IJRCS
International Journal of Robotics and Control Systems
Vol. 2, No. 4, 2022, pp. 666-691
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs
ASCEE

Robotic Motion Planning in Dynamic Environments and its Applications

M. G. Mohanan [1,*], Ambuja Salgaonkar [2]

Department of Computer Science, University of Mumbai, Mumbai, India
[1] mgmohanam@gmail.com, mohanan@udcs.mu.ac.in; [2] ambujas@udcs.mu.ac.in
* Corresponding Author

ARTICLE INFO

Article history
Received May 10, 2021
Revised August 28, 2021
Accepted September 25, 2022

Keywords
Motion planning;
Dynamic environments;
Collision avoidance;
Harvesting

ABSTRACT

The fundamental problem of robot motion planning in a dynamic environment (RMPDE) is to find an optimal collision-free path from the start to the goal in a dynamic environment. Our literature survey of over 100 papers from the last four decades reveals that there are more than 30 models of RMPDE, and there is no benchmarking criterion to select one that is the best in a given situation. In this context, generating a regression-based model with 10 attributes is the first and foremost contribution of our research. Given a highly human-interactive environment like a cafeteria or a bus stand, the gross hidden Markov model has special importance for modeling a robot path. A variant of the growing hidden Markov model for a serving robot in a cafeteria is the second contribution of this paper. We simulated the behavior of GHMM in a cafeteria with static and dynamic obstacles (static obstacles were both convex and concave) and with three different arrangements of the tables and obstacles. Robots have been employed in mushroom harvesting. A novel proposition discussed in this paper is probabilistic road map planning for a robot that finds an optimum path for reaching the ripened mushrooms in a randomly planted mushroom farm and a dexterous hand to pluck the selected mushrooms by employing inverse kinematics. Further, two biologically inspired meta-heuristic algorithms, ant colony optimization, and firefly has been studied for their application to latex collection. The simulation results with this environment show that the firefly algorithm outperforms ant colony optimization in the general case. Finally, we have proposed a few pointers for future research in this domain. The compilation and comparison of various approaches to robot motion planning in highly dynamic environments, and the simulation of a few models for some typical scenarios, have been the contributions of this paper.

1. Introduction

Robot motion planning in dynamic environments is an important thrust area of computer science and computational geometry. The fundamental problem of motion planning is to compute a collision-free path from a 'start' to a 'goal' for a robot that moves in a static and totally known environment, consisting of one or many obstacles. Robot motion planning in dynamic environments (RMPDE) has been studied extensively for the last four decades. Motion planning in dynamic environments with moving obstacles and moving targets is a variant of RMPDE. The RMPDE problem is NP-complete, i.e., a complete planner that reports a solution if one exists, or exits otherwise, takes exponential time in the worst case [1]. Dynamic motion planning for a point in the plane with bounded velocity and many arbitrary obstacles is itself an intractable NP-hard problem [2]. The main approaches to the RMP problem are velocity-based, probability-based, and artificial intelligence-based [3, 4].

Motion planning in dynamic environments was originally addressed by adding the time dimension to the robot's configuration space. Obstacles were assumed to have bounded velocities and known trajectories. A solution to the planar problem for a polygonal robot among many moving polygonal obstacles is provided by searching a visibility graph in the configuration space. The configuration-time space is discretized, resulting in a sequence of configuration space slices at successive time intervals. The static planning problem is solved at every slice, and the adjacent solutions are joined to take the robot to the given goal. A configuration is the environment of a robot at a given time. The configuration-time space is discretized into "cells," such that a cell is empty if there is no obstacle for the robot. A method of cell decomposition joins empty cells to get a collision-free path from start to goal [5].

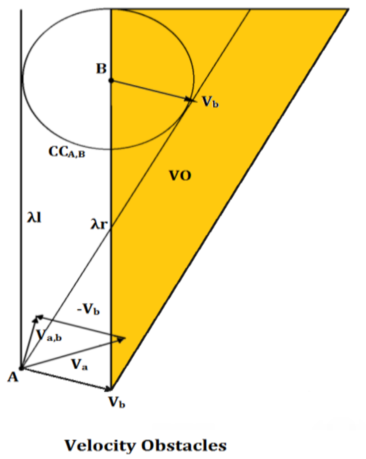
Another approach to dynamic motion planning is to decompose the problem into smaller problems: namely, path planning and velocity planning. This method first computes a feasible path among the static obstacles and represents it as a parametric curve in the arc length, i.e., a path with arc length as its parameter. The intersections of the moving obstacles with the path are represented as forbidden-regions in an arc length-time plane. The velocity of the robot should be planned so as to avoid the forbidden regions. The complete problem of motion planning can be divided into two separate problems: kinematic and dynamic [6]. The kinematic problem consists of finding a trajectory that takes into account the position and velocity of the obstacles as well as an approximation of the dynamic constraints of the robot. The dynamic problem consists of computing an optimal trajectory that satisfies the full set of kinematic and dynamic constraints and is in a close neighborhood to the solution to the kinematic problem. The trajectories of robots moving in a time-varying environment are computed by using the concept of velocity obstacles (VO) [7]. The colored part of Fig. 1(a) denotes the robot's velocities that would cause a collision with obstacles at some near-future time. An avoidance maneuver is computed by choosing velocities that are outside of the velocity obstacles.

DWA is for robots that follow a circular path with translational and rotational velocities. If "look ahead time" is the time taken by the robot to stop and no collisions occur during the interval, then the velocity of the robot at that time is considered to be an admissible velocity Va [8] (see Fig. 1(b). In other words, a velocity is admissible if it allows the system to stop before hitting an obstacle. DWA is a velocity space-based local avoidance scheme where the search for admissible control is carried out in the space of velocities (VS). Time-varying dynamic window (TVDW), as given in Fig. 1(c), is a variant of DWA that computes a set of immediate future obstacles trajectories in order to check for collisions in the near future. Velocities causing a collision after a given time horizon are considered acceptable. VO was extended by NLVO to consider a known velocity outline for the moving object. NLVO consists of all velocities of the robot that would result in a collision with an obstacle at any time to $\leq t \leq tH$. NLVO(t) is a scaled B(t), bounded by the cone formed between the robot and obstacle at time t. NLVO is a warped cone, as in Fig. 1(d) [8-19].

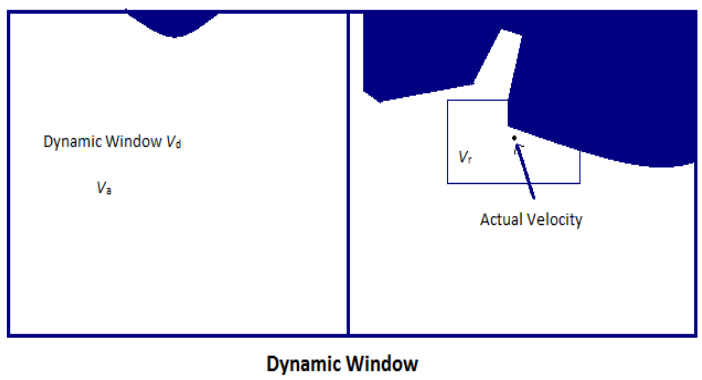
Motion planning methods for robot navigation in dynamic and uncertain environments by using probabilistic approaches involve two problems: (i) safety assessment of roadmaps in uncertain environments by computing the smallest expected collision probability, and (ii) safety assessment beyond the planning horizon of trajectories, since RMPDE approaches generally create partial trajectories towards the goal because motion prediction is not reliable over a long time period. The problem of reliable and efficient navigation in an uncertain and crowded environment is discussed, such that the robot and the surrounding objects reciprocally avoid each other [20-22].

The classical RMPDE approaches are less accurate in address spaces that have local minima; they are costly as well. Heuristic approaches produce an acceptable solution faster. Meta-heuristic algorithms such as (i) ant colony optimization and (ii) firefly, from swarm intelligence, have been successfully employed to solve these types of optimization problems [23-25].

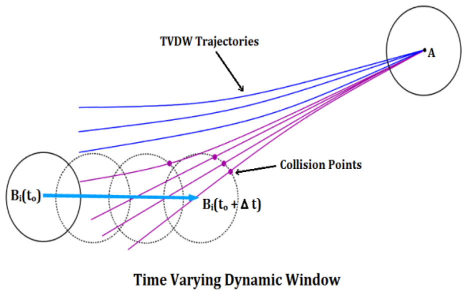
Vol. 2, No. 4, 2022, pp. 666-691



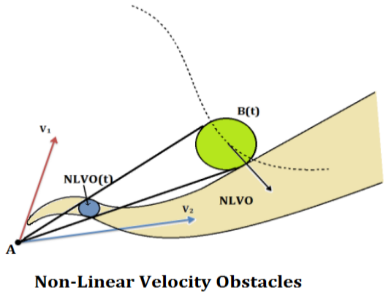
(a) VO



(b) DWA



(c) TVDW



(d) NLVO

Fig. 1. Velocity based Motion planning [8]

We found that the probability road map and hidden Markov model prove better in a dense and unpredictable environment [26]. Models of motion planning for crowded uncertain environments are studied. Three different scenarios in a cafeteria have been simulated. A variant of the growing hidden Markov model has been proposed in this work. The parameters of the path finding in cafeterias are the number of active serving tables and their co-ordinates and the starting coordinates of the robot. The RMPDE has been studied in the context of a specialized domain of agriculture. A system for automation of harvesting in a random agricultural field has been proposed. In particular, the probabilistic road map is recommended for navigation, and an inverse kinematic-based model has been suggested for the plucking of the ripened mushrooms. In contrast to the random field, we studied the performance of ant colony optimization (ACO) and the firefly algorithm (FA) for implementation in a grid-type dynamic environment. A real-life scenario of latex (liquid rubber) collection has been simulated for changing farm sizes, the number of agents, obstacle densities, and hill-locked plantations. Under the assumption of a plain field, our observation is that FF outperforms ACO. Path-length and time are directly proportional to the farm size, which is obvious. What is not obvious is that the number of agents, when increased beyond a certain number, does not add to its performance. For example, in our simulation for a 20×20 farm (i.e., 400 rubber trees), the performance keeps rising till the number of agents is increased to 50. There is no much change in the performance if the agents are increased to 75 or above. The reason is that obstacle density increases, and therefore path length and time increase correspondingly [27].

2. Evaluation of RMPDE

In the paper survey [8], we discussed 31 methods of RMPDE and broadly categorized them into 7 groups on the basis of the prominent parameters used in them. In the literature, several methods have been discussed with their relative pros and cons. However, no evaluation metric seems to have been proposed. In the conclusion of this survey, we attempt a 10-feature characterization of RMPDE and propose a regression-based model for the evaluation of an RMPDE.

The 10 parameters in terms of which we describe an RMPDE are smooth path, safety, path length, run time, accuracy, stability, computational cost, control, efficiency, and future uncertainties. Each of them is given a value on the Likert scale (0-not satisfactory, 1 poor, 2 good, 3-very good, 4-excellent). In the regression analysis, it is observed that three parameters, namely run time, efficiency, and control, do not contribute significantly to the quality metric of the methods. Removing these three, we get a regression model with 7 parameters as

$$\begin{aligned} Quality = {} & 1.08 \times smoothPath + 0.78 \times safety + 0.73 \times accuracy \\ & + 0.51 \times pathLength + 0.5 \times computingCost \\ & + 0.47 \times uncertaintyInPrediction - 0.72 \times stability - 4.6 \end{aligned} \quad (1)$$

The coefficients of the right-hand side of (1) are obtained from the multivariate regression analysis of the attribute matrix in Appendix 1. The R^2 value is 0.82, indicating that the model sufficiently explains the characteristics of the data [8].

