

A Face-off - Classical and Heuristic-based Path Planning Approaches

Vishal Chand

School of Computing, Information and
Mathematical Sciences
University of the South Pacific
Fiji
Email: vishal.chand@usp.ac.fj

Avinesh Prasad

School of Computing, Information and
Mathematical Sciences
University of the South Pacific
Fiji
Email: avinesh.prasad@usp.ac.fj

Kaylash Chaudhary

School of Computing, Information and
Mathematical Sciences
University of the South Pacific
Fiji
Email: kaylash.chaudhary@usp.ac.fj

Bibhya Sharma

School of Computing, Information and
Mathematical Sciences
University of the South Pacific
Fiji
Email: sharma_b@usp.ac.fj

Samlesh Chand

Ministry of Education, Heritage and
Arts
Suva, Fiji
Email: samleshc@yahoo.com

Abstract— Robot path planning is a computational problem to find a valid sequence of configurations to move a robot from an initial to a final destination. Several classical and heuristic-based methods exist that can be used to solve the problem. This paper compares the performance of a classical method based on potential field, Lyapunov-based Control Scheme, with those of the standard and stepping ahead Firefly Algorithms. The performance comparison is based on the optimal path distance and time. The results show that the stepping ahead Firefly algorithm finds a shorter path in lesser duration when compared with the Lyapunov-based method. The LbCS also inherently faces the local minima problem when the start, target, and obstacle's center coordinates are collinear. This problem is solved using the firefly algorithm where the diversification of the fireflies helps escape local minima.

Keywords—Firefly Algorithm, point-mass robot, optimization, Lyapunov based control scheme, heuristic approach, stepping ahead, local minima

I. INTRODUCTION

Path planning of autonomous robots has been an active research area for more than two decades [11], [12], [13]. Path planning attempts to find a short, collision-free path for robots from the starting position towards the predefined ending location [50]. Some notable real-life applications of motion planning include Advanced Driver Assistance Systems [44], bacterial foraging [45], avoidance of autonomous vehicles [42], chemical process regulation [41], obstacle minimum time maneuvering for Uninhabited Combat Air Vehicle [43] and industrial automation [40]. Path planning is important for the navigation and motion control of autonomous robot manipulators [10]. In computational complexity theory, path planning is classified as an NP (nondeterministic polynomial time) complete problem [2]. For example, in Kino dynamic motion planning, where the velocity, acceleration and torque bounds need to be satisfied together with obstacle avoidance, the time can increase exponentially [38].

Robot path planning and obstacle avoidance is also treated as one of the common optimization problems in the literature [46], [47], [48]. Path planning can be seen as finding the solution of an objective function [49], where local and global extrema are of interest in defined regions [6].

According to Yang, optimization algorithms can be classified into two main categories, namely deterministic and stochastic [1]. Deterministic algorithms will produce the same set of solutions if the iterations start with the same initial guess. On the other hand, stochastic algorithms often produce different solutions, even with the same initial starting point. The results will eventually converge to the same optimal solution within a given accurateness [1].

Solving the robot path planning problem can be divided into classical and heuristic approaches [18], [33]. Some of the classical methods used to solve the path planning problem include cell decomposition [34], Artificial Potential Field (APF) [35], virtual force field [36], Subgoal Network (SN) method [37], and Road Map [17]. A popular potential field-based method used to solve the motion planning and control problem is the Lyapunov-based Control Scheme (LbCS) [14], [15], [16]. LbCS is a time-invariant nonlinear algorithm which is used to derive the velocity or acceleration based controllers for a robot to navigate safely in the workspace, reaching its desired target while avoiding the obstacles in the workspace [17].

The heuristic-based method may be defined as a procedure for solving problems by an intuitive approach. The structure of the problem can be interpreted and exploited intelligently using methods such as neural network, genetic algorithm, particle swarm optimization, ant colony optimization to obtain a reasonable solution [19]. While heuristic methods are based on problems, metaheuristic techniques are problem-independent [20]. Metaheuristic algorithms are mostly inspired by nature and have multiple interacting agents [51]. Swarm Intelligence (SI) based algorithms is a subcategory of metaheuristic algorithm. SI was developed by imitating the swarm-intelligence characteristics of humans, birds, bees, ants, fish and other biological agents [2], [3], [21]. Swarm intelligence refers to a research field concerned with collective behavior within self-organized and decentralized systems [4].

