Journal of Applied Engineering and Technological Science
Vol 7(1) 2025: 1-21

A NEW STRATEGY TO IMPROVE THE PERFORMANCE OF INFORMED RRT* ALGORITHM IN SOLVING THE GLOBAL PATH PLANNING OF MOBILE ROBOT

Heru Suwoyo[1], January Dwidasa Winiyoga[2]*, Julpri Andika[3]
Department of Electrical Engineering, Universitas Mercu Buana, Indonesia[123]
heru.suwoyo@mercubuana.ac.id[1], 55420110008@mercubuana.ac.id[2*],
julpri.andika@mercubuana.ac.id[3]

Received: 28 September 2024, Revised: 05 December 2024, Accepted: 18 March 2025
Corresponding Author

ABSTRACT
Informed RRT has been mentioned as a great method to find feasible and optimal solution of path planning. Technically, it uses the prolate hyper-spheroid and a centralized optimization strategy to gain the optimality of path. This optimization process is started when the initial feasible solution is found. Conventionally, the traditional procedure of RRT* is used to connecting starting point and goal point feasibly. Therefore, it is not suppressing if the optimization process begins later in large coverage of path planning problem. For this reason, a new strategy needs to propose with an objective to speed up the convergence rate by reducing the inefficiency of its blind sampling. Sequentially, it is conducted by integrating the bias technique and constraint sampling to replace the traditional sampling method. Next, the nearest node's ancestor is taken into consideration up until the first stage of choosing the parent is less expensive then RRT*. Regarding to these offers and the comparative results, the performance of the proposed method has shown better performance compared to its predecessor in terms of optimality, indicated by a decrease in finding the initial path by an average acceleration of 47.90% and a convergence rate indicated by an average path cost decrease value of 3.94%.*
Keywords : *Informed RRT *, Bias Technique, Constraint Sampling, Convergence Rate, Optimality*

1. Introduction

In certain scenarios, the robot concurrently undertakes exploration and Simultaneous Localization and Mapping (SLAM) tasks Wang et al., 2010; Yu et al., 2014). SLAM facilitates the determination of marginalized poses and the initial unknown map, supporting the robot's autonomy. This assumption arises from the efficiency of these tasks, allowing the robot to primarily focus on deciding the goal point from any initial position (L. Li et al., 2018; Tian, Suwoyo, Wang, & Li, 2019; Tian et al., 2019; Wen et al., 2015). The subsequent finding of a feasible and safe path is termed global path planning, while the tracing process is referred to as path tracking. Effective path planning is crucial before the robot transitions from its current to a desired pose to generate the reference path. Basically, there are two classifications based on how expansion is carried out, namely sampling-based methods and search-based methods. Search-based methods such as Breath First Search (BFS) (S. E. (Sakaria) Ginting & Sembiring, 2019) , Depth First Search (DFS)(El-Ghoul et al., 2008; S. E. Ginting & Sembiring, 2019), Dijkstra (Rachmawati & Gustin, 2020) , and A* (Al-Ansarry & Al-Darraji, 2021; Rachmawati & Gustin, 2020; Ran et al., 2021; Wang et al., 2021), offer a high degree of path optimality. However, it has a slow convergence speed. Meanwhile, sampling-based methods tend to have low optimality but can work quickly. Researchers consider overcoming the speed of search methods more difficult than improving the optimality of sampling-based methods. This is relatively the basis that influences the rapid development of sampling-based methods. Rapidly-Exploring Random Trees (RRT) is the method that is considered to have initiated this development (J. Li et al., 2014; Noreen et al., 2016). As the name suggests, RRT works by generating a random tree consisting of nodes that represent positions in the robot or vehicle configuration space. This tree is generated randomly by adding new nodes based on randomness in the configuration space, then connecting these nodes with the closest nodes that have been created previously. Even though it has been able to solve the path planning problem, the solution provided is far from optimal. This is caused by an undirected random sampling process. So, by implementing the node reconnection stage, RRT develops with the name RRT*. With this reconnection process, RRT* can improve the solution with a recursive iteration process. However, in a large environment, the time required to produce an optimal solution is as if infinite. This became a problem that was then paid attention to. In an effort to increase speed, this method is often combined with methods such as Potential Field (Min et al., 2015; D. Wu et al., 2022; Xinying et al., 2006), Genetic Algorithm (Chen & Qin, 2011; Dain, 1998; Ni et al., 2016; Suwoyo et al., 2018; Yusuf & Musdholifah, 2024; Zhou et al., 2012), Ant Colony Optimization (Bozorg-Haddad et al., 2017; Juang et al., 2018a), Artificial Bee Colony (Chen et al., 2017; Suwoyo et al., 2024), Particle Swarm Optimization (Adriansyah et al., 2019; Agrawal & Shrivastava, 2017; Jiang et al., 2017), and Neural Network (Bingul et al., 2005; Caceres et al., 2017; Ibarra-Pérez et al., 2022), to produce a directed search process. Apart from that, there are also methods named RRT*-Connect (Chen et al., 2021; Kuffner, 2000; Zhang et al., 2018), which adopts the working principle of RRT-Connect expansion while maintaining the node reconnection process. Meanwhile, to increase optimality, RRT* was developed by implementing triangle inequality (Mashayekhi et al., 2020; Noreen et al., 2016; Suwoyo et al., 2023; Z. W u et al., 2021). The use of a loose method like this is intended to optimize node determination when connection is made. So the node connection actually provides the shortest path. However, if you pay attention, both of them only solve partial problems of RRT*. So, a method emerged and was introduced with the name Informed RRT*. Informed RRT* (Mashayekhi et al., 2020; Naderi et al., 2015) uses sampling that is biased towards the informed region, it excels in scenarios where an initial estimate of the optimal path is available, resulting in a more efficient exploration process. Additionally, it uses heuristics to guide the sampling process, by combining prior knowledge or domain-specific information. This heuristic guidance improves the exploration process by intelligently directing the algorithm to promising regions. However, the advantages offered can only be fulfilled if an initial solution is obtained. The initial solution in the Informed-RRT* algorithm still relies on random sampling methods and conventional rewiring processes. Although an optimal solution can be achieved, this often sacrifices the convergence speed at the beginning of the search. As a result, even if an optimal solution is found eventually, the initial progress tends to be slow and inefficient, requiring more iterations to improve the found path. So, it is possible that the optimality offered is not balanced with a good convergence rate at the start.