3. Probabilistic Approaches of Motion Planning in Dynamic Environments

Four major approaches to probabilistic robot motion planning in dynamic environments (PRMPDE) have been briefly discussed: Probabilistic Robot (PR), Probabilistic Collision State (PCS), Partially Closed Loop Receding Horizon Control (PCLRHC) of Stochastic Dynamic Programming (SDP), and Gross Hidden Markov Model (GHMM) [8, 21, 22, 28-50]. We attempt to provide a bird's eye view of this literature and produce a benchmarking model for their evaluation. In the PR model, the processing (i.e., mapping, localization, perception, and control of motion planning) is done by calculating the probability density function. PCS is the probabilistic extension of Inevitable Collision State (ICS). Both of them work in crowded environments. The original algorithm generates a binary Boolean-valued output for the question of whether any path is collision-free, while the extension generates optimum paths with the probability of them being collision-free. Two types of obstacles are considered, (i) passive and (ii) active. Stochastic Dynamic Programming (SDP) is an optimization methodology; in the present context, SDP is used for motion planning in cluttered, dynamic, and uncertain environments; open loop and closed loop refer to the absence or presence of feedback during optimization. We discuss Open Loop (OL) and Partially Closed Loop (PCL) Receding Horizon Control (RHC). The OLRHC is a suboptimal control scheme in which a sequence of control actions is obtained over a finite horizon, and the motion planning problem is resolved at each stage. The measurements beyond the current stage are ignored. Consequently, the expectation in the cost terms can be omitted. This infirmity is overcome in the PCLRHC by generating corresponding cost values for future measurements.

The above three models work for short-term predictions. The behavior of humans, animals, and vehicles varies according to their perception, internal state, intention, etc. In principle, a machine could be trained to predict their behavior by providing the history of these types of objects. The Gross Hidden Markov Model (GHMM), an extension of the Hidden Markov Model (HMM), involves two stages: (i) the learning stage wherein the patterns from the historical data are identified, and a model of the behavior of the object under consideration is built, and (ii) the prediction stage that uses the learned motion patterns in order to predict the future motion of the moving objects. As uncertainty is inherent in prediction, in GHMM, we use a probabilistic framework to model motion patterns as stochastic processes.

A comparison of the four models of RMPDE is given in Table 1. The methods are a probabilistic robot (PR), probabilistic collision state (PCS), partially closed loop receding horizon control

(PCLRHC), and gross hidden Markov model (GHMM). For evaluating the models, some characteristics are ranked between 1 and 5, based on 1-poor, 2-moderate, 3-good, 4-very good, and 5-excellent, whereas others have a binary "Yes" or "No" ranking. The ranks are given based on [21-78].

Table 1. Comparison of probability-based RMPDE models [76]

Sr. No	Methods	Crowded environment (1-5)	Learning approach	long-term prediction	Complexity (1-5)	Safety (1-5)	Waits till obstacle goes away
1	PR	2	No	No	3	3	No
2	PCS	3	No	Yes	3	5	Yes
3	PCLRHC	5	No	No	4	3	Yes
4	GHMM	4	Yes	Yes	3	5	Yes

4. RMPDE Applications in Automation of Cafeteria

A survey of the literature has led us to conclude that GHMM applications include automation of agriculture, manufacturing, and household activities. One emerging domain is the hospitality industry. Restaurant automation aimed at serving an increasing number of customers with a greater variety of choices has the commercial potential [79]. Automation includes placing an order electronically, helping the customer in selecting a suitable menu, assisting the chef in the preparation of dishes and making it to order, serving the orders automatically on a conveyer belt, and informing the customers of the table number where they would be served [80]. A step ahead is a waiter robot. The objective is to serve fast at peak times. Also, it is a technological attraction and therefore fetching business. The waiter robot, a humanoid, has been made for walking through the pathways through the rows of tables and chairs and serving the customers; it also helps remove the used dishes. This robot follows a magnetic tape placed on the floor and serves according to the orders noted in its memory; sensors are employed to know when an obstacle comes across [81, 82]. A simple strategy has been employed: wait till the obstacle gets removed.

A realistic scenario in any cafeteria comprises of the following elements: limited space, well-placed furniture and decors, very few service tables to be served, and a few moving objects, generally humans and sometimes trolleys or the like. Small modifications in the arrangements of the static objects are acceptable. A serving robot has to start a tour and visit all the tables. The task may be taking or serving an order from a customer by coordinating with the kitchen and receipt managing tables.

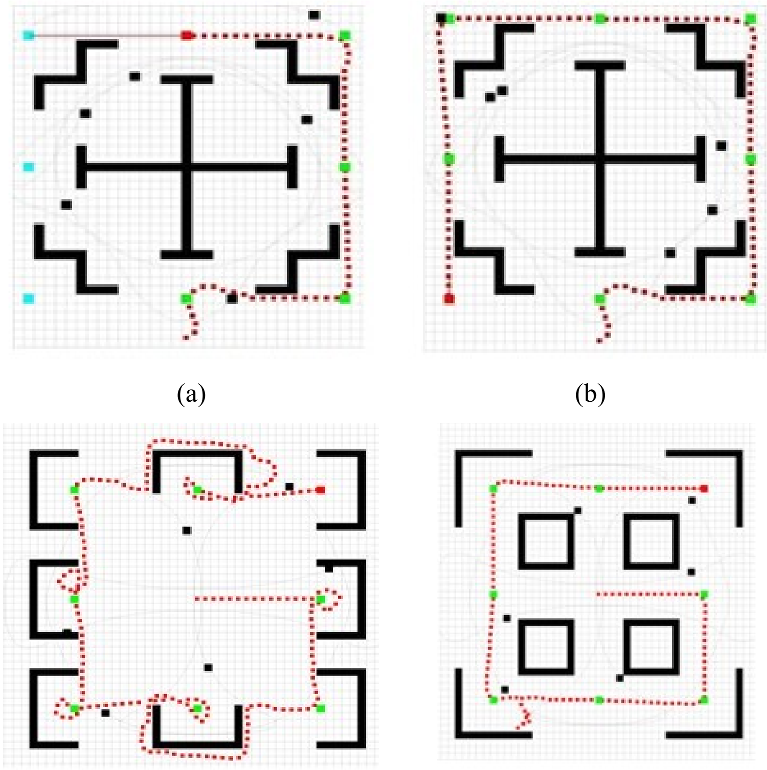
Given this, it becomes a problem for robotic motion planning in a dynamic environment, a small-spaced, highly interactive environment with limited objects in particular. Given the safety requirement in order to avoid spoilage, one may go for PCS or GHMM as a solution. Both could be employed for long-term predictions and could be deployed in a crowded environment such as a cafeteria. However, we rank GHMM the best for its relatively lower complexity and continuously evolving approach due to the learn-and-predict strategy that is its basis.

4.1. Simulation

Basically, a state in GHMM is a cluster of static and moving objects on a known trajectory. The complete state space has been visualized as a collection of clusters that are continuously evolved due to the motions of the dynamic objects. The clusters form Voronoi regions. Transition is possible only in an adjacent Voronoi region. The elements in a cluster are identified with their centroid; the Mahalanobis distance that is an indication of the distribution of element in a cluster is indicative of the path length traversed when a robot selects the next state (i.e., move towards a new cluster) in the direction of its goal (i.e., a table that is to be served very next), i.e., by using the probabilities of HMM.

We have considered three different arrangements with 8 static obstacles in each. See Fig. 2(a), Fig. 2(b), Fig. 2(c), Fig. 2(d). In the simulation results presented, we have considered a serving robot in a cafeteria that has static obstacles like walls, furniture, and fixtures with convex or concave shapes. At any given point in time, there is a minimum of 1 and a maximum of 8 tables that a robot is expected to serve. There are 8 static and 6 moving obstacles that the robot will have to avoid.

Small variations in the locations of the tables are expected. The tables have been moved from their original places twice, and the static obstacles are quite different each time, giving rise to three different arrangements or scenarios for the tables. The task is to find a path to serve all the active tables by avoiding the 8 static and 6 moving obstacles in a given scenario. An instance of completing a Hamiltonian tour in each of the three scenarios has been shown in Fig. 2. The average time of execution per scenario and its corresponding path length for a varying number of tables is listed in Table 2.



(a) (b) (c) (d)

Fig. 2. (a) Partially traversed path by a robot in a cafeteria that has all convex obstacles (Scenario I); (b) Completely traversed path by a robot in Scenario; (c) Completely traversed path by a robot in a cafeteria with all concave obstacles (Scenario II); (d) Traversing in a cafeteria with convex obstacles in its interior and concave in the boundaries (Scenario II1); [76]

Table 2. Average Execution Time and Path Length for GHMM in a cafeteria [76]

Number of tables to be served	Scenario 1		Scenario 2		Scenario 3	
	Time in milliseconds	Path length in # cells	Time in milliseconds	Path length in # cells	Time in milliseconds	Path length in # cells
2	105	113	80	68	95	89
4	127	123	136	126	140	135
6	130	119	191	179	144	133
8	137	113	230	225	156	139
8	135	128	216	206	156	146

In Fig. 2, the various objects are marked in different colors, as follows: (1) Static obstacles are marked with thick black lines, with convex or concave shapes as the case may be. (2) The present position of the robot is marked with a thick red dot. (3) The green and blue dots are the tables where customers could be seated; green dots are tables where the customers have been served, and red dots are where they are yet to be served. (4) The robot's path is marked as a sequence of thin black dots, starting from the middle of the serving area and ending in its present position (thick red dot).

In Fig. 2(a), the robot has served some tables in Scenario 1, but not all. The tables that have been served are marked with a green, but the remaining are blue. In Fig. 2(b), all tables have been served, so they are marked green. In other words, the robot has traversed a Hamiltonian path.

Note that there are two rows corresponding to readings when the number of tables to be served is 8. While the first reading is due to the testing with an increasing number of tables, the second one is due to a separate peak load test. The details of the tables' positions and the time required for completing a Hamiltonian tour, along with the length of the path that the robot traversed in the latter case, are listed in Appendix 2. The two independent runs have resulted in comparable values; therefore, the reliability of the system has been demonstrated. The standard deviation in each case was around 9.

From the observations, it is clear that the time taken for completion of the Hamiltonian tour increases with the number of tables to be served. The average time and path length are the least in scenario 1 in all but the case where the number of tables is 2. Similarly, amongst Scenarios 2 and 3, the values of 2 are higher, except in the case where the number of tables is 4. However, there the figures of scenario 3 are comparable. In light of these observations, one may conclude that Scenario 2 is the most complicated and scenario 1 is the least complicated among the three. From Fig. 2, it could be concluded that the scenario with more concave surfaces is relatively difficult.

The GHMM has been employed to create a collision-free path for pedestrians. An obvious difference between our scenarios is that all but the robots in our scenario are expected to employ their own brains to reach their goals. We do not require to keep track of their progress per se though their movements are important for defining a path for the service robots (SR). There could be more robots serving the goal by working in coordination. Given this, we propose to employ means-end analysis to converge with an arguably more efficient variant of GHMM for the cafeteria scenario. The basic principle is that given a current state, a goal state, and a set of possible transitions, an action is chosen that reduces the difference between the two states.

Imagine an SR with the objective of collecting a dish from a kitchen window for a customer sitting at a table. The solution consists of two parts: one, construct a path from the table to the kitchen window and then back to the table. Let all the bots be broadcasting their status periodically. Call it who-n-whereami protocol. If a bot-i finds bot-j is equipped with the material that is required by bot-i, then the following heuristic is employed:

```
If the distance (bot-i, Kitchen-window) < Threshold OR
distance (bot-i, bot-j) >distance(bot-i, Kitchen-window)
     then findpathto (Kitchen-window)
     else findpathto (bot-j); update (bot-i), update (bot-j)
```

Effectively, this heuristic should help generate a network of short-distance service providers. Those in the relay will help overcome the optimizing the path-finding over a longer distance. It is expected that over the period, the short-distance service providers shall develop expertise in finding the paths in their territory and hence enhance the overall performance of the system.

4.2. The Novelty and Uniqueness of Cafeteria Research

A ranking model has been developed for the four probability-based motion planning approaches and concluded that the gross hidden Markov model (GHMM) is the best-suited method for environments with limited space and highly dynamic due to human interactions, such as the cafeteria.

Using GHMM, we simulated a real-life cafeteria with eight tables to be served by a robot by considering three different arrangements with concave and convex obstacles. For these, we obtained the path length and time of the Hamiltonian path. We found that the concavity of the obstacles makes the scenario more complex for path planning. This is the first time that the variant of the hidden Markov Model algorithm has been proposed for simulating a cafeteria. The SR can be extended to any other service-providing applications.

5. RMPDE Application in Agriculture

5.1. RMPDE in Mushroom Harvesting

Robotic mushroom harvesting in a random field is proposed by employing a probability road map (PRM) for navigation on the farm. (PRM is a sampling-based 2-step method that includes roadmap construction and querying.) Inverse kinematics is employed for plucking the ripened mushrooms.