Firefly algorithm, as proposed by Yang in 2008, is a swarm intelligent algorithm that is based on the grouping behavior of fireflies [5]. It belongs to the stochastic algorithm category, which uses meta-heuristic techniques. The algorithm simulates the characteristic behavior of the

fireflies. Fireflies' population shows characteristic luminary flashing activities to function as attracting the partners, communication, and risk warning for predators [7]. The algorithm works under the assumption that all fireflies are unisexual and all fireflies have attracting potential for each other. The attractiveness is directly proportionate to the brightness level of individuals [8]. Hence, the brighter fireflies attract the less bright ones by moving it towards them which creates the path. If there is no brighter firefly than a certain firefly, it moves randomly [9]. In literature, there are several instances where the firefly algorithm has been modified or hybridized to improve optimization [22], [23], [24]. Some recent examples include Quaternion FA [28], [32], Modified FA [29], New Dynamic FA [30] and Chaotic FA [31]. In [24], Nand et al. modified the firefly algorithm by introducing an additional step to the algorithm that finds the solution by 'Stepping Ahead' from the fireflies' usual movement. This paper will adopt this stepping ahead strategy to find feasible and stabilizing solutions to the motion planning problem.

The objective of this paper is to simulate the path of a point mass robot in a known environment with a fixed obstacle using three different methods: LbCS, firefly algorithm (standard), and modified firefly algorithm (*stepping ahead*). The path distance and time taken for the convergence of each instance is recorded and compared. This paper also attempts to solve the local minima problem faced by LbCS method when exposed to a collinear state using the firefly algorithm method. The contribution of this paper is that it reaffirms the *stepping ahead* firefly algorithm as a better algorithm when compared to LbCS in terms of time complexity and path distance.

The rest of the paper is organized as follows. Section II explains how LbCS works. Section III and IV describes the standard and *Stepping Ahead* firefly algorithm and the objective functions used for path optimization. In section V, we discuss the local minima problem faced by LbCS. Section VI discusses the simulation results, while Section VII concludes the paper and essays suggestions for future work.

II. LYAPUNOV-BASED CONTROL SCHEME (LBCS)

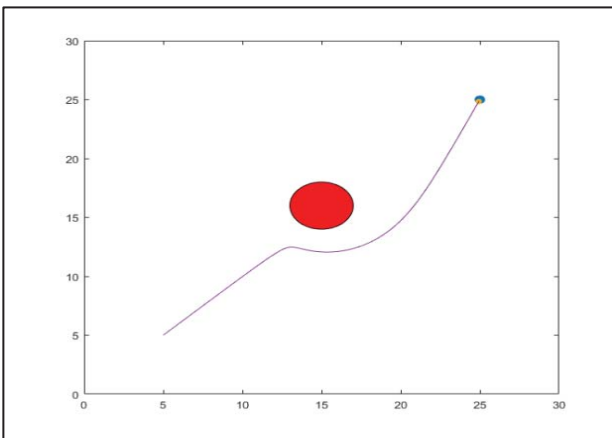


Fig. 1. The trajectory of the robot in the presence of a fixed obstacle using the LbCS.

For path planning, we have utilized target attraction and obstacle avoidance functions based on LbCS [24]. The governing principle behind the LbCS is to design a Lyapunov function (energy function) which comprises of the attractive (for target convergence) and repulsive potential field functions (for obstacle avoidance). These functions are basically the Euclidean measures between the robot and its target, or between the robot and the obstacles. We note that in LbCS, the repulsive potential field function is a fraction consisting of tuning (control) parameter in the numerator and avoidance function in the denominator. Finally, the sum of the attractive and repulsive potential field functions will form a Lyapunov function, from which, the velocity-based controls are extracted.

The following definitions and equations are adapted from [24]:

Definition 1: A point-mass P in the Euclidean space \mathbb{R}^2 is a circular disk of radius $r_p \geq 0$ and is positioned at $(x(t), y(t))$ at time $t \geq 0$. The point-mass is the set (as defined in [24])

$$P = \{(z_1, z_2) \in \mathbb{R}^2: (z_1 - x)^2 + (z_2 - y)^2 \leq r_p^2\}.$$

Suppose the instantaneous velocity of the point-mass be given as $(u_1(t), u_2(t))$, then the kinematic equation for the point-mass is given as:

$$\begin{aligned} \dot{x}(t) &= u_1(t), \\ \dot{y}(t) &= u_2(t). \end{aligned}$$

Here $u_1(t)$ and $u_2(t)$ are treated as controllers, which will be designed such that the point mass robot can navigate safely to the desired target while avoiding collision with stationary obstacles in the workspace.

