Ant colony optimization and firefly algorithms for robotic motion planning in dynamic environments

Mohanan M. Gangadharan | Ambuja Salgaonkar

Abstract
Metaheuristic algorithms such as ant colony optimization (ACO) and firefly (FF) have been successfully employed to solve the optimization problems such as robot motion planning in dynamic environments. The systematic plantation of rubber trees on a rectangular grid motivated us to explore application of grid search algorithms. We compared the ACO and FF algorithms in various scenarios by changing simulation parameters like density of the environment, land size, number of robots simultaneously available, and hillock plantations. In all different scenarios, we evaluated the performance of ACO and FF in terms of path length and time of execution, we found that later is outperforming the former. Regression equations are framed to establish the contributions of different parameters. Statistical significance of the results has been in favor of this hypothesis. The shortest path on a plain land is the relatively simplest scenario, while the Hamiltonian on a concave surface is arguably the most difficult. The novelty of this work lies in the very idea of an autonomous robot for the rubber tapping and then path optimization by employing soft computing techniques. This proposal of rubber harvesting robot if implemented for latex collection, has a potential to drive the rubber farming and allied businesses to scale up the economy of the coastal areas of India, say for example, Kerala.

Keywords
dynamic environments, ant colony optimization, firefly algorithm, rubber harvesting robot

1 | INTRODUCTION

The classical approaches of robotic motion planning in dynamic environments (RMPDE) have limitations due to high computational cost and lack of accuracy for getting trapped into local minima. Heuristic approaches converge faster to an acceptable solution.\textsuperscript{1–8} Heuristic methods have outperformed the classical approaches of robotic motion planning and have become popular in the last four decades.\textsuperscript{9} Metaheuristic algorithms such as (i) ant colony optimization (ACO) and (ii) firefly algorithm (FF) have been successfully employed to solve these types of optimization problems.\textsuperscript{10} The neighborhood for finding an optimal path in the case of ACO and FF is of rectangular shape divided into rectangular cells, assuming that each cell follows eight-connectivity as shown in Figure 1.
Robotic motion planning for an agricultural robot has been discussed in References 11-13. A combination of probabilistic road map for navigation and inverse kinematics for plucking mushrooms has been suggested as a model for a mushroom harvesting robot. Rubber is one of the cash crops of South India. Rubber trees are cultivated systematically in a matrix, with around 225 plants grown in an acre of land. Tapping the plant, that is, taking a cut on the outer skin of the stem, in the early morning and collecting the latex (i.e., liquid rubber) after 3 hours requires the services of one person for two acres of land. Getting skilled labor is a problem faced by rubber farm owners. Employing robots for rubber harvesting is proposed as a solution. ACO and FF are the natural choices due to their inheritance properties of pathfinding on a rectangular grid.

We have simulated the rubber plantations for robotic motion planning by employing ACO and FF. Comparison between the two is the topic of this research. In all different scenarios, we evaluated the performance of ACO and FF in terms of path length and time of execution, we found that FF is outperforming ACO. Regression equations are framed to establish the contributions of different parameters. Statistical significance of the results has been in favor of this hypothesis. The shortest path on a plain land is the relatively simplest scenario, while the Hamiltonian on a concave surface is supposedly the most difficult. The novelty of this work lies in the very idea of an autonomous robot for the rubber tapping and then path optimization by employing soft computing techniques. This proposal of rubber harvesting robot (RHR) if implemented for latex collection, has a potential to drive the rubber farming and allied businesses to scale up the economy of the coastal areas of India.

We first give a brief literature survey of the rubber harvesting, followed by a survey of ACO and FF research in robotics motion planning. Our model for a RHR is discussed, followed by the results of simulation experiments. The findings have been presented toward the end.

2 | LITERATURE SURVEY

2.1 | Robotics in rubber harvesting

Rubber plants are cultivated in open field in the arrangement of rows and columns forming a matrix. The shortage of skilled rubber tappers and the high labor cost prevailed in the rubber farming has been the major problem faced by the 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, which is tied to the stem of the tree. After 2 to 3 hours of incision, the tapper collects the deposited latex at each and every plants in sequence and submits the same in the specific collection center for further processing of rubber. Every morning the rubber tapper empties the cup tied to each tree and makes another cut just above or below the previous cut, thus extracting more latex from the rubber tree. After about 25 years the rubber tree will stop producing latex, on which a new sapling is planted in its place.

Automation could help in overcoming the shortage of skilled rubber tappers and the resulting high labor costs. The idea of employing robots in rubber harvesting dates back to 1990. An implementation of an automated machine is given in Reference 16 that helps the logistics like at a rubber farm by following a predetermined path. Location estimates
are obtained by combining information from visual and odometric sensors as well as a real-time differential global positioning system and gyro compass. An extended Kalman filter\textsuperscript{17} has been employed for detection of obstacles. A redundant set of sensors combines multiple measurements into a good quality estimate and allows the detection of sensor failures.

A navigation control system developed for an autonomous tractor using a laser range finder (LRF) is given in Reference \textsuperscript{18}. LRF was used as a single sensor to detect objects and navigate as an autonomous agricultural tractor. The tractor navigation was decided based on point-to-go calculations, once the positions of the landmarks were recognized. It determines the steering angle while travelling by using the principle of relative position. The autonomous navigation system can manage operations efficiently in environments like rubber and palm oil plantations, while travelling between plantations along straight, curved, and U-shaped paths.

We propose to have the sensors and camera mounted on robots to identify the proper place of tapping on a rubber tree. ACO and FF are proposed for path finding; a rectangular grid is assumed with a cell representing a plant, an obstacle or a free space. Thus, a robot has to find a sequence of free cells to cover each cell that has a plant, take a cut or collect the latex whichever is the case. A dexterous robotic arm that employs inverse kinematics techniques\textsuperscript{20} has been proposed for carrying out the actually cutting and collection activities.

To the best of our understanding, this is the first proposal for the complete automation of the rubber collection activity.

3 | ACO AND FF

ACO is a nature-inspired algorithm. The concept has been derived from the behavior of biological ants in the real world. ACO was first introduced for various combinatorial optimization problems.\textsuperscript{21} The application of ACO for RMPDE has been investigated.\textsuperscript{2} The goal is to find the shortest collision-free path (if it exists) between a starting point and a destination point in a grid network. To simulate a dynamic environment, obstacles with different shapes and sizes are added after the optimal path is found in the original, that is, obstacle-free, network. Two different pheromone reinitialization schemes, (i) local initialization and (ii) global initialization are evaluated and compared with establish location. (In the present context, a pheromone is a chemical left behind by ants to mark a path or trail that other ants follow; the simulated version does something similar, see below.)

Computer simulation results are presented to show the effectiveness of the ACO algorithm.\textsuperscript{22} An improved ant colony algorithm uses a probability based on the number of obstacles around a cell to help the ant in its selection of the next cell and employs new heuristic information based on the principle of unlimited step length, where new global information is established based on an unlimited heuristic search to expand the vision field and to increase its accuracy.

The algorithm adopts a new pheromone updating rule and dynamic adjustment of the evaporation rate to speed up the convergence rate.\textsuperscript{23} Yet another variant of the ACO for robot global path planning is proposed,\textsuperscript{24} consisting of three stages: (i) adjust the pheromone evaporation rate dynamically to enhance the global search ability and convergence speed, (ii) modify the heuristic function to improve the state transition probabilities in order to find the optimal solution quickly, and (iii) change the pheromone update strategy to avoid premature solutions by strengthening the pheromone on the optimal path and limiting pheromone level.

For mobile robots in dynamic environments, a comparison of ACO and genetic algorithms (GA) with variable length chromosomes shows that ACO outperforms GA.\textsuperscript{25} Both algorithms work with global path planning and use a general map of the environment. A similar benchmarking of GA and ACO carried out in static environments also comes to the same conclusion.\textsuperscript{26} ACO has been employed for real-time globally optimal path planning of mobile robots.\textsuperscript{27} The MAKLINK graph is used to establish the free space model of the mobile robot (Figure 2). Dijkstra’s shortest path algorithm is used one part at a time to find a suboptimal collision-free path. ACO is employed to optimize the suboptimal path and generate a globally optimal path. A hybrid ACO algorithm for path planning in dynamic environment has been proposed.\textsuperscript{28} ACO and artificial potential field (APF) are employed, respectively, for global and local path planning. A pheromone generated by ACO is utilized to prevent APF from getting into a local minimum.

In Figure 2, the colored polygons are the obstacles, the thin black borders specify their boundaries, while the other thin black lines specify possible motion paths for the robots which using which appropriate motion planning is done.

