

Enhanced RRT* with APF and Halton Sequence for Robot Path Planning

Mohammed T. Hameed ^{1*}, Firas A. Raheem ², Ahmed R. Nasser ^{3*}

^{1, 2, 3} Control and Systems Engineering Department, University of Technology, Iraq, Baghdad 10067

Email: ¹ cse.22.10@grad.uotechnology.edu.iq, ² 60124@uotechnology.edu.iq, ³ ahmed.r.nasser@uotechnology.edu.iq

*Corresponding Author

Abstract—This paper presents a new path planning method (APF-IRRT*-HS), which relies on the optimization process of the conventional RRT* algorithm and combined with the APF method where the sampling process of the RRT* algorithm is improved using the Halton sequence, which is known to be deterministic and repeatable and provides more efficient coverage than other low discrepancy sequences with the modified goal-based method which provides a probabilistic approach to decide whether to sample from a point directly at the target or to choose a random point from the Halton sequence based on the current distance. We implemented the proposed method in two cases of mass point and two-link robots. The proposed method compares path length with the conventional RRT* algorithm and APF-RRT*, as well as time efficiency and number of iterations. The technique proves effective in various dynamic environments. Specifically, the APF-IRRT*-HS algorithm achieved an improvement of approximately 21.88% and 7.5% in path length, 79.75% and 49.2% in computation time, and 57.39% and 40% in the number of iterations compared with the RRT* and RRT*-APF algorithms, respectively. We can use this method in everyday applications such as robotic arms, drones, self-driving cars, etc. More advanced methods, such as multi-link robots and real-time constraints, can be used in the future.

Keywords—Path Planning; Rapid-Exploring Random Tree; Artificial Potential Fields; Halton Sequence; Free Cartesian Space (FCS).

I. INTRODUCTION

In recent years, robotics has developed significantly, advancing the field in various possible applications, including healthcare, manufacturing, logistics, self-driving car technologies, and more industries [1], [2]. In this sense, path planning is one of the fundamental challenges in robotics. It can be defined as finding the perfect path from a starting point to a goal point, avoiding all obstacles in the surrounding environment, reducing energy, determining the shortest route, and achieving smoothness of the path [3], [4]. Path planning helps ensure the robot's mobility and safety when moving between complex environments efficiently and accurately. It is key for successful applications such as moving freely in space and avoiding obstacles when driving in vehicles [5], [6].

As shown in Fig. 1, researchers classify path-planning techniques into grid-based techniques, like Dijkstra [7], D* [8], and A* [9] algorithms, representing the environment as a grid of cells to find the shortest path using this grid structure. Sampling-based techniques, such as rapid random tree exploration (RRT) [10] and probabilistic road mapping

(PRM) [11] methods, select random samples in the robot's external space to explore the most efficient paths in complex or high-dimensional environments. Artificial intelligence-based techniques, such as deep reinforcement learning [12] and genetic algorithms [13], use AI methods to train the robot to adapt to complex environments to find the best paths. Mathematical-based methods, such as linear control [14] and optimal control [15], determine the optimal route of the robot in the environment according to mathematical constraints and mathematical equations. Potential field methods, such as APF, guide the robot toward a goal while avoiding obstacles using attractive and repulsive force fields [16]. Fig. 1 A comprehensive overview of popular path-planning techniques.

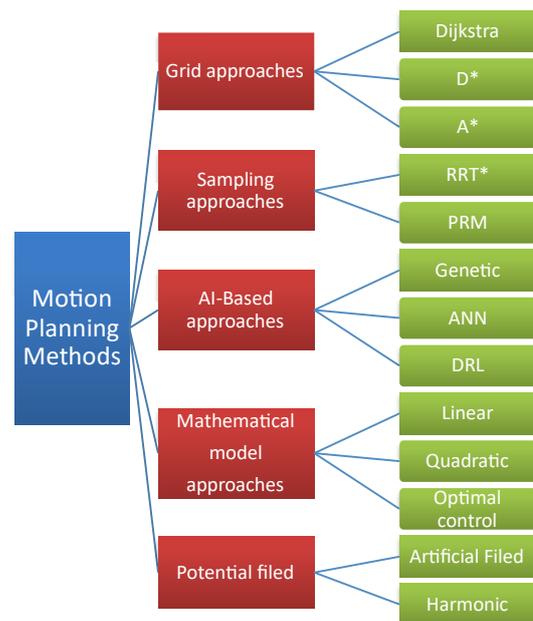


Fig. 1. An overview of commonly used motion planning methods

LaValle at Iowa State University proposed one of the most renowned global motion planning methods, the RRT algorithm, in 1998 [17]. RRT values for its effectiveness in using incremental sampling to identify global solutions. The key concept behind the RRT [18] algorithm is a discovery tree formed at the starting point and expanded by drawing random samples from the available space. The RRT algorithm is straightforward and capable of finding solutions in intricate environments, making it widely adopted in mobile robot navigation. Nevertheless, reliance on random sampling often results in solutions that are not optimal [19], [20].

Unlike the RRT, the RRT* [21] enhances state space exploration by optimizing the tree through extension rewiring, aiming for near-optimal paths [22], [23]. However, the algorithm becomes inefficient due to the amplified sampling and the iterative process. In addition to the difficulty of exploring random samples, this reduces their efficiency in narrow corridors and crowded environments.

The APF can be integrated with RRT* [24], which is extensively used for path-smoothing control [25]. In particular, the APF algorithm employs the effects of attractive and repelling forces, controlling the robot's trajectory by applying potential fields to accomplish obstacle avoidance in changing environments. However, the proximity of obstacles to the target can lead to conflicts between attractive and repulsive forces, causing the APF algorithm in local minima to be trapped, which renders the target unattainable [26].

The APF-IRRT*-HS algorithm is proposed in this paper to solve the problems of random point selection and the difficulty of discovering close paths quickly in the path planning process. APF-IRRT*-HS involves additional modifications to select random points compared to the APF-RRT* approach. We applied the proposed method in two cases: a two-link system and a mass point to demonstrate the efficiency of the proposed algorithm in both methods in terms of time taken, path length, and the number of iterations used, each and compare its results with the rest of the improved RRT algorithm.

The main contribution of this work is the development of the APF-RRT* algorithm, an enhanced version of the RRT algorithm. More specifically, the APF-IRRT*-HS method improves the search and the selection of random points by adding a set of methods; the first approach is the goal-basing [27] method, which uses a probabilistic approach to select samples toward the goal with adding some modification for this technique. In addition to the Halton sequence algorithm [28], which selects quasi-random points close to the target to reduce the path length, this method depends on the distance of the current point to the target to determine the probability ratio. The closer the current point is to the target, the greater the percentage of sample selection toward the target. The further it is from the target, the greater the probability of selecting the sample from among the Halton sequence points, considering the number of repetitions and generating smoother paths compared to other methods. The proposed algorithm enables application in many real-world settings, such as robotic arms, drones, and all applications that require precise and dynamic control and unknown environments that consider the length of the path and the time taken.

We organize the rest of the paper as follows. Section 2 provides a summary of the related works. Subsequently, Section 3 clarifies the algorithms and the two-link system theoretically, supported by equations. Afterward, the fourth section showcases the proposed APF-IRRT*-HS method, including the construction process and the significance of this approach. Section 5 demonstrates the simulation result of the mass point and the two-link system with a discussion to illustrate the extent of the enhancement that the suggested method shows compared with various approaches. Finally,

the last section demonstrates the percentage of improvement and concludes after performing the proposed method.

II. RELATED WORKS

Numerous study efforts have focused on enhancing the efficiency and performance of the RRT algorithm. In this regard, some significant contributions are summarized in this area as follows.

The Informed RRT*-Connect [23] is an improvement over the conventional RRT* algorithm, introducing the idea of integrating informed sampling with the RRT-connected framework. The method focuses the search on an ellipsoid subset of the configuration space after finding the first solution to improve the efficiency of the resulting path. Nevertheless, potential inaccuracies in the ellipsoid approximation used for the search space may either miss promising areas or be overly restrictive. Moreover, the authors in [29] proposed a planning method based on integrating artificial potential fields (APF) and RRT*, which uses attractive and repulsive forces to attract the robot towards the target and help it avoid obstacles. The study first analyzed the advantages and disadvantages of three popular path-planning algorithms and then presented the extended APF-RRT* algorithm. Simulation data showed that the improved APF-RRT* algorithm is the most effective approach in determining the optimal path and effectively solves the problems of slow convergence and low scheduling efficiency in complex environments. However, there are several problems, such as oscillation, local minima in complex environments, and increased collision rates with obstacles, which we will discuss how to avoid in our proposed method. Huang [30] discussed the RRT algorithm combined with the APF algorithm to improve the path-planning process. This approach proposed the goal-basing factor with a gravitational component to speed up the path toward the target and reduce randomness. In addition, it restricted the search area to a repulsive field. It reduced the path length and iterations in this hybrid method of the RRT* algorithm. However, owing to the problematic repulsive forces from the APF, the resulting paths might still exhibit oscillations close to obstacles. This oscillatory attitude can lead to jagged paths, especially in cluttered environments.

Furthermore, the authors in [31] presented an improved path-planning algorithm for intelligent mobile robots to navigate complex environments. This algorithm solves problems such as low search efficiency, redundant nodes, and uneven paths. It incorporated Target bias sampling and the adaptive step size strategy into the RRT framework, which aims to optimize path generation while considering robot constraints. It used evaluation metrics like search time, path length, and number of sampling nodes to assess performance. A comparative study involving the proposed algorithm and the RRT, RRT-connect, and RRT* algorithms showed significant improvements in path length reduction and planning time. In another work, the authors in [33] proposed a new algorithm known as IRRT-SSA by incorporating the salp swarm algorithm (SSA) to enhance the RRT algorithm. This algorithm improves efficiency and path-finding capabilities by strengthening the inherent issues of the traditional RRT. It performed a comparative analysis to

assess the performance of the IRRT-SSA compared to the RRT, where the results demonstrated that the proposed method is effective by approximately 49% in length, 54% in the number of nodes, and 54% in the average number of iterations. However, this algorithm remains ineffective in complex environments and applications that require high computational costs.

On the other hand, F. N. Irzoqe et al. [32] proposed a hybrid approach referred to as the Modified Deep Q-Network-Artificial Potential Field (MDQN-APF), which combines the APF method and a modified version of Deep Q-Network that uses a dynamic reward model to improve mobility and 25 connected neighbors. Compared to the traditional DQN-APF algorithm, the proposed MDQN-APF algorithm showed a performance improvement of 42.867%. However, this work may have difficulty dealing with rapidly changing obstacles, and since it mainly focuses on static environments, it limits adaptability. In addition, the increased computational complexity may hinder using multiple connected neighbors.

Despite advancements in research on the sampling process in the RRT algorithm, it still suffers from the problem of random selection of points that delays the planning process and reduces its efficiency, in addition to the issue of fluctuation in the APF. Therefore, this paper proposes adding semi-random sequences in addition to the modified goal-basing method, where selected samples by applying a probabilistic approach based on the distance between the current point and the target to determine what points will be chosen towards the target or from the points of the Halton sequence, which is known to be deterministic and repeatable and covers a less random area to help the RRT algorithm produce faster and more efficient paths and better path length and avoid passing through points far from the target, which takes more time.

III. THEORETICAL FRAMEWORK

This section provides theoretical descriptions of the RRT* algorithm, the APF algorithm, the Halton sequence algorithm, and the proposed algorithm that combines the three algorithms.

A. The RRT Algorithm

The RRT, or the Rapid-exploring Random Tree, is a pervasively adopted probabilistic technique for solving motion planning problems. Meanwhile, the RRT* is a version that guarantees asymptotic optimality. The fundamental process of the RRT consists of iteratively applying the following steps:

- Randomly selecting a node in space. This node is called a sampling point x_{rand} .
- Selecting x_{near} , which is the most proximate node in the tree to the x_{rand} .
- Based on the step size h with x_{rand} and x_{near} , a new node is selected as x_{new} , also referred to as the primary node, as shown in Eq. (1):

$$x_{\text{new}} = x_{\text{near}} + h \cdot \frac{x_{\text{rand}} - x_{\text{near}}}{\|x_{\text{rand}} - x_{\text{near}}\|} \quad (1)$$

Eq. (1) indicates how a new point x_{new} is created near the random point x_{rand} relies on the nearest point x_{near} , using the step size h , controlling the movement step to expand the path, or using the search process in space.

- If there are no collisions between x_{rand} and x_{new} . Then, it inserted the path into the node tree. However, if a collision occurs, it will reject the path, and the loop will continue [34], [35].

An enhancement in the RRT* method over the basic RRT incorporates a cumulative cost feature, which adds up the lengths of all edges from the initial point to a particular node. The strategy involves a revision where the parent node is for x_{new} . Rather than employing x_{near} . Next, the nodes within a designated radius around x_{new} are analyzed, and it picked the one with the lowest cost as the preferred parent node. After identifying the most suitable parent node, the algorithm traverses the remaining tree nodes, determining the path costs reaching both x_{new} and x_{near} from each node. It then rebuilds the tree by selecting the path with the least accumulated cost [36], [37], [38].

B. The Traditional APF

Path planning methods based on artificial potential fields are frequently applied to robotics, autonomous ships, uncrewed aerial systems, and various other fields to address planning problems [39], [40]. Mainly, the APF method constructs a repulsive potential field around obstacles to prevent the robot from colliding with them and forms an attractive potential field around the target to pull the robot towards it. By exploiting the combination of repulsive and attractive potential fields, the algorithm maps out a path from the initial position to the destination, ensuring safety from collisions. Nevertheless, situated in situations where obstacles are near the target, the robot might face the issue of being unable to reach the target because of the repulsive forces [41], [42], [43]. Specifically, U_{att} , the attractive potential field function, and U_{rep} , the repulsive potential field function is specified in Eq. (2) and Eq. (3).

$$U_{\text{att}}(q) = 0.5k_{\text{att}} \|q - q_g\|^2 \quad (2)$$

$$U_{\text{rep}}(q) = 0.5k_{\text{rep}} \left(\frac{1}{\|q - q_o\|} - \frac{1}{d_o} \right)^2 \quad (3)$$

Where k_{att} and k_{rep} represent the gain coefficients for the attractive and the repulsive field dynamics, respectively, q marks the position, q_g is the destination coordinate, and d_o determines the obstacle's influence range.

The combined potential field function $U(q)$ is formulated as the sum of the gravitational and the repulsive potential fields, with the resultant force $F(q)$ defined as follows in Eq. (4):

$$\begin{aligned} F(q) &= F_{\text{att}}(q) + F_{\text{rep}}(q) = -\nabla U_{\text{rep}}(q) - \nabla U_{\text{att}}(q) \\ &= -\nabla U(q) \end{aligned} \quad (4)$$

where $F_{\text{att}}(q)$ corresponds to the attractive force, while $F_{\text{rep}}(q)$ represents the repulsive force.

Attractive forces are the forces that attract the robot to move towards the target. The closer the robot gets to the

target, the more these forces increase until it reaches it. The forces that work to keep the robot away from obstacles in the environment are known as repulsive forces. They increase the closer the robot gets to the obstacles so that a collision does not occur. [44], [45], [46], [47].