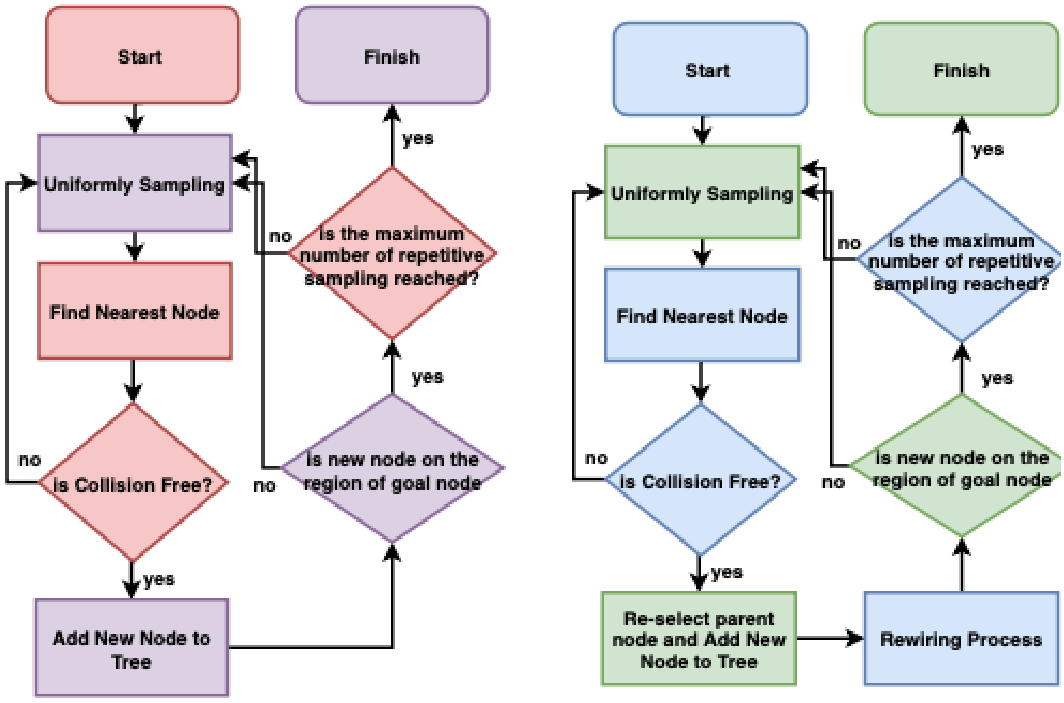
Based on this phenomenon, this research proposes an integrated approach between the bias technique and constraint sampling in the exploration process. Moreover, the nearest node's ancestor is considered in supporting the rewiring process. Furthermore, based on this approach, the novel element that will be obtained is a hybrid algorithm to solve path planning problems with a better level of effectiveness, convergence speed and path optimality. Then based on this novelty, it can be stated that the proposed method contributes to the development of the path planning method. Conceptually, the path planning's core objective is to efficiently explore the search space, providing a high-quality and optimized path from the initial point to the goal while considering environmental constraint. The practical implications of this study are improving the efficiency of path planning for real systems such as robots, autonomous vehicles, or drones, allowing them to operate more effectively in complex environments. Theoretical implications focus on developing and refining algorithms in path planning, providing new insights and innovations in the theory underlying the methods used. Thus, this study not only provides direct benefits for practical applications, but also enriches the knowledge base in the field of route planning. Informed RRT* incorporates an informed sampling strategy, focusing exploration efforts on regions that are likely to contribute to the optimal path. This is achieved by biasing sampling towards areas that have not been adequately explored. It utilizes heuristics to guide the sampling process, leveraging prior knowledge about the environment or problem-specific information. This guidance helps in achieving a more directed and efficient exploration. This process is conducted as an optimization stage when the initial feasible path found. That makes, the informed RRT* has been known as the method cares about optimality path. It maintains the core principle of RRT* by ensuring asymptotic optimality. In other words, as the number of iterations increases, Informed RRT* converges towards the optimal solution, providing a high-quality path. Moreover, it allows for incremental refinement of the existing tree structure, adapting to new information and dynamically adjusting the search strategy. This flexibility is particularly beneficial in dynamic environments. Because of it, the demonstration of effectiveness in high-dimensional configuration spaces, making it suitable for complex robotic systems with numerous degrees of freedom. However, the algorithm's complexity may be a limiting factor, especially in scenarios with real-time constraints or resource-limited platforms. Besides that, the effectiveness of Informed RRT* is contingent on the quality of the heuristic guidance, and choosing appropriate heuristics can be a non-trivial task contains brief and concise research backgrounds, and objectives.

The rest of this paper is organized as follows: Section II presents the theoretical of informed-RRT*, the bias techniques adopted for speeding up the exploration, and constraint sampling to deals with slow convergence speed before the initial path is found. Moreover, in the same section, the wiring process adopted from RRT* is again explained; Section III presents how the proposed method is designed. Section IV presents the results of several type of algorithm such as RRT*, Fast-RRT (Jeong et al., 2015; Q. Li et al., 2022; Z. Wu et al., 2021b), Informed-RRT*, and the proposed method; Section V presents the conclusion stated based on the comparative analysis from Section IV.

2. Literature Review

Let $X \in R^n$ is representation of state space for a path planning problem, with $n \in N$ is space dimension, thus $X = \{X_{obs}, X_{free}\}$ is state space with $X_{obs} \in X$ refers to obstacle coordinates and $X_{free} \in X$refers to the free space. Moreover, if the start node $x_{init} \in X_{free}$ and goal node x_{goal} $x_{goal} \in X_{free}$are given, then referring to X_{obs}, the path planning algorithm has to find the ideal path from-to those nodes, denoted as $\sigma = [0, T] \to X_{free}$ with $\sigma(0) = x_{init}$ and $\sigma(T) = x_{goal}$ where $X_{goal} = \{x \in X \big| x - x_{goal} \big| < r\}$ for r is radius around x_{goal}.