Extending the earlier work stated in the survey, we discuss PRM for the planning of a robotic motion within the mushroom maze or random plantation with static obstacles. The method is extended to find the roadmap in environments with dynamic obstacles. It checks whether or not a robot is in an obstacle-free configuration and proceeds accordingly. The method is capable of dealing with robots with many degrees of freedom and having diverse constraints, and it has been shown to be probabilistically complete, i.e., the probability of failure for a planner to find a solution trajectory, if one exists, converges rapidly to zero as the number of collision-free samplings of the workspace increases [8]. The core of our mushroom harvesting robot is an algorithm for encountering static obstacles by a path finder robot [47]. The PRM computes a collision-free path between two ripened mushrooms with a local planner.

The basic idea is to check if the roadmap constructed to avoid static obstacles also works with dynamic obstacles, i.e., obstacles moving at a given instant. If it works, then the path is built. Else the edges that meet the moving obstacles are marked as blocked, and construction of alternative paths is attempted. A five-step procedure for the PRM in the such environment has been listed [83-85].

The design of a dexterous robot hand is driven by the task of plucking the targeted mushroom assigned to it. We propose a two-step process: first, an assembly of two fingers that is analogous to a thumb and a pointing finger of a human hand to get a grip on the stem of the mushroom bud that is to be plucked; in the next step, the stem is uprooted. The joint angles of fingers are calculated by employing Inverse kinematics.

A mushroom harvesting robot (MHR) consists of three units: (i) a recognition system that identifies mature mushrooms and confirms their locations, (ii) a moving system with wheels that moves through the paths to reach the mature mushrooms, and (iii) picking system that grasps and plucks the mushrooms at the given location [86-91, 146]. Assuming inputs from a recognition system, this paper presents and develops a novel robotic model to perform the moving and picking activities efficiently.

5.1.1. Probabilistic Road Map for Motion Planning of a Robot within a Random Field

First, we discuss the probabilistic road map (PRM) method for the planning of robotic motion with static obstacles. Then the logic is extended to find a roadmap with dynamic obstacles. PRM is a sampling-based 2-step iterative method that includes roadmap construction and querying (see algorithm and Fig. 3 and Fig. 4). It checks if a robot is in an obstacle-free configuration space Q_{free}, (for details see Fig. 3 and Fig. 4). The following algorithm for navigation of a robot through a mushroom farm is an implementation of [92-109].

5.1.2. Algorithm for Static Obstacles

A roadmap is an undirected graph $G = (V, E)$, where the nodes in V represent a set of ripened mushrooms, and each edge in E is a collision-free path between two nodes computed by a local planner, see Fig. 2. Nodes q_{init} and q_{goal} are user-provided inputs. They are, respectively, the initial and final nodes in a path to be discovered by the algorithm.

Querying: Let ConnectQinit be a list of neighboring nodes in the roadmap in the order of their distances from q_{init} and similarly, let ConnectQgoal be a list of neighboring nodes in the same roadmap in the order of their distances from q_{goal}. Try connecting q_{init} to each of its neighboring nodes, and q_{goal} to its neighboring nodes; call the nodes a′ and a″, respectively. Search the graph G for a sequence of edges in E connecting a′ to a.″ Convert this sequence into a feasible path for the robot by computing the corresponding local paths and concatenating them. The local paths can be stored in the roadmap. The whole sequence, $q_{init}- a'- \ldots - a''- q_{goal}$ is a feasible path for a robot. Among the feasible paths, find the shortest path on the roadmap between q_{init} and q_{goal} by employing an appropriate algorithm—one of the A*, D, and D*Lite algorithms [110-112].

Algorithm for static obstacles

Repeat steps S1 and S2 until all mushrooms in the node-set G are covered.

S1 (construction). For a given workspace, construct a roadmap in a probabilistic manner, i.e., randomly select a configuration of nodes (provided by image processing) using some sampling distribution.

S2 (querying). Given an initial configuration q_{init} and goal configuration q_{goal}, find the shortest path connecting q_{init} and q_{goal}.

Remark: The robot is supposed to move and pluck all the mushrooms along this path and remove them from the graph. Then the two steps of the algorithm are to be repeated.

Fig. 3 and Fig. 4 illustrate the two phases of the iterative path-finding algorithm. In Fig. 4, the shortest path from q_{init} to q_{goal} is marked with thick lines.

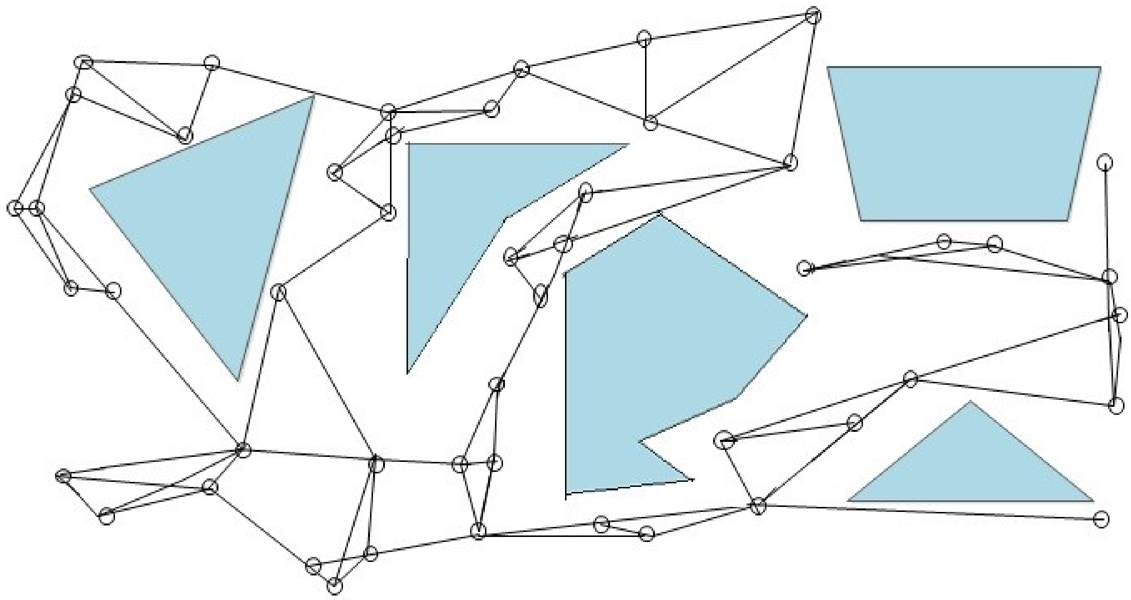


Fig. 3. Example of a roadmap for a point robot in two-dimensional Euclidean space. Shaded areas are obstacles. The small circles are nodes of a graph, and the edges represent obstacle-free paths between adjacent nodes.

5.1.3. Algorithm for Dynamic Obstacles

The basic idea is to check if the roadmap constructed to avoid static obstacles also works despite dynamically moving obstacles at a given instant. If it works, then the path is built. Otherwise, the edges that meet the moving obstacles are marked as blocked, and the construction of an alternative path is attempted. The two ends of the blocked edges are connected locally by employing a rapidly-exploring random tree (RRT) algorithm [110-112]. In a dynamic environment, the initial and goal configurations are also moving entities, and therefore the new path has to be constructed by considering their new positions. A five-step procedure for the PRM in dynamic environments is described as (i) Roadmap labeling and solution path search, (ii) Query node connections, (iii) Local reconnections, (iv) Node insertion and cycle creation, and (v) Edge labeling [113-117].

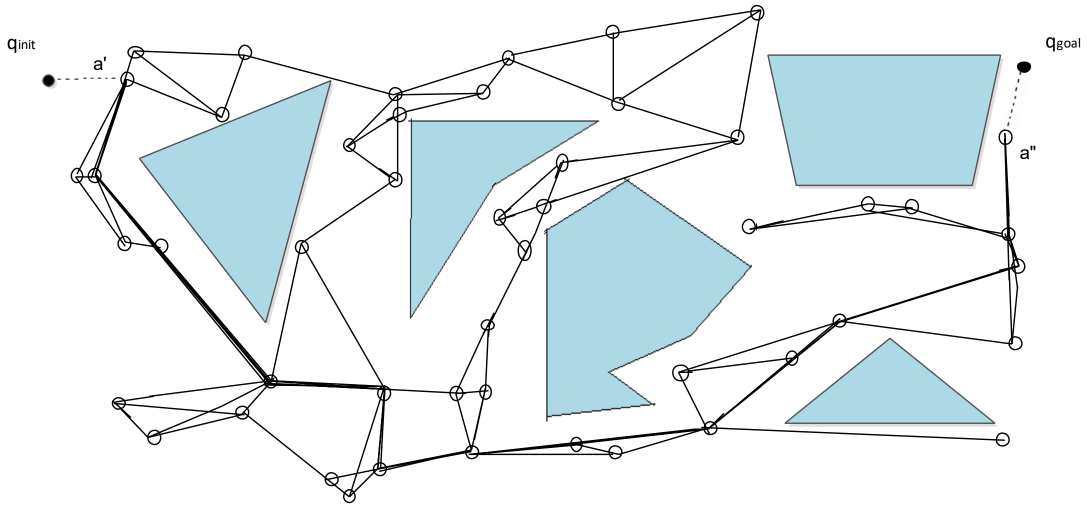


Fig. 4. Example of a query with the roadmap. Nodes q_{init} and q_{goal} are first connected to the existing roadmap through nodes a' and a". The search algorithm returns the shortest path, denoted by a thick dark line [146].

5.1.4. Kinematics for Robotic Hand Motion

Getting a roadmap ready is a task that enables a robot to reach the ripened mushrooms at the nearest possible place from its current location. The next task is to model the motion of the robot hand (the end effector) to reach a ripened mushroom. The design of a dexterous robot hand is driven by the task assigned to it. Diverse models have been discussed [118, 119]. We discuss an assembly of two fingers (analogous to a thumb and a pointing finger of a human hand) that gets a grip on the stem of the mushroom bud to be plucked, and next, the process of uprooting the stem. (Arguably, a five-finger hand like that of a human will be a too-heavy and complicated assembly for the present purpose, as it would take more space and may harm the neighboring buds.) A diagram of this proposed mushroom plucking robot hand is shown in Fig. 5. The first finger-link in the structure is known as the base, and the end link is known as the end effector.

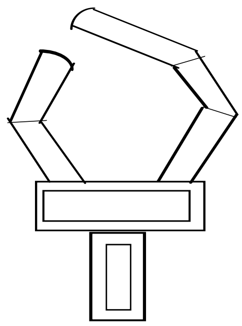


Fig. 5. Model of a two-finger robot hand.

The required angular displacements in the finger-links through the motions at the finger-joints are computed by employing kinematics, i.e., the study of the motion of bodies without consideration of the forces that cause the motion. The inverse kinematics computations for the finger simulating the pointing finger are shown below. The coordinates of a mushroom to be plucked are the driving parameter. The computation of the thumb follows the same logic. The difference is that the thumb has one less joint. The links have an ordered structure in which each link has its own coordinate system and is positioned relative to the coordinate system of the previous link. The position of the link i in the coordinate system of its ancestor is obtained by computing the joint angle [120].

Denavit-Hartenberg (DH) Frame for Joints

The transformation matrix between two adjacent connecting joints is calculated using the D-H parameters in Formula (1) and Table 3, where s_i indicates sin θ_i, c_i indicates cos θ_i $(i = 1, 2, 3)$, α_{i-1} is the twist angle, a_{i-1} is the length of linkages, and d_i is the offset of the linkages. The transformation matrix of the manipulator's finger is obtained by multiplying the transformation matrix of each connecting link ${}^{i-1}_{i}T$ $(i = 1, 2, 3, 4)$, which is a function with the three joint variables $(\theta_1, \theta_2, \theta_3, \theta_4)$ where $\theta_4 = 0$.

Step 1: Holding a mushroom stem:

(X_i, Y_i, Z_i) represents an axial frame of reference. For $i = 0$, it represents the coordinates of the base; the value of i increases by 1 to denote the coordinates of the next joint. The link after the last joint is the tip, i.e., the end-effector. Hence (X_3, Y_3, Z_3) is the frame of reference for the end-effector of a pointing finger (Fig. 7), while for the thumb, it is (X_2, Y_2, Z_2).

Step 1.1: Compute the D-H parameters of the pointing finger.