C. Halton Sequences

The Halton sequence is the most fundamental low-discrepancy sequence in numerous dimensions [48]. In particular, the Halton sequences rely on assorted prime bases, with each dimension correlating to a prime number. It frequently chooses These prime numbers individually to make it easier to control the distances between sample points after combining different dimensions [49]. Classify the Halton sequence as an infinite sequence, and it is an upgrade upon the Vander sequence [50], imitating it confined when the Halton sequence is to one dimension. Halton sequence improves the sampling process in the path planning process in a way that ensures uniform distribution and low variance in an easy and less complex way compared to random distribution or second pseudo-random methods, as it converts the number n based on the base b to its representation in a specific numerical system. The resulting point is expressed through the following mathematical as demonstrated in Eq. (5):

$$n = \sum_{i=0}^{b-1} a_i(n)b_i \quad (5)$$

Assume that the function is $H_b(n)$, and the resulting point in the sequence is at the number using the base b . You can show the definition in the formula below in Eq. (6) [51], [52].

$$H_b(n) = \sum_{i=0}^{b-1} a_i(n)b_i \quad (6)$$

D. The Two-link Robot Arm Kinematic Model

A robotic structure includes multiple rigid parts linked by joints, and the combined position and orientation of these rigid bodies in space are collectively known as a pose. Robot kinematics covers the pose, speed, and acceleration of the rigid components in the robotic system. In this regard, a connection between two elements that enables relative motion is called a kinematic joint. The robot's kinematics encompasses forward and inverse pose calculations [53].

In forward kinematics, we use the joints' variables to find the position and orientation of the robot's end-effector [54]. Solving this issue is generally straightforward, with the derivation of the equation being simple and free of complications.

As shown in Fig. 2, by employing the geometric approach, the forward kinematics equations for the 2-DOF planar robot can be stated as given below:

$$x = a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) \quad (7)$$

$$y = a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) \quad (8)$$

In Equations (7) and (8), we specified the coordinates (position) of the end-effector in the frame attached to the base of the robot $O(x_0, y_0)$.

The rotation angle between the frame and the fixed base frame characterizes the end-effector orientation. The angle θ_E is connected to the actual joint displacements as shown below:

$$\theta_E = \theta_1 + \theta_2 \quad (9)$$

Consequently, the fixed coordinate system attached to the robot's base defines the end-effector position and orientation in Equations (7) to (9) in terms of θ_1 and θ_2 .

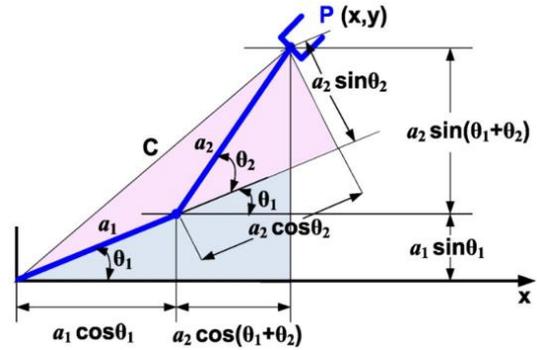


Fig. 2. Forward and inverse kinematics of a 2-link Robot planar manipulator

On the other hand, in the method that solves the joint variables of a given position, we referred to the direction of the end-effector as the inverse kinematic method. However, this method is difficult, complex, and costly [56].

The inverse kinematics problem can be addressed by adhering to the parts below (see Fig. 2):

Part 1: Specification of the length between the end-effector and manipulator base as shown in Eq. (10).

$$c^2 = x^2 + y^2 \quad (10)$$

Part 2: it can obtain the elbow joint angle θ_2 :

$$D = \cos \theta_2 = \left(\frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1 a_2} \right) \quad (11)$$

can be found θ_2 by:

$$\theta_2 = \tan^{-1} \frac{\pm \sqrt{1-D^2}}{D} \quad (12)$$

Part 3: The angle of the shoulder joint θ_1 determined as shown:

$$\theta_1 = \tan^{-1} \left(\frac{x}{y} \right) + \tan^{-1} \left(\frac{a_2 \sin \theta_2}{a_1 + a_2 \cos \theta_2} \right) \quad (13)$$

It introduces the 2-DOF planar robot with a straightforward design. However, the process of determining the inverse kinematics is regarded as quite complex [55], [57], [58]. In the proposed method, inverse equations are used to analyze and determine the workspace and the angles required to locate a point in the free Cartesian space (FCS).

E. Free Cartesian Space (FCS)

The results from the inverse kinematics are employed to develop and analyze the workspace in this study. The free Cartesian space (FCS) outlines the exact positions accessible by the end-effector. The form and volume of the FCS within an environment with numerous obstacles are influenced by the obstacles' number, position, and size, as well as the restrictions imposed by the joints. For the path planning and

simulation of a two-link robot arm's motion, consult the theoretical range of the arm provided in Table I. These constraints influence and confine the manipulator's motion, further separating the workspace into reachable and unreachable areas.

TABLE I. THE THEORETICAL RANGE OF THE TWO-LINK ROBOT ARM

Link No.	Ranges
Joint 1	$0 \leq \theta_1 \leq 360$
Joint 2	$-150 \leq \theta_2 \leq 150$

To compute all possible solutions, the FCS is calculated by examining the points in the environment. The obstacle center and the diameter are critical to the checking function. Furthermore, every point in the environment can exist in one of four possible states. The solution in the first case is the elbow-up solution, in which the arm takes an upward path to reach the target point. In the second state, the elbow-down solution, the arm takes a downward path to reach the target point. The third state represents a combination, which means the arm can reach the target with either the elbow-up or the elbow-down configuration. The last state is the no-solution scenario, caused by the point being either out of the manipulator's range, located on an obstacle, or due to a collision of one or both links with an obstacle. The length of the arm link is the most influential factor in the Cartesian free space [59].

F. The Proposed Method, the artificial potential field, and the enhanced RRT* based on the Halton sequence (APF-IRRT*-HS)

The proposed (APF-IRRT*-HS) method presented in Algorithm 1 aims to enhance the performance of the traditional APF-RRT* algorithm via generating quasi-random points better distributed in multi-dimensional spaces compared to the conventional randomness (such as rand). More specifically, the points generated by the Halton sequence tend to cover the space more uniformly than the standard APF-RRT* algorithm, producing short paths while reducing the number of iterations utilized for optimization.

In addition, it utilized the developed goal-basing method. Instead of relying on sampling random points in the space, the algorithm can preferentially sample points closer to the goal or in a direction that moves towards the goal. For each iteration, the code uses a probabilistic approach to decide whether to sample a point directly at the goal or choose a random point from the Halton sequence, which increases the chance of sampling toward the goal as the algorithm progresses. If the random value is less than this probability, X_{rand} is set to the goal node itself. Otherwise, it is set to a point from the Halton sequence, scaled to fit within the bounds of the environment. This approach ensures that the tree explores more towards the goal as it gets closer, improving the efficiency of finding a feasible path.

The method does not only improve the selection of samples in the RRT* but also uses smoothing to solve the problems of oscillation or deviation that occur in the APF algorithm because of repulsive forces, where the path is modified, unnecessary points are removed, shortcut smoothing is used, and then the new path is saved.

Compared with the standard RRT* algorithm, when adding the Halton sequence method with the modified Goal-basing in addition to smoothing, the proposed method has ensured improving the process of selecting random points in a more regular manner, which leads to flexibility in the path, reducing its length and the time taken, and avoiding selecting points far from the goal and not necessary. This leads to increased productivity, especially in industrial and logistical systems, due to reducing the time that ensures completing more tasks in a shorter period, saving energy, improving production costs, and system response.

Mathematically, the random point represented the conditional statement as follows:

$$x_{rand} = \begin{cases} X_{goal} & \text{if } rand < P \\ p(iter) & \text{otherwise} \end{cases}$$

The equation below outlines the probability formula:

$$P = a \times \left(1 - \frac{D_{current}}{D_{total}}\right) \quad (14)$$

Where a is the coefficient that controls the likelihood of selecting the goal node.

$$D_{current} = \text{norm}(X_{goal} - x_{new})$$

It is the Euclidean distance between the goal node and the current last node of the tree.

$$D_{total} = \text{norm}(X_{goal} - X_{init})$$

It is the Euclidean distance between the goal node and the start node.

Eq. (11) represents the process of selecting points in this method, where a probabilistic approach is applied to this selection process. The closer the current point is to the target, the greater the percentage of point selection coming towards the target X_{goal} . If it is not, it is selected from the semi-random points selected from the Halton sequence $H(iter)$.

The flowchart in Fig. 3 illustrates the methodology of the proposed algorithm.

Algorithm 1: APF-IRRT*-HS

1 – Setup APF – IRRT* – HS parameters:

Input: $X_{init}, X_{goal}, X, n, k_{att}, k_{rep}, d0, \delta, a$

Output: $G = (V, E)$

$V \leftarrow \{X_{init}\}, E \leftarrow \emptyset, G \leftarrow (V, E)$

$H \leftarrow \text{HaltonSequence}(n)$

$iter \leftarrow 1, goalReached \leftarrow \text{False}$

2 – Random point selection:

for $iter \leftarrow 1$ to n do

if $\text{rand}() < a * (1 - X_{goal} - X_{near} / \text{norm}(X_{goal} - X_{init}))$

$X_{rand} \leftarrow X_{goal}$

else

$X_{rand} \leftarrow H(iter) * \text{dimension}$

3 – Find the nearest point:

$X_{nearest} \leftarrow \text{Nearest}(V, X_{rand})$

4 – steer the tree towards the new point:

$\alpha \leftarrow (X_{rand} - X_{nearest}) / \text{norm}(X_{rand} - X_{nearest})$

$\beta \leftarrow U_{att} + U_{rep}$

$\theta \leftarrow \alpha + \beta / \text{norm}(\beta)$

$X_{new} \leftarrow \text{Steer}(X_{nearest}, \delta, \theta)$

5 – Obstacle checking process:

if $\text{CollisionFree}(X_{nearest}, X_{new})$ then

```

 $X_{near} \leftarrow Near(V, X_{new}, 2 * \delta)$ 
 $X_{parent} \leftarrow BestParent(X_{near}, X_{new}, X_{nearest})$ 
 $V \leftarrow V \cup \{X_{new}\}$ 
 $E \leftarrow E \cup \{(X_{parent}, X_{new})\}$ 
then

```

```

6 – Verification of Goal Reached:
if  $X_{new} - goalNode < \delta * 10$  then
goalReached  $\leftarrow True$ 
 $V \leftarrow V \cup \{X_{goal}\}$ 
TraceBack( $X_{goal}$ )
break
end for

```

7 – Path smoothing process:

```

if  $d(X_i, X_{i+2}) > \epsilon \Rightarrow delet X_{i+1}$ 
Smoothing( $X_1, X_2, \dots, X_n$ )
PathSafe  $\leftarrow True$ 
Pathfinal  $\leftarrow X_{smooth}$ 
return  $G = (V, E)$ 

```

IV. RESULTS AND DECISIONS

This section will discuss the simulation results and compare the performance of the RRT*, the APF-RRT*, and the proposed method, the APF-IRRT*-HS, utilizing two cases, including the mass point and the two-link motion. In this regard, the quality of the path length, the number of search attempts, and the computation time are the performance measures of the three methods. These methods were implemented on a computer with a core i5-12450H CPU and 8Gb RAM using MATLAB R2023a. The above specifications enhance the performance of algorithms by accelerating parallel processing and improving the handling of extensive data. The multi-core processor speeds up parallel calculations while sufficient memory provides space to execute tasks without slowing down. The 8 GB RAM also provides enough capacity to store data and temporary paths without affecting performance, which contributes to accelerating the path planning calculation process in the MATLAB R2023a environment.

A. Mass Point (MP) Simulation Results and Decisions

The first case in which the proposed approach will be implemented is the Masa Point, which has four different environments. Three of these test environments contain several known or fixed obstacles of varying complexity. In contrast, the fourth environment is a dynamic environment consisting of a fixed maze with two moving obstacles. In these four environments, an 80 by 80 boundary was used. Utilizing iterative testing, Table II shows the selected values for the attraction coefficients determine how to force the path attracted toward the target, the repulsion coefficients show how strongly the path is pushed away from obstacles, and the effect of the range of barriers warns it when the path reaches a certain distance between it and the obstacle. The control coefficient determines how to combine the repulsive and attractive forces and tries to balance them, N_{max} , which is the maximum number of iterations the robot allows to engage the target, and the step size represents the number of steps in each iteration. These values were elected for each type of test environment. The starting point is depicted as a green star, while the target point is shown as a black plus sign. The purple circles indicate obstacles, and the blue branches represent the tree. Finally, the path is displayed in red. These settings remain consistent across all the test environments.

The black maze and the red moving circles represent the dynamic environment.