A hybridization method is proposed for the path planning and navigation of humanoid robots in a cluttered two-dimensional environment by using regression technique and adaptive ACO with the applications in the domain of engineering.\textsuperscript{29} An improved ACO algorithm is proposed in Reference 30 for path planning of mobile robots in the grid environment, in which the pheromone diffusion and geometric local optimization are combined in the process of
searching for the globally optimal path. The fuzzy ant colony optimization method is proposed in Reference 31 in order to minimize the iterative learning error of ACO algorithm using fuzzy control in the path planning problem.

A swarm intelligence approach inspired by the biological behavior of fireflies (glow worms) has been applied to the robot path optimization problem for both static and dynamic environments. Computer simulation results show that the FF algorithm can be successfully employed for optimal path finding in a dynamic environment. A multiobjective path planning algorithm by using FF has been proposed. Path safety, path length, and smoothness of the path are the major concerns. Patle et al explores an environment and improves the global search in a small number of iterations and hence it can be implemented for real-time obstacle avoidance in dynamic environments. It addresses the challenges of navigation, minimizes the computational calculations, and avoids random movement of fireflies. The performance of the proposed controller is better in terms of path optimality when compared with other intelligent navigational approaches. FF is used for multimodal function optimization. A benchmarking of FF, particle swarm optimization, and GA shows that FF is superior in terms of efficiency and success rate and hence more powerful for solving NP-complete problems.

An improvised FF that involves the Q-learning, a model-free reinforcement algorithm, which tells an agent what action to take under the specific circumstances, is proposed and the optimal parameter values for each FF of a population are learnt. Tighzert et al introduces a set of new compact FF algorithms with minimal computational costs to a legged robot with a specific application. A new three-dimensional path planning method for autonomous underwater vehicles using a modified FF algorithm is proposed in order to improve the speed of convergence in FF algorithm by adjusting the brightness and attraction parameters. Reactive approaches such as FF and ACO perform better than classical approaches because they have a higher capability to handle uncertainty present in the environment. Reactive approaches are used for real-time navigation problems and classical approaches can be improved by hybridizing with the reactive approaches.

3.1 Basis data structure of ACO and FF

The working environment is visualized as a grid with many small rectangular cells that are treated as the unit steps for a robot. A robot can move to any of the adjacent cells. A cell occupied by an obstacle is shown in black or 1 and a free cell, with no color, is represented as 0. A robot has to reach its goal by traversing through the free cells. A cell can be considered as an intermediate cell (eight-connected) or a boundary cell (neighbors restricted by the grid boundary).

3.2 ACO algorithm

ACO has been inspired by the behavior of ants while finding an optimal path between the food source and their nest. The pheromone the ants leave behind the path they traverse is a kind of message for their followers. Initially groups of ants explore diverse paths randomly. They communicate the shortest partial paths through the pheromone. As more ants
follow a path the pheromone becomes more concentrated or “thicker.” Conversely, the thicker the pheromone, the greater
is the probability of follower ants selecting the path. The ants can adapt to environmental changes like obstacles on
their path and accordingly find new paths.

The probability of the ant \( k \) moving from the cell \( i \) to \( j \) is defined as follows.

\[
p_{ij}^k(t) = \begin{cases} 
\frac{\tau_{ij}^k(t) \eta_{ij}^k(t)}{\sum_{s \in \text{allowed}_k} \tau_{is}^k(t) \eta_{is}^k(t)}, & j \in \text{allowed}_k \text{(white grid)}, \\
0, & \text{black grid}
\end{cases}
\]  

(1)

where \( p_{ij}^k(t) \) is the transition probability in which the ant \( k \) will traverse from cell \( i \) to cell \( j \) at time \( t \), \( \tau_{ij}^k(t) \) is the intensity of the pheromone between cell \( i \) and \( j \) at time \( t \), \( \eta_{ij}^k(t) \) is the heuristic function between cells \( i \) and \( j \), \( \alpha \) is the information inspiration factor and \( \beta \) is the hope inspiration factor (these factors are weights) and \( \text{allowed}_k \) is the set of cells that ant \( k \) selects in the next step. When the target position is known, the ant can calculate distances from its surrounding eight cells to the target point. The heuristic function is given by \( \eta_{ij}^k(t) = c \frac{d_{ij}}{d_{ijg}} \), where \( c \) is the heuristic factor (a constant) and \( d_{ijg} \) is the distance of the next grid to the target grid \( g \). The path with obstacles (black cells) is avoided by the value of the transition probability \( p_{ij}^k(t) \), which is 0 as per Equation (1) and the pheromone concentration level is very low.

### 3.3 Pheromone update strategy

In traditional ACO, the evaporation rate \( \rho \) is a constant in the range \((0,1)\), and it has a direct impact on the global search ability and the convergence speed of the algorithm. If \( \rho \) is too small, the pheromone will evaporate very slowly, which reduces the global search ability and brings the algorithm to local convergence. If \( \rho \) is too large, the global search will be improved and the convergence speed will be reduced.

Pheromone update by using the ant-cycle model, which can be expressed as follows:

\[
\Delta \tau_{ij}^k(0) = 0.
\]

\[
L \leq \frac{L_{\text{best}} + L_{\text{worst}}}{2} \quad \text{(average length of the best and worse paths)},
\]

\[
\tau_{ij}(t + 1) = (1 - \rho) \tau_{ij}(t) + \Delta \tau_{ij},
\]

\[
\Delta \tau_{ij}^k = \begin{cases} 
\frac{Q}{L}, & \text{if ant } k \text{ uses edge } (i,j) \text{ in its tour} \\
0, & \text{otherwise}
\end{cases}
\]

where \( Q \) is the pheromone constant.

### 3.4 FF algorithm

FF algorithm is based on three rules:

1. All fireflies are unisex, one FF will be attracted to other irrespective of their sex.
2. Attractiveness is proportional to their brightness, dimmer will move toward the brighter one. Random movements when they are equally bright.
3. Brightness of fireflies is proportional to the value of the Objective function.

The light intensity \( I \) varies with the distance \( r \), given by \( I = I_0 \exp(-\gamma r) \), where \( I_0 \) is the original light intensity, and \( \gamma \) is the absorption coefficient that ranges between 0 and 1.

Attractiveness is given by \( \beta = \beta_0 \exp(-\gamma r^2) \), where \( \beta_0 \) is the attractiveness when \( r = 0 \).

The distance in \( d \)-dimensions between two fireflies \( i \) and \( j \) at \( x_i \) and \( x_j \), respectively, is given by

\[
r_{ij} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}.
\]
and it is \( r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \) in two dimensions.

The movement of FF \( i \) attracted to a brighter FF \( j \) is given by

\[
x_i = x_i + \beta_0 \ e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \left( \text{rand} - \frac{1}{2} \right),
\]

where the second term is due to attraction and the third one is due to randomization with control parameter \( \alpha \), which lies between 0 and 1.

The flashing behavior of a FF is used to determine the optimal path planning when the robot encounters static and dynamic obstacles. Path planning for a robot involves capturing suitable features for a feedback-based objective function that helps to attain a certain performance requirement for the following tasks: obstacle detection, obstacle avoidance, overcoming a trap-like situation, avoiding random walks, and hence generating an optimal path. The sensors that are mounted on a robot provide the information about the surroundings and help the robot to locate its position in an unknown environment.

The navigational path optimization problem is a minimization problem, of which the objective function, based on the goal and obstacle positions, is a desired parameter. The location of a globally brighter FF is computed in each iteration and the robot reaches the same in a series of movements. This function is recursively called in case the robot encounters an obstacle while reaching its immediate goal.

As soon as a robot encounters an obstacle, a specific number of random fireflies are generated in the vicinity of the obstacle, based on the robot’s sensor inputs. The Euclidean distance from each FF to the obstacle and to the goal is computed. Out of all the available fireflies, the brightest FF, the one that is at the maximum safe distance from the obstacle and at the minimum distance from the goal is selected, and the robot moves in its direction.

The path planning optimization problem can be formalized as follows:

\[
f_i = K_1 \min_{a_n \in \alpha} \frac{1}{D_{fo} || D_{fg} ||} + K_2 \ ||D_{fg}||,
\]

where \( D_{fo} \) is the distance of the FF \( f_i \) from the obstacle and \( D_{fg} \) is its distance from the goal, \( K_1 \) is the fitting parameter for the path safety and \( K_2 \) is the parameter that decides the maximum and minimum path length of the navigation. When the value of \( K_1 \) is maximum, the robot can safely avoid the obstacle. The minimum value of \( K_2 \) maximizes the path length and maximum value of \( K_2 \) minimizes the path length. The proper selection of the fitting parameters decides the optimality of the objective function, which can be obtained by trial and error.

