Bayesian probabilistic modeling in robosoccer environment for robot path planning

Diana Steffi[1], Shilpa Mehta[1], Kanyakumari Ayyadurai Venkatesh[2]
[1]Department of Electronics and Communication Engineering, Presidency University, Bangalore, India
[2]Department of Mathematics and Computer Science, Alliance University, Bangalore, India

Article Info

Article history:

Received Feb 24, 2023
Revised Jul 13, 2023
Accepted Aug 2, 2023

Keywords:

Bayesian optimization algorithm
Dynamic environments
Optimization algorithms
Path planning
Robotics

ABSTRACT

The main goal of a route planning approach is to find a trajectory that safely transports the robot from one site to the next. Furthermore, it should provide an energy-efficient path so the computer can calculate it rapidly. This study develops a path-planning system for robots to approach the ball without collision. The Bayesian optimization algorithm (BOA) is used to identify the shortest path between the robot and the ball. BOA employs a probabilistic model to seek the optimum of an uncertain objective function efficiently. The performance of the BOA-based path planning system is compared to other optimization algorithms such as genetic algorithm, ant colony optimization, and firefly algorithm. BOA's acquisition functions such as expected improvement, probability of improvement (PI), and upper confidence bound, are investigated. The exact locations of the robots and the ball are fed into optimization problems to discover the optimum path. The results reveal that the BOA system outperforms other systems in terms of computational time for planning the optimum path in dynamic situations and BOA-PI is the fastest algorithm.

Corresponding Author:

Diana Steffi
Department of Electronics and Communication Engineering, Presidency University
Bangalore 560064, India
Email: steffiseelan@gmail.com

1. INTRODUCTION

The topic of path planning or motion planning is one of the most basic in robotics, and it has been the focus of study for the last two decades. It is essential to map a path between two places or points for a robot if such a thing exists. The path should steer clear of any obstacles in the surrounding area and is often optimized in some manner, for instance, concerning the amount of time it takes or the amount of energy it requires [1]. This issue may be seen in many different contexts, such as a vehicular car-like robot and a translational robot. The difficulty of designing a route or path free from collisions becomes much more complex as the degree of freedom increases. After the robot has chosen a path that is appropriate for travel, for example, by giving power to the wheels, the robot will proceed along that path. When the robot follows the course that was programmed for it, it may conclude that the path that was mapped initially out is no longer viable because an obstruction is now in its way. This might be due to a need for more information about the surrounding area or to the obstruction being moveable and having migrated into a new location after the path was established. Some obstacle avoidance is required to stop the robot when a collision occurs. The obstacle avoidance, motion, and task planning processes may be carried out randomly. A network of interconnected components comprising a hybrid robotic planning system makes it possible to immerse a robot in a constantly changing environment. If the sub-tasks are planned, the visualization of the complexity of a scenario is a simple task [2]. In this work, an efficient path-planning strategy is developedusing a probabilistic modeling approach. Recently, the need for capable path-planning systems has been increasing. The manual operations by people or animals are supplemented by robots that improve productivity and safety. For instance, in industrial assembly factories, robots are employed to move components into positions, do welding, and carry out several other tasks. They are controlled by programs to accomplish a purpose. The proposed path-planning strategy helps robots reach the ball without collision.

A path planning technique is discussed in [3] under obstacles moving along arbitrary trajectories. The non-linear velocity obstacle considers the moving obstacle's velocity, its form, and the curvature of its path. It makes it possible to choose a single maneuver for obstacle avoidance (assuming such a thing even exists), preventing any number of obstacles from moving along any specified trajectory. The nonlinear v-obstacle may be formed as a time integral of the collision velocities, or its borders can be computed using analytic formulas. A novel strategy based on artificial potential fields (APF) is discussed in [4]. It offers a real-time methodology particularly useful for practical motion planners in uncertain dynamic situations. Compared to other methods, the use of Maxwell's equations to develop artificial magnetoquasistatic fields (AMF) as an extension of APF results in a more intelligent, natural, and predictable behavior than the alternatives. In addition to avoiding stationary obstructions, one of the primary goals of the AMF is to circumvent those in motion. The concept of adaptive dimensionality to accelerate the path planning process in dynamic situations is applied in [5]. It makes no assumptions about the dynamic model that describes its behavior. To be more specific, it takes time dimension only where there is a possibility that a collision will take place, and it plans in a state-space that has a low number of dimensions everywhere else. An approach for motion planning in dynamic settings is described in [6]. Motion planning refers to determining a path for a robot to follow within a scenario that contains both stationary and dynamic moving objects. A realistic approach based on a roadmap for the static section of the scene is developed. This road plan computes an estimated time-optimal course from a starting configuration to a destination configuration, ensuring the robot does not run into any moving obstacles along the way.