Definition 2: The target of the point-mass P is a disk in \mathbb{R}^2 with center (p_1, p_2) and radius r_T , and is described as the set

$$T = \{(z_1, z_2) \in \mathbb{R}^2: (z_1 - p_1)^2 + (z_2 - p_2)^2 \leq r_T^2\}$$

The point-mass is required to move from an initial position to its designated target and remain there as $t \rightarrow \infty$. Hence a tentative attractive potential function is

$$V(x, y) = \frac{1}{2} [(x - p_1)^2 + (y - p_2)^2], \quad (1)$$

which is a measure of the Euclidean distance between the point mass position and the target position.

Definition 3: The l th stationary circular solid obstacle is a disk with center (o_{l1}, o_{l2}) and radius r_{o_l} which is defined as the set

$$O_l = \{(z_1, z_2) \in \mathbb{R}^2: (z_1 - o_{l1})^2 + (z_2 - o_{l2})^2 \leq r_{o_l}^2\}$$

for $l = 1, 2, \dots, q$.

The robot must avoid stationary obstacles while on its way to the target. Hence, the following avoidance function is used, which will appear in the Lyapunov function as the denominator of the repulsive potential function.

$$W_l(x, y) = \frac{1}{2} [(x - o_{l1})^2 + (y - o_{l2})^2 - (r_p + r_{o_l})^2] \quad (2)$$

The attractive and avoidance functions as given in (1) and (2) are combined to form the total potentials or the Lyapunov function as

$$L(x, y) = V(x, y) + \sum_{l=1}^q \frac{\beta_l V(x, y)}{W_l(x, y)},$$

where $\beta_l > 0$ are called control parameters.

According to the Direct Method of Lyapunov, the derivative of the Lyapunov function (with respect to time t) should be non-negative. Hence, by introducing the convergence parameters $\delta_1 > 0$ and $\delta_2 > 0$, we let $\dot{L}(x, y) \leq 0$ to obtain the controllers $u_1(t)$ and $u_2(t)$ as:

$$u_1(t) = -\frac{1}{\delta_1} \left(1 + \sum_{l=1}^q \frac{\beta_l}{W_l} \right) (x - p_1) - \sum_{l=1}^q \frac{\beta_l}{W_l^2} (x - o_{l1}),$$

$$u_2(t) = -\frac{1}{\delta_2} \left(1 + \sum_{l=1}^q \frac{\beta_l}{W_l} \right) (y - p_2) - \sum_{l=1}^q \frac{\beta_l}{W_l^2} (y - o_{l2}).$$

III. FIREFLY ALGORITHM

```

Objective function  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ 
Initialize a population of fireflies  $x_i$  ( $i = 1, 2, \dots, n$ )
Define light absorption coefficient  $\gamma$ 
while ( $t < MaxGeneration$ )
for  $i = 1: n$  all n fireflies
for  $j = 1: i$  all n fireflies
    Light intensity  $I_i$  at  $x_i$  is determined by  $f(x_i)$ 
    if ( $I_j > I_i$ )
        Move firefly  $i$  towards  $j$  in all  $d$  dimensions
    end if
    Attractiveness varies with distance  $r$  via  $exp(-\gamma r)$ 
    Evaluate new solutions and update light intensity
end for  $j$ 
end for  $i$ 
Rank the fireflies and find the current best
end while
Post-processing the results and visualization;

```

Fig. 2. Firefly Algorithm as adapted from [1]

The point mass robot, the target of the robot and the obstacle are constructed as defined in II. Each firefly is compared with the rest of the fireflies based on the cost function. The cost function is defined as:

$$C_x = \omega_1 / \sqrt{((o_{l1} - x_1)^2 + (o_{l2} - x_2)^2)} + \omega_2 / \sqrt{((p_1 - x_1)^2 + (p_2 - x_2)^2)} \quad (3)$$

where C_x in (3) is the cost of the firefly which is determined by the distance. o_{l1} and o_{l2} are the center coordinates of the obstacle. x_1 and x_2 are the coordinates of the firefly for which the cost is calculated. p_1 and p_2 are the coordinates of the target. ω_1 is the weightage of the distance between the firefly and the obstacle and ω_2 is the weightage of the distance the firefly has with the target. Both weights are essential in determining each firefly's

cost, which eventually decides on the most optimized firefly. If $C_{x_j} > C_{x_i}$ then the firefly i moves towards firefly j . The convention here is of the objective function to minimize the cost. The magnitude of the movement is based on the attractiveness coefficient, β , of the attracting firefly, which roots from the flashing characteristics of the firefly.

