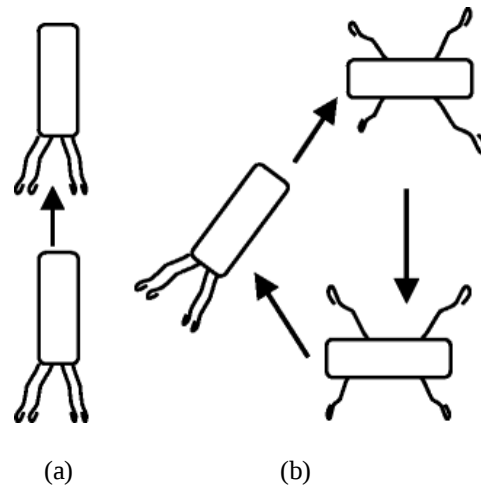


Fig.2 Schematic Diagram of *E. coli* Cell Movement Behavior (a) Counterclockwise Rotation of Flagella “Swimming” (b) Clockwise Rotation of Flagella “Tumbling”

2.2 Swarming

Interesting group behavior has been observed for several motile species of bacteria including *E. coli*, where stable spatiotemporal patterns (swarms) are formed in the semisolid nutrient medium. The bacteria placed in the richest food location tend to attract other bacteria so that they reach the desired place rapidly. The cells when stimulated by the high level of succinate release an attractant aspartate, which helps them to aggregate

into groups and thus move as concentric patterns of swarms of high bacterial density. The cell to cell, signaling in *E. coli* swarm may be represented with the function.

2.3 Reproduction

After the end of the chemotactic process, all the bacteria are sorted according to their health values in an ascending order. The least healthy bacteria that have found fewer amounts of nutrients during the chemotactic will die during the reproductive step and the healthiest bacteria will be asexually split up into two collocated bacteria. To simplify the algorithm, the number of the bacteria keeps constant in the whole process. As mathematically inferred from the aforementioned statement, generally S is chosen to be equal to $S/2$.

The fitness values for the i -th bacterium in the chemotaxis loop are accumulated and calculated by:

$$J_{health}^i = \sum_{j=1}^{N_{C+1}} J^i(j, k, l)$$

where J_{health}^i represents the health of i -th bacterium. The smaller the J_{health}^i is the healthier the bacterium is. To simulate the reproduction character in nature and to accelerate the swarming speed, all the bacteria are sorted according to their health values in an ascending order and each of the first bacteria splits into two bacteria. The characters including location and step length of the mother bacterium are reproduced to the children. Through this selection process the remaining healthier bacteria keeps constant in the whole process.

2.4 Elimination and Dispersal

In the last step, the bacteria are eliminated and dispersed to random positions in the optimization domain according to elimination-dispersal probability. A bacterial colony may be completely eliminated due to environmental extremities or may be dispersed to a nutrient-rich or noxious environment. The dispersal event takes place after a definite number of reproduction processes. First, the probability of elimination and dispersal is chosen for each bacterium and then based on the selected probability, it moves to another position in the environment. These events help the bacterium avoid being trapped into local optima [17].

3. MODEL ENVIRONMENT

The problem should be represented to search for an optimal robot path successfully. The model of the environment is inspired by Hossain [18].

The environment is a two-dimensional rectangular workspace where all the objects

including robot, obstacles, start and target point are located. It shows in Figure 3. The mobile robot is defined as a square object, which is represented by $C (C_x, C_y)$. Obstacles can be of any shape and size with a representation of each point by $O (O_x, O_y)$. It is wrapped by a circle with radius, R of the circle is chosen in accordance with the size of the obstacle. The goal is defined as a triangle object, which is represented by $G (G_x, G_y)$.