The use of inverse kinematics to create a dexterous hand and a probabilistic road map planner for a robot is explored in [7]. The robot is designed to locate the most efficient path for harvesting ripe mushrooms from a field of plants planted at random. In addition, the applications of the biologically motivated meta-heuristic algorithms firefly algorithm (FA) and ant colony optimization (ACO) [8] have been researched and evaluated. An in-depth analysis and critical examination of the important contributions to path planning in dynamic contexts is conducted in [9]. It is the safest approach for velocity-based motion planning and has the capability for an autonomous robotic system to avoid collision with obstacles in the environment. The probabilistic robot, the probabilistic collision state (PCS), the partially closed-loop receding horizon control (PCLRHC), and the gross hidden Markov model are the four main approaches that are discussed in [10] for robot motion planning in dynamic environments. Markov model, the expectation and maximization method, and the Kalman filter are discussed and compared. A method for planning robot maneuvers in dynamic, crowded, and unpredictable surroundings is described in [11], [12]. Following the dynamic programming (DP) formulation, a partially closed-loop receding horizon control algorithm is implemented. This algorithm's approximation to the DP solution integrates prediction, estimation, and planning while also considering chance constraints arising from the robot's uncertain location and other moving agents.

A comprehensive procedure for robot path planning is described in [13]. It uses random sample-based methods that can be effectively used to solve practical complex path-planning problems. It also considers complexity, completeness, effectiveness, and optimality. The potential field approach is often used in autonomously planning the movement of mobile robots [14]. A potential field approach is suggested as a solution for autonomous mobile robot path planning in a dynamic environment. The robot's location, velocity, acceleration, and physical size are all considered by the improved. A decoupling motion planning approach for two manipulators of a live operating robot is described in [15]. It is based on a hybrid implementation of the rapidly-exploring random trees (RRT) algorithm. The target offset RRT method, the bidirectional RRT algorithm, and the dynamic expansion RRT algorithm are all combined into one hybrid RRT algorithm. An optimization technique based on A* and the dynamic window approach is described in [16]. First, during global path planning, redundant turning points are removed to cut down on the total number of turning points. Second, by developing the motion model of the obstacle, the range of the dynamic obstacle is updated to the static environment, and the DWA algorithm's capacity to avoid dynamic obstacles is increased by updating the motion model.

The framework suggested in [17] is capable of planning and optimizing collision-free robot trajectories by considering safety distance, path length, and execution time. When a new optimum path has been identified, the framework will direct the robot to modify its movement pathways accordingly. The most generally used path planning approaches for mobile robot navigation in static and dynamic situations are discussed in [18]. These methodologies have been utilized to navigate mobile robots in various environments. Global and local path planning methods and traditional and heuristic-based methodologies are also investigated. An effective approach to the problem of creating optimum pathways in dynamic settings is described in [19]. Traditional route planning algorithms struggle to adapt to dynamic environmental changes, such as shifting obstacles. It uses a genetic algorithm (GA), which is a search heuristic influenced by the process of natural selection, to solve this problem. The challenges of dynamic environments, where obstacles and environmental conditions can change over time are discussed in [20]. It incorporates a mechanism to prevent ants from revisiting recently explored paths. A detailed description of the FA and its application to path planning is described in [21]. It adapts the FA algorithm to path planning by defining the fitness function and incorporating obstacle avoidance constraints.