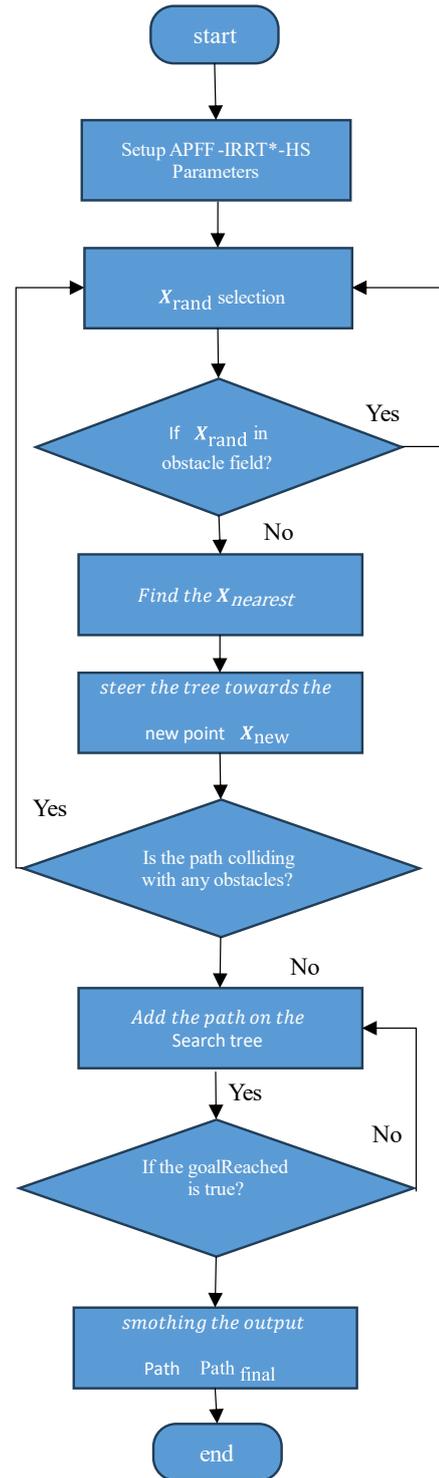


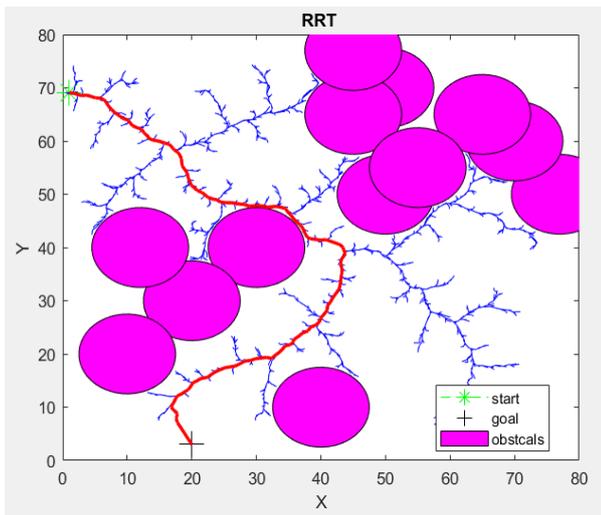
Fig. 3. Flowchart APF-IRRT*-HS

TABLE II. THE PARAMETERS EMPLOYED IN EACH TEST ENVIRONMENT

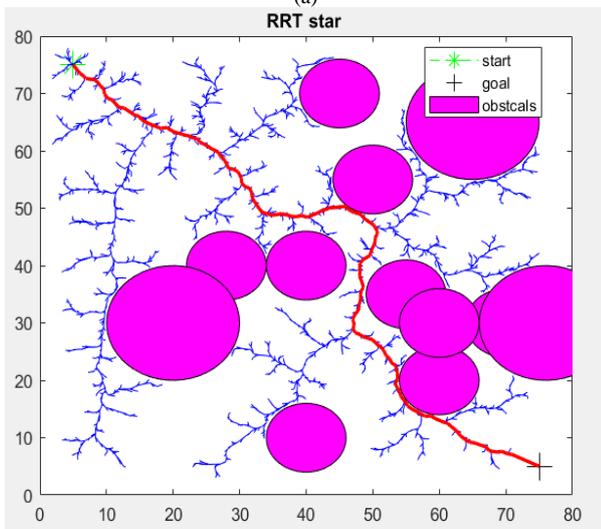
Parameter	Value
k_{att}	10
k_{rep}	2
d_0	10
h	0.45
α	0.4
N_{max}	$8 * 10^4$

1) Simulation Results of the RRT*

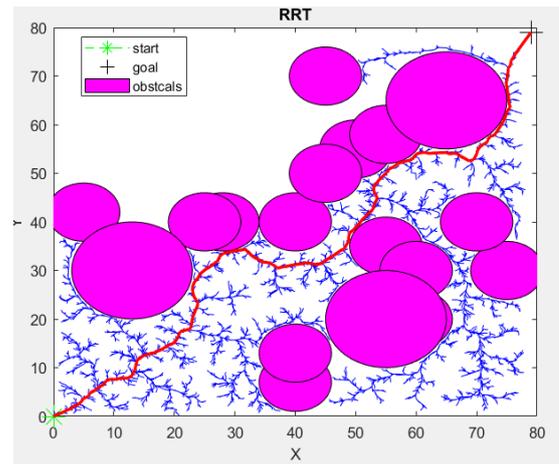
Here, the RRT* method is applied across different test environments, and the planning results for implementing the RRT* algorithm are shown in Fig. 4. Specifically, In the first environment, the obstacles are in the form of fixed circles, the starting point is (1,69), and the target point is (20,3), as shown in (Fig. 4a). After 10 test simulations, the mean path length was to reach the target 117.46m, 2767 iterations, and 3.8735 seconds. The second environment has static obstacles that produce a path starting from point (5,75) to the goal (75,5) as depicted in (Fig. 4b) with a mean path length of 122.5366m, 3025 iterations, and 5.5841 seconds. The third static environment produces a path from the start point (0,0) to (79,79) is illustrated in (Fig. 4c), in which the mean path length is 132.5263m with 9266 iterations and took 8.6537 seconds to accomplish the target. Finally, the RRT* algorithm was applied in the dynamic maze environment where two moving circles were in the maze, as shown in green. The path planning process in this environment produced a path from the starting point (12,12) to (68,68), as can be seen in (Fig. 4d). The results depicted that the path length values, number of iterations, and time taken were 93.87m, 2468, and 17.0561seconds, respectively.



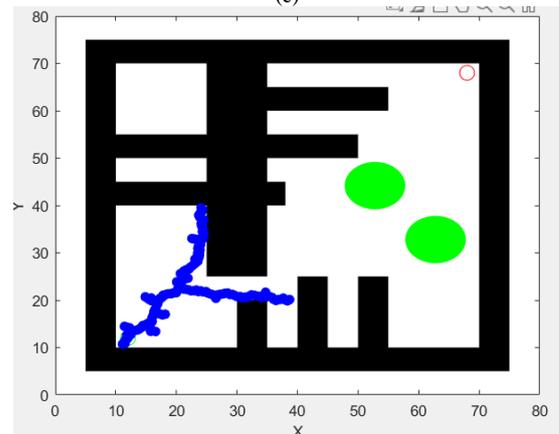
(a)



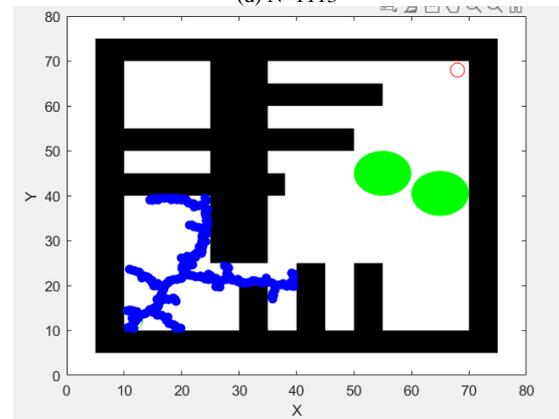
(b)



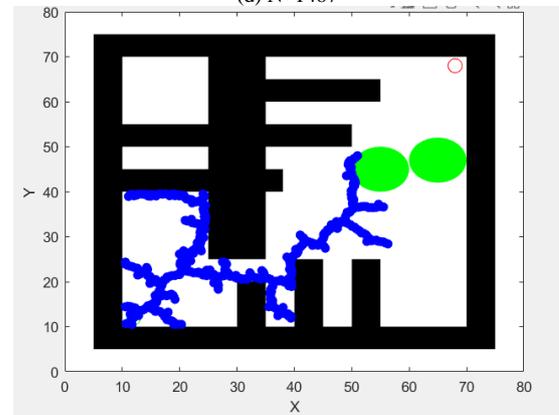
(c)



(d) N=1113



(d) N=1467



(d) N=2177

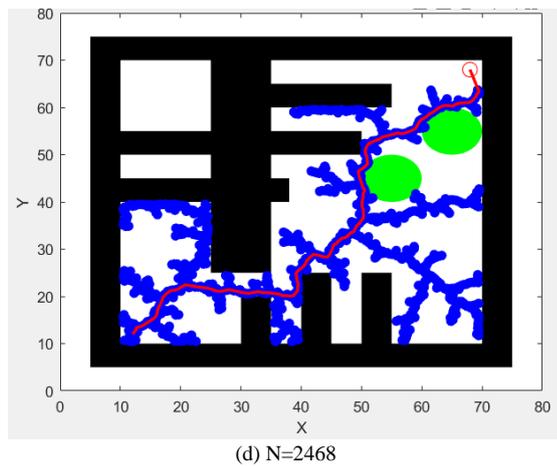
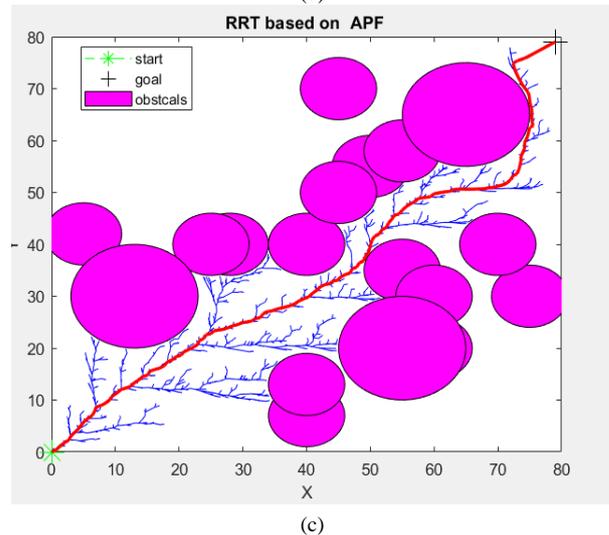
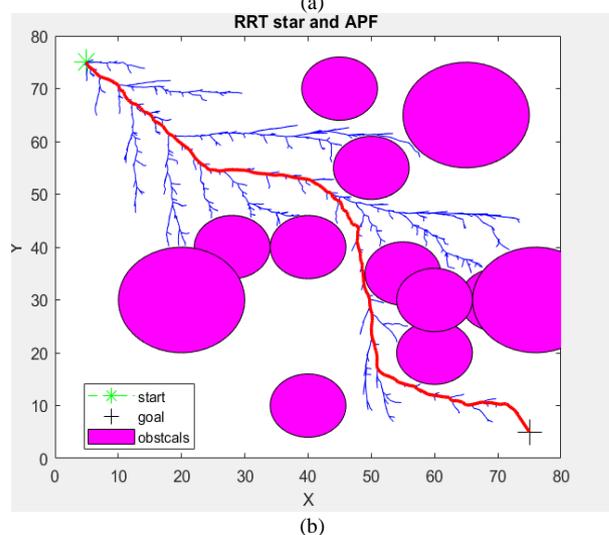
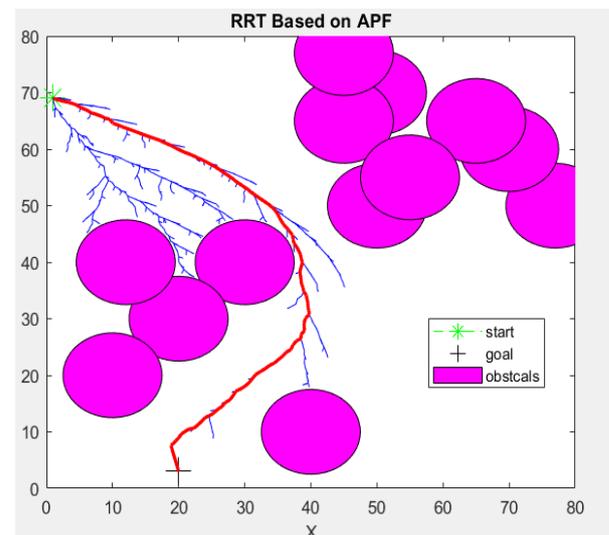


Fig. 4. Evaluation and Performance Analysis of the RRT* Method in Complex Environments with static obstacles and dynamic environment: (a) the first environment, which shows the circles obstacles with purple color, start point in green star, goal point in black plus sign, and the final path in red color. (b) the second environment shows the circles' obstacles with purple color, the start point in green star, the goal point in a black plus sign, and the final path in red color. (c) the third environment, which shows the circles obstacles with purple color, the start point in green star, the goal point in a black plus sign, and the final path in red color, and (d) the fourth environment, which is the black maze obstacles with two dynamic circle obstacles in green color.

2) Simulation Results of the RRT* based on the APF Method

This section presents the results of applying the APF-RRT* algorithm using the same four previous environments. The planning results of implementing the APF-RRT* algorithm are shown in Fig. 5. In the first environment and after the simulation of 10 training iterations, the mean path length is 96.25m with 1920 iterations and took 1.5155 seconds, as depicted in Fig. 5(a). which shows an improvement compared to the previous algorithm by 18.05% in terms of length, 60.78% in computation time, and 30.58% in terms of the number of iterations. The second environment results indicate that the average path length after ten simulations is 112.41m with 2556 iterations and 1.327 seconds, as shown in Fig. 5(b), which exhibits 8.26% enhancement over the RRT* algorithm in terms of the path length, 76.04% in the time and 15.46% in the number of search attempts. The third environment results show an improvement of 4.133%, 70.5%, and 51.57% in the length, computation time, and expansion steps, as shown in Fig. 5 (c). The mean lengths resulting from ten simulations are 127.048m with 4487 iterations and a computation time of 2.7554 seconds. The last dynamic environment produced a path with an average length, number of iterations used, and time taken of about 89.387m, 1053, and 10.8121 seconds, respectively. This method showed an improvement of 4.4%, 12.8%, and 39.4% in each path length, number of searches, and computation time.



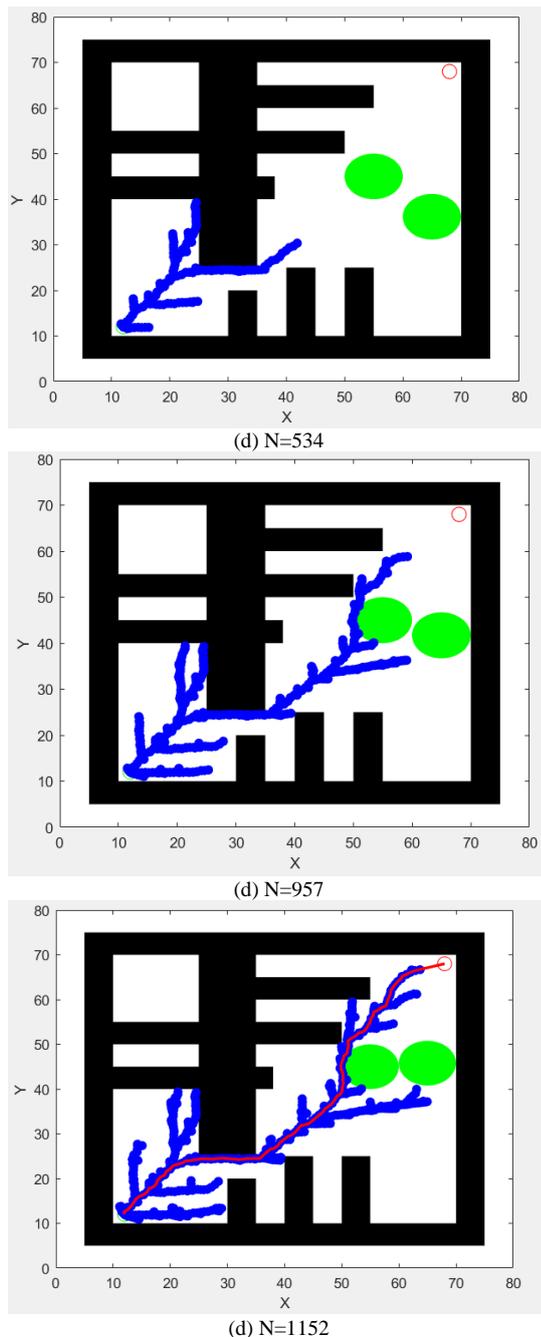


Fig. 5. Evaluation and Performance Analysis of the APF-RRT* Method in Complex Environments with static obstacles and dynamic environment: (a) the first environment, which shows the circles obstacles with purple color, the start point in green star, goal point in a black plus sign, and the final path in red color, (b) the second environment, shows the circles' obstacles with purple color, the start point in green star, the goal point in a black plus sign, and the final path in red color, (c) the third environment, which shows the circles obstacles with purple color, the start point in green star, the goal point in a black plus sign, and the final path in red color, and (d) the fourth environment, which is the black maze obstacles with two dynamic circle obstacles in green color