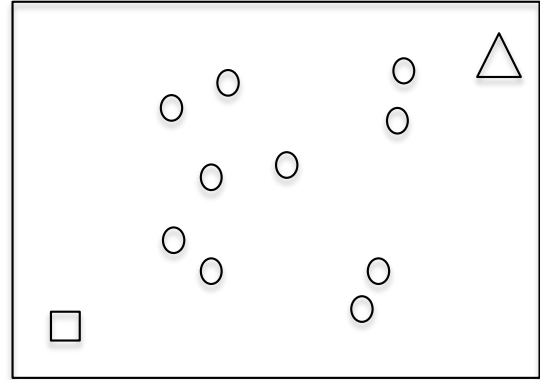


Fig.3 Model of Environment

There are some assumptions in a model environment that are constructed: the obstacles are detected by the mobile robot sensor, the path planning program runs until the goal has been achieved, and both the goal and the obstacles are dynamic. The inputs of the proposed algorithm for solve problem are initial location of the robot, the location of the destination, and position of the obstacles detected by the sensor. While next step of robot towards the feasible path by avoiding obstacle/s become the outcome of each step. The shortest pathway, which is not crashed with given obstacle/s, is the final output.

4. THE PROPOSED ALGORITHM

The flowchart of the proposed algorithm is shown in Fig. 4. The detail algorithm steps are discussed in brief.

Step 1: ($S = 1, 2, \dots, n$) virtual particles $P(P_x, P_y)$ are generated and distributed randomly on a determined area with radius R around the robot's current position, (initially, $C(C_x, C_y)$). P_x is defined with the function: if $C_x \leq O_x$ then $C_x \leq P_x \leq O_x$, otherwise if $C_x \geq O_x$ then $O_x \leq P_x \leq C_x$. And P_y is defined: if $C_y \leq O_y$ then $C_y \leq P_y \leq O_y$, otherwise if $C_y \geq O_y$ then $O_y \leq P_y \leq C_y$. The particle position (S) in time t could be defined as $P_s(t)$ and the next step is calculated as

$$P_s(t + dt) = P_s(t) + R(\Delta(t) / \|\Delta(t)\|)$$

where $\Delta(t)$ is a unit length random vector which is used to define the direction of particle and $||\Delta(t)||$ is the magnitude of the vector.

Step 2: two different strategies are combined for choosing the best particle. A repellant Gaussian cost function is assigned to each obstacle ($i = 1, 2, \dots, n$) when the sensor of robot detects obstacles. So, the formula of the function is defined as,

$$J_{obstacle} = H_{obstacle} * \exp(-W_{obstacle}(\|\theta_i(t) - P_0(t)\|^2))$$

where $H_{obstacle}$ and $W_{obstacle}$ are constant values defining height and width of the repellant, $P_0(t)$ as the obstacle position.

$$J_{goal} = H_{goal} * \exp(-W_{goal}(\|\theta_i(t) - P_G(t)\|^2))$$

where H_{goal} and W_{goal} are height and width of the attractant. So, the total cost function can be calculated as,

$$J = J_{obstacle} + J_{goal}$$

The distance error to the target could be calculated for making a decision of the best particle which can be showed as,

$$e_s^d(t) = d_s(t + dt) - d_s(t)$$

where $d_s(t)$ is the distance from particle to goal at time t which can be computed by $(d_s(t) = \|\theta_s(t) - P_G(t)\|)$, and $d_s(t + dt)$ is the distance from the same particle to the goal at time $(t + dt)$ which can be calculated by the formula, $(d_s(t + dt) = \|\theta_s(t + dt) - P_G(t)\|)$.

The cost function error should also be determined by using this one,

$$e_s^J = J(\theta_s(t + dt)) - J(\theta_s(t))$$

Step 3: all particles are sorted in ascending order of cost error. So the particle with the smallest distance error is at first, which is reflected in the best particle.

Step 4: when an obstacle is not in the robot's range, $J_{obstacle}$ is zero and the robot freely moves towards the target by choosing the first element of vector, S_{sort} , as all of them have $e_s^J(t) < 0$. Otherwise, when the sensor detects an obstacle, $J_{obstacle}$ gets a value and consequently, $124 - 2H_{goal} < e_s^J(t) < 2H_{obstacle}$. In this case, a search on S_{sort} from top to bottom is performed and the first particle with $e_s^J(t) < 0$ is selected as the best one. To get more accuracy, some modifications would have to be done, i.e. distances from robot to detected obstacles which can be evaluated as, $d_{so}(t) = \|\theta(t) - P_0(t)\|$, is also sorted in O_{sort} .