The paper's organization is as follows: an overview of the various strategies currently employed in path planning is provided in section 1. Section 2 describes the Bayesian optimization algorithm (BOA)-based path planning system. Section 3 discusses the performance of the proposed path planning strategy with other existing works. Section 4 concludes this work by summarizing the work carried out.

2. METHODS AND MATERIALS

This section discusses the proposed path planning strategy in more detail. Computers are incredibly competent in a wide variety of application areas, and as a result, the world in which we live has been completely revolutionized. Implementing these technologies has helped improve the quality of work that has historically been performed by people and has entirely automated some jobs, eliminating the need for humans to do such tasks. But, computer-controlled automated systems, such as in sports, are not yet at the point where they can completely replace humans. One of the areas in which computers still need to catch up to their human counterparts is their inability to design exact trajectories for the motions that a robot has to carry out to accomplish a certain job. It is not being claimed that computers are not capable of path planning; rather, the methods that are currently in use do not make it possible for computers to demonstrate the same level of speed in path planning, ability to generalize and accuracy of human skills.

Evolutionary algorithms are a class of optimization algorithms that are inspired by the process of natural selection and evolution. They are often used to solve problems that are difficult or impossible to solve using traditional optimization techniques. The basic idea behind evolutionary algorithms is to create a population of potential solutions to a problem and then use a process of selection, crossover, and mutation to evolve that population towards better solutions over multiple generations. This process is designed to mimic the process of natural selection, in which individuals with better-adapted traits are more likely to survive and pass on their genes to the next generation. Many evolutionary algorithms exist, including genetic algorithms (GA) [22], FA, ACO, and DP. These algorithms can solve a wide range of optimization problems, including problems in engineering, finance, logistics, and more. Overall, evolutionary algorithms have proven to be a powerful and flexible tool for optimization and are widely used in many fields of research and industry. Figure 1 shows the proposed system architecture.

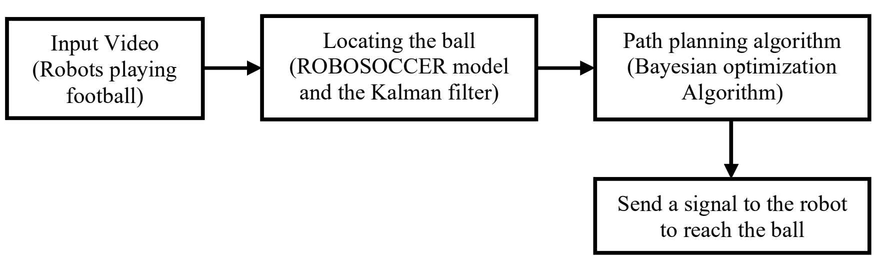

Figure 1. Proposed system architecture

Though the proposed system in this paper finds the shortest path to help the robot to reach the ball, the overall flow of the system shown in Figure 1 includes the object detection approach to find the ball in the image. To locate the ball in the robosoccer environment, the algorithm in [23] is employed before planning the best path without collision. It is a three-class classification problem using a convolution neural network [24] and Kalman filter as tracking algorithms in a robosoccer simulation environment. With classifying an autonomous, programmable, medium-sized humanoid robot (NAO), the ball, and the goalpost, the tracking of the NAO and the ball is also achieved from the first to the final framewithout manual intervention.

One of the main advantages of BOA is that it can find the global optimum with a relatively small number of objective function evaluations. This is achieved by balancing exploration (generating new candidate solutions) and exploitation (using the model to generate solutions expected to perform well). Another advantage is that the probabilistic model provides a measure of uncertainty, which can guide the search toward promising regions of the search space. Bayesian optimization has been successfully applied to various optimization problems in various fields, including engineering, finance, and machine learning [25]. The BOA to find the best path is discussed in the following sub-section.