Before RRT* was introduced, RRT was first known to researchers as a method that solves path planning problems, by applying a sampling-based technique to the space of planning problems. If the information on the starting point, the end point or target, and all obstacles in the environment, RRT first performs path expansion. This expansion begins by generating random nodes in the planning problem area. Based on these random nodes, the nearest node is then detected as a reference for connecting new nodes. A new node is a node that is placed in a position between a random node and the nearest node. If the distance between the random node and the nearest node is smaller than the specified distance, then the random node is automatically considered a new node. The new node is then connected to the nearest node by considering whether there is a collision with an existing obstacle or not. If a collision occurs, the new node is not registered, and instead the new node is registered as a member node of the existing candidate path. By applying this method, expansion is carried out in the next iteration and will stop when the new node is in the area around the goal point. This process produces a safe path quickly even in large areas. However, the obstacle of the RRT method itself is its weakness towards optimality. So that RRT* is introduced with the addition of a rewiring process. This process is the process of determining the relationship between a new node and the nearest node. By considering other nodes, the route that will be generated is considered. The new node will be connected to the node closest to it and will be connected to the node that will produce a shorter distance. Rewiring that is done simultaneously every time a new node is found makes RRT* guarantee convergence to produce an optimal solution along with increasing sampling repetitions. The flowchart representing the working principle of both algorithms can generally be described as shown in Fig. 1.



(a) (b)

Fig. 1. Working Principle of RRT (a) and RRT* (b)

The working principle of RRT* can be seen in the pseudocode of Algorithm 1. As seen in Fig. 1, the difference between RRT and RRT* is in the rewiring process. So, it can be said that removing the 12th line means that Algorithm 1 becomes RRT.

Algorithm 1 RRT*

Require: Z, z_{start}, z_{goal}

1. $T \leftarrow initializeTree(\square)$
2. $T \leftarrow insertNode(z_{start}, T)$
3. $\boldsymbol{while}\ \|z - z_{goal}\| \geq r\ \boldsymbol{do}$
4. $\quad z_{rand} \leftarrow Sample(Z)$
5. $\quad z_{near} \leftarrow Near(z_{rand}, T)$
6. $\quad z_{new} \leftarrow Steer(z_{rand}, z_{near}, StepSize)$
7. $\quad \boldsymbol{if}\ CollisionFree(z_{new}, z_{near}, Z_{obs})\boldsymbol{then}$
8. $\qquad Znear \leftarrow NearN(T, z_{new})$
9. $\qquad x_{min} \leftarrow ChooseParent(Z_{near})$
10. $\quad \boldsymbol{end\ if}$
11. $\quad T \leftarrow insertNode(z_{new})$
12. $\quad T \leftarrow Rewire(z_{min}, z_{new})$
13. $\boldsymbol{end\ while}$

Procedurally, this is what causes the optimality of RRT* to only be obtained when there is a continuous iteration. Or simply, optimality is achieved when the maximum number of samplings is increased and goes to infinity. On this basis, several algorithms such as Smart-RRT* and Informed-RRT* emerged. Although, it also highly depends on the large number of samplings, the optimality of the path produced by Smart-RRT* can be achieved by utilizing sampling in the area near the beacon, the nodes connecting the start to goal point (Islam et al., 2012; Nasir et al., 2013; Noreen et al., 2016; Suwoyo et al., 2023). This sampling is limited by a circle with a radius of r_{beacon} and centered on

z_{beacon}. This sampling is an alternative to uniform sampling from RRT* when the initial path is found. Every time an iteration in this optimization process shows a specified multiple, sampling is performed. Thus, the optimality of the path is more promising with several repetitions that are not as large as RRT*. Technically, this smart sampling will replace the command in line 4 of Algorithm 1, which was previously marked by the existence of an initial path and a specified multiple. While in informed-RRT*, admissible ellipses limit the sampling area that is allowed when the initial path is found. Every time there is a unique path with a better cost value, this ellipse then shrinks so that it can focus sampling in the area around the path. The ellipse formed is highly dependent on the path formed, the cost of the path, and the direct distance between the starting point and the goal point. The ellipse is formed by assuming the distance between the starting and ending points as the focus, while the cost of the path is the major axis. This arrangement makes sampling centralized and the efficiency of path search increases. High sample size remains a major challenge. In environments with many obstacles or with many narrow spaces, Smart-RRT* often requires many samples just to find an initial feasible path. This results in high computational costs and slows down convergence in the early stages. This problem arises due to the nature of sample-based algorithms, where the process of randomly sampling and connecting valid configurations can be very slow, especially in narrow spaces. The pseudocode of this sampling method can be seen in Algorithm 2.

Algorithm 2 $Sample(z_{start}, z_{goal}, a)$

if $a < \infty$ **then**
    $a \leftarrow \left\| z_{start} - z_{goal} \right\|_2$
    $z_{centre} \leftarrow (z_{start} + z_{goal}) * 0.5$
    $R \leftarrow RotationtoWorldFrame(z_{start}, z_{goal})$
    $b \leftarrow \left(\sqrt{(a^2 - c^2)}\right) * 0.5$
    $L \leftarrow diag\{a, b\}$
    $z_{ball} \leftarrow SampleUnitBall$
    $z_{rand} \leftarrow RLz_{ball} + z_{centre}$
else
    $z_{rand} \cup (Z)$
end if
return z_{rand}

In this case, the ellipse's semi-major axis length is denoted by a, its semi-minor axis length by b, its half focal length by c, its centre by xcentre, its rotation matrix by R, its diagonal matrix for stretching transformation by L, its sampling point in the unit circle by z_{ball}, and its final sampling point by z_{rand}. To create the sampling point (x, y) inside the ellipse, point (x_b, y_b) is randomly sampled from a unit circle using the Informed-RRT* algorithm and then stretched. The transformation equation is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x_b \\ y_b \end{bmatrix} \tag{1}$$

The ellipse has to be rotated after this conversion transformation such that its major axis lines up with the line that connects the starting point and the target point. By knowing the middle point coordinates connecting the starting and goal points, and the rotated coordinate (x, y), the sampling node in the region of ellipse (x_{rand}, y_{rand}) can be obtained as

$$z_{rand} = \begin{bmatrix} x_{rand} \\ y_{rand} \end{bmatrix} = R \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{2}$$