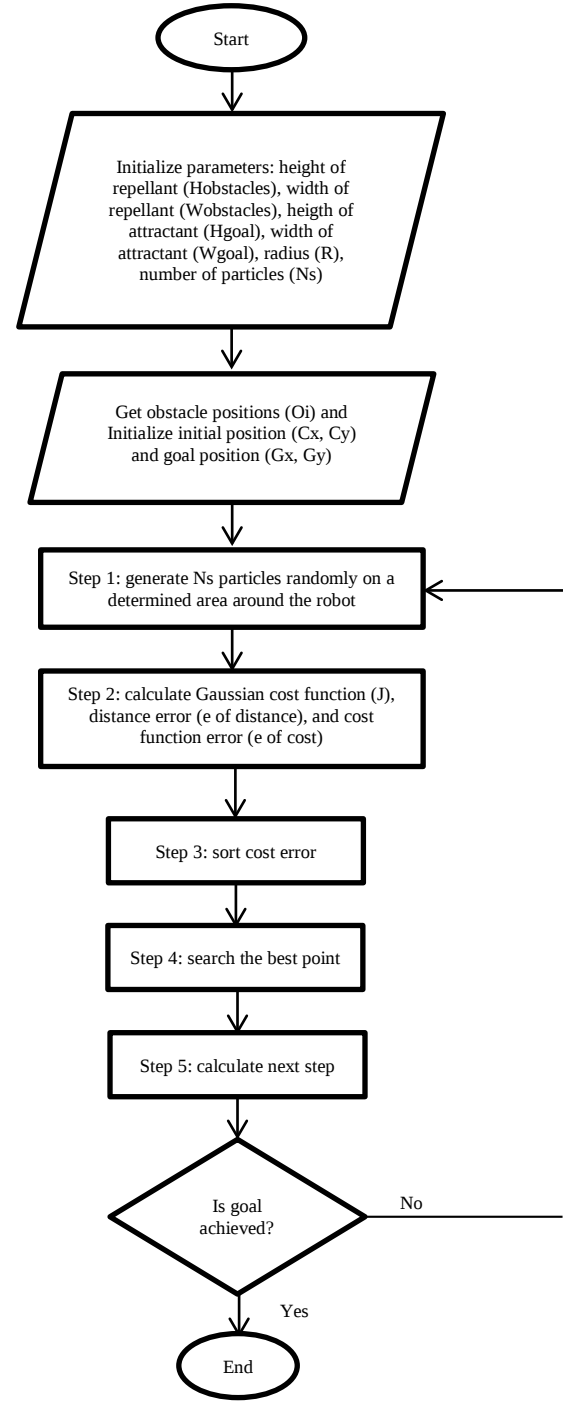


Fig. 4 Flowchart of The Proposed Algorithm

Again, if both the first and second elements of O_{sort} are on the right side, then, sort the cost errors $e_s^J(t)$ in ascended order and pick the first particle with negative error. But, if the first element is on the right side and the second on the left at a safe distance or vice versa, then, sort the cost error from the middle in ascended order and as usually select the first one with negative error. Otherwise, calculate the best one in normal procedure. Again to get optimal path, a new parameter, η can be

introduced which is a co-efficient of $H_{obstacle}$ in decision equation $e_s^j(t) < \eta$ and let the particle not to be omitted like before which can provide optimal path.

Step 5: Finally, add the particle to the preferred path and set the new position as the robot's current position. The process will be repeated until it reaches the convergence condition.

5. CONCLUSION

The topic of navigation is one of the focused problems in the correlation of autonomous mobile robots. It is necessary to have a methodology to accomplish a collision path of navigation from an initial position (state) to a target position (goal). This paper presents a new approach to optimization technique for path planning to determine a feasible, optimal, and safe path. The proposed algorithm, which inspired by BFA, generates the enhanced final path with respect to terms of time and path's feasibility. This algorithm finds a path towards the target and avoiding the obstacles using particles, which are randomly distributed, on a determined area around a robot. The criterion on which it selects the best particle is the distance to the target. By the proposed approach, the next positions of each robot are designed. In the future attempt, the authors propose many Nature Inspired Algorithm (NIA) and comparison for path planning. Some NIA can be hybridized with our proposed algorithm to improve the performance.

6. ACKNOWLEDGMENTS

Please note that this will be double-blind submission. The author's identities will be removed before sending to reviewers. The authors will not know who the reviewers are. Try to follow the reference style below. Some references from International Journal of GEOMATE would be appreciated. The benefit will be returned to all authors by increasing impact factor of the journal.