Table 3. D-H parameters of pointing finger (From [121])

# Joint i	d_i (Joint distance)	a_{i-1} (Link length)	α_{i-1} (Link twist)	θ_i (Joint angle)
1	0	0	0	θ_1
2	0	l_1	0	θ_2
3	0	l_2	0	θ_3
4	0	l_3	0	0

Explanation: The transformation matrix between two adjacent connecting joints can be calculated by the D-H parameters in Formula (1) and Table 1, where s_i indicates sin θ, c_i indicates cos θ_i (i = 1, 2, 3, αi-1 is the twist angle, a_{i-1} is the length of linkages, and d_i is the offset of linkages. The transformation matrix of the manipulator's finger can be obtained by multiplying the transformation matrix of each connecting link continuously ${}^{i-1}_{i}T$ (i = 1, 2, 3, 4), which is a function with the three joint variables (θ1, θ2, θ3, θ4). where θ4=0. Note that $i - 1$ is the base of the link, and i is the successor link.

$$ {}^{i-1}_{i}T = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S_{\alpha_{i-1}} & -S_{\alpha_{I-1}}\, d_i \\ C\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{I-1} & C\alpha_{i-1}\, d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

where S and C represent the sine and cosine functions, by using (2), we can compute the joint angles, θ1, θ2, θ3 [122-126, 146].

Hence the joint angles are

$$ \theta_1 = atan(p_y, p_x) - atan\,(k_2, k_1) \tag{3} $$

$$ \theta_2 = atan\,(S\theta_2, C\theta_2) \tag{4} $$

$$ \theta_3 = atan(S_\omega, C_\omega)\; - \;(\theta_1 + \theta_2) \tag{5} $$

where $p_x = k_1 C\theta_1 + k_2 S\theta_1$, $\omega = (\theta_1 + \theta_2 + \theta_3$, $p_y = k_1 S\theta_1 + \; k_2 C\theta_1$ with $k_1 = \; l_1 + l_2 C\theta_2$ and $k_2 = l_2 S\theta_2$.

5.1.5. The Novelty and Uniqueness of Mushroom Harvesting Research

This is the first time that PRM algorithms are being proposed for navigation inside mushroom farms. Unlike previous research in mushroom harvesting, mushrooms are not planted in a grid or some pattern but are randomly distributed. No human intervention is required at any stage of harvesting by

Mushroom Harvesting Robot (M HR). Robotic automation reduces crop wastage due to the unavailability of labor and the untimely harvesting of mushrooms. Harvesting and other expenses are reduced compared to those with human labor. A kinematic model of a two-finger dexterous hand with 3 degrees of freedom for plucking mushrooms was developed using the Denavit-Hartenberg method. Inverse kinematics techniques for reaching the ripened mushroom give more precision to plucking. There will be no limitation or restriction on large-scale cultivation and harvesting, and this will provide economies of scale.

5.2. RMPDE in Rubber Harvesting

We considered a case again from costal India of rubber plantations, a crop of commercial value and available for harvesting in multiple cycles a year. The systematic plantation of rubber trees on a rectangular grid motivated us to explore the application of grid-search algorithms. We compared the ant colony optimization (ACO) and firefly (FF) algorithms in various scenarios by changing simulation parameters like the density of the environment, land size, and the number of robots simultaneously available [27, 127-131]. We also discussed the effect of land type on the performance of a path-finding system. Our findings may form guidelines for applying ACO and FF for harvesting land cultivated on an imagined grid. We name these robots rubber harvesting robots (RHR) A-RHR follows ACO, and F-RHR follows FF for path finding in a grid environment. The objective of this research is to introduce the concept of employing soft computing to reduce the cost of agri-robotics by taking advantage of a regular topology of cultivation. This topic could be explored in greater detail by considering plausible topologies for the cultivations of other crops. The dexterous hand of the mushroom harvesting robot (MHR) could be tuned to work as a latex collector in RHR in our case or according to the requirement of the situation.

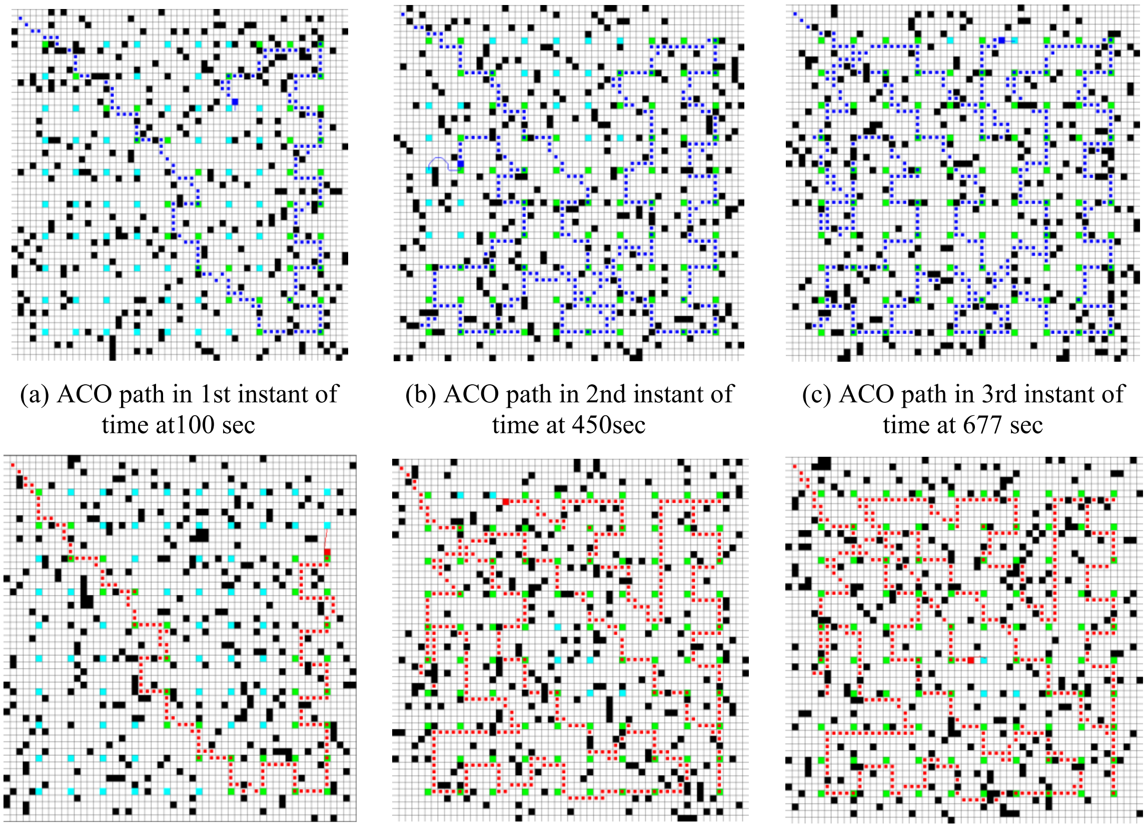
Rubber plants are cultivated in the open field in the arrangement of rows and columns forming a matrix. The shortage of skilled rubber tappers and the high labor cost that prevailed in rubber farming has been the major problem faced by rubber cultivating farmers that automation could help overcome. The rubber is harvested by rubber tappers by making a long curving incision on the outer bark of the trunk of the rubber tree. The white-colored liquid rubber or latex from within the tree seeps to the surface of the cut and down the cut into a container that is tied to the stem of the tree. After two to three hours of incision, the tapper collects the deposited latex at each and every plant in sequence and submits the same to the specific collection center for further processing of rubber [132, 133].

Sensors and cameras are mounted on robots to identify the proper place to tap within the rubber stem. The robot has to navigate by using ant colony optimization (ACO) or firefly algorithm (FF), where cells of the roadmap correspond to the matured rubber plant to be tapped. The robot stops at every matured plant and cuts the outer skin of the plant in a specified way with a sharp-edge knife, using a dexterous robotic arm that employs inverse kinematics techniques. In the literature and to the best of our understanding, obstacle avoidance while navigation is done through human intervention, but in our proposed method, it has been done automatically within the algorithm itself [134-141].

5.2.1. Simulation of a Hamiltonian Tour of a Latex Collector Robot in the Rubber Plantations of Various Farm Sizes

Starting at the upper left corner, a robot reaches one of the 8 nearest neighbors. The reached node becomes the new start, and the new goal is one amongst its non- served nears neighbors. The procedure is called recursively till the robot serves the last non-served plant on a given farm.

Progress of path finding in an instance of ACO has been shown in Fig. 6(a), Fig. 6(b), and Fig. 6(c) with blue color and that of FF in Fig. 6(d), Fig. 6(e) and Fig. 6(f) has been shown in red. A farm of $10 \times 10 = 100$ plants has been simulated. Initially, all plants are blue; as soon as a robot reaches a plant, it turns green. Obstacles are shown in black.



(a) ACO path in 1st instant of time at100 sec
(b) ACO path in 2nd instant of time at 450sec
(c) ACO path in 3rd instant of time at 677 sec
(d) FF path in 1st instant of time at 90 sec
(e) FF path in 2nd instant of time at 500 sec
(f) FF path in 3rd instant of time at 542 sec

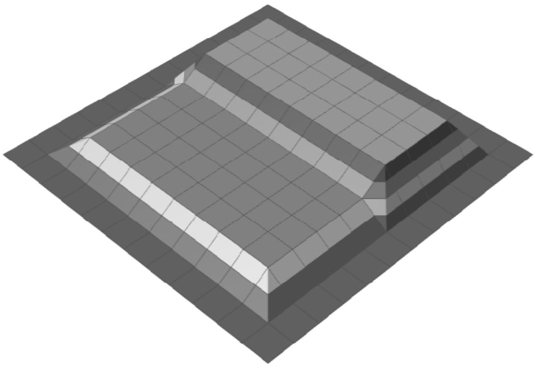
Fig. 6. Hamiltonian tour of ACO and FF in plain rubber field [27]

The average time taken to complete a tour over 5 observations is 678 seconds in the case of ACO; it is 544 seconds in the case of FF. The path lengths are 416 and 380 cells, respectively. Therefore, FF optimization outperformed the ACO by 19.8% in terms of time and 8.7% in terms of path length. The simulation was executed for five different grid sizes: 20×20, 40×40, 60×60, 80×80, and 100×100; for each grid starting with 25, the number of agents increased to 50, 75, 100, and 200. Random dynamic environments were created with obstacle densities of 10%, 30%, 50%, 70%, and 90%. The starting point is the left uppermost corner, and the goal is the right lowermost corner of the respective grids. The average of the best's path-length and the corresponding time over 5 runs have been compiled.

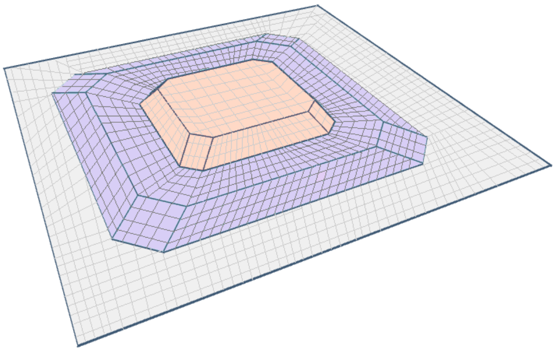
5.2.2. Simulation of Rubber Matrix Grids in Sloped Terrain with Different Height Maps

In the simulation, two different sizes of a grid network of rubber plants are considered, i.e., 10×10 and 20×20, with each grid size with a number of ants and fireflies 25, 50, and 75. Random dynamic environments are created with obstacle densities of 10%, 30%, 50%, and 70%. The simulation is done with three heights within the grids. The normal sea level is h1, the next level is h2, and the final level is h3. The robot transition path is from h1 to h2 and h2 to h3 and vice versa, as shown in Fig. 7(a). There is no direct transition from h1 to h3 or h3 to h1, and these types of nodes will be treated as obstacles. In the grid, three different colors are shown for the height variation within the grids, white for h1, blue for h2, and red for h3, as shown in Fig. 7(b) and Fig. 7(c).

