Path Planning of Mobile Robot With Improved Ant Colony Algorithm and MDP to Produce Smooth Trajectory in Grid-Based Environment

Hub Ali, Dawei Gong*, Meng Wang and Xiaolin Dai

School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China

This approach has been derived mainly to improve quality and efficiency of global path planning for a mobile robot with unknown static obstacle avoidance features in grid-based environment. The quality of the global path in terms of smoothness, path consistency and safety can affect the autonomous behavior of a robot. In this paper, the efficiency of Ant Colony Optimization (ACO) algorithm has improved with additional assistance of A* Multi-Directional algorithm. In the first part, A* Multi-directional algorithm starts to search in map and stores the best nodes area between start and destination with optimal heuristic value and that area of nodes has been chosen for path search by ACO to avoid blind search at initial iterations. The path obtained in grid-based environment consist of points in Cartesian coordinates connected through line segments with sharp bends. Therefore, Markov Decision Process (MDP) trajectory evaluation model is introduced with a novel reward policy to filter and reduce the sharpness in global path generated in grid environment. With arc-length parameterization, a curvilinear smooth route has been generated among filtered waypoints and produces consistency and smoothness in the global path. To achieve a comfort drive and safety for robot, lateral and longitudinal control has been utilized to form a set of optimal trajectories along the reference route, as well as, minimizing total cost. The total cost includes curvature, lateral and longitudinal coordinates constraints. Additionally, for collision detection, at every step the set of optimal local trajectories have been checked for any unexpected obstacle. The results have been verified through simulations in MATLAB compared with previous global path planning algorithms to differentiate the efficiency and quality of derived approach in different constraint environments.

Keywords: mobile robot, ant colony algorithm, Markov decision process model, motion planning, obstacle avoidance

INTRODUCTION

For decades, the concept of autonomous driving has become familiar with the public due to its vast applications in multi-dimensional aspects (Gu et al., 2017). Therefore, ambitious research has taken place to improve the autonomous behavior of mobile robots. Mobile robots have applications in different domains such as industry, military, transportation, etc. (Duchoe et al., 2014). In

industries, autonomous mobile robots are used in logistics warehousing, flexible manufacturing, and intelligent inspection are much more efficient than humans and are available at a relatively lower cost. In mobile robot autonomous behavior is achieved through perception, decision, and actions (Sarkar et al., 2018). Perception contains sensor data from the environment, which has been actuated to low-level control according to decision ability. The decision stage is known as the motion-planning stage. Many approaches have been introduced to improve the motion-planning ability of mobile robots. Motion planning is a middle stage between perception and actuation and is responsible for the autonomous behavior of a robot. Motion planning is mainly categorized into Global Path Planning (GPP) and Local Path Planning (LPP) according to problem estimation.

GPP is a key technique for robots to acquire an optimal route from the initial position to the final destination. Several different approaches have introduced to acquire an optimal global path. Most GPP techniques use discrete search optimization and has applied in the grid-based environment. The path obtained in the grid map consists of suboptimal points connected though straight line segments and, the path contains sharp bends. The bends present in the global path can cause a jerky behavior in a robot's motion and, instant change in velocity and acceleration also effect on energy consumption of robots. In contrast, a smooth route can deliver a mobile robot a comfortable and safe drive. Therefore, the curvilinear global route has considered a suitable choice for better navigation and used in real-time scenarios. For example, In Defense Advanced Research Projects Agency (DARPA) autonomous challenge (2012), a vehicle used the pre-defined curvilinear route (Chu et al., 2012). The global path planning techniques including simulated annealing algorithm (Miao and Tian, 2013), potential function theory (Cetin and Yilmaz, 2014) has considered as traditional approaches due to their limited functionalities. The genetic algorithm has applied to get the global path in Bakdi et al. (2017), but it shows poor computation ability in complex maps. On the other hand, Ant Colony Optimization (ACO) (Wang et al., 2016), genetic algorithm (Huang and Fei, 2018), neural network (He et al., 2015, 2017), and particle swarm algorithm (Song et al., 2017) have considered as intelligent approaches. ACO has considered as one of the popular evolutionary approaches to solve optimization problems. Due to its advantages, such as good feedback information, robustness, better-distributed computing (Akka and Khaber, 2018) and, the ability to be easily combined with many path-planning approaches, it is often used to deliver an optimal solution of global path planning issues.

Similarly, LPP has applied to localize the robot's motion along a global path to avoid unknown obstacles. The predefined global waypoints have given to the robot in various LPP techniques and, it has to maintain an offset position on the centerline to avoid dynamic obstacles. Meanwhile, it plans smooth trajectories to move forward toward destination. DAPRA international auto-driving challenge has organized to present many state of the art technologies to achieve autonomous smooth driving features. Sampling-based approach as RRT* has been used in

path planning (Hwan Jeon et al., 2013). the algorithm used in Pivtoraiko and Kelly (2005) is a discrete representation of the planning area with a grid of states (usually a hyper-dimensional one). Reinforcement learning has been applied in a hierarchical path planning approach to achieve local planning and navigation (Zuo et al., 2015). Different approaches have advantages as well as drawbacks in different complex situations. Such as few mentioned techniques provide local path planning features and, few can only work in global path planning.

CONTRIBUTION

The motivation comes from the research work presented in state-of-art literature to introduce a combined approach for enhancing the path planning abilities of a mobile robot in a known and unknown statics constraint environment. The global trajectory obtained through pre-defined waypoints in local path planning algorithms like (Walambe et al., 2016; Hu et al., 2018). This approach delivers global and local trajectory planning features. For global path planning an improved ACO version has presented that deliver an optimal trajectory with efficient computational ability. The ACO has enhanced with A* multi-directional algorithm. In ACO, starting iterations don't have pheromone concentration and ants have to move randomly (blind search) to reach a goal position. This makes it computationally expensive and time-consuming. Hence, the ant colony algorithm needs to improve its computational efficiency. Therefore, A* Multi-directional algorithm is introduced to sort the area of nodes having a high possibility of obtaining an optimal global path. ACO utilized that information of A* multi-directional algorithm as closest area for search and allow ants to move in that directions. It increases the efficiency of ACO to deliver global path in complex maps efficiently. The quality of path has improved using MPD trajectory evaluation model. ACO gives a sequence of optimal grids and each grid is represented by its center point and the global path consists of straight-line segments containing sharp bends formed through connecting grid points in sequential order. Trajectory evaluation model filter the path points and arc-length parametrization bring path consistency and smoothness in final trajectory. To attain optimal driving behavior, a set of lateral and longitudinal coordinates has been generated. Lateral coordinates refer to offset distance d from reference points to navigate, whereas longitudinal coordinates represent distance covered at each waypoint in sense of cumulative arc length s, respectively. Both lateral and longitudinal coordinates are unified to produce a feasible set of trajectories along the reference path. Each forward move of the robot on the reference path will generate a set of trajectories to avoid unexpected obstacles. Every trajectory has to checked with defined cost constraints that include curvature limit, speed limit, and obstacle check.

RELATED WORK

In recent years, tremendous research efforts have made for improving the path planning efficiency of mobile robots in

both global and local path planning techniques. For global path planning, ACO has chosen in this paper because of its advantages. Although, ACO has drawbacks of slow convergence and pheromone update. To address this problem, many approaches have been proposed (Stützle and Hoos, 2000; Zeng et al., 2016; Zhao et al., 2016). Pheromone rate has been updated after each successful iteration of ant in ACO to improve the convergence rate (Zhao et al., 2016). In Zeng et al. (2016) convergence rate with search ability has been increased through upgrading pheromone update formula and adaptively varying volatilization rate. To avoid blind search in Stützle and Hoos (2000) an initial path has produced and transformed into initial pheromone distribution in ACO. Geometric method has been introduced to optimize the global route and also local diffusion of pheromones has obtained from a force factor defined in artificial potential field to enhance the ability of obstacle detection (Liu et al., 2017). In Yen and Cheng (2018), fuzzy logic is combined with ACO to reduce repetitive learning errors. In Long et al. (2019), A* Heuristic characteristics improved ACO optimization performance in various complexity maps. In the discrete-search algorithm, linear interpolation has been performed to bring smoothness in the global path (Ferguson and Stentz, 2006). In Zuo et al. (2015), the MDP model has been used with the A* algorithm to achieve a smooth path and improve navigation. These approaches can improve the efficiency of ACO. However, the quality of the path obtained in the grid environment doesn't match with the dynamic properties of a mobile robot due to its roughness and sharp bends. In the local path search, a curvilinear road is formed with cubic spline interpolation and a set of feasible trajectories has been generated along the roadside to avoid static obstacles (Chu et al., 2012). In Hu et al. (2018), curvilinear road shape has been produced from predefined waypoints and model predictive approach is used to avoid static and moving obstacles. Similarly, lateral and longitudinal movements have been introduced within the steer relative coordinates to achieve optimal motion control (Werling et al., 2012). The global reference path is derived from the vision map using the lane-level accurate localization information via the LiDAR-based localization methods (Hwang et al., 2003; Li et al., 2017). In Li et al. (2015), conjugate gradient non-linear optimization and cubic spline curve are used to achieve a smooth global path from digital map and curvilinear coordinates framework is used to obtain optimal trajectories. Likewise, these techniques can efficiently handle local path problem but do not work well to achieve optimal global path in complex constraints environment.