In the following, we produce observations of our simulation of ACO and FF optimization. Essentially, we have recorded the path-length and the time taken for finding the same in the following scenarios: (i) path finding from source to destination in a plain field, (ii) Hamiltonian tour in a field on a plateau, and (iii) Hamiltonian tour in a field that has terrains of three heights. The control parameters are grid size, number of agents and the density of obstacles. The observed values are tabulated, a visualization of the path is provided, and a summary and interpretation has been given in the conclusion.

### 4 | SIMULATION EXPERIMENTS AND THEIR OUTCOME

#### 4.1 | Simulation of ACO and FF in general grids

First, we consider the case of evaluation of ACO and FF optimization algorithms for finding a path from source to destination in a plain field. Figure 3A-D show the path finding activity in progress. The grid size is 20 \( \times \) 20; the cells occupied by obstacles are shown in black. The path found by employing the ACO with 25 ants is shown in blue and that by employing the FF optimization with 25 fire flies has been marked in red. The screen shots show the position of path at the end of 15, 30, 43, and 53 seconds, respectively.

The final path length in the case of ACO is 40 while that is marked by FF is 33 cells. Clearly in this case the FF optimization outperformed the ACO by 17.5% in terms of path length; FF converged in 81.5% of the ACO time.
The simulation has been executed for five different grid sizes: 20 × 20, 40 × 40, 60 × 60, 80 × 80, and 100 × 100. For each grid, the number of agents started with 25, were increased to 50, 75, 100, and 200. Random dynamic environments were created with the obstacle density 10%, 30%, 50%, 70%, and 90%. The starting point is the left uppermost corner and goal is the right lowermost corner of the respective grids. The average of the best’s path-length and the corresponding time over five runs have been compiled as shown in Tables 1-5.

In four of 125 observations ACO has performed slightly better time-wise; it performed better path-length-wise in 15 of the 125 instances. We computed the statistical significance of this result. Let the null hypothesis in this case is that there is no significant difference in the execution time and path-lengths generated by employing ACO and FF in case of the pathfinding problem when worked out on a general grid. (ie, H0: Mean time taken by ACO and FF to compute the optimum path and the mean path-lengths computed by ACO and FF procedures are comparable). The alternative hypothesis (H1) is that there is a significant difference between the results generated by the two algorithms and in this case, the mean time taken by ACO to generate the optimum path and the mean path-length computed by ACO are greater than ones generated by FF. The T-statistics test is performed. The z-values and P-values are shown in the Table 6. Clearly, one can say with 95% confidence level that the null hypothesis is rejected, the results are significantly different in both the methods. In turns it supports the alternative hypothesis.
### TABLE 2  Time and path length variation for ACO and FF over a grid size 40 × 40

| No. of Ants/Fireflies Algorithm | 25 | 50 | 75 | 100 | 200 |
|--------------------------------|----|----|----|-----|-----|
| Density (%)                   |    |    |    |     |     |
| 10                             | 87 | 86 | 87 | 85  | 92  |
| 60                             | 58 | 62 | 62 | 63  | 65  |
| 30                             | 108| 91 | 104| 96  | 119 |
| 75                             | 66 | 75 | 74 | 82  | 66  |
| 50                             | 106| 98 | 116| 114 | 119 |
| 73                             | 70 | 78 | 77 | 74  | 68  |
| 70                             | 113| 100| 114| 116 | 116 |
| 81                             | 79 | 79 | 76 | 77  | 75  |
| 90                             | 127| 104| 126| 115 | 127 |
| 87                             | 87 | 84 | 84 | 85  | 82  |

**Abbreviations:** ACO, ant colony optimization; FF, firefly.

### TABLE 3  Time and path length variation for ACO and FF grid size 60 × 60

| No. of Ants/Fireflies Algorithm | 25 | 50 | 75 | 100 | 200 |
|--------------------------------|----|----|----|-----|-----|
| Density (%)                   |    |    |    |     |     |
| 10                             | 130| 126| 142| 134 | 134 |
| 92                             | 92 | 97 | 88 | 98  | 88  |
| 30                             | 189| 146| 193| 154 | 166 |
| 114                            | 108| 117| 109| 107 | 109 |
| 50                             | 198| 167| 177| 172 | 174 |
| 113                            | 113| 115| 114| 119 | 115 |
| 70                             | 196| 175| 187| 188 | 201 |
| 114                            | 113| 120| 115| 121 | 126 |
| 90                             | 198| 171| 195| 172 | 206 |
| 123                            | 122| 125| 122| 122 | 129 |

**Abbreviations:** ACO, ant colony optimization; FF, firefly.

### TABLE 4  Time and path length variation for ACO and FF grid size 80 × 80

| No. of Ants/Fireflies Algorithm | 25 | 50 | 75 | 100 | 200 |
|--------------------------------|----|----|----|-----|-----|
| Density (%)                   |    |    |    |     |     |
| 10                             | 182| 170| 177| 167 | 173 |
| 120                            | 118| 136| 135| 121 | 122 |
| 30                             | 237| 191| 265| 199 | 252 |
| 152                            | 149| 158| 139| 151 | 146 |
| 50                             | 251| 244| 263| 227 | 251 |
| 155                            | 150| 155| 160| 156 | 150 |
| 70                             | 255| 250| 278| 246 | 255 |
| 144                            | 145| 164| 153| 157 | 161 |
| 90                             | 276| 267| 289| 254 | 278 |
| 167                            | 165| 168| 162| 164 | 163 |

**Abbreviations:** ACO, ant colony optimization; FF, firefly.
### TABLE 5  Time and path length variation for ACO and FF grid size 100 × 100

| No. of Ants/Fireflies | ACO  | FF  | ACO  | FF  | ACO  | FF  | ACO  | FF  | ACO  | FF  |
|----------------------|------|-----|------|-----|------|-----|------|-----|------|-----|
|                      | 25   | 50  | 75   | 100 | 200  |
| Density (%)          |      |     |      |     |      |
| 10                   | 237  | 213 | 216  | 215 | 221  | 215 |
| 170                  | 152  | 171 | 171  | 170 | 169  | 166 |
| 30                   | 321  | 239 | 310  | 241 | 308  | 257 |
| 188                  | 175  | 197 | 173  | 192 | 193  | 188 |
| 50                   | 305  | 280 | 300  | 285 | 314  | 274 |
| 195                  | 187  | 194 | 180  | 198 | 195  | 194 |
| 70                   | 299  | 290 | 304  | 296 | 331  | 298 |
| 199                  | 195  | 208 | 200  | 214 | 208  | 198 |
| 90                   | 308  | 304 | 317  | 304 | 353  | 327 |
| 221                  | 201  | 216 | 204  | 218 | 215  | 212 |

Abbreviations: ACO, ant colony optimization; FF, firefly.

### TABLE 6  The z-values and P-values for the comparison test

| No. | Type               | z-Value | P-Value | Result at \( P < .05 \) |
|-----|--------------------|---------|---------|-------------------------|
| 1   | Time comparison    | 10.3692 | <.00001 | Significant             |
| 2   | Path comparison    | 7.49269 | <.00001 | Significant             |

### 4.2  Simulation of a Hamiltonian tour of a Latex collector robot in the rubber plantations of various farm sizes

Started at the upper left corner a robot reaches to one of the eight nearest neighbors. The reached node becomes the new start and the new goal is one amongst its nonserved nears neighbors. The procedure is called recursively till the robot serves the last nonserved plant in a given farm.

A progress of path finding in an instance of ACO has been shown in the Figure 4A-C with blue color and that of FF in Figure 4D-F has been shown in red. A farm of 10 × 10 = 100 plants has been simulated. Initially all plants are blue; as soon as a robot reaches to a plant it turns into green. Obstacles are shown in black.

The average time taken to complete a tour over five 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 Tables 7-11 show the average of the minimum times and that of the path-lengths as recorded in the simulation of the farms with 100, 400, 900, 1600, and 2500 plants.

Continuing with the last null hypothesis that there is no significant difference in the results when either of the two algorithms ACO or FF is employed the \( T \)-statistics have been computed on the above results that is given in Table 12. It clearly shows that the null hypothesis is rejected. Hence there is a point in recommending FF over ACO.