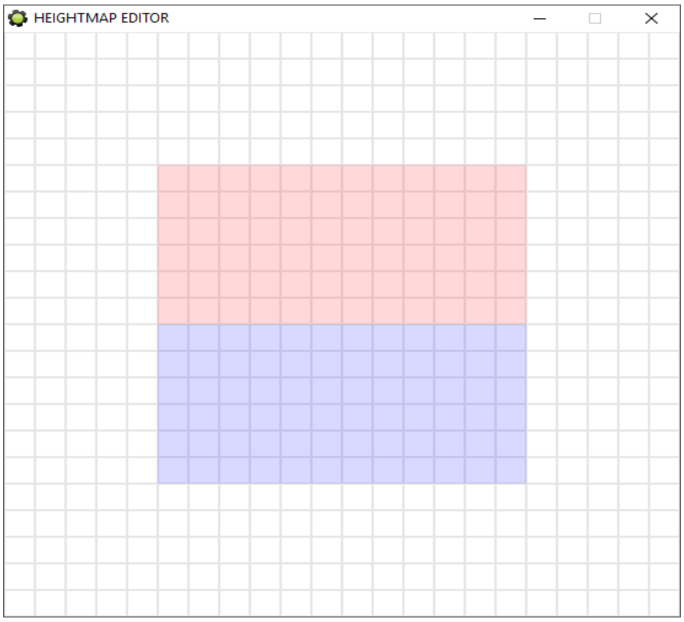
The starting point is the left uppermost corner rubber plant, and the goal point for the robot for simulation is the last rubber plant to be served in the respective grids. The aim of the simulation is to compare the path length and time of execution for obtaining the optimum path for servicing all the plants by using ACO and FF.



(a)



(b)



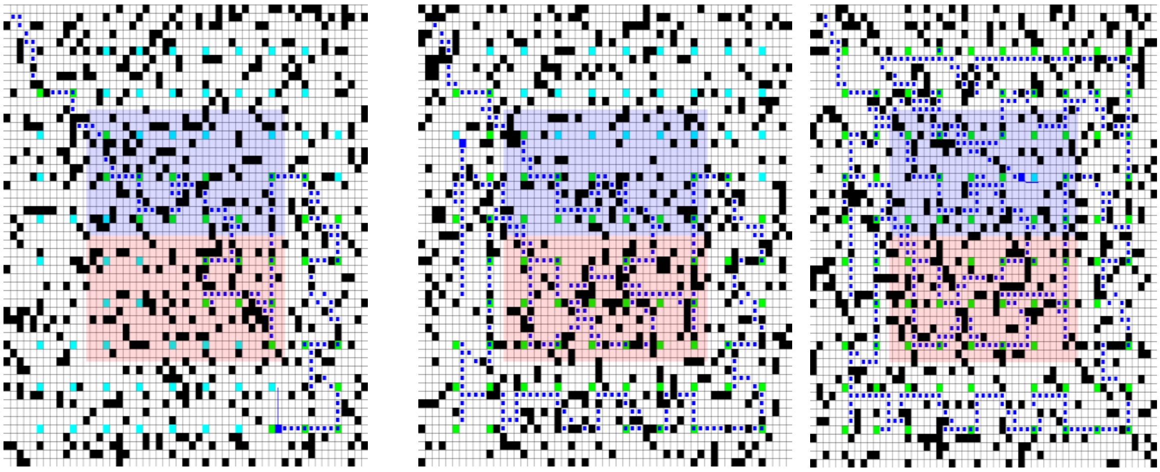
(c)

Fig. 7. Height map of sloped terrain opened up in plain sheets [27]

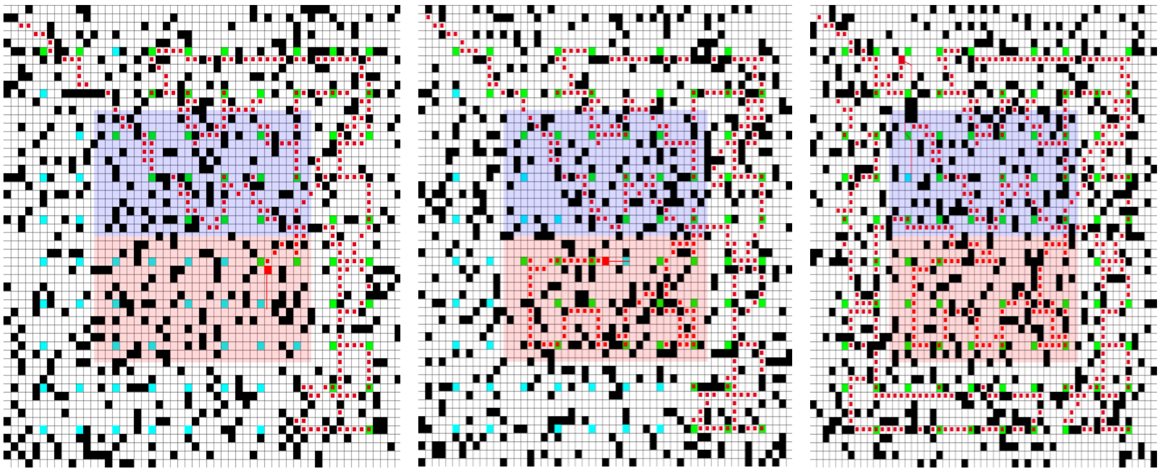
Fig. 8(a), Fig. 8(b), and Fig. 8(c) show the path position at the end of three-time instances for ACO in 10×10 rubber plantation fields with sloped terrain. The same grid size, Fig. 8(d), Fig. 8(e), and Fig. 8(f), show the path-finding activity at the end of three different time instances for FF.ACO completed the path by 878 and FF by 815 seconds, and their path length were 446 and 435 units of cells, respectively. In this case, the FF optimization outperformed the ACO by 7.2% in terms of time and 2.5% in terms of path length. The simulation results of plain rubber grids, sloped rubber terrain, and detailed regression analysis of various parameters contributing to the path planning are given in the paper [27].

5.2.3. The Novelty and Uniqueness of Rubber Harvesting Research

The number of agents above 50 has not shown a substantial contribution to the optimization of path length and time of execution in general. The path length and time increase proportionally with the increase in the grid sizes. In all but the case of terrain, with varying grid sizes, the FF algorithm outperforms ACO in terms of path length and time of execution for the optimum path. Both path length and time are more in the sloped rubber terrain as compared with the normal plain rubber field. In the simulation results and detailed regression analysis, we found the effect of various parameters in motion planning in different dynamic scenarios. The shortest path on plain land is the relatively simplest scenario, while the Hamiltonian on a concave surface is supposedly the most difficult. Our proposal for Rubber Harvesting Robot (RHR) carries novelties in the agricultural domain and gives innovation in the area of automation for the latex collections in rubber plantations [27]. We look forward to the real-life implementation of the RHR proposed here. Research and simulation experiments could be carried out with the continuous sloped rubber terrain fields.



(a) ACO path in sloped terrain (200sec) (b) ACO path in sloped terrain (500 sec) (c) ACO path in sloped terrain (878 sec)



(d) FF path in sloped terrain (200sec) (e) FF path in sloped terrain (400 sec) (f) FF path in sloped terrain (814sec)

Fig. 8. Hamiltonian tour of ACO and FF in sloped rubber terrain [27]

6. Summary and Conclusions

Development of regression-based evaluation models to select the right approach for employing a path-finder robot in a given situation. We simulated the behavior of GHMM in a cafeteria with static and dynamic obstacles (static obstacles were both convex and concave) and with three different arrangements of the tables and obstacles, and the results show that the environments with concave obstacles make the motion planning more complex. Robots have been employed in mushroom harvesting. A novel proposition discussed in this paper is probabilistic road map planning for a robot that finds an optimum path for reaching the ripened mushrooms in a randomly planted mushroom farm and a dexterous hand to pluck the selected mushrooms by employing inverse kinematics. Further, two biologically inspired meta-heuristic algorithms, ant colony optimization, and firefly has been studied for their application to latex collection. The simulation results with this environment show that the firefly algorithm outperforms ant colony optimization in the general case.

7. Future Research Possibilities in RMPDE

The inevitable collision state (ICS) and probabilistic collision state (PCS) concepts guarantee the safety of the robotic system with respect to the model of the future (i.e., how to move from one safe state to another). One research possibility is to employ massively parallel architectures, e.g., the cloud, to implement these models of the future so as to obtain solutions in a reasonable amount of time.

- Multi-robot systems have received much research attention because of their potential to accomplish a variety of complex tasks through cooperation. One viable research topic is a high-level task planner that will increase the autonomy of dynamic robot networks.
- Implementation of the PRM and GHMM in geographically constrained fields, e.g., narrow corridors or wide plots, is itself a research problem.
- Exploring the variants of these models for their robotics applications in other domains (e.g., GHMM has already been employed for gesture recognition [26, 48]) will be a new variety of research extension of this paper.
- Describing the world model in a concise but useful form is necessary to allow for information sharing between robots in the same network. However, the ability to model the world for any general environment is not available. Required for world model fusion is the combining of environment object state estimates acquired through relative sensing. A key issue to address is the "correspondence problem," the difficulty in resolving whether measurements from two sensors or from two different robots are of the same object [142].
- The splitting up of networks into subdivisions in which robots from different subdivisions are explicitly not coordinated raises a research problem: to find a method of determining where the divisions should be made. In the general case, this appears to be a difficult research problem with no obvious solution [143, 144].
- A new field of research has opened up called Cloud Networked Robotics [145]. It deals with the issues of supporting daily activities, e.g., for the elderly and the disabled, throughout various locations in a continuous and seamless manner by abstracting robotic devices and providing a means for utilizing them as a cloud of robots.

In summary, this paper, by proposing a line of robotic solutions to agricultural domains, has contributed to interdisciplinary computational research for social good.

Appendix 1. Evaluation Matrix of RMPDE

METHODS	Smooth Path	Safety	Path Length	Accuracy	Stability	Comp. Cost	Future Uncertainty	RunTime	Control	Efficiency	Weight (out of 10) (y)
ICS & ICS-AVOID	4	4	4	4	3	4	4	2	3	3	9
PVO	4	4	4	4	3	4	3	2	2	3	9
PCS	4	4	4	4	4	3	4	3	3	3	9
PRM, RRT	4	4	4	4	4	3	4	4	4	3	9
ND & GND	4	4	4	4	4	3	4	3	4	3	9
DVS	4	4	3	4	3	3	3	3	3	4	8
MDP & POMDP	3	3	3	4	2	3	4	4	4	4	8
DDN	4	4	3	3	3	3	3	3	2	4	6
PR	3	4	3	4	3	2	4	4	3	3	6
PTMP	3	4	3	4	3	3	3	3	2	3	6
STI	4	3	3	4	4	3	4	3	3	3	7
RCA	3	3	3	4	2	2	4	3	3	3	7
RAMP	3	4	3	4	4	4	4	3	2	2	6
STA, LP	3	4	3	4	3	3	3	3	3	3	7
TSR	3	3	3	4	3	3	4	3	3	3	6
DC	3	4	3	4	4	4	3	4	3	3	7
VO	3	4	3	3	3	3	3	3	3	3	7

METHODS	Smooth Path	Safety	Path Length	Accuracy	Stability	Comp. Cost	Future Uncertainty	RunTime	Control	Efficiency	Weight (out of 10) (y)
ANN	3	3	4	4	4	3	4	3	4	4	6
PSO	3	3	4	4	3	3	2	4	4	4	6
DWA	3	3	3	4	3	3	3	3	4	3	7
RGT	3	4	3	4	4	3	3	4	4	3	7
AODE	3	4	3	4	3	2	3	3	4	4	6
APF	3	4	2	3	1	3	1	4	4	4	6
GA	3	3	4	4	3	3	2	3	3	3	6
GPU	3	3	3	4	3	2	3	3	3	3	5
PMP	3	4	3	4	4	3	3	3	3	3	6
CS, STS	3	3	4	3	3	2	3	3	2	2	5
Diff Constraints.	3	3	3	3	2	2	3	3	2	3	6
GBWFP	3	3	4	3	3	3	2	4	3	4	5
VMP	3	3	3	3	3	3	3	3	4	4	5
AG.	3	2	3	2	1	2	2	2	2	2	5

APPENDIX 2. Simulation Results of GHMM [76]