REFERENCE PATH GENERATION IN STATIC CONSTRAINTS ENVIRONMENT

Global Path Planning is used to obtain a global route from an initial position of the robot to destinations in a static constraint environment. GPP is an essential technique for a mobile robot to find a suitable path in various situations. It is simulated in a grid-based environment with prior knowledge of static constraints. In this section, the grid model has

presented and the serial number method has used to reduce complexity. Multi-Directional A* algorithm has utilized to get the initial searching area, that assist the ACO to get final optimal path. A constraints policy is presented for initial search of ants in ACO and evaluation function of A* algorithm is utilized to accelerate convergence speed in search strategy of ant colony algorithm with better heuristic information. Global search ability has enhanced by MAX_MIN ant system through updating path pheromone information. Moreover, the Trajectory evaluation model has introduced to increase the visibility of the ACO path and filter the corner points through MDP.

Environment Model

This path planning approach is implemented in a grid-based environment. The grid method is simple and effective to create and maintain the grid model. Moreover, the grid method has strong adaptability for obstacles and also convenience for computer storage and processing. It divides the working space into $N \times N$ squares. As shown in **Figure 1A**, white grids are spaces where robot can move freely, in contrast black grids represent constraints area. In order to identify constraint areas, the white grid cell is represented with 0, and black grid unit is represented with 1.

The grid model is consisting of a two-dimensional coordinate system. Each grid is marked according to the serial number method. In $N \times N$ grid map, the starting node is named after Start and the target node is named after Destination. The position coordinates (x, y) corresponding to any grid whose grid number is R is as follow:

$$\begin{cases} x = \begin{cases} mod\,(R, N) - 0.5 & if\ mod\,(R, N)! = 0 \\ N + mod\,(R, N) - 0.5 & otherwise \end{cases} \\ y = N + 0.5 - ceil(\frac{R}{N}) \end{cases} \quad (1)$$

The serial number method is applied to reduce the computational complexity of ACO. The direction of each move from the grid's center point to neighbor grids has been simplified with arithmetic operation. In **Figure 1C**, α represents the number of a central grid point and N is the number of rows and columns of the grid map. n represents all possible the direction.

A* Multi-Directional Algorithm (A*MDA)

In ACO, ants start to search in the map and after every successful iteration, a new pheromone has updated in the network that represents a specific direction toward a goal. After a few iterations, the ants start to converge their directions according to a higher pheromone ratio. At starting iterations, the ants do not have proper guidance through the pheromone ratio. Therefore, they move in different directions to seek a goal node and if the map has a large and complex search space then it consumes more time. A* algorithm has been introduced with multi-direction path search features to assist the ACO. **Figure 1B** has shown possible connecting nodes with the center node. Each node has to be estimated with total cost value (n), which is the sum of $g\,(n)$ and $h(n)$. It defines two matrices to list and mark
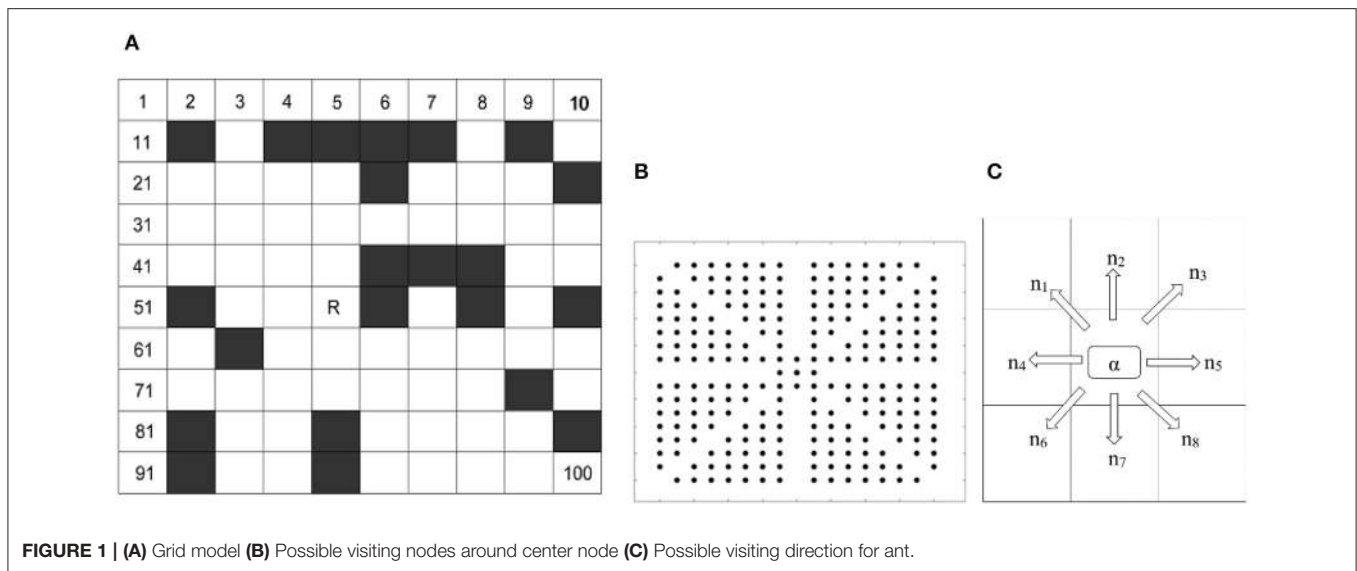
FIGURE 1 | (A) Grid model **(B)** Possible visiting nodes around center node **(C)** Possible visiting direction for ant.

visited node named as Open_ list and Closed_list. Open_list is marked with visited nodes and Closed_list contains the record of obstacles and previously selected nodes to avoid repetition. The next node selection is based on the lowest cost value of $f(n)$. It continues to search, until the destination node arrives. To retrieve the best nodes, the location of each parent node is stored into ParentX and ParentY. This algorithm provides multi-goals on short distances between start and actual destination. Meanwhile, the best nodes selected by A* Multi-Directional algorithm are not enough to complete a global path individually. Therefore, ACO has been utilized to fulfill the missing nodes and complete a global path. **Figure 2A** shows that the multi-goals generated by A*MDA and **Figure 2B** shows the final trajectory. **Table 1** provides algorithm flow:

$$f(n) = g(n) + h(n) \qquad (2)$$

Improved ACO With Initial Constraints Policy

ACO has been applied to achieve optimal global path from an initial position to destination in a grid-based environment. A* Multi-directional algorithm provides the direction guideline to ACO. The ants in traditional ACO have to search all possible grids and every next grid is decided by the roulette wheel method and repeated until the target point is obtained. It was computationally expensive. According to the grid-based environment obstacle grid represented with 1 and free grid represent with 0. In absence of any constraints, Ant α can move in 8 directions from the center grid as shown in **Figure 1C**. In the situation of obstacle detection, the remaining direction grids can be chosen using heuristic information. Initial constraints policy defines a cost for eliminating a specific direction grid, which does not maintain offset distance with obstacle grids. Improved A* heuristic characteristics are used to improve efficiency. To improve premature convergence and phenomenon

update strategy MAX_MIN, the ant system has been utilized. The improvements are given as below:

Initial Constraint Policy

Initial constraints policy is used in case of obstacle detection. When ant α found an obstacle, it has to turn toward remaining directions. Initial constraint policy limits the ant to consider those grids which do not maintain offset distance with obstacle grid. With regards to serial number system, an ant α has 8 neighbor grids directions presented in **Table 2**. Constraints policy is formed based on obstacle grid location and neighbor grids. In **Table 3**, on every obstacle direction constraint policy eliminates following grids to maintain offset distance in the final global path.