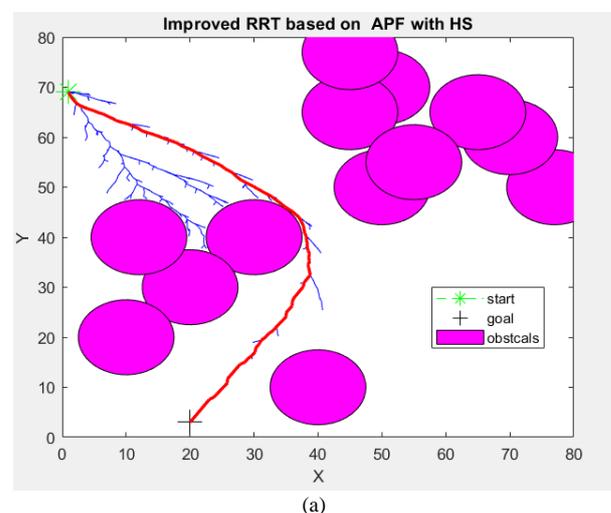
3) Simulation Results of the IRRT* based on APF with the Halton Sequence Method

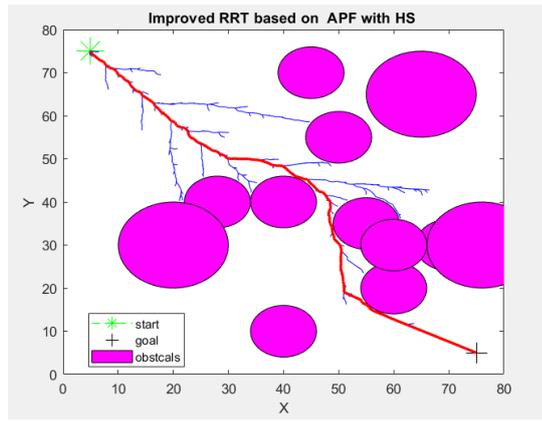
a) Initial Path Generation (Non-Smoothed Path)

The results of the proposed process are shown in this section for the same previous test environments to illustrate the efficiency of the suggested algorithm, which can be seen

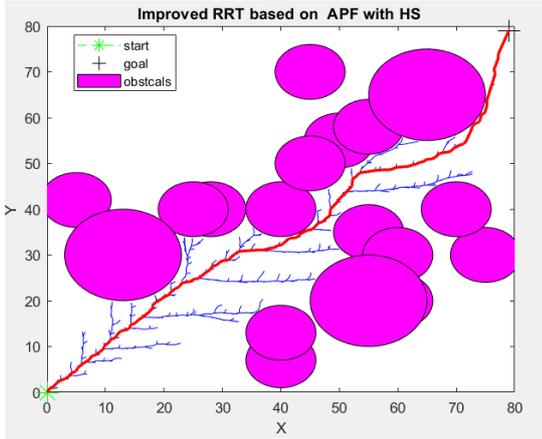
in all planning results of APF-RRT*-HS in Fig. 6. Fig. 6(a) shows the mean final value after 10 simulations of the path in the first test environment utilizing the APF-IRRT*-HS algorithm regarding the path length and the number of search attempts. Specifically, the results indicate that the final length is 92.36m, which is 21.36% better than that of the RRT* method and 3.76% better than that of the APF-RRT* method. At the same time, the search average number reaches 1628 iterations and took 0.97 seconds to reach the target, which refines search numbers at the rates of 55.61% and 36.04% compared to the RRT* and the APF-RRT* algorithms, respectively, the computation time average address reach 80.49% and 50.33% compared with previous methods. As illustrated in Fig. 6(b), which presents the final path length in the second test environment, the outcome of the path length showed that the final length value is 109.7m, which is considered an improvement of 10.4% compared to the RRT* method and 2.4% compared to the APF-RRT* algorithm.

On the other hand, the results of the iterations show 1428 iterations and took 0.71785 seconds, which is 52.79% superior to the RRT* algorithm and 44.13% superior to the APF-RRT* algorithm in the numbers of iterations and 87.14%, 46.74% for the percentage of enhancement in time. The results of the final path, when tested in the third test environment, are shown in Fig. 6(c), which shows that the path length value is 123.086m with 2975 iterations and took 1.0665 seconds. This outcome means that the length has differed by 7.321% and 3.11% compared to the RRT* and the APF-RRT* methods, respectively. The percentage of difference in the number of iterations considering the proposed algorithm is 67.89% compared to the RRT* approach and 33.69% compared to the APF-RRT* approach. The improvement rate in elapsed time is 87.64% and 61.28% compared to the previous methods. Finally, as shown in (Fig. 6(d)), the proposed algorithm is implemented in the dynamic environment, and the results show that the path length, elapsed time, and the number of iterations were 85.5827 meters, 5.474 seconds, and 636 iterations, respectively. These results show that the algorithm has improved by 36.9% and 7.8% compared to RRT and APF in path length. It has been enhanced by 63.73% and 40.16% in elapsed time. Finally, the number of search iterations improved by 53.3% and 46.48% compared to the same methods.

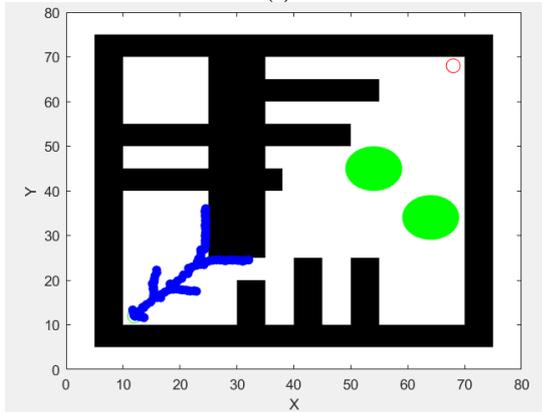




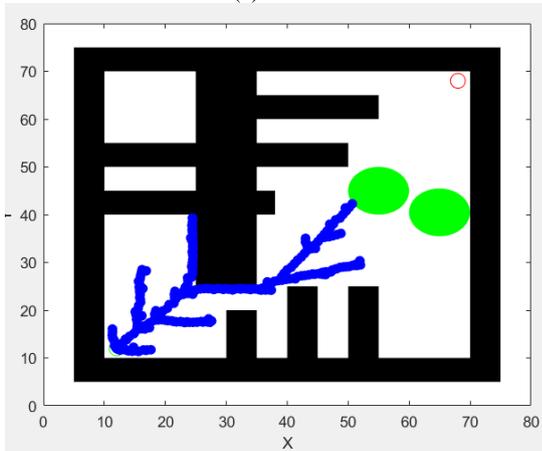
(b)



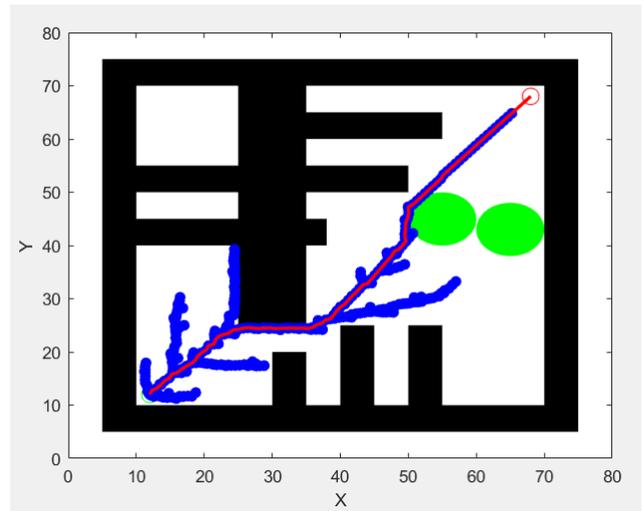
(c)



(d) N=254



(d) N=478



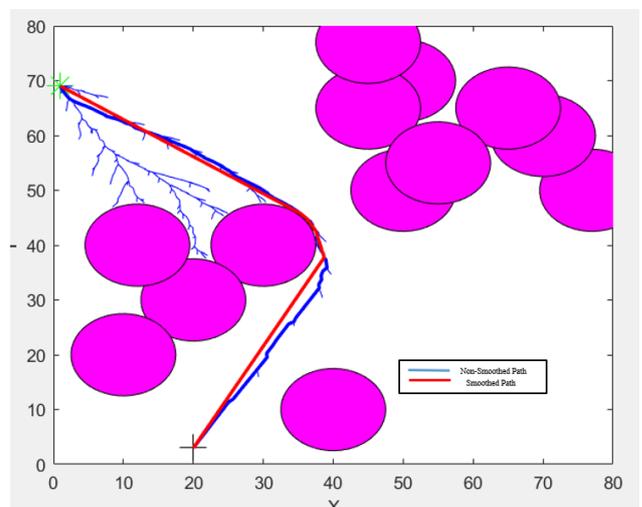
(d) N=636

Fig. 6. Evaluation and Performance Analysis of the APF-IRRT*-HS with smoothing Method in Complex Environments with static obstacles and dynamic environment : (a) the first environment, which shows the circles obstacles with purple color, the start point in green star, goal point in a black plus sign, and the final path in red color, (b) the second environment, shows the circles' obstacles with purple color, the start point in green star, the goal point in a black plus sign, and the final path in red color, (c) the third environment, which shows the circles obstacles with purple color, the start point in green star, the goal point in a black plus sign, and the final path in red color, and (d) the fourth environment, which is the black maze obstacles with two dynamic circle obstacles in green color.

b) Path Smoothing (Smoothed Path)

After creating the path using the proposed APF-IRRT*-HS method, a smoothing stage was added to solve problems that may negatively affect the system performance, such as sharp corner challenges or sudden changes in successive points. We used the spline smoothing method, which removes unnecessary points in the path, making the path smoother and more efficient.

The results of using the smoothing path on the path are shown in Fig. 7. As you can see, the smooth path was applied to the previous four environments and is compared with the path before the smoothing process. The two paths are displayed together to show the size of the differences.



(a)

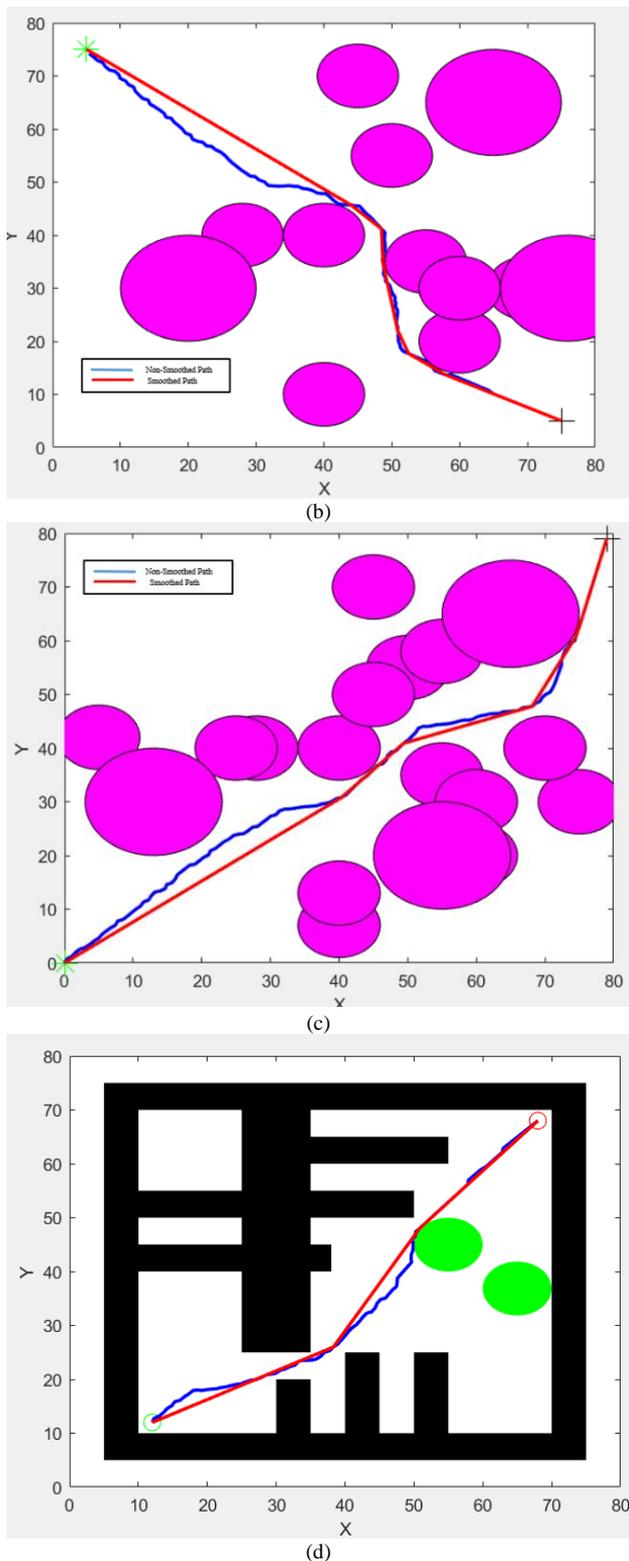


Fig. 7. Comparison of the APF-IRRT*-HS path before and after Smoothing: (a) the first environment, which shows the circles obstacles with purple color, the start point in green star, goal point in black plus sign, and the final path in red color. (b) the second environment shows the circles' obstacles with purple color, the start point in green star, the goal point in a black plus sign, and the final path in red color. (c) the third environment, which shows the circles obstacles with purple color, the start point in green star, the goal point in a black plus sign, and the final path in red color, and (d) the fourth environment, which is the black maze obstacles with two dynamic circle obstacles in red color.

4) Discussion of Results

In all test environments, the path length, computation time, and the number of search attempts were calculated, and the proposed method produced the shortest path and had the least number of iterations compared with the RRT* and the APF-RRT*. Table III shows the path planning results for the three algorithms for path length, computation time, and number of iterations.

TABLE III. THE PATH LENGTH IN METERS, COMPUTATION TIME IN SECONDS, AND THE SEARCH ATTEMPTS (N) COMPARISON

Environment	Methods	Path Length (meters)	Time (seconds)	search attempts
E1	RRT*	117.46m	3.8735s	2767
	APF-RRT*	96.25m	1.5155s	1920
	APF-IRRT*-HS	89.7376m	0.7557s	1228
E2	RRT*	122.5366m	5.5841s	3025
	APF-RRT*	112.41m	1.327s	2556
	APF-IRRT*-HS	104.264m	0.71785s	1428
E3	RRT*	132.5263m	8.6537s	9266
	APF-RRT*	127.048m	2.7554s	4487
	APF-IRRT*-HS	116.4933m	1.347s	2975
E4	RRT*	92.98m	17.834s	1468
	APF-RRT*	88.81m	10.8121s	1280
	APF-IRRT*-HS	81.903m	6.474s	685

The improvement percentages in the path length, the number of iterations, and time elapsed in Tables 4 and 5 illustrate the improved path when applying the proposed method, namely, the APF-IRRT*-HS. Moreover, the average enhancement percentages between the RRT* method and the APF-IRRT*-HS method are about 21.88%, 79.75%, and 57.39% in the path length, computation time, and the number of iterations, respectively.