Sr	Node	X	Y	Time	Time (ms)	Path Length	Scenario
1	-	-	-	156	3120	146	1
2	-	-	-	216	4320	206	2
3	-	-	-	135	2700	128	3
4	1	130	100	186	3720	174	1
5	1	130	100	243	4860	230	2
6	1	130	100	147	2940	139	3
7	1	100	130	162	3240	149	1
8	1	100	130	236	4720	223	2
9	1	100	130	146	2920	138	3
10	1	130	130	183	3660	170	1
11	1	130	130	235	4700	223	2
12	1	130	130	156	3120	147	3
13	2	280	100	177	3540	165	1
14	2	280	100	233	4660	222	2
15	2	280	100	156	3120	148	3
16	2	250	130	184	3680	172	1
17	2	250	130	242	4840	230	2
18	2	250	130	146	2920	136	3
19	2	280	130	177	3540	164	1
20	2	280	130	231	4620	220	2
21	2	280	130	157	3140	148	3
22	3	370	100	175	3500	163	1
23	3	370	100	231	4620	218	2
24	3	370	100	163	3260	155	3
25	3	400	130	167	3340	155	1
26	3	400	130	223	4460	212	2
27	3	400	130	136	2720	126	3
28	3	370	130	181	3620	169	1
29	3	370	130	220	4400	209	2
30	3	370	130	136	2720	128	3
31	4	130	250	180	3600	169	1
32	4	130	250	241	4820	229	2
33	4	130	250	159	3180	150	3
34	4	100	280	185	3700	173	1
35	4	100	280	222	4440	210	2
36	4	100	280	160	3200	151	3
37	4	130	280	166	3320	153	1
38	4	130	280	225	4500	214	2
39	4	130	280	150	3000	142	3
40	5	370	250	159	3180	148	1
41	5	370	250	234	4680	221	2
42	5	370	250	164	3280	155	3
43	5	400	280	157	3140	146	1
44	5	400	280	219	4380	207	2
45	5	400	280	142	2840	134	3
46	5	370	280	185	3700	173	1

Sr	Node	X	Y	Time	Time (ms)	Path Length	Scenario
47	5	370	280	241	4820	229	2
48	5	370	280	136	2720	126	3
49	6	130	400	174	3480	163	1
50	6	130	400	221	4420	210	2
51	6	130	400	156	3120	148	3
52	6	100	370	185	3700	174	1
53	6	100	370	228	4560	217	2
54	6	100	370	153	3060	144	3
55	6	130	370	184	3680	171	1
56	6	130	370	246	4920	233	2
57	6	130	370	160	3200	151	3
58	7	280	400	171	3420	158	1
59	7	280	400	244	4880	233	2
60	7	280	400	164	3280	155	3
61	7	250	370	179	3580	166	1
62	7	250	370	242	4840	231	2
63	7	250	370	159	3180	149	3
64	7	280	370	176	3520	165	1
65	7	280	370	219	4380	208	2
66	7	280	370	156	3120	148	3
67	8	370	400	164	3280	153	1
68	8	370	400	222	4440	211	2
69	8	370	400	153	3060	143	3
70	8	400	370	171	3420	158	1
71	8	400	370	221	4420	210	2
72	8	400	370	137	2740	129	3
73	8	370	370	177	3540	165	1
74	8	370	370	240	4800	227	2
75	8	370	370	156	3120	147	3

References

[1] T. Lozano-Perez, “Spatial Planning: A Configuration Space Approach,” *IEEE Transactions on Computers*, vol. C-32. no. 2, 1983, https://doi.org/10.1109/TC.1983.1676196.

[2] J. F. Canny, “The Complexity of Robot Motion Planning,” *Cambridge MA: MIT Press*, 1988, https://books.google.co.id/books?id=_VRM_sczrKgC.

[3] K. Fujimura and H. Samet, “Planning a Time-Minimal Motion Among Moving Obstacles,” *Algorithmica*, vol. 10, no. 1, 1993, https://doi.org/10.1007/BF01908631.

[4] J.-C. Latombe, “Robot Motion Planning,” *Kluwer Academic Publishers Boston MA,* 1991, https://doi.org/10.1007/978-1-4615-4022-9.

[5] S. M. LaVall, “Planning Algorithms,” *Cambridge University Press*, 2006, https://doi.org/10.1017/CBO9780511546877.

[6] P. Fiorini, “Robot Motion Planning Among Moving Obstacles,” *Ph.D. Thesis University of California Los Angeles*, 1995, http://hdl.handle.net/2014/32127.

[7] P. Fiorini and Z. Shiller, “Robot Motion Planning in Dynamic Environments Using Velocity Obstacles,” *International Journal of Robotics Research*, July 1998, https://doi.org/10.1177/027836499801700706.

[8] M. G. Mohanan and A. Salgoankar, “A Survey of Robotic Motion Planning In Dynamic Environments,” *Elsevier Robotics and Autonomous Systems*, vol. 100, pp. 171-185, 2018, https://doi.org/10.1016/j.robot.2017.10.011.

[9] E. Owen and L. Montano, “Motion Planning in Dynamic Environments Using the Velocity Space,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, https://doi.org/10.1109/IROS.2005.1545110.

[10] D. Fox, W. Burgard, and S. Thrun, “The Dynamic Window Approach to Collision Avoidance,” *IEEE Robotics and Automation Magazine*, vol. 4, pp.23-33, 1997, https://doi.org/10.1109/100.580977.

[11] F. Large, C. Laugier, and Z. Shiller, "Navigation Among Moving Obstacles Using the NLVO: Principles and Applications to Intelligent Vehicles," *Autonomous Robots,* vol. 19, no. 2, pp. 159-171, 2005, https://doi.org/10.1007/s10514-005-0610-8.

[12] F. Large, S. Shekavat, Z. Shiller, and C. Laugier, "Using Non Linear Velocity Obstacles to Plan Motion in A Dynamic Environment," *ICARCV Conference: Control Automation Robotics and Vision on 7th International Conference,* vol. 2, pp. 734-739, 2022, https://ieeexplore.ieee.org/abstract/document/1238513/.

[13] S. Petti and T. Fraichard, "Safe Motion Planning in Dynamic Environments," *Proceedings of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2210-2215, 2005, https://doi.org/10.1109/IROS.2005.1545549.

[14] L. Martinez Gomez and T. Fraichard, "Collision Avoidance in Dynamic Environments," *ICS-Based Solution and its comparative evaluation IEEE Int. Conf. on Robotics and Automation,* pp. 100-105, 2009, https://doi.org/10.1109/ROBOT.2009.5152536.

[15] O. Gal, Z. Shiller, and E. Rimon, "Efficient and Safe On-Line Motion Planning in Dynamic Environments," *IEEE on ICRA International Conference on Robotics and Automation*, pp. 88-93, 2009, https://doi.org/10.1109/ROBOT.2009.5152330.

[16] Z. Shiller, O. Gal, and T. Fraichard. "The Nonlinear Velocity Obstacle Revisited: The Optimal Time Horizon," *Guaranteeing Safe Navigation in Dynamic Environments workshop,* 2010, https://doi.org/10.1007/978-3-7091-0277-0_26.

[17] P. Fiorini and Z. Shiller, "Robot Motion Planning in Dynamic Environments," *The International Symposium of Robotic Research*, pp. 237-248, Oct-1995, https://doi.org/10.1007/978-1-4471-1021-7_27.

[18] P. Fiorini and Z. Shiller, "Time Optimal Trajectory Planning in Dynamic Environments," *IEEE International Conference of Robotics and Automation*, vol. 2, pp. 1553-1558, April 1996, https://doi.org/10.1109/ROBOT.1996.506925.

[19] O. Gal and Z. Shiller, "Mapping Obstacles to Collision States For On-Line Motion Planning In Dynamic Environments," IEEE International Conference on Robotics And Automation, 2009, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.186.6578&rep=rep1&type=pdf.

[20] D, Althoff, "Safety Assessment For Motion Planning In Uncertain And Dynamic Environments," *Ph.D. Thesis Technical University of Munich,* 2014, https://mediatum.ub.tum.de/doc/1173997/file.pdf.

[21] N. E. Du Toit, "Robot Motion Planning in Dynamic Cluttered and Uncertain Environments: The Partially Closed-Loop Receding Horizon Control Approach," *Ph.D. Thesis California Institute of Technology*, 2010, https://doi.org/10.1109/ROBOT.2010.5509278.

[22] K. Fujimura, "Motion Planning in Dynamic Environments," *Springer Science and Business Media*, 2012, https://doi.org/10.1109/ROBOT.2010.5509278.

[23] M. Dorigo, T. Stiitzle, "Ant Colony Optimization," *MIT Press*, 2004, https://doi.org/10.1504/IJBIC.2010.032124.

[24] X. Yang, "Firefly Algorithm Stochastic Test Functions and Design Optimisation," *Int. J. Bio-Inspired Computation,* vol. 2, no. 2, pp. 78-84, 2010, https://arxiv.org/pdf/1003.1409.pdf.

[25] X. S. Yang, *Nature-inspired Metaheuristic Algorithms*, Luniver Press, 2010, https://books.google.co.id/books?id=iVB_ETlh4ogC.

[26] D. Vasquez, "Incremental Learning for Motion Prediction of Pedestrians and Vehicles" *Ph.D. dissertation Institut Nationale Polytechnique de Grenoble Grenoble FR,* 2007, https://www.theses.fr/2007INPG0023.

[27] M. M. Gangadharan and A. Salgaonkar, "Ant Colony Optimization and firefly Algorithms for robotic Motion Planning in Dynamic Environments," *Engineering Reports*, vol. 2, no. 3, 2020, https://doi.org/10.1002/eng2.12132.

[28] L. Martinez-Gomez, "Safe Navigation for Autonomous Vehicles in Dynamic Environments: An Inevitable Collision State (ICS) Perspective," *PhD Thesis Université de Grenoble*, 2010, https://tel.archives-ouvertes.fr/tel-00600578/document.

[29] M. Bennewitz, "Mobile Robot Navigation in Dynamic Environments," *Ph.D Thesis University of Freiburg Germany*, 2004, http://ais.informatik.uni-freiburg.de/publications/papers/thesis_bennewitz.pdf.

[30] D. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 2012, https://books.google.co.id/books?id=qVBEEAAAQBAJ.

[31] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable Collision States: A Probabilistic Perspective," *Proc. IEEE Int. Conf. Robotics and Automation,* pp. 4022 –4027, May 2010, https://doi.org/10.1109/ROBOT.2010.5509233.

[32] R. J. Mackay, "Hidden Markov Models: Multiple Processes and Model Selection," *Ph.D. Thesis The University of British Columbia*, 2003, http://hdl.handle.net/2429/16862.

[33] C. Fulgenzi, "Autonomous Navigation in Dynamic Uncertain Environment Using Probabilistic Models Of Perception And Collision Risk Prediction," *Ph.D. Thesis Institut National Polytechnique de Grenoble-INPG*, 2009, https://tel.archives-ouvertes.fr/tel-00398055/.

[34] L. Montesano, J. Minguez, and L. Montano, "Modeling Dynamic Scenarios for Local Sensor-Based Motion Planning," *Autonomous Robots,* vol. 25, no. 3, pp. 231-251, 2008, https://doi.org/10.1007/s10514-008-9092-9.

[35] P. Ondruska, "Learning Complex Markov Models for Mobile Robotics," *DPhil Thesis Draft St. Hilda's College University of Oxford*, July 2016, https://www.robots.ox.ac.uk/~mobile/Theses/ondruska_thesis.pdf.

[36] K. Chai Sim, "Structured Precision Matrix Modelling for Speech Recognition," *Dissertation submitted to the University of Cambridge for the degree of Doctor of Philosophy,* 2006, http://mi.eng.cam.ac.uk/~mjfg/thesis_kcs23.pdf.

[37] Q. Zhu, "Hidden Markov Model for Dynamic Obstacle Avoidance of Mobile Robot Navigation," *IEEE Transactions on Robotics and Automation,* vol. 7, no. 3, 1991, https://doi.org/10.1109/70.88149.

[38] T. Fraichard and H. Asama, "Inevitable Collision States: A Step Towards Safer Robots," *Advanced Robotics*, vol. 18, no. 10, July 2004, https://doi.org/10.1163/1568553042674662.

[39] Z. Shiller, F. Large, S. Sekhavat, and C. Laugier, "Motion Planning in Dynamic Environments: Obstacles Moving Along Arbitrary Trajectories," *IEEE International Conference on Robotics and Automation,* vol. 4, pp. 3716-3721, 2001, http://www.liralab.it/IIT_school/files/Papers/WF-05-06.pdf.

[40] A. Sud, E. Andersen, S. Curtis, M. Lin, and D. Manocha, "Real-time Path Planning for Virtual Agents In Dynamic Environments," *ACM SIGGRAPH*, pp. 1-9, 2008, https://doi.org/10.1145/1401132.1401206.

[41] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics," *MIT Press,* 2005, https://www.emerald.com/insight/content/doi/10.1108/03684920610675292/full/html.