Heuristic Information and Path Strategy Information

In ACO, poor pheromone distribution at initial steps of ant search causes slow convergence and increase search time. The direction information and path strategy of ACO has been improved with A* algorithm characteristics to avoid blind search at initial steps of ant (Duchoe et al., 2014). The estimated function of A* has been used to enhance directional information expressed as Equation (2). $g(n)$ represents distance from source grid to current grid and $h(n)$ represents distance from current grid to destination grid, respectively.

In order to achieve reduced number of sharp bending in global path, a bending reducing operator has been derived in Equation (3) and included in heuristic search information of ACO. $cost(bend)$ is bending reducing operator. Number of moves from previous node to next node represented with *turn* and *theta* contains the angle formed among previous node and next node with respect to current node. φ and ψ represents cost of transforming turning times and angle into grid length, respectively.

$$cost(bend) = \varphi * turn + \psi * thita$$
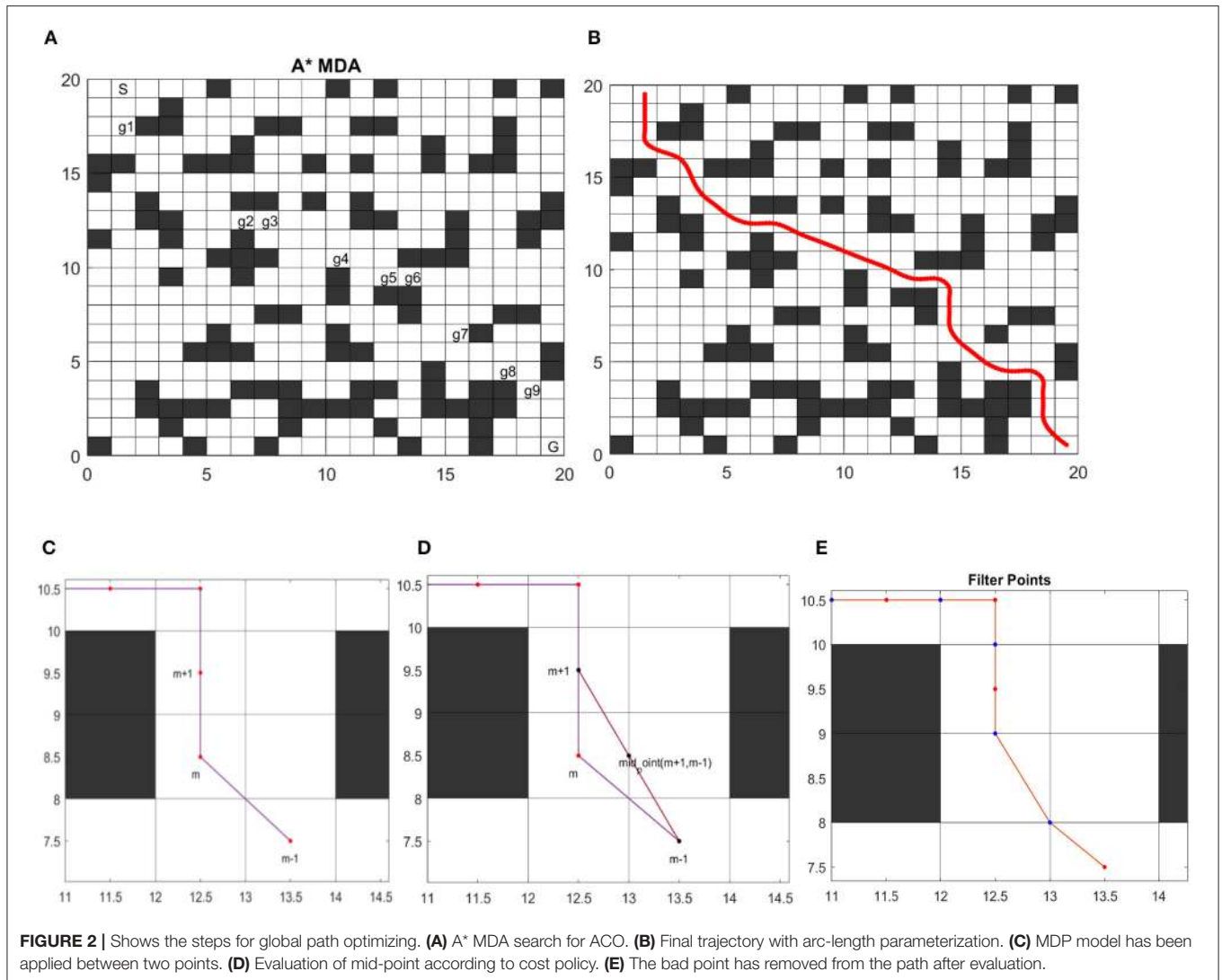$$\eta_{ij}(t) = \frac{Q_2}{f(n) + cost(bend)} \qquad (3)$$

FIGURE 2 | Shows the steps for global path optimizing. **(A)** A* MDA search for ACO. **(B)** Final trajectory with arc-length parameterization. **(C)** MDP model has been applied between two points. **(D)** Evaluation of mid-point according to cost policy. **(E)** The bad point has removed from the path after evaluation.

MAX_MIN Ant System (MMAS)

MAX_MIN system has been used in Stützle and Hoos (2000) to enhance pheromone strategy in ACO. Pheromone is updated after each iteration in conventional ACO, with MMAS the optimum route pheromone has been updated to the pheromone trial exclusively presented in Equation (4). Q_1, Q_2 are coefficient and each contains constant value less than one. L^{best} contains cost value for present shortest route. $Cals(l)$ and $Turns(l)$ represents sum of bending angles changed in path and sum of turns for optimal route. Similarly, ω_1 and ω_2 are considered as weights and defined according to robot's structure (Wu et al., 2011).

$$\tau_{ij}(t+1) = (1 - \rho)\,\tau_{ij}(t) + \Delta\tau_{ij}^{best}$$
$$\Delta\tau_{ij}^{k}(t) = \frac{Q_1}{L^{best}} + \frac{Q_3}{\omega_1 Cals(l) + \omega_2 Turns(l)} \qquad (4)$$

In conventional ACO, an ant during search may fall into local optima and mislead the search process. In this context, MMAS

pheromone trials has been bounded to upper limits and lower limits $\left[\tau_{\min}, \tau_{\max,}\right]$ as:

$$\begin{cases} \tau_{\min}, \tau \leq \tau_{\min} \\ \tau, \tau_{\min} < \tau \leq \tau_{\max} \\ \tau_{\max}, \tau > \tau_{\max} \end{cases} \qquad (5)$$

The above procedure aims to optimize the performance of ACO.

Markov Decision Process (MDP) Model With Novel Reward Policy

MDP consists of state-action transition model, which takes sequence of optimal path grids and evaluate every grid with respect to defined reward policy. Each optimal path grid is referred to as state and possible action has been taken based on reward policy. Reward policy is designed in order to check every middle point and its direction with respect to first two neighbor grids. Global route achieved through the Improved Ant colony algorithm. ACO is applied in a grid-based environment

TABLE 1 | A* Multi-directional algorithm.

Procedure: A*MDA

Initialization: Start(S), Goal(E), MAP, Open_list, Closed_list, PerentX_list, PerentY_list

// Setting up matrices Q representing neighbors to be investigated

// Add the start node

put the start Node on the Open_list (it's $f(S)$ equals to $h(s)$)

// Loop until you find the goal node

While the current node is equal to goal node

//Initializing current node with min $f(n)$ value and opening first node

if $f(n)$ equals to infinity

// Open_list is empty or goal not found

Break; // path not found

end if

for Current node equals to goal node

Break; // Retrieve the path

end for

// visit the Q neighbors and calculate $f(n)$ for each

for Q **do**

Using Equation (2) calculate $f(n)$

// save visited nodes x, y values in PerentX_list and PerentY_list.

end for

end While

While // Retrieve the path,

Currentnode_X = ParentX(Current node);

Currentnode_Y = PerentY(Current node);

If current node is equal to goal node

// Output = {Start, goal$n$, goaln_+1, goaln_+2,…goal$_t$}

end if

end while

TABLE 2 | Arithmetic equations for direction.