As explained by Hashmi et al. [25], FA is based on the flashing characteristic of fireflies. The basic concept is that the light intensity is maximum at the source and gradually decreases with distance. The brightness of a firefly relative to another is defined by the inverse square law (4) where the intensity decreases at the rate of the squared distance between the fireflies.

$$I(r) = I_s / r^2 \quad (4)$$

In (4), $I(r)$ is the light intensity at a distance r from the firefly and I_s is the intensity of the firefly at the source. The equation can be redefined to include the inverse square law and the light absorption coefficient of the medium using the following Gaussian form:

$$I(r) = I_0 e^{-\gamma r^2} \quad (5)$$

where $I(r)$ is the light intensity at a distance r , I_0 is the light intensity at the source, γ is the light absorption coefficient, and r is the viewer's distance from the source. As we know, a firefly's attractiveness is proportional to the light intensity seen by adjacent fireflies; thus, the attractiveness β of firefly can be determined by the equation:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (6)$$

where β is the attractiveness of a firefly and β_0 is the constant and presents the attractiveness at $r = 0$.

IV. STEPPING AHEAD FIREFLY ALGORITHM

```

Step 1: Initialize population of fireflies
Random partition  $x_i$  ( $i = 1, 2, \dots, n$ )
Step 2: Formulate light intensity  $I$  so that it is associated with  $f(x)$ 
Step 3: Define absorption coefficient  $\gamma$ 
while  $t < MaxGeneration$  do
    foreach  $i = 1: n$  (all n fireflies) do
        foreach  $j = 1: i$  (n fireflies) do
            if  $I_j \leq I_i$  then
                Vary attractiveness with distance  $r$  via  $exp(-\gamma r)$ ;
                Step 4: Stepping ahead
                move firefly  $i$  further than  $j$  that is  $j$  distance plus
                distance difference  $j$  and  $i$ ;
                Evaluate new solutions and update light intensity;
                if new solution not better then
                    move firefly  $i$  in relation to  $j$  that is  $i$  distance plus
                    distance difference  $j$  and  $i$ ;
                    Evaluate new solutions and update light intensity;
                end
            else
                nothing
            end
        end
        Rank fireflies and find the current best;
    end
    Post-processing the results and visualization;

```

Fig. 3. Stepping Ahead FA. Adapted from [23].

Fig. 3 shows *Stepping Ahead* Firefly Algorithm where the original algorithm has been modified for better optimization. In this algorithm, the original 3 steps of the FA have been retained. Specifically, they are 1. Initializing the firefly population, 2. Formulating the light intensity, I so that it is associated with the cost function and 3. Defining of the absorption coefficient.

The 4th step has been modified so that the new solution is found in relation to the firefly j where the movement of the firefly is further away than the usual movement of firefly i closer to the best solution as in standard FA. The term coined for this approach is *Stepping Ahead*.

V. LOCAL MINIMA PROBLEM FACED BY LbCS

Local Minima is one of the issues faced by the APF methods including LbCS in robot path planning, as shown in Fig. 4. This happens when the coordinates of the start point, target, and obstacle are in collinear. A local minimum or minima is a suboptimal equilibrium point at which system error is non-zero, and the hidden output matrix is singular [26]. As stated in [27], when the Lyapunov function includes the obstacle situation, it may have local minima in the open subset. If the controlled system closes near the local minima, it will fall in the minima, and it cannot escape.