The average improvement percentages in the path length, computation time, and the number of iterations are about 7.5%, 49.2%, and 40.085%, respectively, between the APF-RRT* method and the APF-IRRT*-HS method. This improvement could be significant in various applications that demand low search and length ratios in the path-planning process.

TABLE IV. THE REFINEMENT PERCENTAGE IN THE PATH LENGTH (L) AND THE SEARCH ATTEMPTS (N)

Environment	APF-RRT* VS RRT*		APF-IRRT*-HS VS RRT*		APF-IRRT*-HS VS APF-RRT*	
	L (%)	N (%)	L (%)	N (%)	L (%)	N (%)
E1	18.05	30.58	21.36	33.74	3.76	4.54
E2	8.26	15.46	10.4	45.77	2.4	35.855
E3	4.133	51.57	7.123	71.07	3.11	40.26
E4	4.4	12.8	36.9	53.3	7.8	46.48

TABLE V. THE REFINEMENT PERCENTAGE IN COMPUTATION TIME (T)

Environment	APF-RRT* VS RRT*	APF-IRRT*- HS VS RRT	APF-IRRT*-HS VS APF-RRT*
	T	T	T
E1	60.78%	80.49%	50.33%
E2	76.04%	87.14%	46.74%
E3	70.5%	87.64%	61.28%
E4	39.4%	63.73%	40.16%

Finally, the effect of adding smoothing to the proposed algorithm is explained in both Table VI and Table VII, where the results show an improvement in path length of up to about 5%, in the number of iterations to a percentage of up to about 20%, and in the time taken to a rate of up to 15%. These improvements are considered an additional advantage of the proposed method. They can be used in several daily robotics applications, especially in applications that depend on short path lengths to avoid delays and passing through unimportant paths.

TABLE VI. COMPARISON OF SMOOTHED AND NON-SMOOTHED PATHS IN TERMS OF THE PATH LENGTH IN METERS, COMPUTATION TIME IN SECONDS, AND THE SEARCH ATTEMPTS (N) COMPARISON

Environment	Methods	Path length (L)	Number of iteration	Time (S)
E1	Non-Smoothed Path	92.36m	1833	0.923s
	Smoothed Path	89.7376m	1228	0.7557s
E2	Non-Smoothed Path	109.7m	1640	0.837s
	Smoothed Path	104.264m	1428	0.7178s
E3	Non-Smoothed Path	123.086m	3146	1.347s
	Smoothed Path	116.4933m	2975	1.0665s
E4	Non-Smoothed Path	85.5827m	636	6.574s
	Smoothed Path	81.9035m	456	5.474s

TABLE VII. THE REFINEMENT PERCENTAGE IN THE PATH LENGTH (L), THE SEARCH ATTEMPTS (N), AND TIME (T)

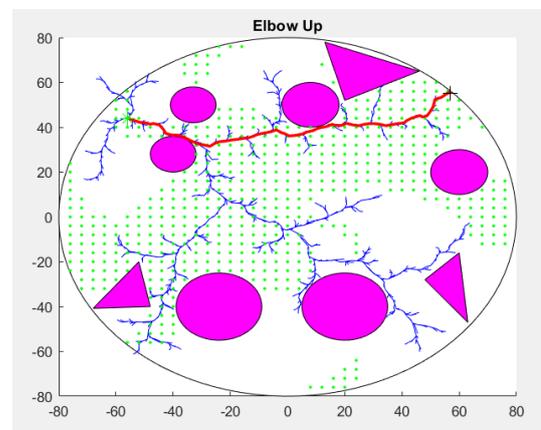
Environment	Non-Smoothed Path Vs Smoothed Path		
	L	N	T
E1	2.84 %	33%	18.12 %
E2	4.96%	12.92%	11.92%
E3	5.359%	5.43%	20.82%
E4	7.8%	28.3%	16.73%

B. The Two-Link Robot Simulation Results and Discussions

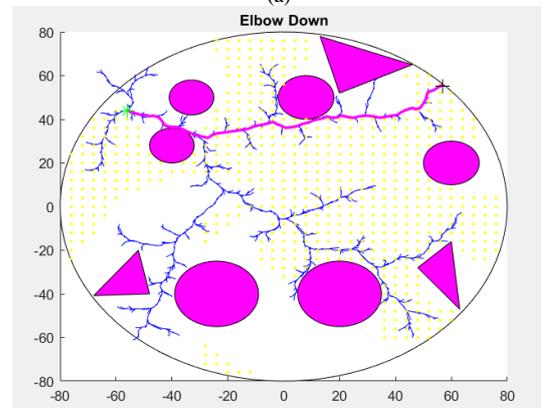
Like the mass point simulation, a two-joint robot will utilize the previous three techniques. The algorithms will be applied in three environments to test the efficiency of the proposed algorithm. Specifically, each link has a length of 40 meters, which is half the total size of the environment, and the robot is capable of full 360-degree rotation.

1) Simulation Results of the RRT*

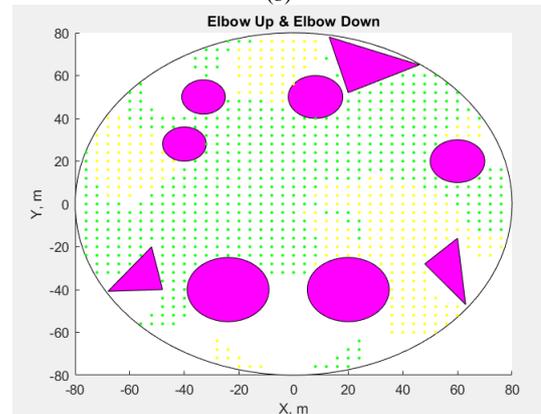
The RRT* path's free space in environment one is represented in Fig. 8, highlighting the elbow-up posture in (a), the elbow-down posture in (b), both configurations in (c), and the arm's motion in (d). The free space of this environment displays only the elbow-up configuration that can solve this task, the situation in which the arm takes an upward path to reach the target point known. When applied to more extensive and complex environments, the RRT* algorithm results in the robot failing to achieve the goal, becoming trapped in a local minimum, as illustrated in Fig. 9 the second environment (a) the first solution elbow-up (b) elbow-down solution (c) elbow-up and elbow-down solution, and finally (d) showing the two-link motion. The third environment is shown in Fig. 10, (a) showing the first solution elbow-up (b) elbow-down solution (c) elbow-up and elbow-down, and finally (d) showing the two-link motion.



(a)



(b)



(c)

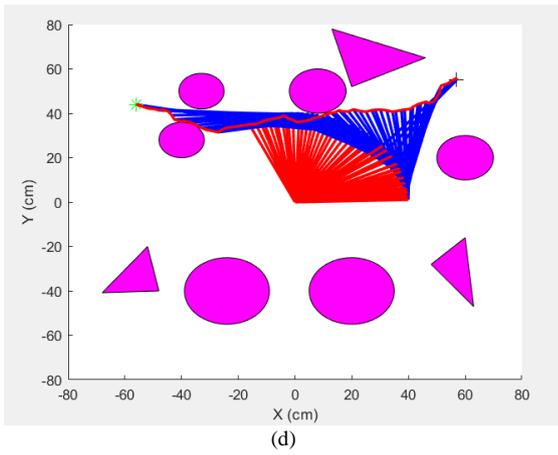


Fig. 8. Applying RRT* in the first environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion.

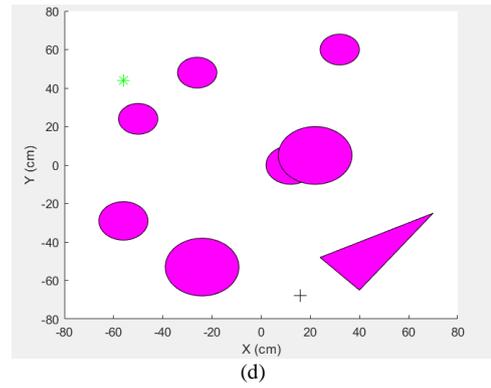
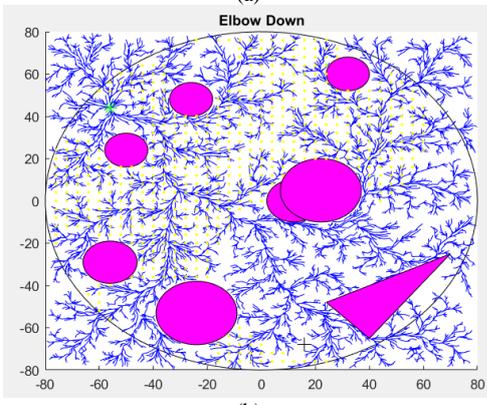
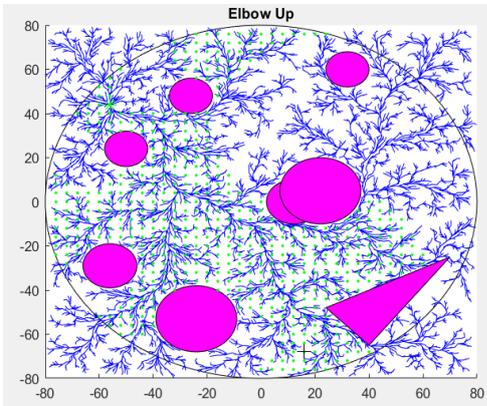
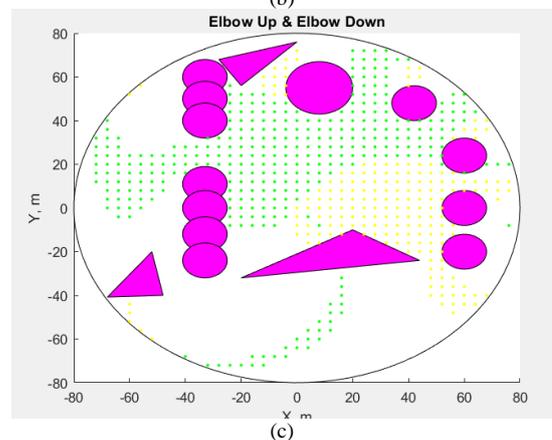
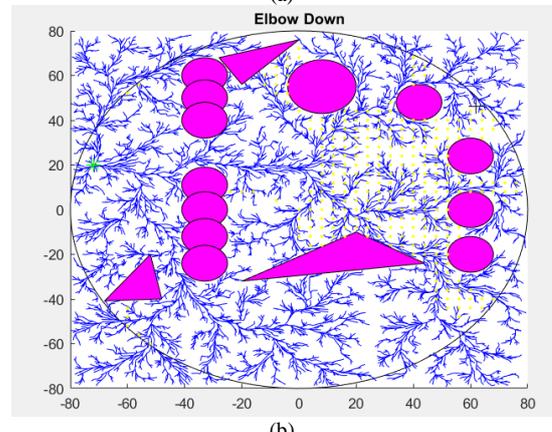
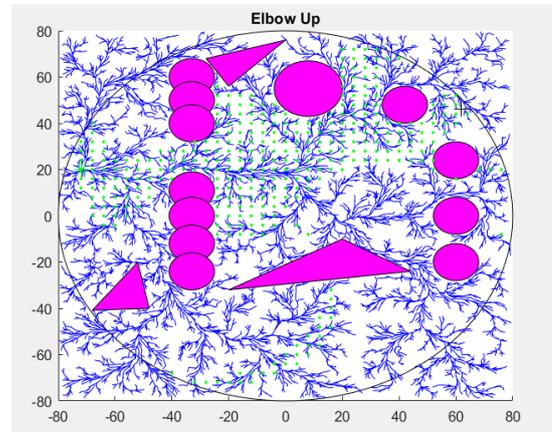


Fig. 9. Applying RRT* in the second environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion



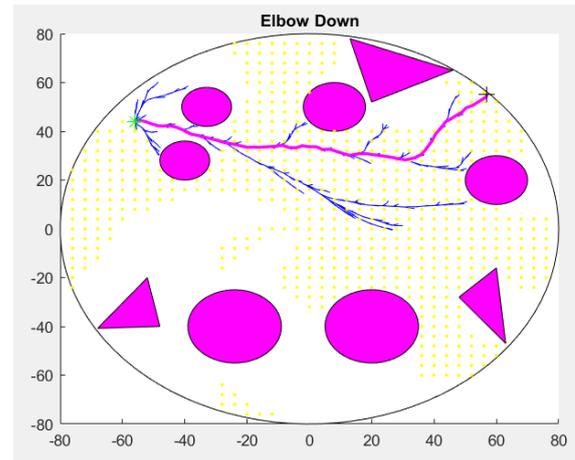
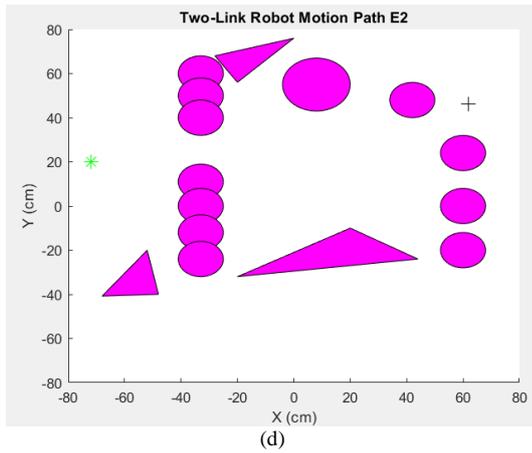


Fig. 10. Applying RRT* in the third environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion

2) Simulation Results of the RRT* based on the APF Method

The APF-RRT* algorithm was applied in the same previous environments. It is noted that the algorithm successfully reached the target in all test environments, with all environments executed using the elbow-up method only. The results of applying the APF-RRT* algorithm in the first environment are shown in Fig. 11, (a) showing the first solution elbow-up, (b) elbow-down solution, (c) elbow-up and elbow down, and finally (d) showing the two-link motion. The second environment is shown in Fig. 12, (a) showing the first solution elbow-up, (b) elbow-down solution (c) elbow-up and elbow-down, and finally, (d) showing the two-link motion), and the third environment is shown in Fig 13, (a) showing the first solution elbow-up, (b) elbow-down solution, (c) elbow-up and elbow-down, and finally (d) showing the two-link motion.