Direction	Arithmetic operation
n1	$\alpha - (N+1)$
n_2	$\alpha - N$
n3	$\alpha - (N-1)$
n4	$\alpha - 1$
n5	$\alpha + 1$
n6	$\alpha + (N-1)$
n7	$\alpha + N$
n8	$\alpha + (N + 1)$

TABLE 3 | Initial constraints policy for ant.

Obstacle direction	Eliminating grids
n_2	n1 & n3
n4	n1 & n6
n7	n6 & n8
n5	n3 & n8

to achieve a sequence of optimal grids points in the (x, y) coordinates. These center points are known as path candidates. The global route produced by path candidates consists of straight suboptimal line segments that contain sharp bending and the result is a rough path shown in **Figure 2C**. To deal with the non-holonomic properties of robot, this path is not feasible to

maintain a smooth and safe drive. To achieve path consistency and smoothness a novel evaluation function is introduced. This evaluation function examines path point sequence and remove bad path candidates according to the given cost policy. The evaluation technique follows the given steps:

Linear Interpolation

To increase the visibility of global path extra points has generated through Linear interpolation. Linear interpolation has a broad area of applications (Zheng et al., 2012). In Chapra (2012), interpolating function $f(x)$ is used to generate new point between every two global path points. In Equation (6) (x_0, y_0) and (x_1, y_1) are two consecutive global path points. To compute new points in a straight line, the equation of straight slope is used. In **Figure 2C** LI is used to generate a midpoint between every two path candidates showed by dot line.

$$f(x) = y = \frac{y_1 - y_2}{x_1 - x_0}(x - x_0) + y_0 \qquad (6)$$

Cost Policy

This paper introduces a new cost policy to filter the path obtained in grid environment. Path has considered of points m.

$$m_i = m_1, m_2, m_3, m_4, \ldots m_l \text{ and } l = total\ number\ of\ points$$

In order to achieve efficient computation results a novel cost policy is formed to evaluate grid points. The main objective of this evaluation function is to remove path points, which do not satisfy the cost policy. This cost policy is mainly consisting of following steps:

1 The Mid-Point Evaluation method has introduced to analyze the direction of each point among sequence. The evaluation path point has denoted by m_i. $m_i - 1$ and $m_i + 1$ are the first and second neighbor, respectively. **Figure 2D** shows the path points obtained using ACO present the sharp corners. Cost policy will perform midpoint evaluation on each grid point to recognize its direction to neighbors.
2 The path points have been passed through MDP evaluation model presented in **Table 4**. The cost has set 0 and that point will eliminate from path if the midpoint value of first two neighbors is equal to centered point as shown in cost policy **Table 5**. In **Figure 2E** the m point doesn't satisfy the cost policy and allotted 0 value, hence deleted from path.

Arc-Length Parametrization With Cubic Spline

In order to achieve optimal route from initial position to destination, several different approaches have used arc-length parameterization technique in predefined global waypoints to generate a reference road path. In grid-based environment, the obtained global path doesn't contain path consistency and robot is unable to localize itself precisely on derived route. This is considered as a drawback for solving path planning problem in grid-based environment. Therefore, the arc-length parameterization is applied to get curvilinear smooth route from refined path points. **Figure 2B** shows the smoothness and consistency in path generated with equally spaced points having
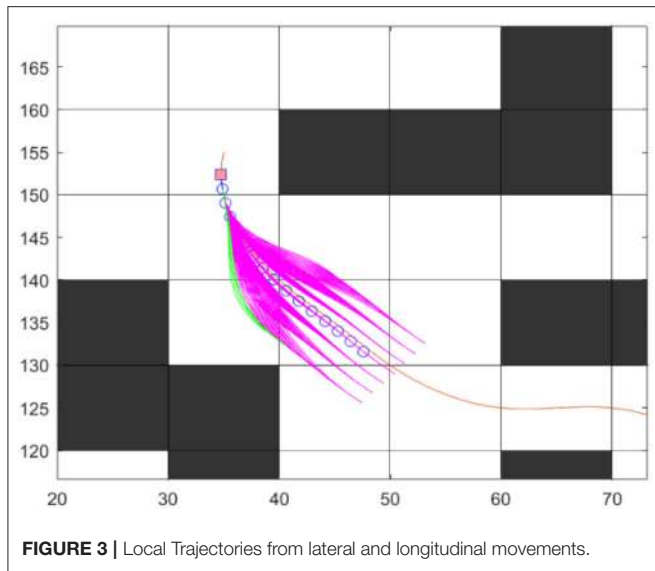
FIGURE 3 | Local Trajectories from lateral and longitudinal movements.

TABLE 4 | MDP state-action Model.

State	{m| m ϵ set of optimal path points }
Action	(remove m from path, Keep m in path)
Reward	0 is for removed m and 1 is for kept m

TABLE 5 | Novel reward policy.

If direction of first neighbor grids with m	Annotation	Reward
Midpoint($m_i - 1$&&$m_i + 1$) $\neq$ m_i where i is number of path point	m_i is a bending point	Assign 0 else 1

arc length (s) along global route defined thorough Equation (7) (Wang et al., 2002). The k_r curvature of each point present in optimal reference line has been derived through (8).

$$x_s = a_x (s)^3 + b_x (s)^2 + c_x (s) + d_x$$
$$y_s = a_y (s)^3 + b_y (s)^2 + c_y (s) + d_y \qquad (7)$$

$$k_r = \frac{x_s' y_s'' - x_s'' y_s'}{\sqrt{(x_s' + y_s')^3}} \qquad (8)$$

a_x, b_x, c_x, a_y, b_y, c_y, d_x, d_y has considered as coefficient. x_s', x_s'', y_s', y_s'' are first and second derivatives of x_s, y_s coordinates.

UNEXPECTED CONSTRAINTS AVOIDANCE AND TRAJECTORY PLANNING

After obtaining an optimal reference global route, robot starts to move toward destination. Meanwhile, it has to avoid the unexpected constraints and also has to maintain its position on reference trajectory. To obtain optimal behavior in unknown obstacle environments, curvilinear coordinates framework approached human like driving behavior. In human

like behavior, distance covered along road and offset to its center is taken into consideration. With this regard, Barfoot and Clark (2004) and Hu et al. (2018) curvilinear road path derived though cubic interpolation between predefined global waypoint. In Li et al. (2015), curvilinear path is adopted from digital map and B-spline curve used to bring smoothness in center line. This paper introduced an efficient approach to derive curvilinear reference route autonomously in complex grid maps using Improved ACO with MDP model and arc length parameterization. To define robot motion on global reference line, lateral and longitudinal coordinates generated and unified to produce set optimal trajectories.

Formation of Lateral and Longitudinal Coordinates

The obtained global route consists of smoothly distributed waypoints produced through arc-length parameterization. To maintain its position on reference route and avoid obstacles, robot has to move along offset of center line and offset distance represented with d. Arc-length s is taken as parameter to localize the position of robot at each waypoint on global route. Both s and d has been considered as lateral and longitudinal coordinates. For each forward move on reference route, a set of lateral and longitudinal coordinates are generated separately and unified to form a set of trajectories as shown in **Figure 3**. Time parameterization is used in producing lateral and longitudinal coordinates. In Werling et al. (2010), Quantic and quartic polynomial coefficient are derived to calculate a set of longitudinal and lateral trajectories with different final times and different final states (relative to the reference line). At initial ε_0 and final ε_t states lateral and longitudinal movements are considered as:

$$\varepsilon_0 = [\varepsilon_0, \dot{\varepsilon}_0, \ddot{\varepsilon}_0], \quad \varepsilon_t = [\varepsilon_t, \dot{\varepsilon}_t, \ddot{\varepsilon}_t] \qquad (9)$$

Furthermore, it can be minimized through considering square of jerk J_t with defined time integral from t_0 to t_1.