The above experiment is the proof of concept model for implementation of a robot for the real RHR. The results are encouraging. A real robot needs to be engineered and to be deployed accordingly. Yet another parameter that is the evenness of the ground also needs to be computed.

### 4.2.1  Simulation of rubber matrix grids in sloped terrain with different height maps

In the simulation, grids of two different sizes, namely, 10 × 10 and 20 × 20, representing the rubber planted at three different heights, the normal sea-level: \( h_1 \), the next level: \( h_2 \), and the final level: \( h_3 \), are considered. With each grid size with number of ants and fireflies are 25, 50, and 75. Random dynamic environments are created with the obstacle density 10%, 30%, 50%, and 70%. The robot transition path is from \( h_1 \) to \( h_2 \) and \( h_2 \) to \( h_3 \) and vice versa as shown in Figure 5A.
FIGURE 4 Hamiltonian tour of ACO and FF in plain rubber field (Figure 4). ACO, ant colony optimization; FF, firefly.

TABLE 7 Time and path length variation for ACO and FF over a plain grid, size 10x10

| No. of Ants/Fireflies Algorithm | 25 | 50 | 75 |
|--------------------------------|----|----|----|
|                                | ACO | FF | ACO | FF | ACO | FF |
| Density (%)                    |     |    |     |    |     |    |
| 10                             | 678 | 544 | 680 | 551 | 661 | 561 |
| 30                             | 1112| 984 | 1184| 972 | 1184| 988 |
| 50                             | 1532| 1341| 1521| 1367| 1521| 1362|
| 70                             | 2174| 1896| 2184| 1905| 2184| 1874|

Abbreviations: ACO, ant colony optimization; FF, firefly.
### Table 8
Time and path length variation for ACO and FF over a plain rubber grid of size 20 × 20

| No. of Ants/Fireflies Algorithm | 25 | 50 | 75 |
|-------------------------------|----|----|----|
| ACO              | FF          | ACO  | FF  | ACO  | FF  |
| Density (%) 10        | 2735         | 2204 | 2765 | 2218 | 2796 | 2244 |
|                   | 1602         | 1558 | 1614 | 1564 | 1602 | 1553 |
|                   | 30           | 3140 | 2574 | 3177 | 2588 | 3186 | 2598 |
|                   | 1724         | 1584 | 1738 | 1597 | 1724 | 1574 |
|                   | 50           | 3595 | 2987 | 3617 | 3021 | 3621 | 3004 |
|                   | 1684         | 1601 | 1688 | 1635 | 1684 | 1617 |
|                   | 70           | 4028 | 3475 | 4075 | 3504 | 4082 | 3497 |
|                   | 1705         | 1600 | 1713 | 1640 | 1705 | 1650 |

Abbreviations: ACO, ant colony optimization; FF, firefly.

### Table 9
Time and path length variation for ACO and FF over a plain rubber grid of size 30 × 30

| No. of Ants/Fireflies Algorithm | 25 | 50 | 75 |
|-------------------------------|----|----|----|
| ACO              | FF          | ACO  | FF  | ACO  | FF  |
| Density (%) 10        | 6504         | 4949 | 6524 | 5007 | 6594 | 5064 |
|                   | 3880         | 3535 | 3896 | 3563 | 3847 | 3594 |
|                   | 30           | 6847 | 5519 | 6888 | 5545 | 6871 | 5591 |
|                   | 4274         | 3894 | 4215 | 3841 | 4295 | 3814 |
|                   | 50           | 7481 | 5873 | 7469 | 5881 | 7431 | 5846 |
|                   | 4721         | 4431 | 4793 | 4595 | 4725 | 4481 |
|                   | 70           | 8096 | 6374 | 8137 | 6367 | 8081 | 6374 |
|                   | 5145         | 4861 | 5147 | 4864 | 5184 | 4835 |

Abbreviations: ACO, ant colony optimization; FF, firefly.

### Table 10
Time and path length variation for ACO and FF over a plain rubber grid of size 40 × 40

| No. of Ants/Fireflies Algorithm | 25 | 50 | 75 |
|-------------------------------|----|----|----|
| ACO              | FF          | ACO  | FF  | ACO  | FF  |
| Density (%) 10        | 11159        | 9211 | 11123 | 9515 | 11139 | 9596 |
|                   | 6717         | 6527 | 6793 | 6511 | 6889 | 6542 |
|                   | 30           | 11954 | 9814 | 11956 | 9853 | 11999 | 9834 |
|                   | 7124         | 6912 | 7174 | 6927 | 7274 | 6991 |
|                   | 50           | 12684 | 10314 | 12697 | 10354 | 12681 | 10349 |
|                   | 7567         | 7426 | 7558 | 7447 | 7468 | 7598 |
|                   | 70           | 13540 | 10974 | 13513 | 10961 | 12573 | 10977 |
|                   | 8085         | 7868 | 8031 | 7851 | 8036 | 7819 |

Abbreviations: ACO, ant colony optimization; FF, firefly.
## Table 1

| No. of Ants/Fireflies Algorithm | 25 | 50 | 75 |
|--------------------------------|----|----|----|
| Density (%)                   |    |    |    |
| 10                             | 18018 | 16345 | 18094 | 16345 | 18012 | 16341 |
|                               | 10749 | 10546 | 10771 | 10583 | 10897 | 10665 |
| 30                             | 18417 | 16842 | 18548 | 16825 | 18533 | 16851 |
|                               | 11248 | 10805 | 11355 | 10875 | 11357 | 10875 |
| 50                             | 19025 | 17348 | 18953 | 17391 | 18947 | 17334 |
|                               | 11648 | 11149 | 11623 | 11212 | 11615 | 11215 |
| 70                             | 19431 | 17945 | 19580 | 17983 | 19487 | 17334 |
|                               | 12074 | 11691 | 12081 | 11641 | 12041 | 11624 |

Abbreviations: ACO, ant colony optimization; FF, firefly.

## Table 2

| No. | Type                              | z-Value | P-Value     | Result          |
|-----|-----------------------------------|---------|-------------|-----------------|
| 3   | Time comparison for plain rubber grid | 7.74597 | <.00001     | Significant at $P < .05$ |
| 4   | Path comparison for plain rubber grid | 6.71317 | <.00001     | Significant at $P < .05$ |

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 heights variation within the grids, white for h1, blue for h2, and red for h3 as shown in Figure 5B,C.

The starting point is the left uppermost corner rubber plant and 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. The Hamiltonian tour of ACO and FF in sloped rubber terrain is given in Figure 6.

### 4.3 Effect of motion planning in dynamic sloped rubber plantation terrain

The following Tables 16 and 17 show the effect of motion planning in sloped terrain with respect to time and path length by comparing with the plain normal rubber grids.

### 4.4 Results and findings

1. In all cases of general grids, rubber plain field grids, and sloped terrain rubber field grids the path length and time of the optimal path increases with the increase in the obstacle density in both ACO and FF optimization algorithms. In the case of sloped rubber field terrain 20 x 20 in our study, when obstacle density increases from 50% to 70%, it has been observed that the corresponding path length and time increases very sharply for both ACO and FF (188% for ACO and 204% for FF for path length; 299% for ACO and 350% for FF for the time).
2. Number of agents above 50 has not shown substantial contribution in the optimization of path length and time of execution in general.
3. The path length and time increases proportionally with the increase in grid sizes.
4. In all the situations of different grid sizes, FF optimization algorithm outperforms ACO in terms of path length and time of execution for the optimum path.
5. Both path length and time is more in the sloped rubber terrain as compared with the normal plain rubber field one. The percentage change in path length is 14.6 for ACO and 16.9 for FF in the sloped rubber terrain of size 10 x 10. The average percentage increase in path length is 95.5 for ACO and 116.6 for FF in the grid size 20 x 20. The average...
The overall percentage change in path length by taking average of both grid size is 55.1 for ACO and 66.73 for FF and the overall percentage change in time for the optimal path is 76.4 for ACO and 82.1 for FF. The results are tabulated as follows in Table 18. The sign test results are summarized in Table 19.

5 | REGRESSION ANALYSIS OF ACO AND FF

In this article, the pathfinding problem is considered for a robot in a row- and column-wise uniformly cultivated farm. A typical case is a rubber plantation, a commercially important crop in coastal India that is facing growing scarcity of labor. The requirement of moving through rubber trees is mapped to moving through a rectilinear grid. The robot has to serve trees located at nodes in the grid, and it can cross an edge provided the edge is free of obstacles. This resembles the scenario in the ACO and fireflies algorithms, that have been developed for optimal path finding on a grid.