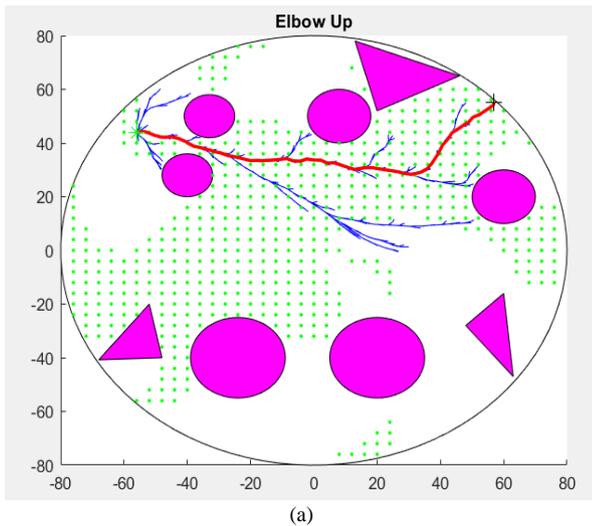
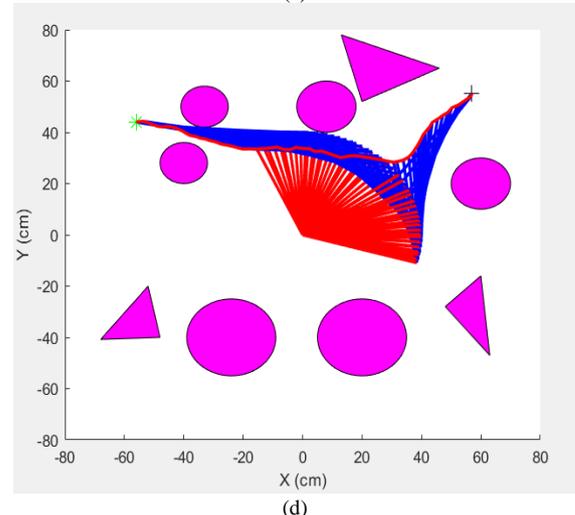
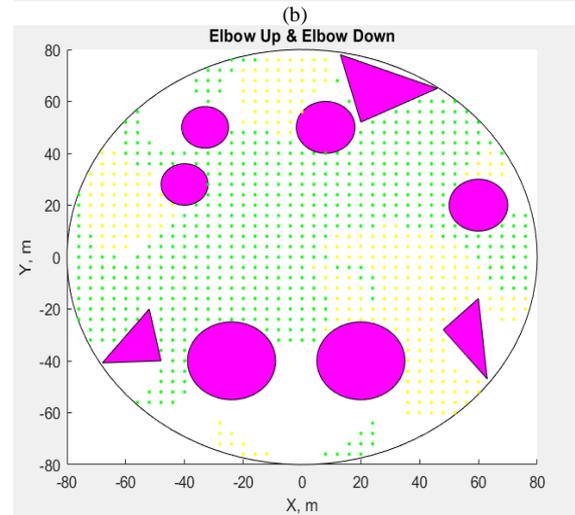


Fig. 11. Applying APF-RRT* in the first environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion

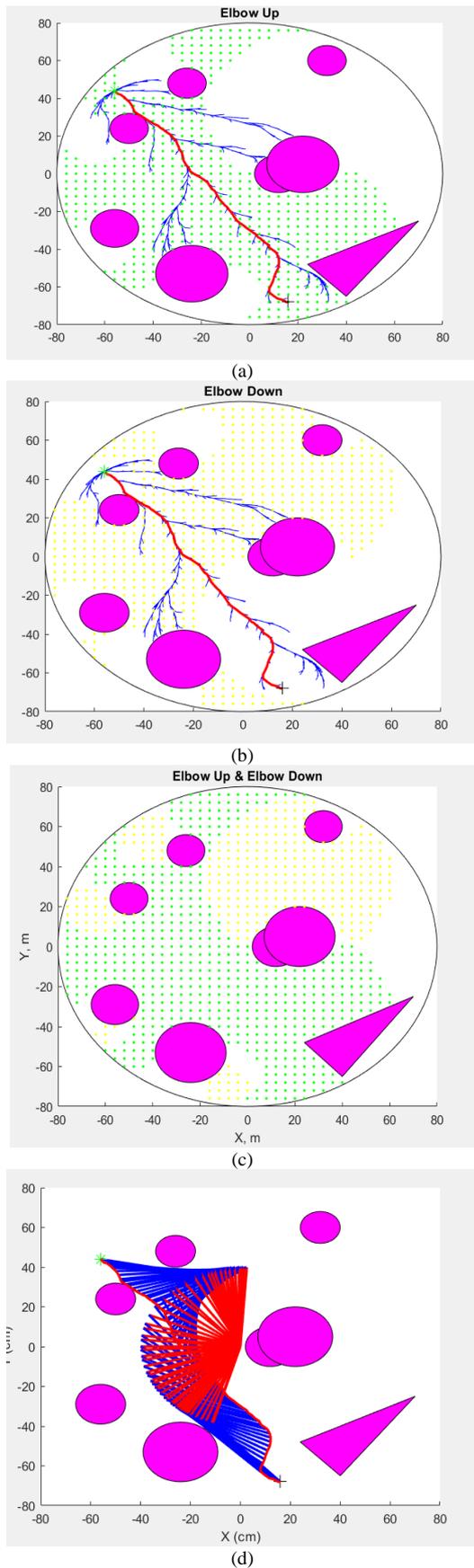


Fig. 12. Applying APF-RRT* in the second environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion

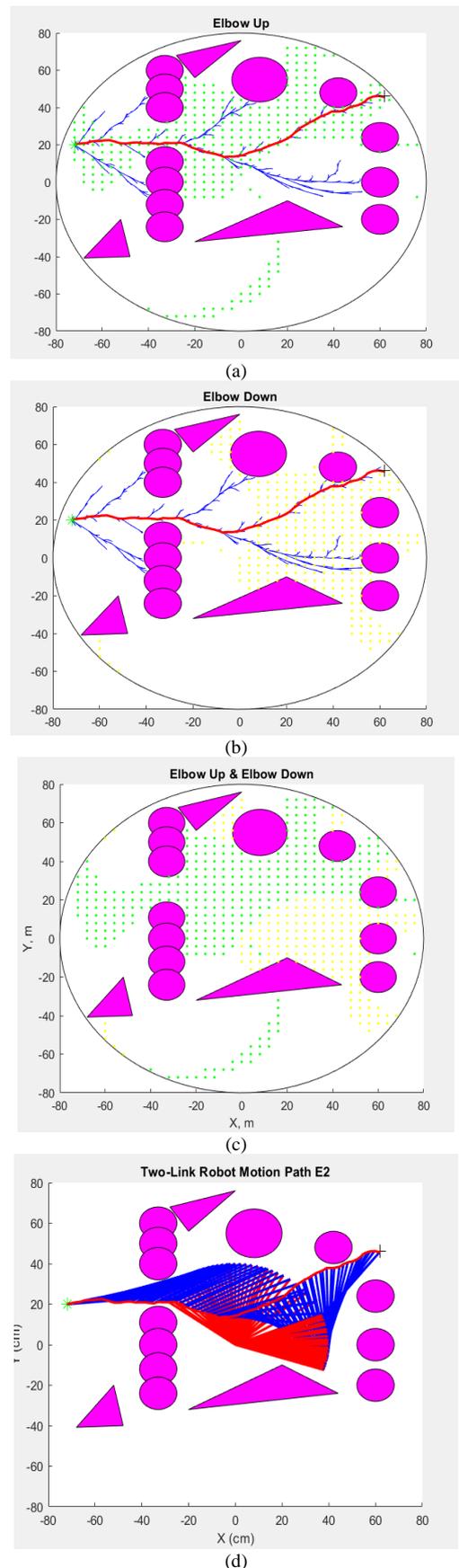


Fig. 13. Applying APF-RRT* in the third environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion

3) *Simulation Results of the IRRT* based on APF with the Halton Sequence Method*

Here, the two-link robot simulation is presented using the proposed APF-IRRT*-HS method in the same three environments used for the previous algorithms. As with the previous algorithm, all paths successfully reached the target. However, this time, the paths were shorter and were executed faster. The free space indicates that the algorithm operates solely in the elbow-up area in all environments. The results of applying the APF-IRRT*-HS algorithm in the first environment are presented in Fig. 14, (a) elbow-up solution, (b) elbow-down solution, (c) elbow-up and elbow-down solution, and finally (d) showing the two-link motion. The second environment is shown in Fig. 15, (a) showing the first solution elbow-up, (b) elbow-down solution, (c) elbow-up and elbow-down, and finally (d) showing the two-link motion), and the third environment is shown in Fig. 16, (a) showing the first solution elbow-up, (b) elbow-down solution, (c) elbow-up and elbow-down, and finally (d) showing the two-link motion.

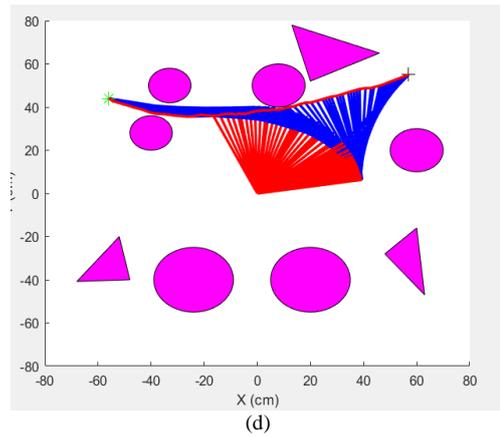
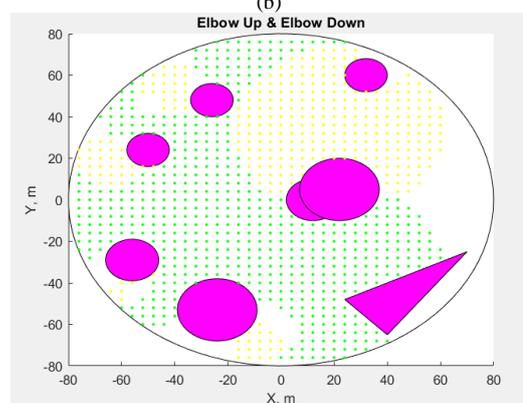
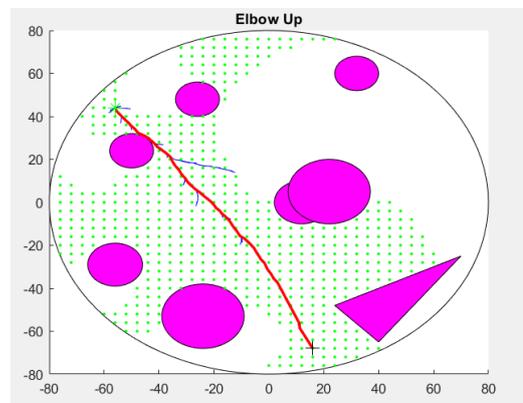
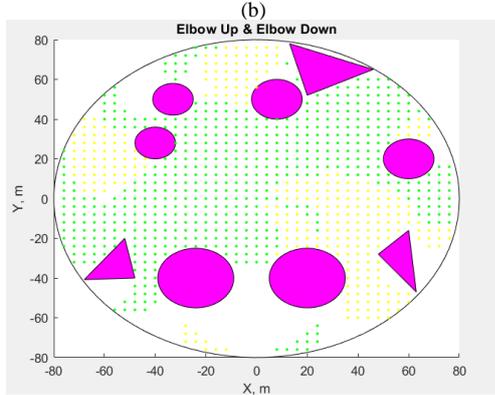
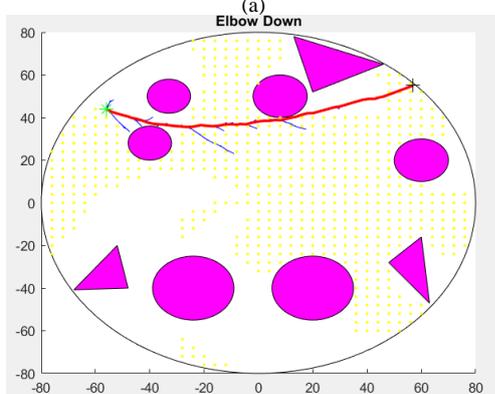
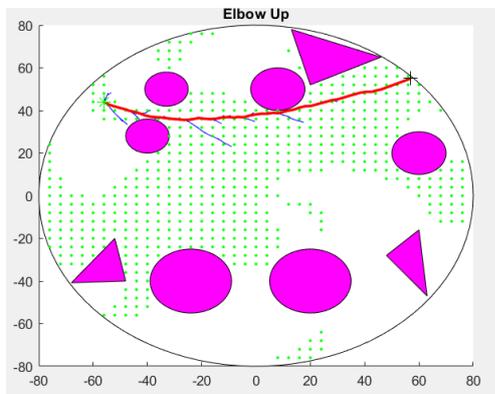


Fig. 14. Applying APF-IRRT*-HS in the first environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion



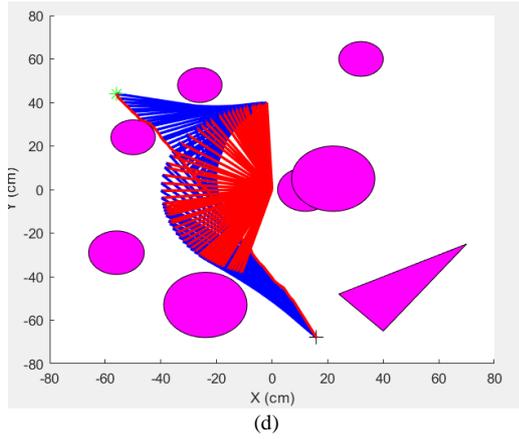


Fig. 15. Applying APF-RRT*-HS in the first environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion

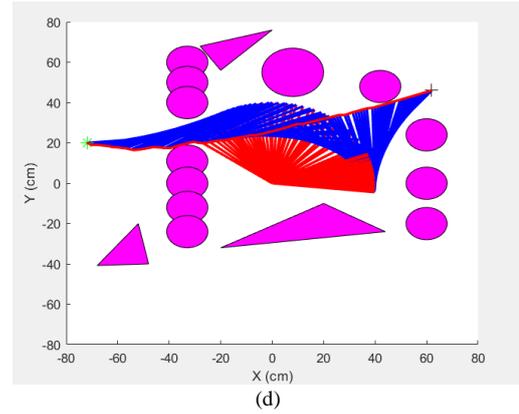


Fig. 16. Applying APF-RRT*-HS in the third environment (a) Workspace analysis with various obstacle shapes relied on the elbow-up solution (b) Workspace analysis with various obstacle shapes relied on the elbow-down solution (c) Workspace analysis with various obstacle shapes relied on elbow-up and elbow-down solution (c) two-link motion

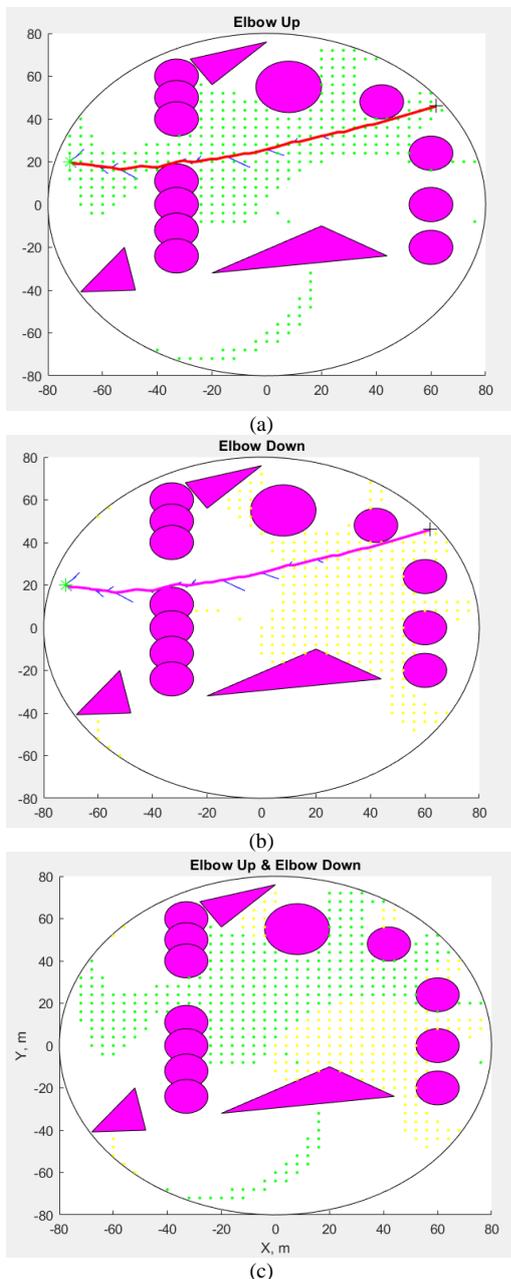


Table VIII and Table IX present the path length (L), number of iterations (N), and time elapsed (T) comparison between the RRT*, APF-RRT*, and proposed approach APF-IRRT*-HS in the three test environments (E). The percentage of improvement in path length (L), search attempts (T), and time elapsed (T) can be observed in Table X and Table XI.