2.1. BOA

Bayesian optimization is a popular optimization technique that uses a probabilistic model to search for the optimum of an unknown objective function efficiently. It is particularly useful when the objective function is expensive to evaluate or when the search space is high-dimensional. The basic idea behind BOA is to construct a probabilistic model, such as a Gaussian process, that represents the unknown objective function. The model is updated iteratively as new objective function evaluations become available. The model is then used to generate new candidate solutions, which are evaluated to update the model again. The algorithm continues this process until a satisfactory solution is found. The Bayesian optimization algorithm involves the following steps:

a. Initialize a probabilistic objective function model using an appropriate surrogate model, such as a Gaussian process or a tree-based model. This study uses Gaussian process as an objective function for path panning.
b. Choose an acquisition function that determines which point in the search space should be evaluated next. Examples of acquisition functions include expected improvement (EI), probability of improvement (PI), and upper confidence bound (UCB).
c. Using the acquisition function and the probabilistic model, identify the next point in the search space to evaluate.
d. Evaluate the objective function at the chosen point.
e. Update the probabilistic model with the new observation.
f. Repeat steps (a-e) until a stopping criterion is met, such as a maximum number of function evaluations or a satisfactory solution.
g. Return the best solution found during the search

2.1.1. Objective function

Let us consider the minimization problem in (1):

$$\text{find} p * \in \underset{p \in X}{argmin} f(p) \quad (1)$$

where the objective function is $f: X_f \subset \Re^D \to R$. The ambiguity set X is referred to as the optimization problem's search space with $X_f \equiv X$ i.e., without the loss of generality and it is a box-constrained set. The observations (y) is corrupted by noise, which is formulated as (2):

$$y = f(p) + \alpha \quad (2)$$

where $\alpha \sim N(0, \sigma_n^2)$and σ_n represents standard deviation. The BOA's first building block is the response surface. It includes a surrogate model to learn the objective function. Given an input point x, a surrogate model should deliver the predicted value of the genuine objective function and the associated uncertainty in the forecast.

A response surface gives a probabilistic representation of an unknown objective function. Let us consider the posterior probabilities $p(f|X, y)$ for N number of inputs (X) and observations (y) obtained by updating the prior probability $p(f)$ by the likelihood $p(f|X, f)$. Then the posterior probability is defined as (3):

$$p(f|X, y) \propto p(f|X, f)p(f) \quad (3)$$

The likelihood $p(f|X, f)$ in (3) updates $p(f)$ based on the collected data points. In this case, the data points represent the positions of NAOs and the ball's location. The initial assumptions about th$p(f)$ is are very important to reduce the complexity of the BOA.

2.1.2. Acquisition function

The second main component of BAO is the acquisition function. The values of the response surface are fed to the acquisition function to search for the best fit for the problem, i.e., find the best path for the NAO to reach the ball without collision.

Throughout the process of finding the optimal balance, acquisition functions must make trade-offs between exploration and exploitation. During the exploration phase, objective assessments are focused on areas of the response surface with a high degree of uncertainty, while the exploitation phase prioritizes the selection of input sites at which predictions achieve small values (in the case of a minimization problem). The acquisition function acts as a map that directs the search for the optimal solution and contributes to the data efficiency of the optimization process that is applied to the optimization function *f*.

a. Expected improvement (EI)

EI is a commonly used acquisition function in Bayesian optimization, an iterative algorithm for the global optimization of black-box functions [26]. The basic idea behind EI is to find the next point to evaluate that is expected to have the greatest potential for improvement over the current best-observed value. In other words, EI evaluates the expected improvement of a potential new point in the search space over the current best value. To calculate EI, the algorithm first fits a probabilistic model to the observations of the objective function. This model is typically a Gaussian process, which captures the uncertainty of the function values and the correlations between points in the search space.