The work in this article is aimed at evaluating these algorithms in a typical environment that is characterized by (i) a constrained (or a nonconstrained) path, (ii) the dimensions of the workspace, (iii) the obstacle-density of the environment, and (iv) the number of robots available simultaneously to explore a path. An additional contextual constraint of the open farm is that the floor may not be plain; it may have hillocks or depressions that add convexity or concavity to the surface.

To meet these objectives, the algorithm that takes less computational time and finds the shorter path length is the better algorithm. This appears to be obvious, except for a possible trade-off between time complexity and shorter path length.
(A) ACO path in sloped terrain at 200 sec  
(B) ACO path in sloped terrain at 500 sec  
(C) ACO path in sloped terrain at 878 sec  
(D) FF path in sloped terrain at 200 sec  
(E) FF path in sloped terrain at 400 sec  
(F) FF path in sloped terrain at 814 sec

**FIGURE 6** (A–C) show the path position at the end of three time instances for ACO in a 10 × 10 rubber plantation field with sloped terrain. The same grid size, (D–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 are 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 Tables 13 and 14 show the average of the minimum times and that of the path-lengths as recorded in the simulation of the farms with 100 and 400 plants in the sloped terrain. The Table 15 gives the z-values and P-values for the comparison test. ACO, ant colony optimization; FF, firefly

The simulation has been worked out with varied values of density, number of robots, and farm-size, for a flat form and for a hilly farm, by assuming two types of service robots: (i) a courier robot (CR) that picks up desired material from a specific location (source) to dispatch it to another specific location (destination) and (ii) a latex collector robot (LCR) that starts at a tree in a given farm and completes its tour by collecting the latex from each of the trees in the farm. Depending on the situation, the LCR could be modified for any other activity like taking a picture of each tree or selected trees, watering plant-pots in a nursery where the pots are arranged in rows and columns, and so on. The CR is set for the shortest path and LCR is set to find an optimal Hamiltonian tour.

The simulation is run till we find some trait in each of the scenarios. Therefore, the number of observations in the later scenarios are few. However, each event is tested with both ACO and FF. The findings in each case have been discussed with the description of the case above. Here is an attempt to do some exploratory analysis by taking all the observations together.
### Table 13: Time and path length variation for ACO and FF over a sloped rubber terrain grid size 10 × 10

| No. of Ants/Fireflies Algorithm | 25     | 50     | 75     |
|---------------------------------|--------|--------|--------|
|                                 | ACO    | FF     | ACO    | FF     | ACO    | FF     |
| Density (%)                     |        |        |        |        |        |        |
| 10                              | 878    | 815    | 865    | 826    | 848    | 863    |
|                                 | 446    | 435    | 425    | 421    | 452    | 469    |
| 30                              | 1253   | 1064   | 1269   | 1065   | 1296   | 1141   |
|                                 | 465    | 468    | 496    | 485    | 472    | 485    |
| 50                              | 1864   | 1565   | 1847   | 1547   | 1945   | 1596   |
|                                 | 591    | 468    | 584    | 462    | 632    | 598    |
| 70                              | 2369   | 2254   | 2264   | 2354   | 2410   | 2217   |
|                                 | 574    | 483    | 582    | 631    | 614    | 643    |

Abbreviations: ACO, ant colony optimization; FF, firefly.

### Table 14: Time and path length variation for ACO and FF over a sloped rubber terrain grid size 20 × 20

| No. of Ants/Fireflies Algorithm | 25     | 50     | 75     |
|---------------------------------|--------|--------|--------|
|                                 | ACO    | FF     | ACO    | FF     | ACO    | FF     |
| Density (%)                     |        |        |        |        |        |        |
| 10                              | 3961   | 3715   | 3749   | 3765   | 4403   | 3801   |
|                                 | 1853   | 1916   | 1976   | 2017   | 1816   | 1935   |
| 30                              | 5644   | 4183   | 5826   | 4568   | 5907   | 4362   |
|                                 | 1947   | 2001   | 1928   | 2048   | 1970   | 2015   |
| 50                              | 7601   | 6033   | 7252   | 6044   | 8147   | 5833   |
|                                 | 1876   | 2061   | 2342   | 2160   | 2202   | 2120   |
| 70                              | 16240  | 14154  | 15371  | 14458  | 17172  | 15926  |
|                                 | 7060   | 7357   | 7647   | 8270   | 7082   | 7372   |

Abbreviations: ACO, ant colony optimization; FF, firefly.

### Table 15: The z-values and P-values for the comparison test

| No. | Type                                        | z-Value | P-Value | Result                |
|-----|---------------------------------------------|---------|---------|-----------------------|
| 5   | Time comparison for sloped rubber terrain  | 3.67423 | <.00001 | Significant at P < .05 |
| 6   | Path comparison for sloped rubber terrain   | 1.22474 | .11034  | Not significant at P < .05 |

### Table 16: Percentage increase in time and path length for obtaining optimal paths in sloped terrain of 100 plants

| No. of Ants/Fireflies Algorithm | 25     | 50     | 75     |
|---------------------------------|--------|--------|--------|
|                                 | ACO    | FF     | ACO    | FF     | ACO    | FF     |
| Density (%)                     |        |        |        |        |        |        |
| 10                              | 29.5   | 49     | 27.2   | 49.9   | 28.3   | 53.8   |
|                                 | 7.2    | 8      | 0.7    | 6.9    | 4.9    | 22.8   |
| 30                              | 12.7   | 8.1    | 7.2    | 9.6    | 9.5    | 15.5   |
|                                 | 11.8   | 10.1   | 15.9   | 16.3   | 9.8    | 13.3   |
| 50                              | 21.7   | 16.7   | 21.4   | 13.17  | 27.9   | 17.2   |
|                                 | 30.8   | 17.9   | 21.4   | 20.3   | 28.2   | 21.8   |
| 70                              | 9      | 18.9   | 3.7    | 23.57  | 10.4   | 18.3   |
|                                 | 8.9    | 15.9   | 13.2   | 21.6   | 21.8   | 21.4   |

Abbreviations: ACO, ant colony optimization; FF, firefly.
### Table 17 Percentage change in time and path length in the slope terrain of 400 plants

| No. of Ants/Fireflies Algorithm | 25  | 50  | 75  |
|---------------------------------|-----|-----|-----|
|                                 | ACO | FF  | ACO | FF  | ACO | FF  |
| Density (%)                     |     |     |     |     |     |     |
| 10                              | 44.9| 68.6| 35.6| 69.8| 57.5| 69.4|
|                                 | 15.7| 23  | 22.4| 29  | 13.4| 24.6|
| 30                              | 79.8| 62.5| 83.4| 76.5| 85.4| 67.9|
|                                 | 12.9| 26.3| 10.9| 28.2| 14.3| 28  |
| 50                              | 111.4| 102 | 100.5| 100.07| 125 | 94.1|
|                                 | 11.4| 28.7| 38.7| 32.1| 30.8| 31.1|
| 70                              | 303.2| 307.3| 277.2| 312.6| 320.7| 355.4|
|                                 | 314.1| 367.4| 346.4| 422.1| 315.4| 358.2|

Abbreviations: ACO, ant colony optimization; FF, firefly.

### Table 18 Overall percentage change in path and time in sloped rubber terrain

| Grid Size | Path | Time |
|-----------|------|------|
|           | ACO  | FF   | ACO  | FF   |
| 10 × 10   | 14.6 | 16.9 | 17.4 | 24.6 |
| 20 × 20   | 95.5 | 116.6| 135.4| 140.5|
| Overall average percentage | 55.1| 66.73| 76.4 | 82.1 |

Note: The units of path length is grid cells, 1 grid cell = 10 pixels, and time is in seconds. Abbreviations: ACO, ant colony optimization; FF, firefly.