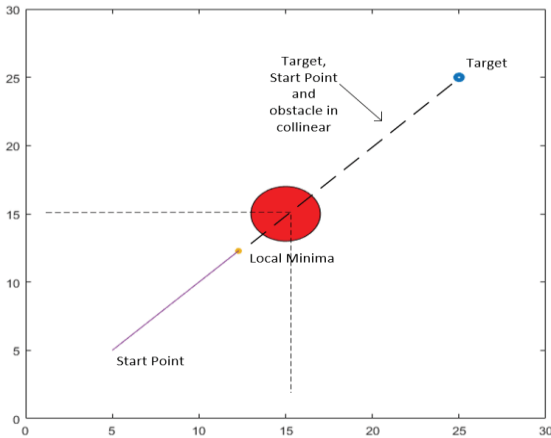


Fig. 4. Local minima problem faced by LbCS

The heuristic-based optimization algorithm like FA solves this issue.

VI. SIMULATION RESULTS AND DISCUSSION

TABLE I. AVERAGE PATH DISTANCE AND CONVERGENCE TIME USING VARIOUS ALGORITHMS

LbCS		Stepping-Ahead Firefly		Standard Firefly	
Distance	Time(s)	Distance	Time(s)	Distance	Time(s)
30.386	62.059	29.818	3.207	30.297	19.564

Table I shows the point-mass robot's performance using each algorithm in a 30×30 grid with a starting coordinate of (5,5) and ending coordinate of (25,25). Stepping-ahead

firefly and standard firefly algorithm were iterated 90 times each. The average distance and time were recorded. LbCS did not require any iterations as all runs are in an exact scenario and provide the same results.

TABLE II. BEST AND WORST CASE PATH DISTANCE AND CONVERGENCE TIME OF FIREFLY ALGORITHMS

	Stepping-Ahead Firefly		Standard Firefly	
	Distance	Time(s)	Distance	Time(s)
Best Case	29.161	2.075	29.984	19.007
Worst Case	30.627	15.290	30.699	20.486

Table II shows the best and worst-case performance of all the firefly algorithms. The best and worst case of LbCS is acquired from Table I. The best-case time of *stepping ahead* was 2.075s, which is considerably low compared to the best case timing of LbCS and standard FA. Another interesting observation is that the range of standard FA's best and worst cases is quite small compared to the *stepping ahead* algorithm.

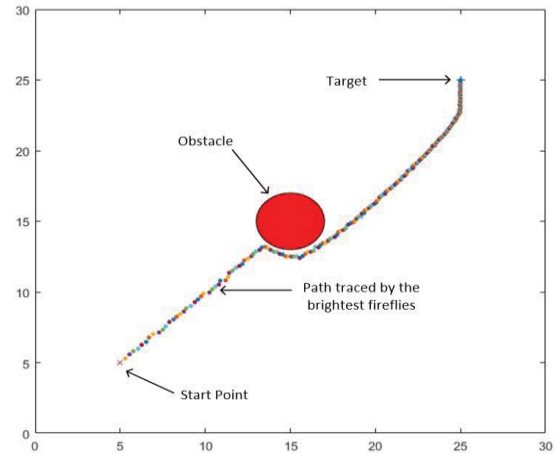


Fig. 5. Robot path traced by standard FA.

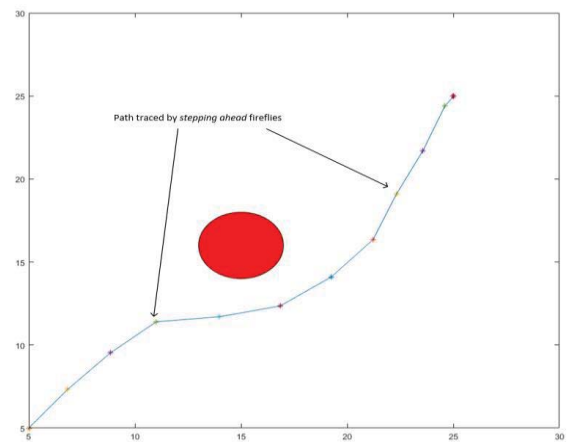


Fig. 6. Robot path traced by *stepping-ahead* FA

Both *stepping-ahead* and standard firefly exhibited better performance both in time complexity and path distance when compared to LbCS. Overall, *stepping-ahead*

firefly's performance was slightly better than the other two methods. The execution time of 3.207s was just 5% of LbCS's time and 16% of standard firefly's time. Also, the path distance of stepping-ahead firefly algorithm was 1.86% less when compared with LbCS. We note that in LbCS, the control/convergence parameters play a huge role in determining the time and safety of the robot. A large control parameter ensures that the robot avoids an obstacle from a greater distance. Similarly, large values of convergence parameters will decrease the rate of convergence of the robot to its target whereas very small or unsuitable values will enable the robot to converge to the target quickly and thus compromising its safety. In literature, brute-force procedure is mainly used to determine the values of these parameters and this method can adversely affect time optimality.