$$J_t (\varepsilon (t)) := \int_{t_0}^{t_1} \dddot{\varepsilon}^2(\tau) d\tau \qquad (10)$$

cost of trajectory C in constraint free environment is based upon sum of individual cost of lateral (C_d) and longitudinal (C_s) components, respectively.
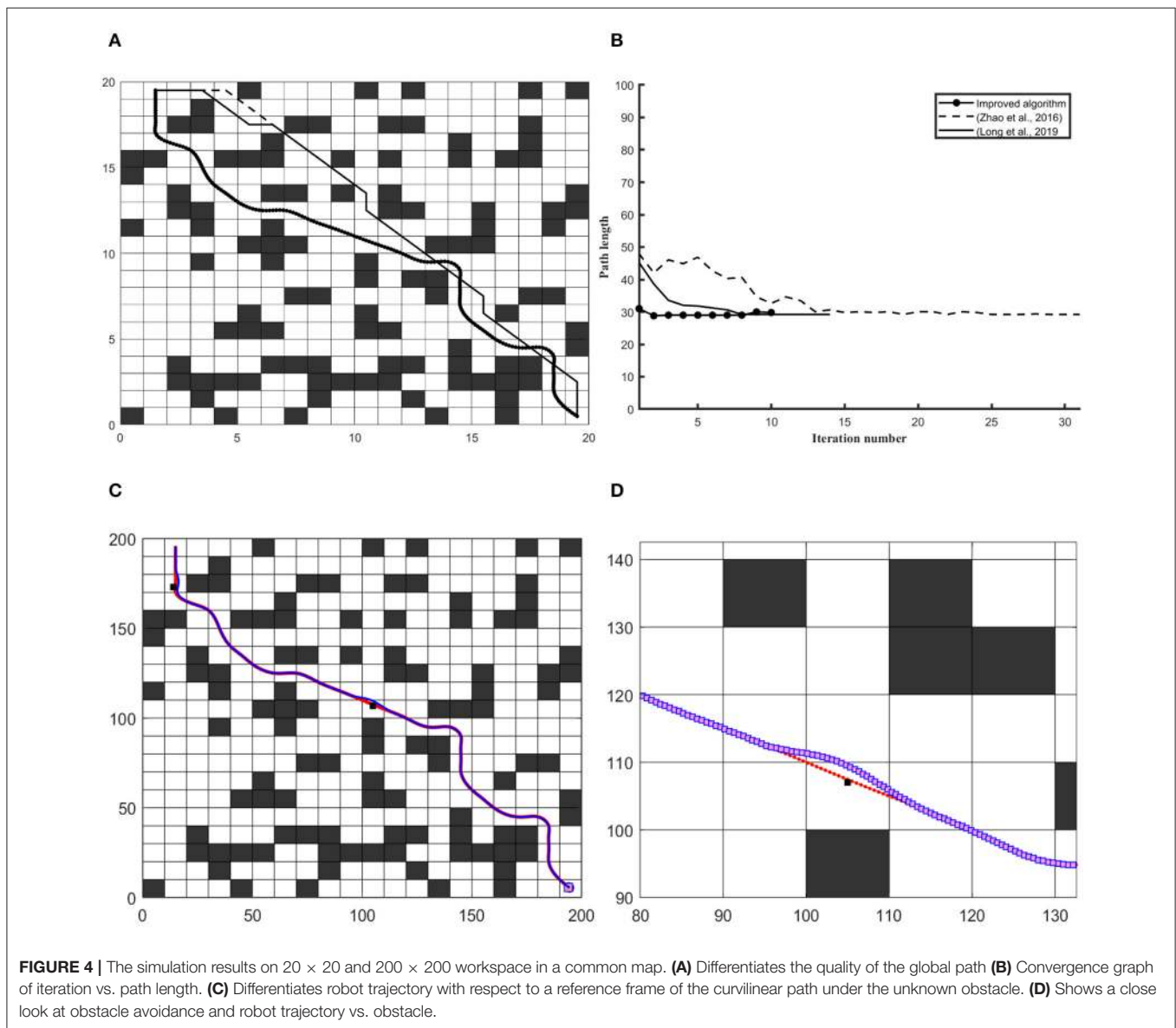
$$C = k_j J_t + k_t g(T) + k_p h(\varepsilon_t) \qquad (11)$$

T is referred as time interval of transforming state ε_0 to ε_t and g and h has been assumed arbitrary functions with parameters k_j, k_t and k_p. At first, lateral coordinates have been generated to specify the vehicle steering direction and to maintain offset distance parallel to center line. Initial lateral position of robot is d_0, $\dot{d}_0$, and $\ddot{d}_0$ refers to initial lateral velocity and acceleration, respectively. In set of final trajectories, lateral movements have been unified with longitudinal movements. In this content, Quantic polynomial (Rathgeber et al., 2015) has been used to get smooth lateral deviations suitable to longitudinal terminal manifolds in different

TABLE 6 | Simulation results.

	Common map			Tunnel map			Through map			Baffle map		
Algorithm	1	2	3	1	2	3	1	2	3	1	2	3
Number of sharp bends	10	10	0	12	10	0	–	13	0	–	10	0
Average path length	29.45	29.38	29.23	38.48	38.12	37.99	–	51.84	52	–	41.48	41.89
Number of iterations	33	12	9	35	16	10	–	40	11	–	16	10
Time of reference trajectory generation (sec)	7.26	4.89	1.49	20.62	17.97	1.601	–	88.20	1.778	–	10.98	1.39
Number of unknown obstacles	–	–	2	–	–	2	–	–	5	–	–	5
Time taken by robot to respond unknown constraints (millisec)	–	–	60	–	–	60	–	–	60	–	–	60

*Algorithm.
1-described in Zhao et al. (2016).
2-described in Long et al. (2019).
3-Derived approach in this paper.

FIGURE 4 | The simulation results on 20 × 20 and 200 × 200 workspace in a common map. **(A)** Differentiates the quality of the global path **(B)** Convergence graph of iteration vs. path length. **(C)** Differentiates robot trajectory with respect to a reference frame of the curvilinear path under the unknown obstacle. **(D)** Shows a close look at obstacle avoidance and robot trajectory vs. obstacle.

modes of operations. $\dot{d}_0$ and $\ddot{d}_0$ has taken 0 in starting, as the robot has to move parallel with center line.

$$D_0 = \left[d_0, \dot{d}_0, \ddot{d}_0, T\right] = [d_0, 0, 0, T] \tag{12}$$

Longitudinal coordinates have been utilized to localize the robot on reference line. At each forward move on global route, offset distance from centered waypoint to robot position is considered as d. Waypoint robot covers arc length s and estimate s_t position in time interval T paired with set of lateral movement using quartic polynomials (Guan et al., 2005). Initial longitudinal state of robot is $S_0 = [S_0, \dot{S}_0, \ddot{S}_0]$ and time t. After T interval it estimates terminal state $S_t = \left[\dot{S}_t, \ddot{S}_t\right]$ and time changes to t_t. Longitudinal trajectory generation can be achieved efficiently with quartic polynomials through varying the $\Delta \dot{s}_0$ and T.