Given the model, the EI for a candidate point is computed as the expected improvement over the current best value. This expected improvement is defined as the difference between the candidate point's value and the current best value, conditional on the observations and the model. The EI function is designed to balance exploration and exploitation, encouraging exploration of the search space to find promising regions while exploiting the information gained from previous evaluations to refine the search. By iteratively applying the EI function, Bayesian optimization can converge to the global optimum of the objective function with relatively few evaluations.

b. Probability of improvement (PI)

The PI function is designed to find the next point to evaluate in the search space with the highest probability of improving upon the current best value [27]. The PI function is defined as the probability that a candidate point will improve the current best value, given the observations of the objective function and the probabilistic model of the function. Specifically, the PI function is defined as:

$$PI(x) = P(f(x) > f(x^*)) \tag{4}$$

where x is a candidate point, *f(x)* is the value of the objective function at *x*, x^* is the current best-observed point, and $P(f(x) > f(x^*))$ is the probability that the value of the objective function at x is greater than the current best value. The PI function can be computed using the cumulative distribution function (CDF) of the Gaussian process model, which provides a probabilistic estimate of the likelihood that the objective function value at *x* is greater than the current best value.

The PI function encourages exploration of the search space by assigning high values to points with a high probability of improvement while also exploiting the information gained from previous evaluations to refine the search. Bayesian optimization can converge to the global optimum of the objective function with a very small number of evaluations if the PI function is applied repeatedly throughout the process.

c. Upper confidence bound (UCB)

The UCB function [28] is designed to balance the exploration and exploitation trade-off by selecting the next point to evaluate in the search space based on a trade-off between the estimated mean and the uncertainty of the objective function value. The UCB function is defined as:

$$UCB(x) = \mu(x) + k\sigma(x) \tag{5}$$

where *μ(x)* is the estimated mean of the objective function value at *x*, *σ(x)* is the estimated uncertainty of the objective function value at *x*, and *k* is a constant that controls the balance between exploration and exploitation.

The UCB function encourages search space exploration by assigning high values to points with high uncertainty and high expected improvement while exploiting the information gained from previous evaluations to refine the search. The constant k determines the balance between exploration and exploitation, with larger values of k promoting exploration and smaller values of k promoting exploitation. The UCB function is widely used in multi-armed bandit problems and other decision-making problems where the goal is to balance the trade-off between exploration and exploitation. In Bayesian optimization, the UCB function can be combined with a probabilistic model of the objective function to search for the global optimum with relatively few evaluations efficiently.

3. RESULTS AND DISCUSSIONS

The performances of the proposed BOA system are discussed in this section. The BOA system finds a time or energy-efficient trajectory for robots in Robosoccer environment. Thus, the time complexity of the

BOA system is analyzed using different acquisition functions in BOA and compared to other optimization algorithms. Each randomly selected NAObegins in a different configuration within the environment. The trajectory of a NAO is generated by selecting random objectives and then using the BOA to build the pathways to reach the ball. The start and target configurations for each NAO are chosen such that the resultant path would be sufficiently lengthy, guaranteeing that they would traverse a significant portion of the surrounding environment. If NAOs of different sizes are used, then the moving NAO will pass through any little gaps that the robot attempts to pass through, which will cause congestion at those small gaps. To avoid congestion, all NAO is considered to be the same size. Table 1 shows the time complexity to reach the ball without collision for five NAO using different acquisition functions in BOA. It is noted that the distance between the NAO and the ball is fixed for computing the time complexity. If another NAO blocks the path, the BOA finds an alternate path to reach the ball without collision. The time shown for each NAO is the average time from the results of ten runs.

Table 1. The time complexity of the BOA approach for different acquisition functions

NAO	Time complexity in milliseconds (ms)		
	BOA-EI	BOA-PI	BOA-UCB
1	6.8	5.3	7.9
2	7.2	5.45	8.5
3	6.5	4.7	8
4	8.1	6	9.8
5	10.2	7.9	12
Overall average	7.76	5.87	9.24

It can be seen from Table 1 that the BOA-PI system computes the path to reach the ball very quickly than other combinations such as BOA-EI and BOA-CUB. The BOA-EI approach requires ~1.89 ms more than BOA-PI to compute the best path, whereas BOA-UCB requires ~3.3 ms more than BOA-PI. To compare the performance of the system, different optimization algorithms such as GA [26], [27], ACO [28], [29], and FA [30] with the same setting is analyzed. Figure 2 shows the performances of different optimization algorithms in terms of time complexity to find the best path.