Where R is a two-dimensional rotation matrix expressed as follows

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3}$$

where θ is the angle formed by the line between the beginning point, the target point, and the x-axis. Meanwhile for the middle coordinates of the line that directly connects the starting and goal points can be calculated as follows

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \frac{1}{2} * \left(\begin{bmatrix} x_{start} \\ y_{start} \end{bmatrix} + \begin{bmatrix} x_{goal} \\ y_{goal} \end{bmatrix} \right) \tag{4}$$

3. Proposed Method

Basically, informed-RRT* utilizes RRT* in the exploration process to find the initial path. Random and uncertain sampling makes this process take a long time. Instead of being able to determine the optimal path, blind search like this is unable to handle complex environmental problems such as mazes with close distances between Z_{free} and Z_{obs}. On this basis, a sampling method that prioritizes expansion speed is proposed. In addition to sampling restrictions on areas that have not been explored, this sampling method also applies a bias technique. Both have a role to ensure that there is a process of connecting the start to the goal point quickly and ensuring that each determination of a new node can avoid obstacles in narrow areas. Both of these sampling methods are adopted from improved-RRT which contains fast-RRT. Unlike fast-RRT, rewiring the process is applied to ensure that in addition to being able to get a path quickly, the resulting path is also shorter. Conceptually, this sampling can be done by first generating uniform sampling, and producing z_{rand}. After z_{rand} is obtained, the nearest node is searched for, to determine the location of z_{new}. After the coordinates of z_{new} are found, its position in relation to the collection of nodes in the tree is then observed. If the z_{new} position is a smaller distance calculated from any node in the tree, then z_{new_1} is declared invalid, and uniform sampling is performed again. This cycle repeats until z_{new} is more than or equal to r_{fs} to any node in the tree. This process will make z_{new} always in an unexplored space. This concept can be described as shown in Fig. 2.

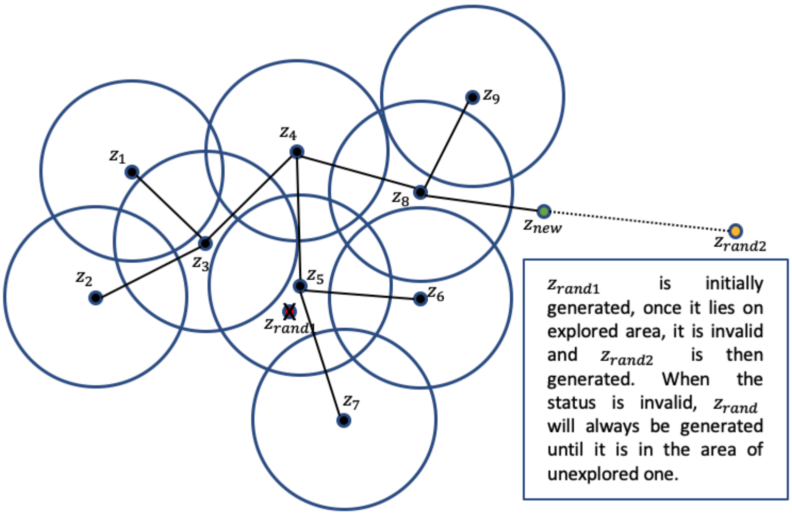


Fig. 2. Concept of Sampling Restriction

Not only is the sampling repeated in an explored area, every z_{new} obtained is indicated as not collision-free, it also makes the expansion process slow and even stack when planning in corridors with a width smaller than the sample size. Referring to this problem, random steering is proposed as a method to bias z_{new} in a direction that is free from collisions. In addition to z_{new} being confirmed to be in an unexplored area, z_{new} is also forced to be able to penetrate obstacles if a collision is indicated to occur with any obstacle in the environment. This process can be illustrated as follows in Fig. 3.

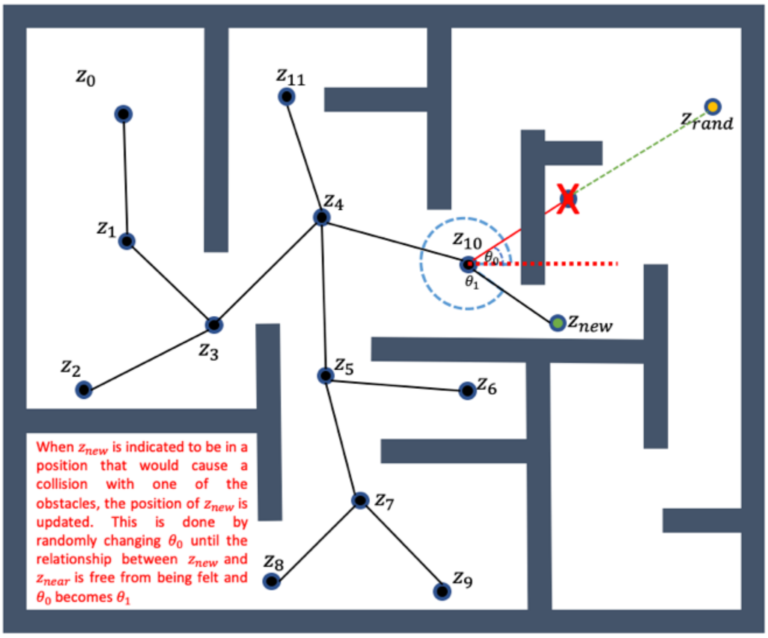


Fig. 3. Bias Sampling Principle

Bias sampling ensures that the new node placement is in Z_{free}, and its connections are collision-free. As shown in Fig. 3, this process is done by generating theta_new when the theta_old (connection direction z_{new} to z_{near}) results in an invalid or collides to any obstacle. This bias sampling is repeated in the exploration process after the random node is generated untill the initial path is found. With bias sampling and sampling restriction techniques, the direction of path exploration can be improved with better effectiveness of random node placement. Random steering adjusts the direction of the search tree to be more focused, while fast sampling prioritizes regions that have a higher chance of finding valid paths. The combination of these two techniques speeds up the search and convergence process without sacrificing the quality of the solutions found. In the implementation of the limitation of the maximum number of sampling is often a measure of whether an algorithm can complete a task quickly or not. So when the initial path is found, the remaining amount allowed for sampling can be used to optimize the path, namely by applying sampling restrictions only to the ellipses on the informed-RRT*.