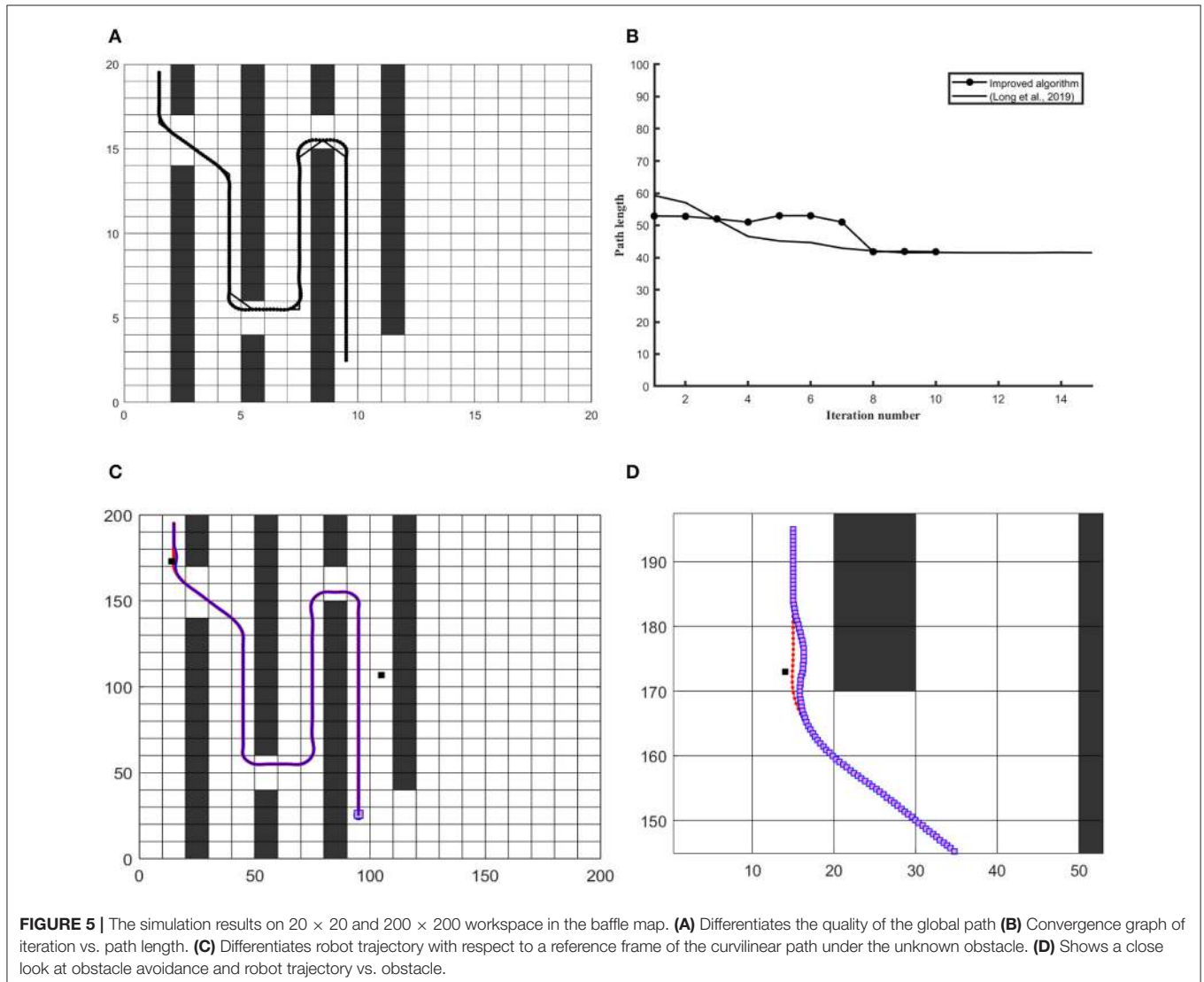$$[\dot{s}_t, \ddot{s}_t, T]_{0t} = [[\dot{s}_d + \Delta \dot{s}_0], 0, T] \tag{13}$$

In case of moving robot with constant velocity $\dot{s}_d$ it requires to a constant velocity instead of defining position to define its motion along reference path. The functional cost can be minimized through quartic coefficient.

$$C_d = k_j J_t\left(d\left(t\right)\right) + k_t\left(T\right) + k_d h(d_t) \tag{14}$$

$$C_s = k_j J_t\left(s\left(t\right)\right) + k_t(T) + k_s[\dot{s}_t + \dot{s}_d]^2 \tag{15}$$

In order to achieve a unified pair of trajectories on reference path, each lateral and longitudinal set has been passed through initial check. And trajectories coupled with unsuitable lateral and longitudinal accelerations has been removed based on the values of $\ddot{d}$ and $\ddot{s}$. The final cost of each optimal trajectory has been minimized through summing individual costs C_d and C_s (Werling et al., 2010). Similarly, every trajectory has been verified with respect to its cost value and minimum jerk J_t.

The final cost of each optimal trajectory has been minimized through summing individual costs C_d and C_s.

FIGURE 5 | The simulation results on 20 × 20 and 200 × 200 workspace in the baffle map. **(A)** Differentiates the quality of the global path **(B)** Convergence graph of iteration vs. path length. **(C)** Differentiates robot trajectory with respect to a reference frame of the curvilinear path under the unknown obstacle. **(D)** Shows a close look at obstacle avoidance and robot trajectory vs. obstacle.

In constraints avoidance scenario, a safety distance has defined to keep away collision. Every trajectory has been checked with respect to its cost value. With this regard, the heading $\theta(t)$, curvature $\kappa(t)$, velocity $v(t)$, and acceleration $a(t)$ has been taken in consideration to give underlying control of actuators. Therefore, curvilinear coordinates have been transformed to Cartesian coordinates as in Werling et al. (2012).

SIMULATION RESULTS

Path length has been considered a key element in the optimization of the global path planning problem. With the shortest and path smoothness, the robot consumes less amount of fuel to perform the task in minimum time. This paper improves both features to achieve an optimal trajectory. To verify the efficiency of the combined approach, simulations have been performed in MATLAB on different complex maps and robot size

has assumed smaller than grid resolution. In first section, optimal global path has been shown and compared with previous versions of ACO in sense of path consistency, smoothness and safety. In second part, the results presented under both pre-defined and unknown static constraint environments along with details have been given in **Table 6**. In **Figure 4**, the algorithm has been applied on common map with static constraint environment in 20×20 workspace and compared with Long et al. (2019) and Zhao et al. (2016). The figure shows that the obtained path has consistency in order to localize the robot along global route, as well as, the graph in **Figure 4B** verifies the efficiency of algorithm in reducing No. of iterations along with shortest distance considering safety and comfort. In **Figure 4C**, reference global path is represented through red dotted line and blue dotted line representing robot's trajectory under placement of unexpected static obstacles.

In **Figure 5**, the baffle map has been selected to optimize the global path with a static constraints environment in the workspace. This approach gives an optimal path in baffle
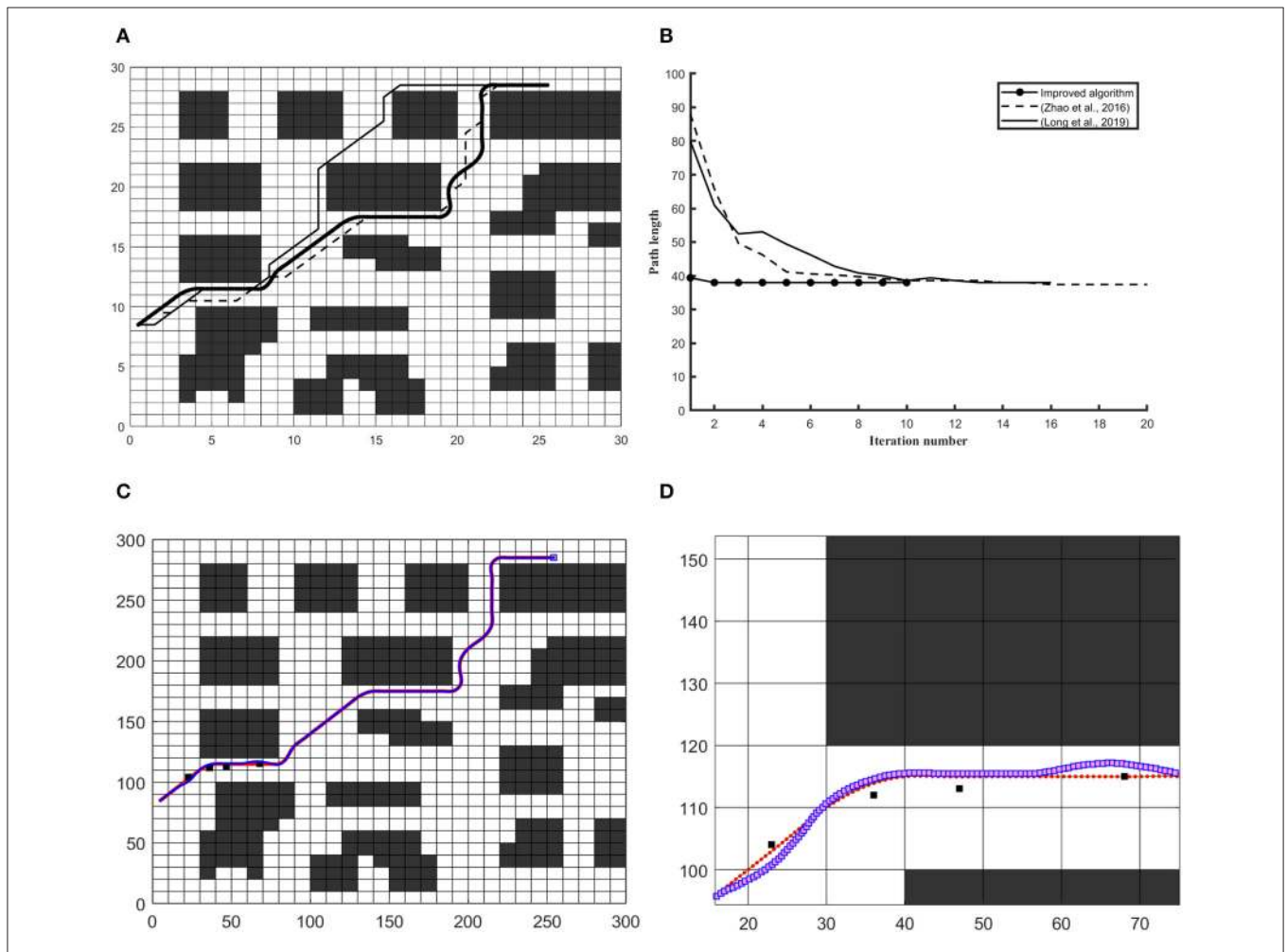