7. REFERENCES

- [1] Kubelka V, Reinstein M, Svoboda T. Improving multimodal data fusion for mobile robots by trajectory smoothing. *Robotics and Autonomous Systems*, 2016, 84: 88-96.
- [2] Lozano-Pérez, T., and M. A. Wesley. 1979. "An algorithm for planning collision-free paths among polyhedral obstacles." *Communications of the ACM* 22 (10):560-570.
- [3] Sedighi K. H., Ashenayi K., Manikas T. W., Wainwright L. R., Tai H. M., Autonomous Local Path Planning for Mobile Robot Using a Genetic Algorithm, *IEEE Evolutionary Computation* (2004).
- [4] Garcia, P., O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin. 2009. "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation." *Applied Soft Computing* 9 (3):1102-1110.
- [5] Martínez-Alfaro, H., and S. Gómez-García. 1998. "Mobile robot path planning and tracking using simulated annealing and fuzzy logic control." *Expert Systems with Applications* 15 (3):421-429.
- [6] Gemeinder, M., and M. Gerke. 2003. "GA-based path planning for mobile robot systems employing an active search algorithm." *Applied Soft Computing* 3 (2):149-158.
- [7] Huang, H., and C. Tsai. 2011. "Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm." *SICE Annual Conference (SICE)*, 2011 Proceedings of..
- [8] Contreras-Cruz, M., V. Ayala-Ramirez, and U. Hernandez-Belmonte. 2015. "Mobile robot path planning using artificial bee colony and evolutionary programming." *Applied Soft Computing* 30:319-328.
- [9] Ma, Q., and X. Lei. 2010. "Dynamic path planning of mobile robots based on ABC algorithm." In *Artificial Intelligence and Computational Intelligence*, 267-274. Springer.
- [10] Guan-Zheng, T., H. Huan, and A. Sloman. 2007. "Ant colony system algorithm for real-time globally optimal path planning of mobile robots." *Acta Automatica Sinica* 33 (3):279-285.
- [11] Liang, Y., and L. Xu. 2009. "Global path planning for the mobile robot based genetic algorithm and modified simulated annealing algorithm." *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*.
- [12] Kala, R., A. Shukla, and R. Tiwari. 2011. "Robotic path planning in the static environment using hierarchical multi-neuron heuristic search and probability based fitness." *Neurocomputing* 74 (14):2314-2335.
- [13] K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Syst. Mag.* 22 (3) (2002) 52-67.
- [14] S. Mishra, A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation, *IEEE Trans. Evolut. Computat.* 9 (1) (2005) 61-73.
- [15] W.M.Korani, Bacterial Foraging Oriented by Particle Swarm Optimization Strategy for PID Tuning, *Genetic and Evolutionary Computation*, Atlanta, GA, USA, 2008, 1-4.

- [16] M. Tripathy, S. Mishra, Bacteria foraging-based solution to optimize both real power loss and voltage stability limit, *IEEE Trans. Power Syst.* 22 (1) (2007) 240–248.
- [17] Hossain, M. A., and Ferdous I. 2015. "Autonomous robot path planning in a dynamic environment using a new optimization technique inspired by bacterial foraging technique." *Elsevier Robotics and Autonomous Systems* Vol. 64: 137-141.

8. AUTHOR'S BIOGRAPHY

Yisti Vita Via is a lecturer in Informatics Engineering, Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur, Indonesia. She obtained her Bachelor from Information Technology, Politeknik Elektronika Negeri Surabaya, and Master from Informatics Engineering, Institut Teknologi Sepuluh Nopember, Indonesia. Her research interests include Artificial Intelligent, Nature Inspired Algorithm, Neural Network, and Path Planning of

Mobile Robots. Her contact E-mail is yistivia.if@upnjatim.ac.id.

9. AUTHOR'S CONTRIBUTIONS

Yisti Vita Via: Conception, design, acquisition, analysis, and interpretation of data and drafting the article. Henni Endah Wahanani and Salamun Rohman Nudin: Critical reviewing and final approval of the version to be submitted.

10. ETHICS

This article is original and contains unpublished material. All of the other authors have read and approved the manuscript and no ethical issues involved.

Copyright © Int. J. of GEOMATE. All rights reserved, including the making of copies unless permission is obtained from the copyright proprietors.