In this part, the RRT* algorithm and the RRT*-APF algorithm are applied in addition to the proposed APF-IRRT*-HS algorithm in the two-link motion process. The outcomes are demonstrated and contrasted to show the enhancement demonstrated by the proposed method. The comparison considers the path length and the number of iterations. In particular, the results revealed an improvement in the proposed algorithm by percentages of up to 69.5% and 94.4% in terms of the length and the number of iterations, respectively, compared to the RRT* algorithm. Moreover, the results show a decrease in the path length and the number of iterations by approximately 5.6% and 82.5%, respectively, compared to the RRT*-APF algorithm. Finally, suggesting areas for future research, such as integrating learning-based techniques (e.g., reinforcement learning) or multi-robot systems, could offer opportunities for further improvement and broader algorithm applicability.

TABLE VIII. THE PATH LENGTH (L) AND THE SEARCH ATTEMPTS (N) COMPARISON

Environment	RRT*		APF-RRT*		APF-IRRT*-HS	
	L (m)	N	L (m)	N	L (m)	N
E1	129.33	1074	126.42	776	118.08	180
E2	No Path	No Path	143.678	1118	135.08	194
E3	No Path	No Path	146.06	1675	139.619	198

TABLE IX. THE COMPUTATION TIME IN SECONDS (T)

Environment	RRT*	APF-RRT*	APF-IRRT*-HS
	T	T	T
E1	2.4376s	0.75188s	0.1263s
E2	No path	2.0156s	0.16071s
E3	No Path	9.6143s	0.1866s

TABLE X. THE ENHANCEMENT PERCENTAGE IN THE PATH LENGTH (L) AND THE SEARCH ATTEMPTS (N)

Environment	APF-RRT* VS RRT*		APF-IRRT*-HS VS RRT		APF-IRRT*-HS VS APF-RRT*	
	L (%)	N (%)	L (%)	N (%)	L (%)	N (%)
E1	2.2	27.74	8.7	83.24	6.59	76.8
E2	100	100	100	100	5.9	82.64
E3	100	100	100	100	4.4	88.17

TABLE XI. THE ENHANCEMENT PERCENTAGE IN THE COMPUTATION TIME (T)

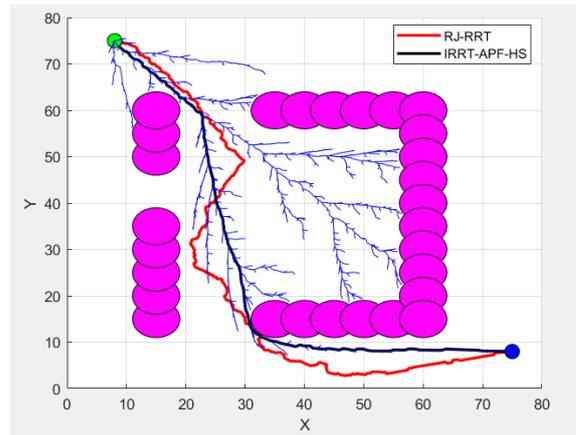
Environment	APF-RRT* VS RRT*	APF-IRRT*-HS VS RRT	APF-IRRT*-HS VS APF-RRT*
	T	T	T
E1	69.15%	94.48%	83.2%
E2	100%	100%	92.4%
E3	100%	100%	98%

V. COMPARISON WITH RELATED METHODS

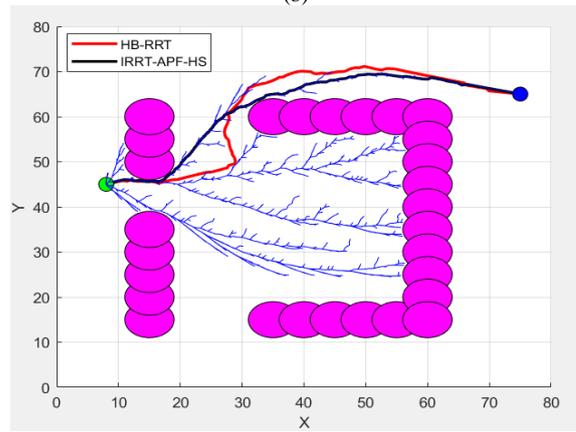
The proposed APF-IRRT*-HS method is compared with the Reduce-Judge RRT (RJ-RRT) [60], a Halton Biased Rapidly exploring Random Tree HB-RRT [48], and the APF-IRRT*[19], to ensure greater efficiency. RJ-RRT reduces configuration space and redundant nodes using environmental judgment techniques and reducing the sampling space. However, it faces problems such as expanded size and additional complexity. The HB-RRT algorithm is based on sampling using the Halton sequence algorithm and enhances it using the (Goal-Oriented Strategy). However, it suffers from difficulty in detection in dense obstacle environments. The APF-IRRT* algorithm combines the APF and the RRT* algorithm to guide the robot towards the goal and avoid obstacles. However, it suffers from oscillation and local minima problems.

As presented in Fig. 17, the proposed algorithm achieved the shortest path length in less time among all algorithms. The red path illustrates the proposed algorithm in both environments, while the blue path illustrates the other methods.

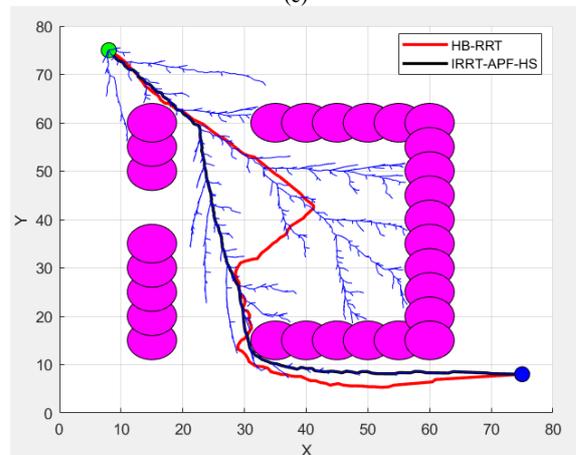
Table XII, Table XIII, and Table XIV present the path length (L), the number of iterations (N), and time elapsed (T) comparison between the three methods and the proposed approach APF-IRRT*-Hs in the two validation environments (E).



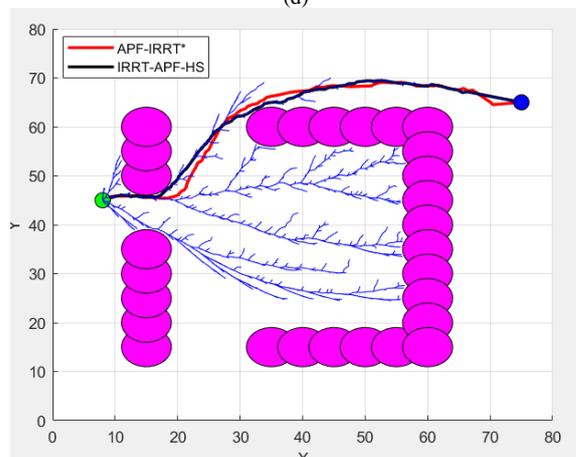
(b)



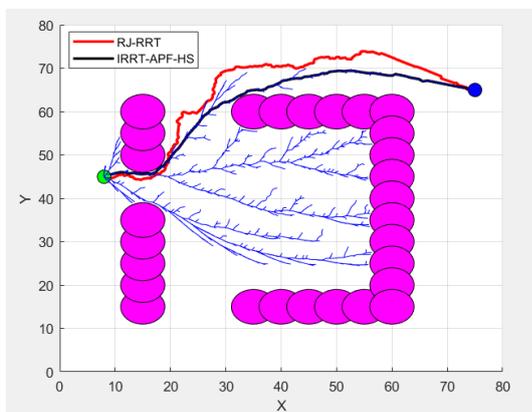
(c)



(d)



(e)



(a)

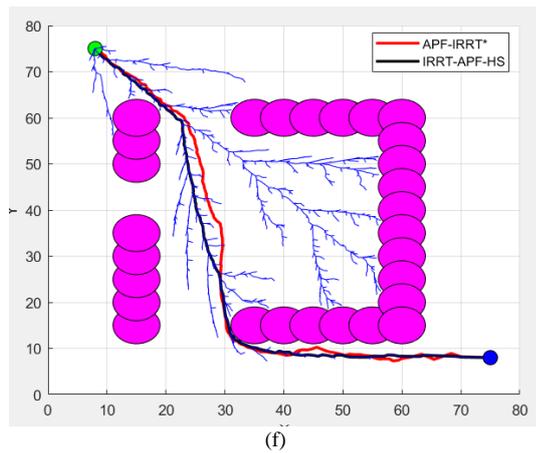


Fig. 17. Comparing the path results in two testing environments: (a)E.1 RJ-RRT X APF-IRRT*-HS, (b)E.2 RJ-RRT X APF-IRRT*-HS (c)E.1 HB-RRT X APF-IRRT*-HS, (d)E.2 HB-RRT APF-IRRT*-HS (e)E.1 APF-IRRT* X APF-IRRT*-HS, (e)E.2 APF-IRRT* X APF-IRRT*-HS

TABLE XII. COMPARISON OF THE PATH LENGTH IN METERS

Environment	Methods	Path length
E1	RJ-RRT	92.30m
	APF-IRRT*-HS	78.924m
E2	RJ-RRT	132.50m
	APF-IRRT*-HS	116.0183m
E1	HB-RRT	86.8566m
	APF-IRRT*-HS	78.924m
E2	HB-RRT	125.7952m
	APF-IRRT*-HS	116.0183m
E1	APF-IRRT*	81.4172m
	APF-IRRT*-HS	78.924m
E2	APF-IRRT*	118.9797m
	APF-IRRT*-HS	116.0183m

TABLE XIII. REFINEMENT NUMBER OF SEARCH ATTEMPTS IN ITERATIONS

Environment	Methods	Number of iterations
E1	RJ-RRT	13145
	APF-IRRT*-HS	1995
E2	RJ-RRT	24128
	APF-IRRT*-HS	3092
E1	HB-RRT	11473
	APF-IRRT*-HS	1995
E2	HB-RRT	21435
	APF-IRRT*-HS	3092
E1	APF-IRRT*	2130
	APF-IRRT*-HS	1995
E2	APF-IRRT*	3453
	APF-IRRT*-HS	3092

TABLE XIV. TIME ELAPSED IN SECONDS

Environment	Methods	Time(s)
E1	RJ-RRT	1.768s
	APF-IRRT*-HS	0.453s
E2	RJ-RRT	2.8169s
	APF-IRRT*-HS	0.813s
E1	HB-RRT	1.532s
	APF-IRRT*-HS	0.453s
E2	HB-RRT	2.481s
	APF-IRRT*-HS	0.813s
E1	APF-IRRT*	0.5812s
	APF-IRRT*-HS	0.453s
E2	APF-IRRT*	0.989s
	APF-IRRT*-HS	0.813s

The proposed APF-IRRT*-HS method has established its effectiveness in all test environments and achieved satisfactory results. Remarkably, it accomplished the target with shorter lengths and significantly reduced iterations compared to RRT* and the APF-RRT* algorithms in complex environments. Moreover, the proposed method was compared with RJ-RRT, HB-RRT, and APF-IRRT* methods, and it constructed better paths than those of these approaches. The enhancement percentage in path length (L) and search attempts (N) is shown in Table XV. The time elapsed (T) percentage computation can be seen in Table XVI.

The robot needs to choose the shortest path and shortest time to save resources while moving between points, especially in applications of mobile robots, maps, vehicles, and drones.

TABLE XV. COMPARISON PERCENTAGE ENHANCEMENT IN THE PATH LENGTH (L) AND THE NUMBER OF ITERATIONS (N)

Environment	APF-IRRT*-HS Vs RJ-RRT		APF-IRRT*-HS Vs HB-RRT		APF-IRRT*-HS VS APF-IRRT*	
	L	N	L	N	L	N
E1	14.48%	84.81%	9.12%	82.61%	3.05%	6.34%
E2	12.44%	87.18%	7.7%	85.57%	2.48%	10.44%

TABLE XVI. COMPARISON PERCENTAGE ENHANCEMENT IN THE COMPUTATION TIME(T)

Environment	APF-IRRT*-HS Vs RJ-RRT	APF-IRRT*-HS Vs HB-RRT	APF-IRRT*-HS VS APF-IRRT*
	T	T	T
E1	74.54%	70.43%	22.1%
E2	71.73%	67.57%	17.7%

VI. CONCLUSION

In this work, the RRT* algorithm was developed by proposing an APF-IRRT*-HS algorithm that uses both the improved target rule and the Halton sequence to enhance the sampling process and reduce randomness, consequently establishing a short final path and exploring it quickly. The outcomes supported the potency of the proposed algorithm compared to the RRT*, the APF-RRT*, the RJ-RRT, HB-RRT, and the APF-IRRT* algorithms. This method has been implemented in two cases of mass point and two link robots in different dynamic environments. The proposed algorithm has shown decreased path length and the number of iterations, producing optimal paths in less time than other traditional methods. The results of the proposed APF-IRRT*-HS showed an improvement of approximately 13% and 4% in path length, 50.2% and 40% in elapsed time, and 5% and 7% in many iterations compared to RRT* and APF-RRT* algorithms, respectively. The proposed method balances finding and discovering new points that may lead to finding more efficient paths (exploration) and exploiting and improving previously discovered points (exploitation). The method can be used to improve the accuracy of obstacle avoidance in drones and autonomous vehicles. Navigation systems can be enhanced by improving the accuracy of maps

and updating them continuously to interact with the environment better. As a future direction, the proposed method can be applied in more complex dynamic moving environments and may be combined with improved integration with reinforcement learning techniques.