Algorithm 3 Enhanced IRRT*

Require: Z, z_{start}, z_{goal}

$T \leftarrow initializeTree(\square)$

$T \leftarrow insertNode(z_{start}, T)$

$\boldsymbol{while}\ it < maxIt\ and\ \|z - z_{goal}\| \geq r\ \boldsymbol{do}$

$\quad \boldsymbol{if}\ initialPathFound$

$\quad\quad z_{rand} \leftarrow IRRTStarSampling(Z)$

$\quad \boldsymbol{else}$

$\quad\quad z_{rand} \leftarrow FastSampling(Z)$

$\quad \boldsymbol{end\ if}$

$\quad z_{near} \leftarrow Near(z_{rand}, T)$

$\quad z_{new} \leftarrow BiasSteer(z_{rand}, z_{near}, StepSize)$

$\quad \boldsymbol{if}\ CollisionFree(z_{new}, z_{near}, Z_{obs})\boldsymbol{then}$

$\quad\quad Znear \leftarrow NearN(T, z_{new})$

$\quad\quad x_{min} \leftarrow ChooseParent(Z_{near})$

10. $\boldsymbol{end\ if}$
11. $T \leftarrow insertNode(z_{new})$
12. $T \leftarrow Rewire(z_{min}, z_{new})$
13. $\boldsymbol{end\ while}$
14. $pathOptimized(T)$

Conversely, if the initial path is found slowly or requires a large sampling repetition, the sampling available for optimization becomes limited and the optimization cannot work optimally. This factor is the basis for the proposal of a fast and effective sampling method to improve the performance of the informed-RRT* in obtaining the optimal path in a short time. The concept of the proposed method can be seen in Algorithm 3.

As can be seen from Algorithm 3, the fast sampling and random steering method is used when the initial path is not found yet. Fast sampling prioritizes areas of the space that are more likely to yield valid paths, thus reducing unproductive samples and speeding up the search process. On the other hand, random steering directs the search tree expansion to more promising areas, reducing the time spent on exploration in irrelevant areas. Thus, the sampling efficiency in path planning before the initial path is found can be greatly improved in complex, narrow, and corridor areas. To clearify the optimization process the Algoritm 4 is presented (it represents IRRTStarSampling on Algorithm 3).

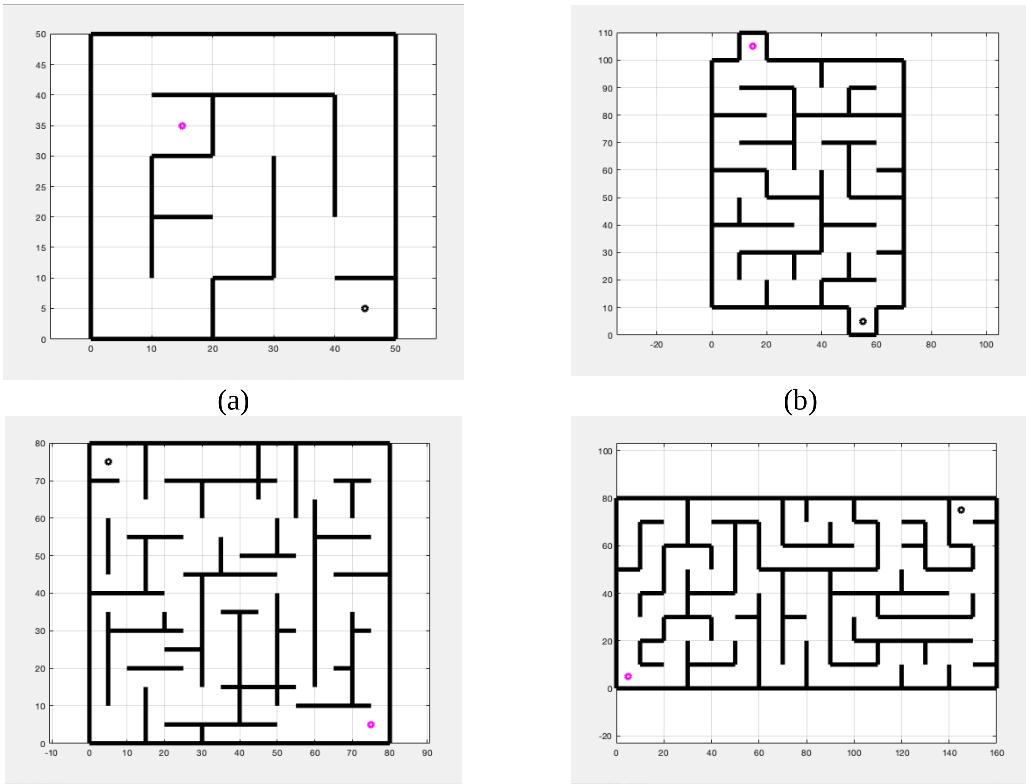
Algorithm 4 IRRTStarSampling

if $c_{max} < \infty$
  $z_{centre} \leftarrow \frac{z_{start} + z_{goal}}{2}$
  $\boldsymbol{C} \leftarrow RotateToWorldFrame(z_{start}, z_{goal})$
  $r_1 \leftarrow \frac{c_{max}}{2}$
  $c_{min} \leftarrow \|z_{goal} - z_{start}\|_2$
  $\{r_i\}_{i=2,\dots,n} \leftarrow 0.5 * (c_{max}^2 - c_m in^2)^{1/2}$
  $\boldsymbol{L} \leftarrow diag(r_1, r_2, \dots, r_n)$
  $z_{ball} \leftarrow SampleUnitBall$
  $z_{rand} \leftarrow (\boldsymbol{CL} z_{ball} + z_{centre}) \cap Z$
else
  $z_{rand} \leftarrow \mathcal{U}(Z)$
return z_{rand}

SampleUnitNBall function, which generates uniformly distributed random samples from the volume of an n-ball of radius one centered at the origin. In this case, an n-ball is a geometric shape that is a generalization of a sphere in a higher-dimensional space, and this function samples points uniformly within the volume of the ball. The notation $xball \sim \mathcal{U}(Z_{ball})$ means that the generated points (x_{ball}) follow a uniform distribution (U) in the unit ball space. Thus, this function is used to sample random points in n-dimensional space that lie within the unit ball. Thus, in the optimization process, the generated random nodes will be on the ellipse formed based on the positions of z_start and z_goal. The use of this ellipse also allows optimization to reduce sampling in unnecessary spaces, so that it still directs the search towards areas that are more likely to produce the best path. Unlike the initial path search process, in this optimization process the BiasSteering function (Algorithm 3 line 6) is replaced with the steer function (see Algorithm 1 line 6). The optimization process ends when the number of allowed sampling has been met. Furthermore, as the end of this method, path reduction is applied by referring to the principle of triangular inequality (see Smart-RRT*).