Fig. 1, Fig. 2, and Fig. 3 show the path traced by the point mass robot when executed under identical conditions. One of the discussion points is the safety parameter of the robot in relation to the obstacle. The safety parameter of the firefly path from the obstacle is based on the weights, ω_1 and ω_2 set in equation (6). These weights are set manually during the simulation. The path traced by all three algorithms also vary. LbCS has the smoothest path traced (since the differential equations governing the robotic motion are continuous for all time $t > 0$) while stepping-ahead FA has an uneven path due to fewer fireflies involved in path creation.

VII. CONCLUSION

This paper used the Lyapunov base Control Scheme (LbCS) and Firefly Algorithm (FA) (standard and modified) to solve the robot path planning problem. Firstly, the workspace for the robot was defined together with the robot and obstacle. Then the objective function for each of the methods was defined and used in the algorithm. Each of the three methods was executed, recording the time of each iteration. The path distance during each execution was also recorded and analyzed.

During the execution using LbCS when the start point, target, and obstacle were in collinear position, the robot got trapped in the local minima. When faced with the same scenario, FA was able to maneuver out of the trap because, during the initialization phase, the fireflies are randomly distributed which promotes diversification [39]. Even during the movement phase the fireflies do a local search and do not memorize the global best. This is one of the reasons why FA does better than LbCS when faced with local minima. It has been inferred that the meta-heuristic based algorithm like FA is more efficient in comparison to classical path planning methods when confronted with problems such as local minima.

The *stepping ahead* FA outperformed the standard FA and LbCS which shows that the new algorithm has the potential to be used and further modified in the future. The idea present in this algorithm is exemplary which generally states that the algorithm should not settle for the best solution in the current region but to take one more step and check for the better solution.

One of the issues which can be solved in future research is to optimize ω_1 and ω_2 values instead of assigning it

manually. The optimization target can be the path distance and the time it takes to form the path. The research can be further extended to fine-tune stepping-ahead algorithm to provides a smoother path for the robot path planning problem.

VIII. REFERENCES

- [1] Yang, X.S., 2010. Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation*, 2(2), pp.78-84.
- [2] Brand M, Masuda M, Wehner N, Yu XH. Ant colony optimization algorithm for robot path planning. In 2010 international conference on computer design and applications 2010 Jun 25 (Vol. 3, pp. V3-436). IEEE.
- [3] Fister, I., Fister Jr, I., Yang, X.S. and Brest, J., 2013. A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, 13, pp.34-46.
- [4] Yang, X.S., 2009, October. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169-178). Springer, Berlin, Heidelberg.
- [5] Törn, A. and Žilinskas, A., 1989. *Global optimization* (Vol. 350). Berlin: Springer-Verlag.
- [6] Golchi, M.M., Saraeian, S. and Heydari, M., 2019. A hybrid of firefly and improved particle swarm optimization algorithms for load balancing in cloud environments: Performance evaluation. *Computer Networks*, 162, p.106860.
- [7] Senthilnath, J., Omkar, S.N. and Mani, V., 2011. Clustering using firefly algorithm: performance study. *Swarm and Evolutionary Computation*, 1(3), pp.164-171.
- [8] Jati, G.K., 2011, September. Evolutionary discrete firefly algorithm for travelling salesman problem. In *International conference on adaptive and intelligent systems* (pp. 393-403). Springer, Berlin, Heidelberg.
- [9] Brand, M., Masuda, M., Wehner, N. and Yu, X.H., 2010, June. Ant colony optimization algorithm for robot path planning. In *2010 international conference on computer design and applications* (Vol. 3, pp. V3-436). IEEE.
- [10] Prasad, A., Sharma, B. and Vanualailai, J., 2012. Motion planning and control of autonomous robots in a two-dimensional plane. *World Acad. Sci. Eng. Technol*, 6(12), pp.1163-1168.
- [11] Janglová, D., 2004. Neural networks in mobile robot motion. *International Journal of Advanced Robotic Systems*, 1(1), p.2.
- [12] Prasad A, Sharma B, Vanualailai J, Kumar SA. A geometric approach to target convergence and obstacle avoidance of a nonstandard tractor - trailer robot. *International Journal of Robust and Nonlinear Control*. 2020 Sep 10;30(13):4924-43.
- [13] Raghunwaiya K, Sharma B, Vanualailai J. Leader-follower based locally rigid formation control. *Journal of Advanced Transportation*. 2018 Feb 25;2018.
- [14] Shen, C., Shi, Y. and Buckham, B., 2017. Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control. *IEEE Transactions on Industrial Electronics*, 65(7), pp.5796-5805.
- [15] Sharma B, Singh S, Vanualailai J, Prasad A. Globally rigid formation of n-link doubly nonholonomic mobile manipulators. *Robotics and Autonomous Systems*. 2018 Jul 1;105:69-84.
- [16] Sharma, B.N., Raj, J. and Vanualailai, J., 2018. Navigation of carlike robots in an extended dynamic environment with swarm avoidance. *International Journal of Robust and Nonlinear Control*, 28(2), pp.678-698.
- [17] Mac, T.T., Copot, C., Tran, D.T. and De Keyser, R., 2016. Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86, pp.13-28.