REFERENCES

- [1] D. K. Muhsen, A. T. Sadiq, and F. A. Raheem, "A Systematic Review of Rapidly Exploring Random Tree RRT Algorithm for Single and Multiple Robots," *Cybernetics and Information Technologies*, vol. 24, no. 1, 2024.
- [2] A. A. Kareem, B. K. Oleiwi, and M. J. Mohamed, "Planning the Optimal 3D Quadcopter Trajectory Using a Delivery System-Based Hybrid Algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 2, pp. 427–439, 2023.
- [3] F. N. Irzoqe, F. A. Raheem, and A. R. Nasser, "Path Planning Improvement Using a Modified Q-learning Algorithm Based on Artificial Potential Field," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 4, 2024.
- [4] Z. A. Ahmed and S. M. Raafat, "Enhanced Global Path Planning Efficiency by Bidirectional A*, Gradient Descent, and Orientation Interpolation Algorithms," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 5, 2023, doi: 10.22266/ijies2023.1031.39.
- [5] R. Kamil, M. Mohamed, and B. Oleiwi, "Path Planning of Mobile Robot Using Improved Artificial Bee Colony Algorithm," *Engineering and Technology Journal*, vol. 38, no. 9, pp. 1384–1395, Sep. 2020, doi: 10.30684/etj.v38i9a.1100.
- [6] A. A. Kareem, B. K. Oleiwi, and M. J. Mohamed, "Unmanned aerial vehicle path planning in a 3D environment using a hybrid algorithm," *Journal of Field Robotics*, vol. 13, no. 2, pp. 905–915, 2024.
- [7] X. Zhou, J. Yan, M. Yan, K. Mao, R. Yang, and W. Liu, "Path Planning of Rail-Mounted Logistics Robots Based on the Improved Dijkstra Algorithm," *Appl. Sci.*, vol. 13, p. 9955, 2023.
- [8] F. A. Raheem, A. T. Sadiq, and N. A. F. Abbas, "Ant colony algorithm improvement for robot arm path planning optimization based on D* strategy," *Int. J. Mech. Mechatron. Eng.*, vol. 21, pp. 96–111, 2021.
- [9] F. A. Raheem and M. Abdulkareem, "Development of A* Algorithm for Robot Path Planning Based on Modified Probabilistic Roadmap and Artificial Potential Field," *Journal of Engineering Science and Technology*, vol. 15, pp. 3034–3054, 2020.
- [10] B. Li and B. Chen, "An adaptive rapidly-exploring random tree," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 283–294, 2021.
- [11] F. Raheem and M. Abdulkareem, "Development of Path Planning Algorithm Using Probabilistic Roadmap Based on Ant Colony Optimization," *Engineering and Technology Journal*, vol. 38, no. 3A, pp. 343–351, Mar. 2020, doi: 10.30684/etj.v38i3a.389.
- [12] S. Rahman, S. Akter, and S. Yoon, "A Deep Q-Learning Based UAV Detouring Algorithm in a Constrained Wireless Sensor Network Environment," *Electronics*, vol. 14, no. 1, 2025, doi: 10.3390/electronics14010001.
- [13] O. Syzovov, S. Tomasiello, and N. Capuano, "New Insights into Fuzzy Genetic Algorithms for Optimization Problems," *Algorithms*, vol. 17, p. 549, 2024, doi: 10.3390/a17120549.
- [14] J. A. P. Ferreira and M. S. A. Teixeira, "Robust Linear Control for Autonomous Systems in Uncertain Environments," *IEEE Transactions on Automatic Control*, vol. 68, no. 10, pp. 5635–5647, 2023, doi: 10.1109/TAC.2023.3156754.
- [15] Y. Zhang and X. Liu, "Energy-Efficient Path Planning for UAVs Using Optimal Control," *Journal of Field Robotics*, vol. 41, no. 3, pp. 345–358, 2024, doi: 10.1002/rob.22317.
- [16] D. K. Muhsen, A. T. Sadiq, and F. A. Raheem, "Memorized Rapidly Exploring Random Tree Optimization (MRRTO): An Enhanced Algorithm for Robot Path Planning," *Cybernetics and Information Technologies*, vol. 24, no. 1, pp. 190–204, 2024.
- [17] K. Ran, Y. Wang, C. Fang, Q. Chai, X. Dong, and G. Liu, "Improved RRT* Path-Planning Algorithm Based on the Clothoid Curve for a Mobile Robot Under Kinematic Constraints," *Sensors*, vol. 24, p. 7812, 2024, doi: 10.3390/s24237812.
- [18] S. Davarzani and M. T. Ejaz, "A 2D path-planning performance comparison of RRT and RRT* for unmanned ground vehicles," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 13, no. 1, pp. 105–112, 2024.
- [19] D. Wu, L. Wei, G. Wang, L. Tian, and G. Dai, "APF-IRRT*: An improved informed rapidly-exploring random trees-star algorithm by introducing artificial potential field method for mobile robot path planning," *Appl. Sci.*, vol. 12, no. 21, p. 10905, Oct. 2022.
- [20] W. Burzynski, and W. Stecz, "Trajectory planning with multiplatform spacetime RRT*," *Appl. Intell.*, vol. 54, 9524–9541, 2024.
- [21] Y. Li, "Comparative Analysis of Hybrid A* and RRT* Algorithms an Transactions on Computer Science and Intelligent Systems Research," *IEEE Access*, vol. 8, pp. 18658–18668.
- [22] J. Liang, W. Luo, and Y. Qin, "Path Planning of Multi-Axis Robotic Arm Based on Improved RRT*," *CMC*, vol. 81, no. 1, 2024, doi: 10.32604/cmc.2024.055883.
- [23] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi, I. Ahmady, and I. Ali, "Informed RRT*-Connect: An Asymptotically Optimal Single-Query Path Planning Method," *IEEE Access*, vol. 8, 2020.
- [24] L. Keyu, L. Yonggen, and Z. Yanchi, "Dynamic obstacle avoidance path planning of UAV Based on improved APF," *Proceedings of the 2020 5th International Conference on Communication, Image and Signal Processing (CCISP)*, pp. 159–163, 2020.
- [25] Q. Diao, J. Zhang, M. Liu, and J. Yang, "A Disaster Relief UAV Path Planning Based on APF-IRRT* Fusion Algorithm," *Drones*, vol. 7, no. 5, p. 323, May 2023.
- [26] J. Gao, X. Zheng, P. Liu, P. Yang, J. Yu, and Y. Dai, "APF-IBRRT*: A Global Path Planning Algorithm for Obstacle Avoidance Robots With Improved Iterative Search Efficiency," in *IEEE Access*, vol. 12, pp. 124740–124750, 2024, doi: 10.1109/ACCESS.2024.3451616.
- [27] X. Jin, Z. Yan, H. Yang, Q. Wang, and G. Yin, "A Goal-Biased RRT Path Planning Approach for Autonomous Ground Vehicle," *Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, pp. 743–746, 2020.
- [28] E. I. Atanassov, "On the discrepancy of the Halton sequences," *Math. Balkanica (NS)*, vol. 18, no. 1–2, pp. 15–32, 2004.
- [29] Y. Zhao, K. Liu, G. Lu, Y. Hu, and S. Yuan, "Path Planning of UAV Delivery Based on Improved APF-IRRT* Algorithm," *J. Phys. Conf. Ser.*, vol. 1624, no. 4, p. 042004, 2020.
- [30] S. Huang, "Path planning based on mixed algorithm of RRT and artificial potential field method," in *2021 4th International Conference on Intelligent Robotics and Control Engineering (IRCE)*, pp. 149–155, 2021.
- [31] L. Wang, X. Yang, Z. L. Chen, and B. R. Wang, "Application of the improved rapidly exploring random tree algorithm to an insect-like mobile robot in a narrow environment," *Biomimetics*, vol. 8, no. 4, 2023.
- [32] F. N. Irzoqe, F. A. Raheem, and A. R. Nasser, "A New Approach for Path Planning Algorithm Utilizing Modified Deep Q-Network Combined with Artificial Potential Field," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 6, 2024.
- [33] D. K. Muhsen, A. T. Sadiq, and F. A. Raheem, "Improved rapidly exploring random tree using salp swarm algorithm," *Journal of Intelligent Systems*, vol. 33, p. 20230219, 2024.
- [34] X. Zhong, Z. Wei, and T. Chen, "Motion Planning and Pose Control for Flexible Spacecraft Using Enhanced LQR-RRT*," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 6, pp. 8743–8751, Dec. 2023, doi: 10.1109/TAES.2023.3309612.
- [35] W. Wu, "Research Process of Path Planning based on RRT Algorithm and Its Improvements," *Transactions on Computer Science and Intelligent Systems Research*, vol. 5, 2024.
- [36] X. Li, G. Li, and Z. Bian, "Research on autonomous vehicle path planning algorithm based on improved RRT* algorithm and artificial potential field method," *Sensors*, vol. 24, p. 3899, 2024.
- [37] X. Jiang, Z. Wang, and C. Dong, "A Path Planning Algorithm Based on Improved RRT Sampling Region," *Computers, Materials, and Continua CMC*, vol. 80, no. 3, 2024, doi: 10.32604/cmc.2024.054640.
- [38] Q. Li, J. Wang, H. Li, B. Wang, and C. Feng, "Fast-RRT*: an improved motion planner for mobile robot in two-dimensional space," *IEEJ Trans Electr Electron Eng*, vol. 17, pp. 200–208, 2022.

- [39] L. Li, W. Wu, Z. Li, and F. Wang, "Collision Avoidance Method for Unmanned Ships by Using a Modified APF Algorithm," *Research Square*, 2024.
- [40] M. M. Badr and F. A. Raheem, "Development of Modified path planning algorithm using artificial potential field (APF) based on PSO for factors optimization," *American Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 37, no. 1, 2017.
- [41] J. Gao, X. Xu, Q. Pu, P. B. Petrovic, A. Rodi'c, and Z. Wang, "A Hybrid Path Planning Method Based on Improved A* and CSA-APF Algorithms," *IEEE Access*, vol. 12, pp. 39139–39151, 2024.
- [42] F. A. Raheem and M. M. Badr, "Smoothed artificial potential field (apf) via points path planning algorithm based on pso," *AL-yarmouk Journal*, vol. 11, no. 1, 2019.
- [43] Y. Chen, G. Bai, Y. Zhan, X. Hu, and J. Liu, "Path Planning and Obstacle Avoiding of the USV Based on Improved ACO-APF Hybrid Algorithm with Adaptive Early-Warning," *IEEE Access*, vol. 9, pp. 36829–36839, 2021.
- [44] A. Ma'Arif, W. Rahmaniari, M. A. M. Vera, A. A. Nuryono, R. Majdoubi, and A. Çakan, "Artificial Potential Field Algorithm for Obstacle Avoidance in UAV Quadrotor for Dynamic Environment," *2021 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, pp. 184–189, 2021, doi: 10.1109/COMNETSAT53002.2021.9530803.
- [45] I. Suwarno, W. A. Oktaviani, Y. Apriani, D. U. Rijalusalam, and A. Pandey, "Potential Force Algorithm with Kinematic Control as Path Planning for Disinfection Robot," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, 2022, doi: 10.18196/jrc.v3i1.11528.
- [46] F. A. Raheem, S. M. Raafat, and S. M. Mahdi, "Robot Path-Planning Research Applications in Static and Dynamic Environments," *Earth Systems Protection and Sustainability*, vol. 1, pp. 291–325, 2022, doi: 10.1007/978-3-030-85829-2_12.
- [47] S. A. Nagy, M. A. El-Beltagy, and M. Wafa, "Multilevel Monte Carlo by Using the Halton Sequence," *Monte Carlo Methods and Applications*, vol. 26, no. 3, pp. 193–203, 2020, doi: 10.1515/mcma-2020-2065.
- [48] H. Zhong, M. Cong, M. Wang, Y. Du, and D. Liu, "HB-RRT: A path planning algorithm for mobile robots using Halton sequence-based rapidly-exploring random tree," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108362, 2024, doi: 10.1016/j.engappai.2024.108362.
- [49] W. H. Bangyal, K. Nisar, A. A. B. Ibrahim, M. R. Haque, J. J. Rodrigues, and D. B. Rawat, "Comparative Analysis of Low Discrepancy Sequence-Based Initialization Approaches Using Population-Based Algorithms for Solving the Global Optimization Problems," *Appl.Sci.*, vol. 11, p. 7591, 2021.
- [50] J. Wang and P. Wu, "An Improved Three-point Method for Power Flow Calculation Based on Halton Sequence," *Frontiers in Computing and Intelligent Systems*, vol. 4, no. 1, 2023.
- [51] N. Kirk and C. Lemieux, "An Improved Halton Sequence for Implementation in Quasi-Monte Carlo Methods," *arXiv:2405.15799v1*, 2024.
- [52] A. Kumar, A. Sahasrabudhe, C. Perugu, S. Nirgude, and A. Murugan, "Kinematics & Dynamics Library for Baxter Arm," *arXiv:2409.00867v1*, 2024.
- [53] A. N. Sharkawy, "Forward and inverse kinematics solution of a robotic manipulator using a multilayer feedforward neural network," *Journal of Mechanical and Energy Engineering*, vol. 6, no. 2, 2022.
- [54] M. Zavar, "Forward and Inverse Kinematics of 4-DoF SCARA: Using Optimization Algorithms," *Journal of Applied Dynamic Systems and Control*, vol. 6, no. 3, pp. 25–34, 2023.
- [55] R. V. N. Kumar and R. Sreenivasulu, "Inverse Kinematics (IK) Solution of a Robotic Manipulator using PYTHON," *Journal of Mechatronics and Robotics*, vol. 3, pp. 542–551, 2019.
- [56] W. E. Abdul-Lateef, Y. N. Ibrahim-Alothman, and S. A. Gitaffa, "An optimal motion path planning control of a robotic manipulator based on the hybrid PI-sliding mode controller," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 727–737, 2023.
- [57] N. A. El-khateeb and R. I. Badr, "Novel PID Tracking Controller for 2DOF Robotic Manipulator System Based on Artificial Bee Colony Algorithm," *Electrical, Control and Communication Engineering*, vol. 13, no. 1, pp. 55–62, Dec. 2017.
- [58] F. Loucif and S. Kechida, "Optimization of sliding mode control with PID surface for robot manipulator by evolutionary algorithms," *Open Computer Science*, vol. 10, no. 1, pp. 396–407, 2020.
- [59] F. A. Raheem, A. T. Sadiq, and N. A. F. Abbas, "Robot arm free Cartesian space analysis for heuristic path planning enhancement," *Int. J. Mech. Mechatron. Eng.*, vol. 19, pp. 29–42, 2019.
- [60] Q. Chai and Y. Wang, "RJ-RRT: Improved RRT for Path Planning in Narrow Passages," *Applied Sciences*, vol. 12, no. 23, p. 12033, 2022.