4. Results and Discussions

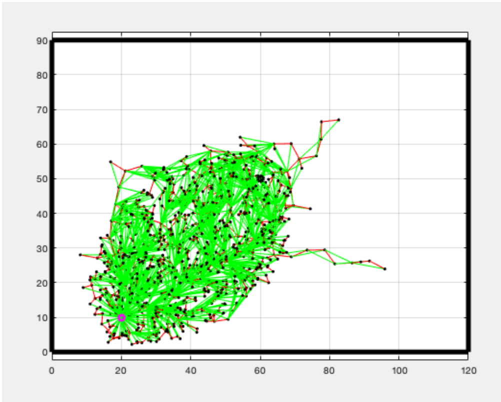
In this section, the results of the different algorithm tests, namely RRT*, Fast-RRT*, RRT*-Smart, Informed-RRT*, and the proposed algorithm, are described. The tests were conducted on a PC with the following specifications: processor: 3.6 GHz Quad-Core Intel Core i3; graphics: Intel UHD Graphics 630 1536 MB, and memory: 8 GB 2667 MHz DDR4. The tests were conducted in various types of maze environments from simple to complex, as seen in Fig. 4.



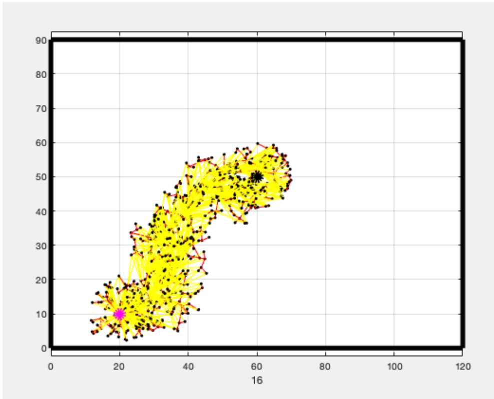
(a) (b) (c) (d)

Fig. 4. Different Maze Environment used to Evaluate the Performance of RRT*, Fast-RRT*, RRT*-Smart, Informed-RRT*, and Proposed Method

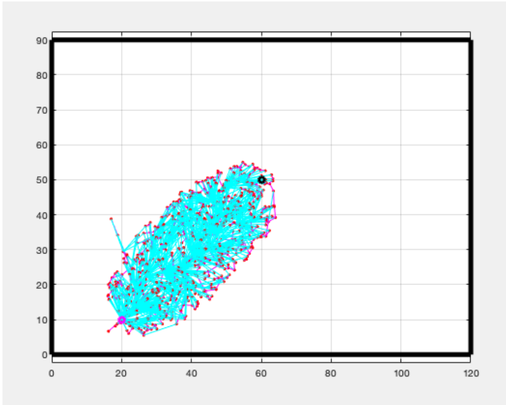
The environment shown in Fig. 4 is commonly used as a benchmark for testing path planning algorithms. Fig. 4 (a) is a maze environment adopted from (Bastapure et al., 2023) by placing the starting position at (15, 35) and (45, 5). With increased complexity, especially in the complexity and distance between the starting point and the goal point, the 2nd maze environment was adopted from (Nasir et al., 2013). In this arena shown in Fig. 4 (b), the starting point is at (15,105) and the goal point is at (75,5). Furthermore, Fig. 1 also shows that the various algorithms are tested for both their working speed and optimality in an environment with high complexity and narrower corridor width, see Fig. 4 (c). This environment is proposed to measure the level of speed in conducting exploration when the corridor is narrowed with the same sampling settings as before. In this arena, the starting point is positioned at (7,75) and the goal point is positioned at (75,5). As for the 4th environment shown in Fig. 4 (d), it is an environment adopted from (Sun et al., 2022) with the designation as a test environment to measure the speed of convergence when the distance between the starting and goal points is stretched further with only 1 type of road available that can connect them. The starting and goal points are respectively at positions (5,5) and (145,75). In this test, all methods are tested to solve the path planning problem in the first maze environment. This test considers the working speed represented by the number of samplings needed to obtain the initial solution. In addition, the optimality of the path is also considered, represented by the lowest distance that can be generated at a given number of samples. Before testing on the maze environment shown in Fig. 4, the first test was conducted to show the difference in sampling methods when the initial path is found. This test involves informed-RRT*, RRT*-Smart, and the proposed method. These sampling techniques can be seen in Fig. 5.



(a)



(b)



(c)

Fig. 5. Different Sampling Technique Conducted After Initial Path Found (a) Informed-RRT* (b) RRT* Smart (c) Proposed Method

As shown in Fig. 5, informed-RRT* utilizes the prolate hyper-spheroid formed based on the initial path found and the distance between the starting and goal points. As seen in Fig. 5 (a), the initial path found greatly determines the size of the sampling distribution boundary ellipse. By only utilizing RRT* in sampling in the exploration process, a non-optimal initial path found will widen the sampling range, so that obtaining optimal paths takes a long time. Unlike informed-RRT*, sampling in RRT*-Smart focuses on nodes that are members of the solution path, see Fig. 5 (b). Thus, the non-optimal node relationship causes RRT*-Smart to require more time to improve the optimality of the final path. While in the proposed sampling method, these two problems can be solved, as shown in Fig. 5 (b). Although it has exactly the same characteristics as informed-RRT*, sampling in the optimization process is greatly assisted by an initial path that has better optimality compared to informed-RRT* and RRT*-smart. This is due to the exploration method that tends to be in line with the need to obtain optimality, by using fast-RRT*. Thus, optimization can be carried out optimally.