FIGURE 6 | The simulation results on 30×30 and 300×300 workspace in the tunnel map. **(A)** Differentiates the quality of the global path **(B)** Convergence graph of iteration vs. path length. **(C)** Differentiates robot trajectory with respect to a reference frame of the curvilinear path under the unknown obstacle. **(D)** Shows a close look at obstacle avoidance and robot trajectory vs. obstacle.

FIGURE 7 | The simulation results on 40 × 40 and 400 × 400 workspace in through map. **(A)** Differentiates the quality of the global path **(B)** Convergence graph of iteration vs. path length. **(C)** Differentiates robot trajectory with respect to a reference frame of the curvilinear path under the unknown obstacle. **(D)** Shows a close look at obstacle avoidance and robot trajectory vs. obstacle.
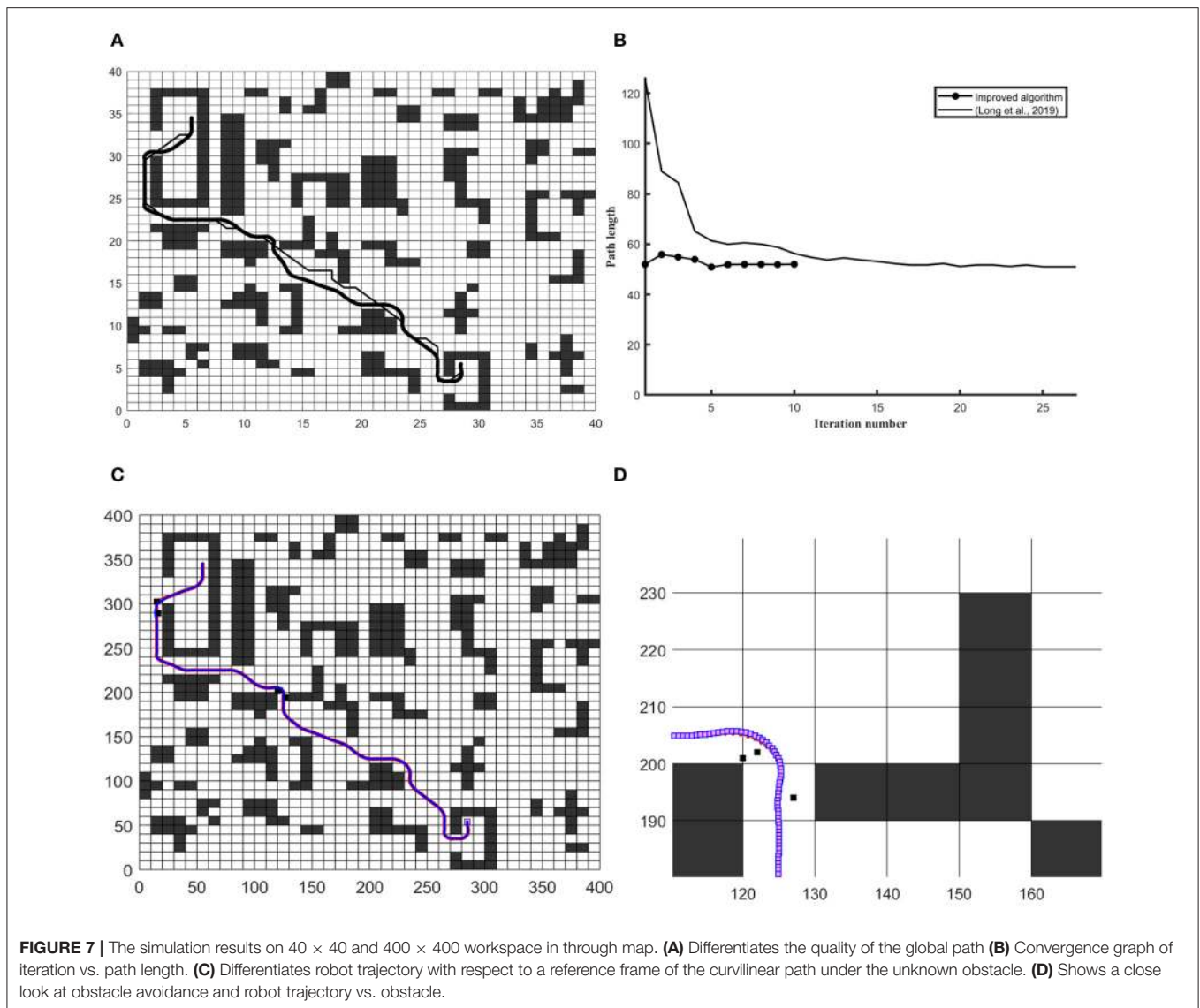
map with smoothness and enhanced path visibility, that assists the robot to localize and navigate efficiently, rather than (Long et al., 2019). In **Figure 5B** Vertical axis represents path length and the horizontal axis represents the number of iterations, respectively. **Figure 5A** differentiate the quality of path and **Figure 5B** shows that the number of iterations is lower than (Long et al., 2019) and, the computation time of this approach has also reduced compared with previous versions.

In **Figure 6**, algorithm has been applied on tunnel map. In **Figure 6A** Global route has shown to differentiate the results and **Figure 6B** further confirms efficiency of this approach with respect to finding optimal global path in 30 × 30 tunnel map. In order to deal with static and dynamic path planning, a 300 × 300 workspace is created. In **Figure 6C**, reference path is shown with red dotted line and robot

trajectory along reference path has been presented with blue dotted line and **Figure 6D** gives a close view of obstacle avoiding trajectory.

Similarly, **Figure 7** demonstrate the efficiency of this derived approach in 40 × 40 search space to solve static and dynamic path planning problems in grid-based environment. The obtained path in **Figure 7A** has been compared with Long et al. (2019) and the quality of derived path founds suitable to robot motion. The workspace has been increased to 400 × 400 in order to optimal trajectory in known and unexpected static constraints environment. In **Figure 7C**, robot's trajectory has been represented with blue dotted line and reference path represented with red dotted line. It indicates that robot follows reference route at unconstraint situation, and at some places robot's trajectory differs from reference route to avoid obstacle appearing ahead of it.

CONCLUSION AND FUTURE WORK

This paper introduces a combined approach for a mobile robot to deal with path planning problems in static and dynamic constraints environment and will be applied on mobile robot in real world environment. In the first part, global reference route is obtained through A* Multi-directional algorithm and improved ACO. MDP model based on novel reward policy has been introduced to evaluate the global path points generated in grid-based environment. Arc-length parametrization generates a curvilinear global route among obtained waypoints. In order to deal the environment with dynamic constraints, a set of lateral and longitudinal coordinates have been generated to maintain a suitable offset distance from global reference path and to produce a set of trajectories along global route. In addition, a cost policy is defined to choose the constraints-free smooth trajectory. In this way, an optimal approach is derived to deal with the complexity of different maps considering safety, path consistency and smoothness of trajectories compared to other grid-based approaches in complex-constraints environment. The simulations have been performed in MATLAB to verify its efficiency.

Although, the ability and functionalities of this approach in GPP and LPP can be extended further to cover verity of mobile robot applications and to avoid dynamic moving obstacles.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are included in the article/supplementary material.