It can be seen from Figure 2 that the BOA approach with different acquisition functions outperforms other optimization algorithms, and among the acquisition function, PI is the fastest algorithm to find the path without collision. The time complexity of the proposed system is ~7 ms lesser than GA [29], [30] and ~5ms, ~4ms lesser than ACO [31], [32] and FA [33], respectively.

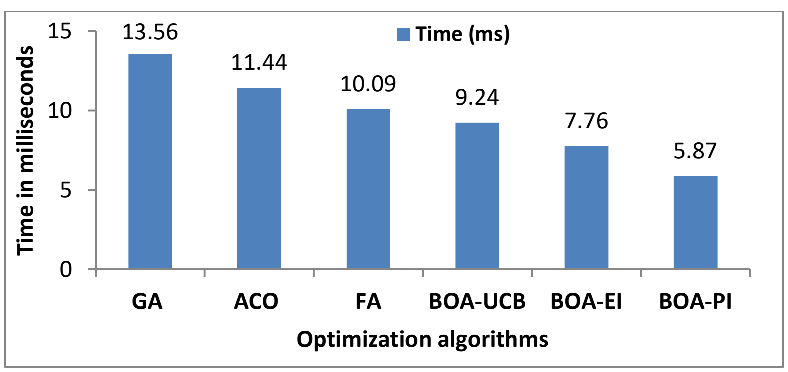

Figure 2. Performance comparison of BOA with other algorithms

4. CONCLUSION

The present study in this paper proposes an effective approach for path planning in robosoccer environment. It develops a rule that takes in the positions of NAOs and ball location as inputs and generates a comprehensive sequence of movements that can proficiently steer the robot along an optimal trajectory from its initial position to the ball location. The system employs BOA to determine the optimal path based on probabilistic modeling. The Gaussian process serves as the objective function, while an analysis is conducted on three distinct acquisition functions, such as EI, PI, and UCB, to determine the optimal trajectory. The comparative analysis of the BOA-based system's performance is conducted in relation to other evolutionary algorithms, including GA, ACO, and FA. The findings demonstrate that BOA-PI offers the optimal route with minimal computational complexity of ~5 milliseconds.

REFERENCES

[1] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: obstacles moving along arbitrary trajectories," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2001, vol. 4, pp. 3716–3721, doi: 10.1109/ROBOT.2001.933196.

[2] R. Iraji and M. T. M.-Shalmani, "AMF: A novel reactive approach for motion planning of mobile robots in unknown dynamic environments," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, pp. 1698–1703, doi: 10.1109/ROBIO.2009.5420410.

[3] A. Vemula, K. Muelling, and J. Oh, "Path Planning in Dynamic Environments with Adaptive Dimensionality," *Proceedings of the International Symposium on Combinatorial Search*, vol. 7, no. 1, pp. 107–115, Sep. 2021, doi: 10.1609/socs.v7i1.18386.

[4] J. P. van den Berg and M. H. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 885–897, Oct. 2005, doi: 10.1109/TRO.2005.851378.

[5] M. G. Mohanan and A. Salgaonkar, "Robotic Motion Planning in Dynamic Environments and its Applications," *International Journal of Robotics and Control Systems*, vol. 2, no. 3, pp. 666–691, Oct. 2022, doi: 10.31763/ijrcs.v2i4.816.

[6] H. Azath, A. K. Velmurugan, K. Padmanaban, A. M. S. Kumar, and M. Subbiah, "Ant based routing algorithm for balanced the load and optimized the AMNET lifetime," 2023, pp. 1–8, doi: 10.1063/5.0110676.

[7] M. G. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, Feb. 2018, doi: 10.1016/j.robot.2017.10.011.

[8] M. G. Mohanan and A. Salgaonkar, "Probabilistic Approach to Robot Motion Planning in Dynamic Environments," *SN Computer Science*, vol. 1, no. 3, pp. 1–16, May 2020, doi: 10.1007/s42979-020-00185-0.