[42] S. M. LaValle, "Rapidly-exploring Random Trees: A New Tool for Path Planning," 2000, https://www.cs.csustan.edu/~xliang/Courses/CS4710-21S/Papers/06%20RRT.pdf.

[43] P. Svestka, "Robot Motion Planning Using Probabilistic Roadmaps," *Ph.D. Thesis Utrecht Univ,*1997. https://cir.nii.ac.jp/crid/1570854176381920896.

[44] L. K. Dale, "Optimization Techniques for Probabilistic Roadmaps," *Ph.D. Thesis Texas A&M University* 2000, https://www.elibrary.ru/item.asp?id=5328429.

[45] N. E. Du Toit and J. W. Burdick, "Probabilistic Collision Checking with Chance Constraints," *IEEE Transactions on Robotics*, vol. 27, no. 4, Aug 2011, https://ieeexplore.ieee.org/abstract/document/5738354/.

[46] www.readbag.com, http://www.readbag.com/robots-stanford-papers-thrun-probrob

[47] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory Algorithms and Implementations*, MIT Press, June 2005, https://books.google.co.id/books?id=S3biKR21i-QC.

[48] D. Vasquez, T. Fraichard, and C. Laugier, "Incremental Learning of Statistical Motion Patterns with Growing Hidden Markov Models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 403-416, 2009, https://doi.org/10.1109/TITS.2009.2020208.

[49] F. Dawood and C. Kiong Loo, "Robot Behaviour Learning Using Topological Gaussian Adaptive Resonance Hidden Markov Model," *Neural Computing and Applications*, vol. 27, no. 8, pp. 2509-2522, 2016. https://doi.org/10.1007/s00521-015-2021-x.

[50] E. L. Thorn Jr, "Autonomous Motion Planning Using a Predictive Temporal Method," *Ph.D. Thesis University of Florida*, Dec 2009, https://apps.dtic.mil/sti/pdfs/ADA544950.pdf.

[51] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable Collision States: A Probabilistic Perspective," *IEEE International Conference on Robotics and Automation 2010,* pp. 4022-4027, May 2010, https://doi.org/10.1109/ROBOT.2010.5509233.

[52] J. Hernández-Aceituno, L. Acosta, and J. D. Pinero, "Application of Time Dependent Probabilistic Collision State Checkers in Highly Dynamic Environments," *IEEE International Conference on Robotics and Automation. PLoS One,* vol. 10, no. 3, e0119930, 2015, https://doi.org/10.1371/journal.pone.0119930.

[53] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Readings in speech recognition*, vol. 77, no. 2, pp 267-296, Feb 1989, https://doi.org/10.1109/5.18626.

[54] D. Althoff, D. Wollherr, and M. Buss, "Safety Assessment of Trajectories for Navigation in Uncertain and Dynamic Environments," *IEEE International Conference on Robotics and Automation*, pp. 5407-5412, May 2011, https://doi.org/10.1109/ICRA.2011.5980115.

[55] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation–mobile Robot Navigation with Uncertainty in Dynamic Environment," *Proceedings IEEE international conference on robotics and automation,* pp. 35-40, 1999, https://doi.org/10.1109/ROBOT.1999.769927.

[56] F. Rohrmuller, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic Mapping of Dynamic Obstacles Using Markov Chains for Replanning in Dynamic Environments," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 2504-2510, 2008, https://doi.org/10.1109/IROS.2008.4650952.

[57] C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier, "Probabilistic Navigation In Dynamic Environment Using Rapidly-Exploring Random Trees And Gaussian Processes," *IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp. 1056-1062, 2008, https://doi.org/10.1109/IROS.2008.4650959.

[58] R. E. Bellman, *Dynamic Programming*, Princeton University Press, 2021, https://doi.org/10.1515/9781400835386.

[59] R. S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018, https://books.google.co.id/books?id=uWV0DwAAQBAJ.

[60] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, 2015, https://doi.org/10.1515/9781400874668.

[61] L. K. Platzman, "Finite Memory Estimation and Control of Finite Probabilistic Systems," *Ph.D. Thesis Massachusetts Institute of Technology*, January 1977, https://apps.dtic.mil/sti/citations/ADA038813.

[62] M. Farrokhsiar, "MPC-based Mobile Robots Motion Planning and Control in Uncertain Dynamic Environment," *Ph.D. Thesis University of British Columbia*, April 2015, https://central.bac-lac.gc.ca/.item?id=TC-BVAU-52950&op=pdf&app=Library&oclc_number=1032940869.

[63] D. Held, D. Guillory, B. Rebsamen, S. Thrun, S. Savarese, "A Probabilistic Framework For Real-Time 3D Segmentation Using Spatial Temporal and Semantic Cues," *Robots: Science and Systems (RSS)*, vol. 12, 2016, http://davheld.github.io/segmentation3D/segmentation_extended.pdf.

[64] R. A. Howard, "Dynamic Programming and Markov Process," *MIT Press*, 1960, https://psycnet.apa.org/record/1961-01474-000.

[65] M. Bennewitz, W. Burgard, and S. Thrun, "Learning Motion Patterns of Persons for Mobile Service Robots," *Proc. of the IEEE Int. Conf. On Robotics and Automation Washington USA*, pp. 3601-3606, 2002, https://doi.org/10.1109/ROBOT.2002.1014268.

[66] M. Althoff, O. Stursberg, and M. Buss, "Model-based Probabilistic Collision Detection in Autonomous Driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 299 – 310, 2009, https://doi.org/10.1109/TITS.2009.2018966

[67] J. Hernández-Aceituno, L. Acosta, and José D. Piñeiro, "Application of time dependent probabilistic collision state checkers in highly dynamic environments," *PLoS One,* vol. 10, no. 3, 2015, https://doi.org/10.1371/journal.pone.0119930.

[68] D. Fox, W. Burgard, and S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environment," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391-427, 1999, https://doi.org/10.1613/jair.616.

[69] S. Thrun, "Robotic mapping: A Survey School," *Carnegie Mellon University,* 2002, http://robots.stanford.edu/papers/thrun.mapping-tr.pdf.

[70] S. Thrun, "Probabilistic Algorithms in Robotics," *Ai Magazine*, vol. 21, no. 4, p. 93, 2000, https://doi.org/10.1609/aimag.v21i4.1534.

[71] L. Moreno, S. Garrido, and M. L. Munoz, "Evolutionary Filter for Robust Mobile Robot Global Localization," *Robotics and Autonomous Systems*, vol. 54, no. 7, pp. 590-600, 2006, https://doi.org/10.1016/j.robot.2006.02.009.

[72] D. Fox, "Makov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation," *Ph.D. Thesis, Dept. of Computer Science, University of Bonn. Germany*, December 1998.

[73] D. Althoff, D. Wollherr, and M. Buss, "Safety Assessment of Trajectories For Navigation In Uncertain and Dynamic Environments Proceedings," *IEEE International Conference on Robotics and Automation*, pp. 5407-5412, May 2011, https://doi.org/10.1109/ICRA.2011.5980115.

[74] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal Navigation–Mobile Robot Navigation with Uncertainty In Dynamic Environment," *Proceedings 1999 IEEE international conference on robotics and automation*, pp. 35-40, 1999, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.5110&rep=rep1&type=pdf.

[75] N.E. Du Toit, J.W. Burdick, "Robot Motion Planning in Dynamic Uncertain Environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, 2011, https://doi.org/10.1109/TRO.2011.2166435.

[76] M.G.Mohanan, A. Salgaonkar, "Probabilistic Approach to Robot Motion Planning In Dynamic Environments," *SN Computer Science*, vol. 1, no. 3, pp. 1-6, 2020, https://doi.org/10.1007/s42979-020-00185-0.

[77] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic Obstacle Avoidance In Uncertain Environment Combining Pvos And Occupancy Grid," *Proceedings of the IEEE Int. Conf. on Robotics and Automation*, Apr 2007, https://doi.org/10.1109/ROBOT.2007.363554.

[78] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic Checker for Crowded Environment," *IEEE International Conference on Robotics and Automation (ICRA),* pp. 1492-1498, May 2010, https://doi.org/10.1109/ROBOT.2010.5509369.

[79] A. Jain, S. Chauhan, A. Hirlekar, and S. Sarange, "Automated Restaurant Management System," *International Journal of Innovative Research in Electrical Electronics Instrumentation and Control Engineering*, vol. 4, no. 5, 2016, https://www.ijireeice.com/upload/2016/may-16/IJIREEICE%208.pdf.

[80] D. Mirgal, P. Parab, A. Puro, and B. Dakhare, "Smart Automated Restaurant," *IJESC*, vol. 8, no. 4, 2018, https://ijesc.org/upload/cd012d0986d211503636e18a80f18207.Smart%20Automated%20Restaurant.pdf.

[81] N. Mishra, D. Goyal, A. D. Sharma, "Automation in Restaurants: Ordering to Robots in Restaurant Via Smart Ordering System," *International Journal of Converging Technologies & Management,* vol. 4, no. 1, pp, 1-4, 2018, https://www.gyanvihar.org/researchjournals/ctmpaper2017/IJCTM%20Vol_4_Iss_1_1.pdf.

[82] M. Asif, M. Sabeel, and Z. Khan, "Waiter Robot - Solution to Restaurant Automation, " *1st Student Multi Disciplinary Research Conference (MDSRC)*, Nov 2015.

[83] L. Jaillet and T. Simeon, "A PRM-based Motion Planner for Dynamically Changing Environments," *Proceedings of the IEEE Int. Conference on Robots & Systems*, vol. 2, pp. 1606-1611, 2004, https://hal.laas.fr/hal-01987651/document.

[84] L.E. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Trans. Robot. Autom*, vol. 12, no. 4, p. 5, 1996, https://doi.org/10.1109/70.508439.

[85] L. Jaillet, J. Cortes, and T. Simeon, "Sampling-based Path Planning on Configuration-Space Costmaps," *IEEE Transactions on Robotics*, vol. 26, no. 4, August 2010, https://doi.org/10.1109/70.508439.

[86] H. Ulbrich, J. Baur, J. Pfaff, and C. Schuetz, "Design and Realization of A Redundant Modular Multipurpose Agricultural Robot," *DINAME - Proceedings of the XVII International Symposium on Dynamic Problems of Mechanics Brazil,* pp. 22-27, February 2015, https://mediatum.ub.tum.de/1245752.

[87] J. Baur, C. Schütz, J. Pfaff, T. Buschmann, and H. Ulbrich, "Path Planning for a Fruit Picking Manipulator," *International Conference of Agricultural Engineering*, 2014, http://www.geyseco.es/geystiona/adjs/comunicaciones/304/C03090001.pdf.

[88] N. Reed, S.J. Miles, J. Butler, M. Baldwin, and R. Noble, "Automatic Mushroom Harvester Development," *Journal of Agricultural Engineering,* vol. 78, no. 1, pp. 15-23, 2001, https://doi.org/10.1006/jaer.2000.0629.

[89] J. H. Rowley, "Developing Flexible Automation for Mushroom Harvesting (Agaricus bisporus) : innovation report," *Ph.D. Thesis the University of Warwick*, June 2009, http://wrap.warwick.ac.uk/36893/.

[90] O. Ringdahl, "Automation in Forestry–Development of Unmanned Forwarders," *Ph.D. Thesis Department of Computing Science Umeå University,* 2011, https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A412664&dswid=-8601.

[91] S. Bachche, "Deliberation on Design Strategies of Automatic Harvesting Systems: A Survey," *Robotics,* vol. 4, no. 2, pp. 194-222, 2015, https://doi.org/10.3390/robotics4020194.

[92] H. M. Dee, "Explaining Visible Behaviour," *Ph.D. Thesis University of Leeds,* 2005, https://etheses.whiterose.ac.uk/1338/.

[93] J. P. Van den Berg, "Path Planning in Dynamic Environments," *Ph.D. Thesis Utrecht University*, 2007, https://dspace.library.uu.nl/handle/1874/20873.

[94] P. Svestka, "Robot Motion Planning Using Probabilistic Roadmaps," *Ph.D. Thesis Utrecht Univ,* 1997, https://cir.nii.ac.jp/crid/1570854176381920896.

[95] 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, October 2005, https://doi.org/10.1109/TRO.2005.851378.

[96] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation," *IEEE International Conference on Robotics and Automation (ICRA),* pp. 1928-1935, 2008, https://doi.org/10.1109/ROBOT.2008.4543489.