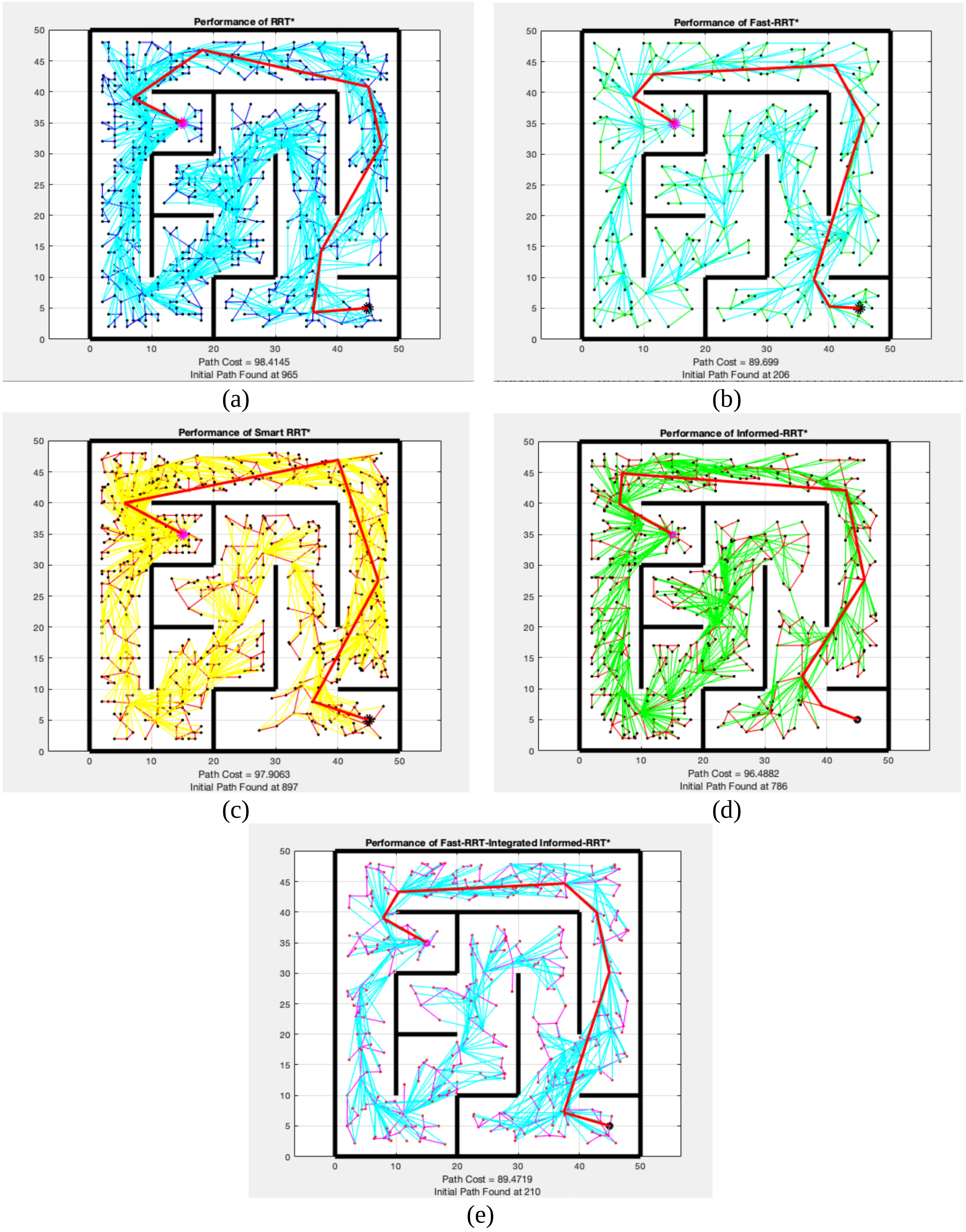


Fig. 6. Performance of Different Algorithm in Solving A Path Planning Problem in Environment 1 (a) RRT*, (b) Fast-RRT*, (c) RRT*-Smart, (d) Informed-RRT*, and (e) Proposed Method