### Table 19 Sign test results summary

| No. | z-Value | P-Value | Result          |
|-----|---------|---------|-----------------|
| 1   | 10.3692 | <.00001 | Significant at $P < .05$ |
| 2   | 7.49269 | <.00001 | Significant at $P < .05$ |
| 3   | 7.74597 | <.00001 | Significant at $P < .05$ |
| 4   | 6.71317 | <.00001 | Significant at $P < .05$ |
| 5   | 3.67423 | <.00001 | Significant at $P < .05$ |
| 6   | 1.22474 | .11034 | Not significant at $P < .05$ |

### Table 20 Gives the scenario-wise distribution of the observation

| No. | Description                        | Density (%) | No. of Agents | Farm-Size Unit | Total Observations |
|-----|------------------------------------|-------------|---------------|----------------|--------------------|
| 1   | Source to destination on plain land| 10, 30, 50, 70, and 90 | 25, 50, 75, 100, and 200 | 400, 1600, 3600, 6400, and 10000 | $5 \times 5 \times 5 = 125 $ $5 \times 2 = 250$ |
| 2   | Visit all nodes on plain land      | 10, 30, 50, and 70 | 25, 50, and 75 | 100, 400, 900, 1600, and 2500 | $4 \times 3 \times 5 = 60 $ $60 \times 2 = 120$ |
| 3   | Visit all nodes on land with hillocks | 10, 30, 50, and 70 | 25, 50, and 75 | 100 and 400 | $4 \times 3 \times 2 = 24 $ $24 \times 2 = 48$ |
| 4   | Visit all nodes on land with only convex hurdles | 10, 30, 50, and 70 | 25, 50, and 75 | 100 | $4 \times 3 \times 1 = 12 $ $12 \times 2 = 24$ |
| Total |                                   |             |               |                | 442                |
**TABLE 21** Regression for observed time and observed path-lengths

| All Attributes Model | $R^2$ | Density | #Agents | Farm Size | Tour Type | Algorithm |
|----------------------|-------|---------|---------|-----------|-----------|-----------|
| Time                 | 0.38  | 9.01    | −10.36  | 0.00      | 1898.83   | −249.52   |
| Path-length          | 0.31  | 4.76    | −5.56   | 0.01      | 1002.92   | 93.68     |

**TABLE 22** Coefficient of parameters in the regression equation for the tours of different types

| Algo_Tour-Type_Y | $R^2$ | Density | #Agents | Farm Size |
|------------------|-------|---------|---------|-----------|
| FF_H-Cave_Pathlength | 0.73126 | 30.08074 | −20.0951 | 7.808366 |
| ACO_H-Cave_Pathlength | 0.74653 | 29.05611 | −18.7483 | 7.433302 |
| FF_H-Cave_Time    | 0.81345 | 70.17186 | −38.5635 | 15.29118 |
| ACO_H-Cave_Time    | 0.84389 | 76.10051 | −43.588  | 18.25111 |
| ACO_P-Plain_Pathlength | 0.973724 | 4.76186 | 0.186519 | 0.017441 |
| ACO_P-Plain_Time   | 0.973222 | 0.962333 | 0.085506 | 0.026727 |
| FF_P-Plain_Pathlength | 0.984774 | 0.537287 | 0.102498 | 0.016427 |
| FF_P-Plain_Time    | 0.977356 | 0.890898 | 0.169373 | 0.023634 |
| FF_H-Plain_Pathlength | 0.997635 | 9.658753 | −4.63537 | 4.533262 |
| FF_H-Plain_Time    | 0.991253 | 40.48333 | 2.735    | 6.071333 |
| ACO_H-Vex_Pathlength | 0.989765 | 4.803333 | 1.025    | 6.087333 |
| ACO_H-Vex_Time      | 0.98923  | 16.01215 | −10.4253 | 6.527888 |
| ACO_H-Plain_Pathlength | 0.988308 | 23.23029 | −16.5466 | 7.341209 |
| ACO_H-Plain_Time     | 0.998249 | −0.40333 | 8.54E-16 | 7.753     |
| ACO_H-Vex_Time       | 0.998546 | 40.97167 | −1.57    | 10.13717 |

Abbreviations: ACO, ant colony optimization; FF, firefly.

We employ regression as a tool. Tables 20-25 are compilations of the regression output at various scenarios, followed by our interpretation.

First in Table 21, without making any assumption about any of the parameters that include the algorithm and the objective, we compute regression for observed time and observed path-lengths, which are the independent variables.

**R0:** The $R^2$ value is not so good. However, it is not merely insignificant.

**R1:** A comparison between the coefficients of other parameters tells us that the tour type (shortest path or Hamiltonian on different surfaces) and Algorithm are largely the factors that determine the model. The density and number of agents have some role to play. However, the farm size hardly matters.

**R2:** Recall from the last section that ACO and FF do not perform identically. The ACO and FF have been labeled as 0 and 1, respectively. Therefore, the positive coefficient in the path-length model says that the path generated by FF is longer than the path generated by ACO. The negative coefficient of the time model tells that FF will help a faster convergence in general.

Next, we run the regression separately on the sets of observations that share the tour type and algorithm. Intuitively, the shortest path on a plain land is the relatively simplest scenario, while the Hamiltonian on a concave surface is supposedly the most difficult. Therefore, we label them 1 and 4, respectively. The Hamiltonian with convex hurdles being the next consideration after the Hamiltonian on plain land, we labeled it 3 and the Hamiltonian on plain land being more difficult than the shortest path we labeled it 2. The mnemonics for 1, 2, 3, and 4 are P-Plain, H-Plain, H-Vex, and H-Cave. The results in the following Table 22 are presented in ascending order of $R^2$ square.

**R3:** There is a significant improvement in $R^2$ square; the model is successful in describing $Y$ almost 100% in all but the concave tours. The low $R^2$ square in concave tours is due to the fact that the time and path-length in the concave tour is dependent of the concavity that could be crossed or bypassed.

**R4:** The Hamiltonian tours on plain and concave surface have shown considerable impact by the number of agents; the coefficients in the case of concave tours are significantly larger than those of the plain tours. This implies that a small
rise in the number of agents in the case of concave tours will drastically improve the performance of the system. The impact of agents in the other cases is significantly low. However, except in the case of modeling time, the coefficients being positive, a rise in the number of agents will cause a rise in time and path-length. It may be that path finding on a plain or a convex surface leaves no scope for parallelism.

**R5:** In all but the case of modeling path length in convex tours that employ FF, the coefficient of density is invariably positive, and in the time models it is double of the path-length model. The observations about the convex tours are inadequate to establish any relationship. It may be that there does not exist one.

**R6:** The coefficient of farmsize is invariably positive and in the time models it is 1.7 times the path-length model. It can be concluded that for a given tour type and algorithm the impact of farm size is almost the same as that of density. The larger the farm size, the more is the time taken for completing the task and the longer is the path.

Furthermore, we explore more to study the impact of multiple agents at a specific tour with a given density.

**R7:** Invariably the coefficient of #Agents at a given density for modeling the time taken by FF is the highest and that of the path length generated by FF is the lowest. The increased number of agents adversely affects the performance. The impact is higher on the time models and is guaranteed on environments with the density 70% and higher. Between the two, it is more for ACO when it comes to path-length and for FF while modeling the time characteristic. Table 24 shows events where performance of ACO is more affected than that of FF when the number of agents are increased.

Table 25 shows the events where performance of FF is more affected than that of AC when the number of agents are increased.

**R8:** About 85% entries in the Table 20 are for modeling path length and the same percentage of entries in the Table 22 is for time modeling. Based on these observations one may recommend FF if path-length is of more concern and ACO otherwise. Group I and II comprise of the events that show positive and negative impacts, respectively, on the system’s performance on increasing the number of agents. The entries in the Group III of Table 24 indicate an increase in the performance when FF is employed and that of Table 25 indicate the same effect when ACO is employed.
**TABLE 24** Group-wise regression results with varying number of agents when ACO is more affected than FF

| Algo_Tour-Type_Density#%_Y | R Square | #Agents | Farm Size |
|---------------------------|----------|---------|-----------|
| Group I                   |          |         |           |
| FF_H_Cave_Density10_Pathlength | 0.99913  | -0.71   | 4.96      |
| ACO_H_Cave_Density10_Pathlength | 0.99870  | -0.60   | 4.78      |
| FF_H_Cave_Density10_Time | 0.99939   | -1.21   | 9.52      |
| ACO_H_Cave_Density10_Time | 0.99457   | -0.79   | 10.13     |
| FF_H_Cave_Density30_Pathlength | 0.99979  | -0.37   | 5.09      |
| ACO_H_Cave_Density30_Pathlength | 0.99986  | -0.04   | 4.87      |
| FF_H_Cave_Density70_Pathlength | 0.98468  | -21.64  | 21.55     |
| ACO_H_Cave_Density70_Pathlength | 0.98794  | -19.82  | 20.35     |
| FF_H_Plain_Density10_Pathlength | 0.99961  | -2.09   | 4.24      |
| ACO_H_Plain_Density10_Pathlength | 0.99986  | -0.58   | 4.30      |
| FF_H_Vex_Density30_Pathlength | 0.99976  | -1.40   | 8.63      |
| ACO_H_Vex_Density30_Pathlength | 0.99443  | -0.04   | 7.58      |
| Group II                  |          |         |           |
| FF_H_Cave_Density50_Pathlength | 0.99934  | 1.40    | 5.10      |
| ACO_H_Cave_Density50_Pathlength | 0.99415  | 2.52    | 5.02      |
| FF_H_Cave_Density50_Time | 0.99962   | 0.60    | 14.88     |
| ACO_H_Cave_Density50_Time | 0.99791   | 1.91    | 18.87     |
| FF_H_Plain_Density70_Pathlength | 0.99755  | 1.31    | 4.75      |
| ACO_H_Plain_Density70_Pathlength | 0.99722  | 2.13    | 4.88      |
| FF_H_Vex_Density70_Pathlength | 0.99973  | 1.24    | 6.85      |
| ACO_H_Vex_Density70_Pathlength | 0.99976  | 2.70    | 7.96      |
| Group III                 |          |         |           |
| FF_H_Plain_Density30_Pathlength | 0.99984  | -1.27   | 4.38      |
| ACO_H_Plain_Density30_Pathlength | 0.99974  | 0.33    | 4.52      |
| FF_H_Vex_Density50_Pathlength | 0.99996  | -0.78   | 7.79      |
| ACO_H_Vex_Density50_Pathlength | 0.99929  | 2.52    | 8.96      |