[9] N. E. Du Toit and J. W. Burdick, "Robotic motion planning in dynamic, cluttered, uncertain environments," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 966–973, doi: 10.1109/ROBOT.2010.5509278.

[10] N. E. Du Toit and J. W. Burdick, "Robot Motion Planning in Dynamic, Uncertain Environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, Feb. 2012, doi: 10.1109/TRO.2011.2166435.

[11] L. Ou, W. Liu, X. Yan, Y. Chen, and J. Liang, "A Review of Representation, Model, Algorithm and Constraints for Mobile Robot Path Planning," in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Dec. 2018, pp. 563–569, doi: 10.1109/ITOEC.2018.8740620.

[12] L. C.-ling, T. Jing, and Y. J.- Yu, "Path planning of mobile robot using new potential field method in dynamic environments," in *2011 Seventh International Conference on Natural Computation*, Jul. 2011, pp. 1011–1014, doi: 10.1109/ICNC.2011.6022190.

[13] Z. Liu, A. Pan, A. Jiang, W. Li, J. Zhang, and C. Bai, "Research on Decoupling Motion Path Planning of Two Manipulators of Live Working Robot Based on Hybrid RRT Algorithm," in *2021 4th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, Nov. 2021, pp. 278–282, doi: 10.1109/WCMEIM54377.2021.00064.

[14] Y. Xia, Y. Zheng, X. Wang, L. Zhang, and L. Wang, "Research on Path Planning Algorithm of Service Robot in Dynamic Environment," in *2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*, Oct. 2021, pp. 478–485, doi: 10.1109/AIAM54119.2021.00101.

[15] M. Bdiwi, S. Hou, and K. Delang, "Human-Robot-Cooperation Real Time Robot Path Planning for Dynamic HRC-Applications," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 5542–5542, doi: 10.1109/IROS.2018.8594014.

[16] S. Campbell, N. O'Mahony, A. Carvalho, L. Krpalkova, D. Riordan, and J. Walsh, "Path Planning Techniques for Mobile Robots A Review," in *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, Feb. 2020, pp. 12–16, doi: 10.1109/ICMRE49073.2020.9065187.

[17] A. Elshamli, H. A. Abdullah, and S. Areibi, "Genetic algorithm for dynamic path planning," in *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513)*, 2004, pp. 677–680, doi: 10.1109/CCECE.2004.1345203.

[18] N. Xiong, X. Zhou, X. Yang, Y. Xiang, and J. Ma, "Mobile Robot Path Planning Based on Time Taboo Ant Colony Optimization in Dynamic Environment," *Frontiers in Neurorobotics*, vol. 15, pp. 1–11, Mar. 2021, doi: 10.3389/fnbot.2021.642733.

[19] B. K. Patle, A. Pandey, A. Jagadeesh, and D. R. Parhi, "Path planning in uncertain environment by using firefly algorithm," *Defence Technology*, vol. 14, no. 6, pp. 691–701, Dec. 2018, doi: 10.1016/j.dt.2018.06.004.

[20] B. E and J. Akpajaro, "Genetic Algorithm With Bagging For Dna Classification," *International Journal Of Advances In Signal And Image Sciences*, vol. 7, no. 2, pp. 31–39, Dec. 2021, doi: 10.29284/IJASIS.7.2.2021.31-39.

[21] M. S. Kumar, H. Azath, A. K. Velmurugan, K. Padmanaban, and M. Subbiah, "Prediction of Alzheimer's disease using hybrid machine learning technique," 2023, doi: 10.1063/5.0110283.

[22] S. Ahmad and A. Shahada, "Multilayer Neural Network Based Fall Alert System Using IOT," vol. 10, no. 4, 2019.

[23] D. Steffi, S. Mehta, and V. Venkatesh, "Object detection on robosoccer environment using convolution neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 1, pp. 286–294, Jan. 2022, doi: 10.11591/ijeecs.v29.i1.pp286-294.