[97] P. Svestka and M. H. Overmars, "Motion Planning for Car-Like Robots a Probabilistic Learning Approach," *Int. Journal of Robotics Research*, vol. 16, pp. 119-143, 1997, https://doi.org/10.1177/027836499701600201.

[98] P. Svestka, M.H. Overmars, "Coordinated Path Planning for Multiple," *Robots Robotics and Autonomous Systems,* vol. 23, pp. 125-152, 1998, https://doi.org/10.1016/S0921-8890(97)00033-X.

[99] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "MAPRM: A Probabilistic Roadmap Planner with Sampling on The Medial Axis of The Free Space," *Proc. IEEE Int.Conf. on Robotics and Automation*, vol. 2, pp. 1024-1031, 1999, https://doi.org/10.1109/ROBOT.1999.772448.

[100] J. Cortes, T. Simeon, and J. P. Laumond, "A Random Loop Generator for Planning the Motions of Closed Kinematic Chains Using PRM Methods," *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 2141-2146, 2002, https://doi.org/10.1109/ROBOT.2002.1014856.

[101] L. K. Dale and N. M. Amato, "Probabilistic Roadmaps -Putting It All Together*," Proc IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1940-1947, 2001, https://doi.org/10.1109/ROBOT.2001.932892.

[102] J. P. Van den Berg and M. H. Overmars, "Roadmap-based Motion Planning in Dynamic Environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, October 2005, https://doi.org/10.1109/TRO.2005.851378.

[103] R. Philippsen, "Motion Planning And Obstacle Avoidance For Mobile Robots in Highly Cluttered Dynamic Environments," *Ph.D. Thesis EPF Lausanne*, 2004, https://infoscience.epfl.ch/record/33615.

[104] L. Han and N. M. Amato, "A Kinematics-Based Probabilistic Roadmap Method for Closed Chain Systems," *Algorithmic and Computational Robotics*, 2001, https://www.taylorfrancis.com/chapters/edit/10.1201/9781439864135-33.

[105] J. Rong, P. Wang, Q. Yang, and F. Huang, "A Field-Tested Harvesting Robot for Oyster Mushroom in Greenhouse," *Agronomy*, vol. 11, no. 6, p. 1210, Jun. 2021, https://doi.org/10.3390/agronomy11061210.

[106] J. Barraquand, L. Kavraki, J. C. Latombe, T. Y. Li, R. Motwani, and P. Raghavan, "A Random Sampling Scheme for Path Planning," *Int. Journal of Robotics Research*, vol. 16, no. 6, pp. 759-774. 1997, https://doi.org/10.1177/027836499701600604.

[107] G. Van Den Bergen, "Collision Detection in Interactive 3D Computer Animation," *Ph.D. Thesis Eindhoven University*, 1999, https://research.tue.nl/en/publications/collision-detection-in-interactive-3d-computer-animation.

[108] R. Bohlin and L. E. Kavraki, "Path Planning Using Lazy PRM," *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 521-528, 2000, https://doi.org/10.1109/ROBOT.2000.844107.

[109] V. Boor, M. H. Overmars, A. F. Van Der Stappen, "The Gaussian Sampling Strategy for Probabilistic Roadmap Planners," *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 1018-1023, 1999, https://doi.org/10.1109/ROBOT.1999.772447.

[110] D. Hsu, R. Kindel, J. C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233-255, 2002, https://doi.org/10.1177/027836402320556421.

[111] A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning," *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI),* vol. 95, pp. 1652-1659, 1995, https://dl.acm.org/doi/abs/10.5555/1643031.1643113.

[112] S. Koenig and M. Likhachev, "Improved Fast Replanning for Robot Navigation in Unknown Terrain," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 968-975, 2002, https://doi.org/10.1109/ROBOT.2002.1013481.

[113] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An Efficient Approach to Single Query Path Planning," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp. 995-1001, 2000, https://doi.org/10.1109/ROBOT.2000.844730.

[114] P. Cheng and S. M. LaValle, "Resolution Complete Rapidly-Exploring Random Trees," *Proceedings IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 1, pp. 267-272, 2002, https://doi.org/10.1109/ROBOT.2002.1013372.

[115] M. Pivtoraiko and A. Kelly, "Fast and Feasible Deliberative Motion Planner for Dynamic Environments," *IEEE ICRA,* vol. 1, no. 2, 2009, https://www.ri.cmu.edu/pub_files/2009/5/pivtoraiko_kelly_icra09.pdf.

[116] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Probabilistic Rapidly-Exploring Random Trees for Autonomous Navigation Among Moving Pedestrians," *IEEE ICRA*, 2009, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.186.9418&rep=rep1&type=pdf.

[117] S. M. LaValle, "Rapidly- Exploring Random Trees: A New Tool for Path Planning," *IEEE ICRA*, 1998, https://www.cs.csustan.edu/~xliang/Courses/CS4710-21S/Papers/06%20RRT.pdf.

[118] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Pearson Education, 2005, https://books.google.co.id/books?id=hRzOp_qdxG8C.

[119] Z. R. Novakovic and B. Nemee, "A Solution of The Inverse Kinematics Problem Using the Sliding Mode," *IEEE Transactions on Robotics and Automation,* vol. 6, no. 2, pp. 247-252, 1990, https://doi.org/10.1109/70.54740.

[120] C. Welman, "Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation," *Ph.D. ThesisSimon Fraser University*, 1993, https://cseweb.ucsd.edu/classes/wi20/cse169-a/readings/welman.pdf.

[121] S. Kucuk and Z. Bingul, "Robot Kinematics: Forward and Inverse Kinematics," *Industrial Robotics: Theory, Modelling and Control, IntechOpen*, 2006, https://doi.org/10.5772/5015.

[122] G. Jin and J. Ni, "Analysis and Simulation on The Kinematics of Robot Dexterous Hand," *2nd International Conference on Electronics Network and Com, Atlantis Press*, pp. 123-127, 2016, https://dx.doi.org/10.2991/icence-16.2016.26.

[123] W. Kwan, "Closed-form and Generalised Inverse Kinematic Solutions for Animating the Human Articulated Structure," *Curtin University of Technology*, 1996, https://documents.uow.edu.au/~/kwanwu/Honours-Thesis.pdf.

[124] R. P. Paul, *Robot Manipulators: Mathematics Programming and Control*, *MIT Press*, 1981, https://books.google.co.id/books?id=UzZ3LAYqvRkC.

[125] A. H. Watt and M. Watt, *Advanced Animation and Rendering Techniques*, ACM Press, 1992, https://books.google.co.id/books?id=HvgYAQAAIAAJ.

[126] L. Barinka and Ing. R. Berka, "Inverse kinematics - basic methods," 2002, https://old.cescg.org/CESCG-2002/LBarinka/paper.pdf.

[127] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimisation by A Colony of Cooperating Agents," *IEEE Transactions on Systems Man and Cybernetics–Part B*, vol. 26, no. 1, pp.1-13, 1996, https://doi.org/10.1109/3477.484436.

[128] M. Brand, M. Masuda, N. Wehner, and X.H. Yu, "Ant Colony Optimization Algorithm for Robot Path Planning," *International Conference on Computer Design and Applications (ICCDA),* vol. 3, pp. V3-436, 2010, https://doi.org/10.1109/ICCDA.2010.5541300.

[129] F. K. Purian, F. Farokhi, and R. S. Nadooshan, "Comparing The Performance Of Genetic Algorithm And Ant Colony Optimization Algorithm For Mobile Robot Path Planning In The Dynamic Environments With Different Complexities," *Journal of Academic and Applied Studies*, vol. 3, no. 2, February 2013, pp. 29-44, 2013, http://academians.org/Media/Default/Articles/February2013/Feb2013-3.pdf.

[130] M. Brand and X. H. Yu, "Autonomous Robot Path Optimization Using Firefly Algorithm," *Proceedings of the 2013 International Conference on Machine Learning and Cybernetics,* vol. 3, pp. 1028-1032, 2013, https://doi.org/10.1109/ICMLC.2013.6890747.

[131] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the Multi-Objective Path Planning Problem in Mobile Robotics with A Firefly-Based Approach," *Soft Comput,* vol. 21, no. 4, pp. 949–964, 2017, https://doi.org/10.1007/s00500-015-1825-z.

[132] W. Li, D. Li, and J. Ni, "Diagnosis of Tapping Process Using Spindle Motor Current," *International Journal of Machine Tools & Manufacturing Elsevier,* vol. 43, no. 1, pp. 73-79, 2003, https://doi.org/10.1016/S0890-6955(02)00142-6.

[133] B. Zhang, F. Yang, and J. Wang, "Fundamental Aspects in Vibration Assisted Tapping*," Journal of Material Processing Technology Elsevier*, vol. 132, no. 1-3, pp. 345-352, 2003, https://doi.org/10.1016/S0924-0136(02)00950-0.

[134] K. Akka and F. Khaber, "Mobile Robot Path Planning Using an Improved Ant Colony Optimization," *International Journal of Advanced Robotic Systems*, vol. 15, no. 3, pp. 1-7, May-June 2018, https://doi.org/10.1177/1729881418774673.

[135] M. R. Zeng, L. Xi, and A. M. Xiao, "The Free Step Length Ant Colony Algorithm in Mobile Robot Path Planning," *Advanced Robotics*, vol. 30, no. 23, pp. 1509-1514, 2016, https://doi.org/10.1080/01691864.2016.1240627.

[136] B. K. Patle, A. Pandey, A. Jagadeesh, and D. R. Parhi, "Path Planning in Uncertain Environment by Using Firefly Algorithm," *Elsevier Defence Technology,* vol. 14, no. 6, pp. 691-701, Jun 2018, https://doi.org/10.1016/j.dt.2018.06.004.

[137] J. Cao, “Robot Global Path Planning Based on An Improved Ant Colony Algorithm,” *Journal of Computer and Communications*, vol. 4, no. 2, p. 11, 2016, https://doi.org/10.4236/jcc.2016.42002.

[138] X. S. Yang, “Firefly Algorithms for Multimodal Optimization,” *Stochastic Algorithms: Foundations and Applications SAGA Lecture Notes in Computer Sciences,* vol. 5792, pp. 169-178, 2009, https://doi.org/10.1007/978-3-642-04944-6_14.

[139] A. Pandey, “Mobile Robot Navigation in Static and Dynamic Environments Using Various Soft Computing Techniques,” *Ph.D. thesis National Institute of Technology Rourkela,* 2016, http://ethesis.nitrkl.ac.in/8038/.

[140] Y. Xue and J. Q. Sun, “Solving the Path Planning Problem in Mobile Robotics with The Multi-Objective Evolutionary Algorithm,” *Applied Sciences*, vol. 8, no. 9, p. 1425, August 2018, https://doi.org/10.3390/app8091425.

[141] L. Luo, H. Wen, Q. Lu, H. Huang, W. Chen, X. Zou, and C. Wang, “Collision-Free Path-Planning for Six-DOF Serial Harvesting Robot Based on Energy Optimal and Artificial Potential Field,” *Complexity,* 2018, https://doi.org/10.1007/978-3-642-04944-6_14.

[142] S. M. Zadeh, “Autonomous Reactive Mission Scheduling and Task-Path Planning Architecture for Autonomous Underwater Vehicle,” *Ph.D. Thesis Flinders University of South Australia*, Dec 2016, https://doi.org/10.48550/arXiv.1706.04189.

[143] C. M. Clark, “Dynamic Robot Networks: A Coordination Platform for Multi Robot Systems,” *Ph. D thesis Stanford University*, May 2004, https://digitalcommons.calpoly.edu/csse_fac/77/.

[144] M. L. Puterman, “Markov Decision Processes: Discrete Stochastic Dynamic Programming,” *John Wiley & Sons*, 2014, https://books.google.co.id/books?id=VvBjBAAAQBAJ.

[145] K. Kamei, S. Nishio, N. Hagita, and M. Sato, “Cloud Network Robotics,” *IEEE Network Magazine*, vol. 26, no. 3, pp. 28-34, May 2012, https://doi.org/10.1109/MNET.2012.6201213.

[146] M. G. Mohanan and A. Salgaonkar, “Robotic Mushroom Harvesting by Employing Probabilistic Road Map and Inverse Kinematics,” *BOHR International Journal of Future Robotics and Artificial Intelligence*, vol. 1, no.1, pp. 1-10, 2021, https://doi.org/10.54646/bijfrai.001.