AUTHOR CONTRIBUTIONS

HA proposed this contribution. DG and MW verified and conclude simulation results. XD gave suggestions in manuscript writing. All authors contributed to the article and approved the submitted version.

FUNDING

ACKNOWLEDGMENTS

REFERENCES

Akka, K., and Khaber, F. (2018). Mobile robot path planning using an improved ant colony optimization. *Int. J. Adv. Rob. Syst.* 15:1729881418774673. doi: 10.1177/1729881418774673

Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O., and Bouzouia, B. (2017). Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Rob. Auton. Syst.* 89, 95–109. doi: 10.1016/j.robot.2016.12.008

Barfoot, T. D., and Clark, C. M. (2004). Motion planning for formations of mobile robots. *Rob. Auton. Syst.* 46, 65–78. doi: 10.1016/j.robot.2003.11.004

Cetin, O., and Yilmaz, G. (2014). Sigmoid limiting functions and potential field based autonomous air refueling path planning for UAVs. *J. Intell. Rob. Syst.* 73, 797–810. doi: 10.1007/s10846-013-9902-y

Chapra, S. C. (2012). *Applied Numerical Methods With MATLAB for Engineers and Scientists.* New York, NY: McGraw-Hill.

Chu, K., Lee, M., and Sunwoo, M. (2012). Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE Trans. Intell. Transp. Syst.* 13, 1599–1616. doi: 10.1109/TITS.2012.2198214

Duchoe, F., Babineca, A., Kajana, M., Beeoa, P., Floreka, M., Ficoa, T., et al. (2014). Path planning with modified A star algorithm for a mobile robot. *Procedia Eng.* 96, 59–69. doi: 10.1016/j.proeng.2014.12.098

Ferguson, D., and Stentz, A. (2006). Using interpolation to improve path planning: the field D* algorithm. *J. Field Rob.* 23, 79–101. doi: 10.1002/rob.20109

Gu, Y., Hashimoto, Y., Hsu, L. T., Iryo-Asano, M., and Kamijo, S. (2017). Human-like motion planning model for driving in signalized intersections. *IATSS Res.* 41, 129–139. doi: 10.1016/j.iatssr.2016.11.002

Guan, Y., Yokoi, K., Stasse, O., and Kheddar, A. (2005). "On robotic trajectory planning using polynomial interpolations," in *2005 IEEE International Conference on Robotics and Biomimetics-ROBIO* (Shatin: IEEE), 111–116. doi: 10.1109/ROBIO.2005.246411

He, W., Chen, Y., and Yin, Z. (2015). Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE Trans. Cybern.* 46, 620–629. doi: 10.1109/TCYB.2015.2411285

He, W., Yan, Z., Sun, C., and Chen, Y. (2017). Adaptive neural network control of a flapping wing micro aerial vehicle with disturbance observer. *IEEE Trans. Cybern.* 47, 3452–3465. doi: 10.1109/TCYB.2017.2720801

Hu, X., Chen, L., Tang, B., Cao, D., and He, H. (2018). Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. *Mech. Syst. Signal Process.* 100, 482–500. doi: 10.1016/j.ymssp.2017.07.019

Huang, C., and Fei, J. (2018). UAV path planning based on particle swarm optimization with global best path competition. *Int. J. Pattern Recogn. Artif. Intell.* 32:1859008. doi: 10.1142/S0218001418590085

Hwan Jeon, J., Cowlagi, R. V., Peters, S. C., Karaman, S., Frazzoli, E., Tsiotras, P., et al. (2013). "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *2013 American Control Conference* (Washington, DC: IEEE), 188–193.

Hwang, J. Y., Kim, J. S., Lim, S. S., and Park, K. H. (2003). A fast path planning by path graph optimization, systems, man and cybernetics, part A: systems and humans. *IEEE Trans.* 33, 121–129. doi: 10.1109/TSMCA.2003.812599

Li, X., Sun, Z., Cao, D., He, Z., and Zhu, Q. (2015). Real-time trajectory planning for autonomous urban driving: framework, algorithms, and verifications. *IEEE/ASME Trans. Mech.* 21, 740–753. doi: 10.1109/TMECH.2015.2493980

Li, X., Sun, Z., Cao, D., Liu, D., and He, H. (2017). Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mech. Syst. Signal Process.* 87, 118–137. doi: 10.1016/j.ymssp.2015.10.021

Liu, J., Yang, J., Liu, H., Tian, X., and Gao, M. (2017). An improved ant colony algorithm for robot path planning. *Soft Comput.* 21, 5829–5839. doi: 10.1007/s00500-016-2161-7

Long, S., Gong, D., Dai, X., and Zhang, Z. (2019). Mobile robot path planning based on ant colony algorithm with A* heuristic method. *Front. Neurorob.* 13:15. doi: 10.3389/fnbot.2019.00015

Miao, H., and Tian, Y. C. (2013). Dynamic robot path planning using an enhanced simulated annealing approach. *Appl. Math. Comput.* 222, 420–437. doi: 10.1016/j.amc.2013.07.022

Pivtoraiko, M., and Kelly, A. (2005). "Efficient constrained path planning via search in state lattices," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space* (Munich), 1–7.

Rathgeber, C., Winkler, F., Kang, X., and Müller, S. (2015). Optimal trajectories for highly automated driving. *Int. J. Mech. Aerosp. Ind. Mech. Manuf. Eng.* 9, 969–975. doi: 10.5281/zenodo.1106815

Sarkar, C., Paul, H. S., and Pal, A. (2018). "A scalable multi-robot task allocation algorithm, 2018," *IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD: IEEE), 1–9.

Song, B., Wang, Z., and Zou, L. (2017). On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm. *Cogn. Comput.* 9, 5–17. doi: 10.1007/s12559-016-9442-4

Stützle, T., and Hoos, H. H. (2000). MAX–MIN ant system. *Fut. Gen. Comput. Syst.* 16, 889–914. doi: 10.1016/S0167-739X(00)00043-1

Walambe, R., Agarwal, N., Kale, S., and Joshi, V. (2016). Optimal trajectory generation for car-type mobile robot using spline interpolation. *IFAC Papers Online* 49, 601–606. doi: 10.1016/j.ifacol.2016.03.121

Wang, H., Kearney, J., and Atkinson, K. (2002). "Arc-length parameterized spline curves for real-time simulation," in *Proceeding 5th International Conference on Curves and Surfaces* (Saint Malo), 387396.

Wang, P., Lin, H. T., and Wang, T. S. (2016). An improved ant colony system algorithm for solving the IP traceback problem. *Inf. Sci.* 326, 172–187. doi: 10.1016/j.ins.2015.07.006

Werling, M., Kammel, S., Ziegler, J., and Gröll, L. (2012). Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. *Int. J. Rob. Res.* 31, 346–359. doi: 10.1177/0278364911423042

Werling, M., Ziegler, J., Kammel, S., and Thrun, S. (2010). "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE International Conference on Robotics and Automation* (Anchorage, AK: IEEE), 987–993.

Wu, J., Wang, J., and You, Z. (2011). A comparison study on the dynamics of planar 3-DOF 4-RRR, 3-RRR and 2-RRR parallel manipulators. *Rob. Comput. Integr. Manuf.* 27, 150–156. doi: 10.1016/j.rcim.2010.07.001

Yen, C. T., and Cheng, M. F. (2018). A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance. *Microsyst. Technol.* 24, 125–135. doi: 10.1007/s00542-016-3192-9

Zeng, M.-R., Xi, L., and Xiao, A.-M. (2016). The free step length ant colony algorithm in mobile robot path planning. *Adv. Rob.* 30, 1509–1514. doi: 10.1080/01691864.2016.1240627

Zhao, J., Cheng, D., and Hao, C. (2016). An improved ant colony algorithm for solving the path planning problem of the omnidirectional mobile vehicle. *Math. Probl. Eng.* 2016:7672839. doi: 10.1155/2016/7672839

Zheng, C., Cai, J., and Yin, H. (2012). A linear interpolation-based algorithm for path planning and replanning on girds. *Adv. Linear Algebra Matrix Theory* 2:20. doi: 10.4236/alamt.2012.22003

Zuo, L., Guo, Q., Xu, X., and Fu, H. (2015). A hierarchical path planning approach based on A* and least-squares policy iteration for mobile robots. *Neurocomputing* 170, 257–266. doi: 10.1016/j.neucom.2014.09.092