- [18] Silver, E.A., Victor, R., Vidal, V. and de Werra, D., 1980. A tutorial on heuristic methods. *European Journal of Operational Research*, 5(3), pp.153-162.
- [19] Mishra, S.K., Puthal, D., Rodrigues, J.J., Sahoo, B. and Dutkiewicz, E., 2018. Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications. *IEEE Transactions on Industrial Informatics*, 14(10), pp.4497-4506.
- [20] Martens, D., Baesens, B. and Fawcett, T., 2011. Editorial survey: swarm intelligence for data mining. *Machine Learning*, 82(1), pp.1-42.
- [21] Kavousi-Fard, A., Samet, H. and Marzbani, F., 2014. A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting. *Expert systems with applications*, 41(13), pp.6047-6056.
- [22] Fister, I., Yang, X.S., Brest, J. and Fister Jr, I., 2013. Modified firefly algorithm using quaternion representation. *Expert Systems with Applications*, 40(18), pp.7220-7230.
- [23] Nand, R., Chaudhary, K. and Sharma, B., 2020, July. Stepping ahead based hybridization of meta-heuristic model for solving Global Optimization Problems. In 2020 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.
- [24] Chand R, Prasad A, Sharma B, Vanualailai J. Landmark aided navigation of a point-mass robot via Lyapunov-based control scheme. In Asia-Pacific World Congress on Computer Science and Engineering 2014 Nov 4 (pp. 1-7). IEEE.
- [25] Hashmi A, Goel N, Goel S, Gupta D. Firefly algorithm for unconstrained optimization. *IOSR J Comput Eng*. 2013 May;11(1):75-8.
- [26] Choi B, Lee JH, Kim DH. Solving local minima problem with large number of hidden nodes on two-layered feed-forward artificial neural networks. *Neurocomputing*. 2008 Oct 1;71(16-18):3640-3.
- [27] INOUE A, SEKIGUCHI K. Lyapunov function-based obstacle avoidance scheme for a two-wheeled mobile robot. *Journal of Control Theory and Applications*. 2008(4):10.
- [28] Fister I, Yang XS, Brest J, Fister Jr I. Modified firefly algorithm using quaternion representation. *Expert Systems with Applications*. 2013 Dec 15;40(18):7220-30.
- [29] Kavousi-Fard A, Samet H, Marzbani F. A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting. *Expert systems with applications*. 2014 Oct 1;41(13):6047-56.
- [30] Wang H, Wang W, Cui Z, Zhou X, Zhao J, Li Y. A new dynamic firefly algorithm for demand estimation of water resources. *Information Sciences*. 2018 Apr 1;438:95-106.
- [31] Yelghi A, Köse C. A modified firefly algorithm for global minimum optimization. *Applied Soft Computing*. 2018 Jan 1;62:29-44.
- [32] Tavakkoli-Moghaddam R, Torabi N, Ghaseminejad A. A Quaternion Firefly Algorithm to Solve a Multi-row Facility Layout Problem (RESEARCH NOTE). *International Journal of Engineering-Transactions B: Applications*. 2015 Nov 19;28(11):1605-13.
- [33] Ab Wahab MN, Nefti-Meziani S, Atyabi A. A comparative review on mobile robot path planning: Classical or meta-heuristic methods?. *Annual Reviews in Control*. 2020 Oct 16.
- [34] Kloetzer M, Mahulea C, Gonzalez R. Optimizing cell decomposition path planning for mobile robots using different metrics. In 2015 19th International Conference on System Theory, Control and Computing (ICSTCC) 2015 Oct 14 (pp. 565-570). IEEE.