Abbreviations: ACO, ant colony optimization; FF, firefly.

**R9:** The distribution of entries in Group II indicates that in most of the events an increase in the number of agents in a high density environment will result in the reduced performance.

### 6 CONCLUSION

In this article, we attempted a comparison of ACO and FF, the two widely employed soft computing algorithms in general, for a specific purpose of path finding in a dynamic environment where the locations to be visited are arranged at uniform distances row and column-wise. A specific case considered is Rubber plantation. The results are equally applicable in any other environment that follows a rectangular grid topology. Findings and contributions of the research are as follows:

- In all cases of general grids, rubber plain field grids and sloped terrain rubber field grids the path length and time of the optimal path increases with the increase in the obstacle density in both ACO and FF optimization algorithms.
- The number of agents above 50 has not shown substantial contribution in the optimization of path length and time of execution in general.
### Table 25

Results when FF is more affected than AC with increasing number of agents

| Algorithm | Type | Density | % | R Square | #Agents | Farm Size |
|-----------|------|---------|---|----------|---------|-----------|
| Group I   |      |         |   |          |         |           |
| ACO_H_Cave_Density70_Time | 0.99245 | -24.35 | 43.22 |
| FF_H_Cave_Density70_Time   | 0.99144 | -16.80 | 38.74 |
| ACO_H_Plain_Density10_Time | 0.98503 | -13.00 | 7.23 |
| FF_H_Plain_Density10_Time  | 0.98840 | -9.51  | 6.45 |
| ACO_H_Plain_Density30_Time | 0.98758 | -6.56  | 7.43 |
| FF_H_Plain_Density30_Time  | 0.98973 | -4.42  | 6.57 |
| Group II  |      |         |   |          |         |           |
| ACO_H_Plain_Density50_Pathlength | 0.99856 | 1.06 | 4.68 |
| FF_H_Plain_Density50_Pathlength | 0.99853 | 1.29 | 4.54 |
| ACO_H_Plain_Density70_Time | 0.98656 | 5.63 | 7.69 |
| FF_H_Plain_Density70_Time   | 0.99106 | 7.41 | 6.83 |
| ACO_H_Vex_Density10_Time | 0.99970 | 0.50 | 13.75 |
| FF_H_Vex_Density10_Time | 0.99986 | 2.36 | 12.03 |
| ACO_H_Vex_Density70_Time | 0.99999 | 0.74 | 37.58 |
| FF_H_Vex_Density70_Time | 0.99903 | 3.50 | 36.51 |
| Group III |      |         |   |          |         |           |
| ACO_H_Cave_Density30_Time | 0.99900 | -1.69 | 14.63 |
| FF_H_Cave_Density30_Time | 0.99881 | 0.94 | 10.79 |
| ACO_H_Plain_Density50_Time | 0.98830 | -0.71 | 7.58 |
| FF_H_Plain_Density50_Time | 0.99073 | 0.50 | 6.7 |
| ACO_H_Vex_Density10_Pathlength | 0.99878 | -1.08 | 7.54 |
| FF_H_Vex_Density10_Pathlength | 0.99943 | 0.94 | 7.1 |
| ACO_H_Vex_Density30_Time | 0.99864 | -3.62 | 22.28 |
| FF_H_Vex_Density30_Time | 0.99949 | 1.86 | 17.69 |
| ACO_H_Vex_Density50_Time | 0.99923 | -3.90 | 32.49 |
| FF_H_Vex_Density50_Time | 0.99923 | 3.22 | 22.84 |

Abbreviations: ACO, ant colony optimization; FF, firefly.

- The path length and time increase proportionally with the increase in the grid sizes.
- In all but the case of terrain, with the 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 is 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 the motion planning in different dynamic scenarios. The shortest path on a plain land is the relatively simplest scenario, while the Hamiltonian on a concave surface is supposedly the most difficult.
- Our proposal of RHR carries novelties in the agricultural domain and gives innovation in the area of automation for the latex collections in the rubber plantations.
- Based on the 10 regression results above we recommend the following:
  - The results of our exhaustive application of regression are consistent; it revealed the power of regression as a tool for exploratory research when you have sufficient results out of a not so well-planned experiment.

We look forward for the real life implementation of the RHR proposed here. Research and simulation experiments could be carried out with the continuous sloped rubber terrain fields with higher complexities.
CONFLICT OF INTEREST
The authors declare no potential conflict of interest.

AUTHOR CONTRIBUTIONS
Mohanan G Conceptualization-Lead, Data curation-Lead, Formal analysis-Lead, Funding acquisition-Lead, Investigation-Lead, Methodology-Lead, Project administration-Lead, Resources-Lead, Software-Lead, Supervision-Lead, Validation-Lead, Visualization-Lead, Writing-original draft-Lead, Writing-review & editing-Lead; Ambuja Salgaonkar Conceptualization-Supporting, Data curation-Supporting, Formal analysis-Supporting, Investigation-Supporting, Methodology-Supporting, Project administration-Supporting, Resources-Supporting, Software-Supporting, Supervision-Supporting, Validation-Supporting, Visualization-Supporting, Writing-original draft-Supporting, Writing-review & editing-Supporting.