[24] C. Qin, D. Klabjan, and D. Russo, "Improving the expected improvement algorithm," *Advances in Neural Information Processing Systems*, vol. 2017, pp. 5382–5392, 2017.

[25] Q. Lin, L. Gong, Y. Zhang, M. Kou, and Q. Zhou, "A probability of improvement-based multi-fidelity robust optimization approach for aerospace products design," *Aerospace Science and Technology*, vol. 128, p. 107764, Sep. 2022, doi: 10.1016/j.ast.2022.107764.

[26] A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, and P. Auer, "Upper-Confidence-Bound Algorithms for Active Learning in Multi-armed Bandits," in *International Conference on Algorithmic Learning Theory*, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 189–203, doi: 10.1007/978-3-642-24412-4_17.

[27] S. D. Immanuel and U. K. Chakraborty, "Genetic Algorithm: An Approach on Optimization," in *2019 International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2019, pp. 701–708, doi: 10.1109/ICCES45898.2019.9002372.

[28] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: concepts and applications [in engineering design]," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 5, pp. 519–534, 1996, doi: 10.1109/41.538609.

[29] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[30] L. Liu, Y. Dai, and J. Gao, “Ant Colony Optimization Algorithm for Continuous Domains Based on Position Distribution Model of Ant Colony Foraging,” *The Scientific World Journal*, vol. 2014, pp. 1–9, 2014, doi: 10.1155/2014/428539.

[31] X. S. Yang, “Firefly algorithm, stochastic test functions and design optimisation,” *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010, doi: 10.1504/IJBIC.2010.032124.

[32] A.Balaji, V. P. Srinivasan, J. Rangarajan, V. Nagaraju, B. Bharathi and S. Murugan, “Smart Technique to Prevent Flood Disaster due to High Rainfall in Coastal Areas using IoT,” *International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics,* pp. 1252-1256, 2023, doi: 10.1109/IITCEE57236.2023.10090999.

[33] P. Arul, M. Meenakumari, N. Revathi, S. Jayaprakash, and S. Murugan, “Intelligent Power Control Models for the IOT Wearable Devices in BAN Networks,” *International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics,* pp. 820-824, 2023, doi: 10.1109/IITCEE57236.2023.10090918

BIOGRAPHIES OF AUTHORS

Diana Steffi is a research scholar in Presidency University, Bangalore. Her research interests include image/signal processing, object detection, robotics and path planning. She can be contacted at email: steffiseelan@gmail.com.

Shilpa Mehta completed her B.E. in Electronics Engineering in 1991 from Government Engineering College Ujjain affiliated to VVV Ujjain, where she bagged the university Gold Medal as the University Merit List Topper. She went on to do her M.E. in Digital Techniques and Instrumentation Engineering from SGSITS under DAVV Indore. She worked as a Lecturer in Electronics and Department Telecommunication Engineering in SGSITS for 6 years and then moved to Hospet, Karnataka where she worked as a Lecturer for 7 years and later as Associate Professor for 8 years. She pursued her PhD research work under JNTU during this tenure. In 2013 she relocated to Bangalore where she worked as a Professor in RITM, and SVCE. She is now working as Professor and Dean-Acadamic in the SOE of Presidency University. She can be contacted at email: shilpamehta@presidencyuniversity.in.

Kanyakumari Ayyadurai Venkatesh pursued his Ph.D. in Discrete Optimization, (A Study on Fractional Half-Domination Number of Graph) at Alagappa University, Karaikudi in 2013. Prof. Venkatesh has a vast knowledge of quantitative techniques, management science, discrete mathematics, numerical analysis, theory of computation, algorithms, artificial intelligence and data science. In addition to his academic credentials, Prof Venkatesh also had a consulting assignment as the Chief Technical Officer and Head – HR for I-Source World-Wide (I) Pvt.Ltd. He has handled international assignments as a Professor at MIIT, Mandalay at Myanmar while on deputation from IIIT Bangalore. He can be contacted at email: prof.kavenkatesh@gmail.com.