- [35] Lee MC, Park MG. Artificial potential field based path planning for mobile robots using a virtual obstacle concept. In Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003) 2003 Jul 20 (Vol. 2, pp. 735-740). IEEE.
- [36] Bortoff SA. Path planning for UAVs. In Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334) 2000 Jun 28 (Vol. 1, No. 6, pp. 364-368). IEEE.
- [37] Baginski B, Eldracher M. Path planning with neural subgoal search. In Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94) 1994 Jun 28 (Vol. 5, pp. 2732-2736). IEEE.
- [38] Hsu D, Kindel R, Latombe JC, Rock S. Randomized kinodynamic motion planning with moving obstacles. *The International Journal of Robotics Research*. 2002 Mar;21(3):233-55.
- [39] He L, Huang S. Modified firefly algorithm based multilevel thresholding for color image segmentation. *Neurocomputing*. 2017 May 31;240:152-74.
- [40] Büskens C, Knauer M. Higher order real-time approximations in optimal control of multibody-systems for industrial robots. *Multibody System Dynamics*. 2006 Feb 1;15(1):85-106.
- [41] Diehl M, Findeisen R, Allgöwer F. A stabilizing real-time implementation of nonlinear model predictive control. In Real-time PDE-constrained optimization 2007 Jan 1 (pp. 25-52). Society for Industrial and Applied Mathematics.
- [42] Frazzoli E, Dahleh MA, Feron E. Real-time motion planning for agile autonomous vehicles. *Journal of guidance, control, and dynamics*. 2002 Jan;25(1):116-29.
- [43] Kumar RP, Dasgupta A, Kumar CS. Real-time optimal motion planning for autonomous underwater vehicles. *Ocean engineering*. 2005 Aug 1;32(11-12):1431-47.
- [44] Bertolazzi E, Biral F, Da Lio M. Future advanced driver assistance systems based on Optimal Control: the influence of "risk functions" on overall system behavior and on prediction of dangerous situations. In IEEE Intelligent Vehicles Symposium, 2004 2004 Jun 14 (pp. 386-391). IEEE.
- [45] Jati A, Singh G, Rakshit P, Konar A, Kim E, Nagar AK. A hybridisation of improved harmony search and bacterial foraging for multi-robot motion planning. In 2012 IEEE Congress on Evolutionary Computation 2012 Jun 10 (pp. 1-8). IEEE.
- [46] Saska M, Macas M, Preucil L, Lhotska L. Robot path planning using particle swarm optimization of Ferguson splines. In 2006 IEEE Conference on Emerging Technologies and Factory Automation 2006 Sep 20 (pp. 833-839). IEEE.
- [47] Castilho O, Trujillo L. Multiple Objective Optimization Genetic Algorithms For Path Planning In Autonomous Mobile Robots. *Int. J. Comput. Syst. Signals*. 2005;6(1):48-63.
- [48] Tuba E, Strumberger I, Zivkovic D, Bacanin N, Tuba M. Mobile robot path planning by improved brain storm optimization algorithm. In 2018 IEEE Congress on Evolutionary Computation (CEC) 2018 Jul 8 (pp. 1-8). IEEE.
- [49] Ratliff N, Zucker M, Bagnell JA, Srinivasa S. CHOMP: Gradient optimization techniques for efficient motion planning. In 2009 IEEE International Conference on Robotics and Automation 2009 May 12 (pp. 489-494). IEEE.
- [50] Atyabi A, Powers DM. Review of classical and heuristic-based navigation and path planning approaches. *International Journal of Advancements in Computing Technology*. 2013 Oct 1;5(14):1.
- [51] Yang XS. Nature-inspired metaheuristic algorithms: Success and new challenges. arXiv preprint arXiv:1211.6658. 2012 Nov 28.