ORCID
Mohanan M. Gangadharan https://orcid.org/0000-0002-0779-4419

REFERENCES
1. Yang X. Firefly algorithm stochastic test functions and design optimisation. Int J Bio-Inspired Comput. 2010;2(2):78-84.
2. Dorigo M, Maniezzo V, Colorni A. The ant system: optimisation by a colony of cooperating agents. IEEE Trans Syst Man Cybern B. 1996;26(1):1-13.
3. Brand M, Masuda M, Wehner N, Yu X-H. Ant colony optimization algorithm for robot path planning. Paper presented at: International Conference on Computer Design And Applications (ICCCDA 2010), Qinhuangdao,China, Vol. 3; June 2010;V3-436-V3-440.
4. Ming-bo W, Qiang F, Nan T, Mengmeng L, Yiming Z. An improved firefly algorithm for traveling salesman problems. Paper presented at: 4th National Conference on Electrical Electronics and Computer Engineering (NCEECE 2015), December 12-13, 2015; Xi’an.
5. Zadeh SM. Autonomous Reactive Mission Scheduling and Task-Path Planning Architecture for Autonomous Underwater Vehicle [PhD thesis]. Flinders University of South Australia; Chapter 2; 2016:19-46.
6. Pandey A. Mobile Robot Navigation in Static and Dynamic Environments Using Various Soft Computing Techniques [PhD thesis]. Rourkela: National Institute of Technology; 2016.
7. Baras JS, Mehta H. A probabilistic emergent routing algorithm for mobile ad hoc networks. Paper presented at: WiOpt’03: Modeling and Optimization in Mobile Ad Hoc and Wireless Networks; March, 2003;10; Sophia Antipolis, France.
8. Xue Y, Sun J. Solving the path planning problem in mobile robotics with the multi-objective evolutionary algorithm. Appl Sci. 2018;8(9):1425. https://doi.org/10.3390/app8091425.
9. Dorigo M, Stiitzle T. Ant Colony Optimization. Cambridge, England: MIT Press; 2004.
10. Yang XS. Nature-Inspired Metaheuristic Algorithms. UK: Luniver Press; 2008.
11. Ulbrich H, Baur J, Pfaff J, Schuetz C. Design and realization of a redundant modular multipurpose agricultural robot. Paper presented at: DINAME 2015 - Proceedings of the XVII International Symposium on Dynamic Problems of Mechanics; February 22–27, 2015; Brazil.
12. Baur J, Schutz C, Pfaff J, Buschmann T, Ulbrich H. Path planning for a fruit picking manipulator. Paper presented at: Teschnische Universität München Institute of Applied Mechanics Germany, International Conference of Agricultural Engineering; 2014; Zurich.
13. Correll N, Arechiga N, Bolger A, et al. Building a distributed robot garden. Paper presented at: IEEE/RSJ International Conference on Intelligent Robots and Systems; October 11–15, 2009;1509-1516; St. Louis, MO.
14. Li W, Li D, Ni J. Diagnosis of tapping process using spindle motor current. Int J Mach Tools Manuf. 2003;43:73-79.
15. Zhang B, Yang F, Wang J. Fundamental aspects in vibration assisted tapping. J Mater Process Tech. 2003;132:345-352.
16. Thanpattranon P, Ahamed T, Takigawa T. Navigation of an autonomous tractor for a row-type tree plantation using a laser range finder – development of a point –to-go algorithm. Robotics. 2015;4:341-364. https://doi.org/10.3390/robots4030341.
17. Luo L, Wen H, Lu Q, et al. Collision-free path-planning for six-DOF serial harvesting robot based on energy optimal and artificial potential field. Complexity. 2018;2018:3563846. https://doi.org/10.1155/2018/3563846.
18. Ringdahl O. Automation in Forestry – Development of Unmanned Forwarders [PhD thesis]. Sweden: Department of Computing Science, Umea University; 2011.
19. Hellstrom T, Ringdahl O. Follow the past: a path-tracking algorithm for autonomous vehicles, Department of Computing Science, Umea University S-901 87, Umea Sweden. Int. J. Vehicle Auton Syst. 2006;4:2-4.
20. Mohanan MG, Salgoankar A. A survey of robotic motion planning in dynamic environments. Robot Auton Syst. 2018;100:171-185.
21. Dorigo M, Gambardella L.M. Ant colony system: a cooperative learning approach to the travelling salesman problem. IEEE Trans Evolut Comput. 1997;1(1):53-66.
22. Akka K, Khaber F. Mobile robot path planning using an improved ant colony optimization. Int J Adv Robot Syst. 2018;26(1):1-7.
23. Zeng MR, Xi L, Xiao AM. The free step length ant colony algorithm in mobile robot path planning. Adv Robot. 2016;30:1-6.
24. Cao J. Robot global path planning based on an improved ant colony algorithm. J Comput Commun. 2016;4:11-19. https://doi.org/10.4236/jcc.2016.42002.
25. Purian FK, Farokhi F, Nadooshan RS. Comparing the performance of genetic algorithm and ant colony optimization algorithm for mobile robot path planning in the dynamic environments with different complexities. *J Acad Appl Stud*. 2013;3(2):29-44.
26. Buniyamin N, Sariff N, Ngah WW, Mohamad Z. Robot global path planning overview and a variation of ant colony system algorithm. *Int J Comput Math Simul*. 2009;5:151-159.
27. Guan-Zheng T, Huan H, Aaron S. Ant colony system algorithm for real-time globally optimal path planning of mobile robots. *Acta Autom Sin*. 2007;33(3):279-285.
28. Mei H, Tian Y, Zu L. A hybrid ant colony optimization algorithm or path planning of robot in dynamic environment. *Int J Inf Technol*. 2006;12(3):78-88.
29. Kumar PB, Sahu C, Parhi DR. A hybridized regression-adaptive ant colony optimization approach for navigation of humanoids in a cluttered environment. *Appl Soft Comput*. 2018;68:565-585.
30. Liu J, Yang J, Liu H, Tian X, Gao M. An improved ant colony algorithm for robot path planning. *Soft Comput*. 2017;21(19):5829-5839.
31. Chih-Ta Y, Ming-Feng C. A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance. *Microsyst Technol*. 2018;24(1):125-135.
32. Brand M, Yu X. Autonomous robot path optimization using firefly algorithm. Paper presented at: Proceedings of the 2013 International Conference on Machine Learning and Cybernetics; July 14-17, 2013; Tianjin.
33. Hidalgo-Paniagua A, Vega-Rodríguez A, Ferruz J, Pavon N. Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach. *Soft Comput*. 2017;21:949-964. https://doi.org/10.1007/s00500-015-1825-z.
34. Patle BK, Pandey A, Jagadeesh A, Parhi DR. Path planning in uncertain environment by using firefly algorithm. *Def Technol*. 2018;14:691-701.
35. X.S. Yang. Firefly algorithms for multimodal optimization. Paper presented at: Stochastic Algorithms: Foundations and Applications, SAGA 2009 Lecture Notes in Computer Sciences, Sapporo, Japan; Vol. 5792; 2009;169-178.
36. Sadhu AK, Konar A, Bhattacharjee T, Das S. Synergism of firefly algorithm and Q-learning for robot arm path planning. *Swarm Evol Comput*. 2018;43:50-68. https://doi.org/10.1016/j.swevo.2018.03.014.
37. Tighzert L, Fonlupt C, Mendil B. A set of new compact firefly algorithms. *Swarm Evolution Comput*. 2018;40:92-115. https://doi.org/10.1016/j.swevo.2017.12.006.
38. Liu C, Zhao Y, Gao F, Liu L. Three-dimensional path planning method for autonomous underwater vehicle based on modified firefly algorithm. *Math Probl Eng*. 2015;2015:1-10. https://doi.org/10.1155/2015/561394.
39. Tiwari AK, Guha A, Pandey A. Dynamic motion planning for autonomous wheeled robot using minimum fuzzy rule based controller with avoidance of moving obstacles. *Int J Innovative Technol Explor Eng*. 2019;8(9):1425. https://doi.org/10.3390/app8091425.
40. Patle BK, Pandey A, Parhi DR, Jagadeesh A. A review: on path planning strategies for navigation of mobile robot. *Def Technol*. 2019;15:582-606.
41. Ning J, Zhang C, Sun P, Feng Y. Comparative study of ant colony algorithms for multi-objective optimization. *Information*. 2019;10(1):11.
42. Mei H, Tian Y, Zu L. A hybrid ant colony optimization algorithm for path planning of robots in dynamic environment. *Int J Inf Technol*. 2006;12(3):78-88.
43. Brendan E Hover JFS. Multi-goal feasible path planning using ant colony optimization. Paper presented at: 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China; 2011;2255-2260.
44. Garcia MAP, Montiel O, Castillo O, Sepúlveda R, Melin P. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl Soft Comput*. 2009;9(3):1102-1110. https://doi.org/10.1016/j.asoc.2009.02.014.
45. Ming-bo W, Qiang F, Nan T, Mengmeng L, Yiming Z. An improved firefly algorithm for traveling salesman problems. Paper presented at: 4th National Conference on Electrical Electronics and Computer Engineering (NCEECE 2015), Xi’an, China; 2015.

**AUTHOR BIOGRAPHIES**

**M. G. Mohanan** Associate Professor in Vivek College, Department of Mathematics and Computer Science, University of Mumbai. Received M.Sc. (Applied Mathematics) from University of Mumbai, Recipient of UGC National Scholarship for doing M.Sc, M. Phil in Graph Theory from Tata Institute of Fundamental Research (TIFR), Mumbai, DCM from NMIMS, Mumbai and DHE from Mumbai University. At present doing PhD in the Department of Computer Science, University of Mumbai. His major research interests are in Robotic Motion planning in Dynamic Environments, Agriculture Applications of Motion Planning, Soft computing, and Graph Theory.
Dr. Ambuja Salgaonkar Head, Department of Computer Science, University of Mumbai, has 24 years of Experience in Teaching, Research and Administration at University level. Received MCA and Ph.D. in Computer Science from Shivaji University, Kolhapur and MBA From Indira Gandhi National Open University, New Delhi. Her research interests include Computational modeling, Optimization, Artificial Intelligence, and Software Engineering. She has worked in diverse fields of Computing, automatic theorem provers, automatic question generation, manuscriptology, music, poetry, and artificial text creation to name a few. Contemporary applications of heritage science is her specialty area.

How to cite this article: Gangadharan MM, Salgaonkar A. Ant colony optimization and firefly algorithms for robotic motion planning in dynamic environments. Engineering Reports. 2020;2:e12132.
https://doi.org/10.1002/eng2.12132