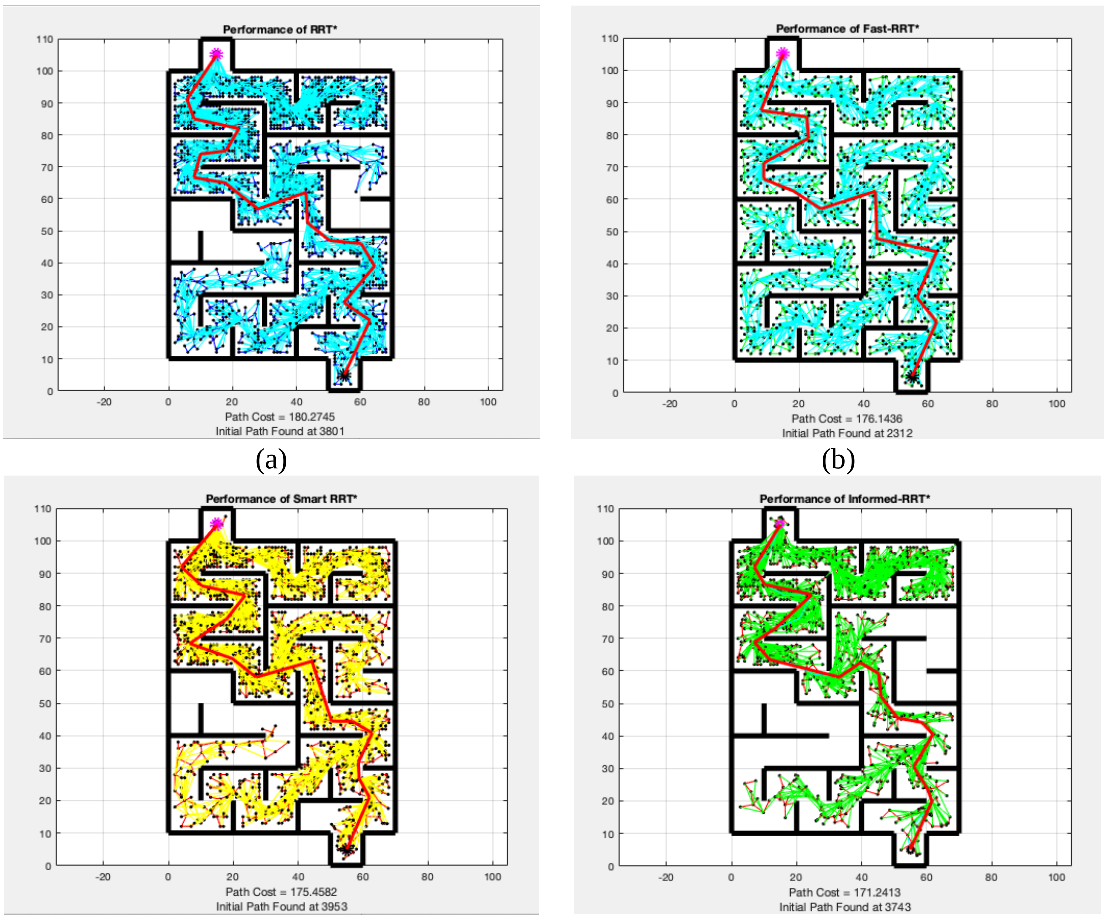
Next, testing is carried out in the maze environment 1. In this test, the sampling allowed is 1500 times. In the fast-sampling in Fast-RRT* and the proposed method is set with a radius value of 5 distance units. To measure optimality, the path cost generated by each algorithm is observed. As seen in Fig. 6 (e), and Table 1, the path cost of the proposed method shows the best value, the lowest of the others. Although too far from informed-RRT* and RRT*-Smart, the cost value obtained is close to the results of Fast-RRT*, Fig. 6 (b). This shows that the path optimization after the initial path is found runs more optimally. So, it is clear, the role of Fast-Sampling in this case has a significant influence, namely in the form of work speed in finding the initial path, and leaving sufficient optimization work duration. Where this condition cannot be achieved well by

RRT*-Smart (in Fig. 6 (c)) and Informed-RRT* (in Fig. 6 (d)) which require identical time as RRT* (in Fig. 6 (a)) to find the initial path.

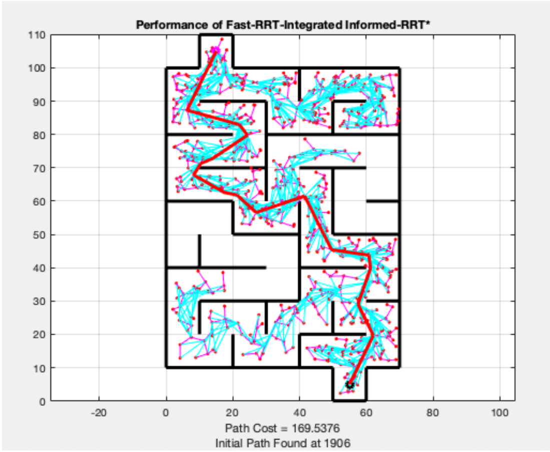
Table 1 - Path Cost of Different Algorithm's Solution in Environment 1

Algorithm	N-th Sampling Used to Obtain Initial Path	Path Cost (Units of Length)
RRT*	965	98.4145
Fast-RRT*	206	89.6990
Smart-RRT*	897	97.9063
Informed-RRT*	786	96.4882
Proposed Method	210	89.4719

Next, testing is carried out on environment 2 with the number of samplings allowed is 6000 times. The setting on fast-sampling in fast-RRT* and the proposed method is 5 units of length, assuming the robot has a dimension of 1 unit, and with the closest distance to the obstacle is 1 unit as tested in all environments. The results of the second test can be seen in Fig. 7.



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



(e)

Fig. 7. Performance of Different Algorithm in Solving A Path Planning Problem in Environment 2 (a) RRT*, (b) Fast-RRT*, (c) RRT*-Smart, (d) Informed-RRT*, and (e) Proposed Method

The results of the second test are shown in Fig. 7. As seen in Fig. 7 and Table 2, RRT*, Informed-RRT*, and RRT*-Smart require around 4000 samplings, three times larger than Fast-RRT* and the proposed method. Thus, the remaining sampling amount that becomes the duration of path optimization in Informed-RRT* and RRT*-Smart becomes very short, which is around 2000 repetitions on more complex problems and far from the starting and goal points. This has an impact on the high cost path, because optimization cannot run properly. On the other hand, with 3 times faster initial path found, informed-RRT* and the proposed method have a better duration to perform optimization. However, optimization techniques that apply fusion, relying on other alternative paths, in narrow corridor areas are no longer ideal. This is due to the limited sampling in the arena to obtain other different paths that are very rare. This incident is a factor, even though it has enough time to perform optimization, fast-RRT* is not good enough compared to the proposed method. In contrast to the incident, by utilizing the ellipse that limits the sampling, the proposed method can maximize the optimality of the formed path even in a narrow corridor. This basis makes the optimality of the proposed method better than all the methods tested. To be able to pay more attention to this difference, Table 2 is presented.

Table 2 - Path Cost of Different Algorithm's Solution in Environment 2

Algorithm	N-th Sampling Used to Obtain Initial Path	Path Cost (Units of Length)
RRT*	3801	180.2745
Fast-RRT*	2312	176.1436
Smart-RRT*	3953	175.4582
Informed-RRT*	3743	171.2413
Proposed Method	1906	169.5376

Next is the test on the 3rd maze environment with the characteristic of inconsistent width of the corridor that limits the starting with the goal point. In this test, the number of samples allowed is 10000. Nothing has changed in the parameter settings for fast-sampling in Fast-RRT* and the proposed method. In this test, each for RRT*, Fast-RRT*, Smart-RRT*, informed RRT*, and the proposed method requires 7753, 5193, 7145, 7383, and 5041 repetitions to obtain the initial path, respectively (see Fig. 8 and Table 3). So, the optimization duration for RRT*, Smart-RRT*, informed-RRT* is around 2000s and for Fast-RRT* and the proposed method is 4000s.

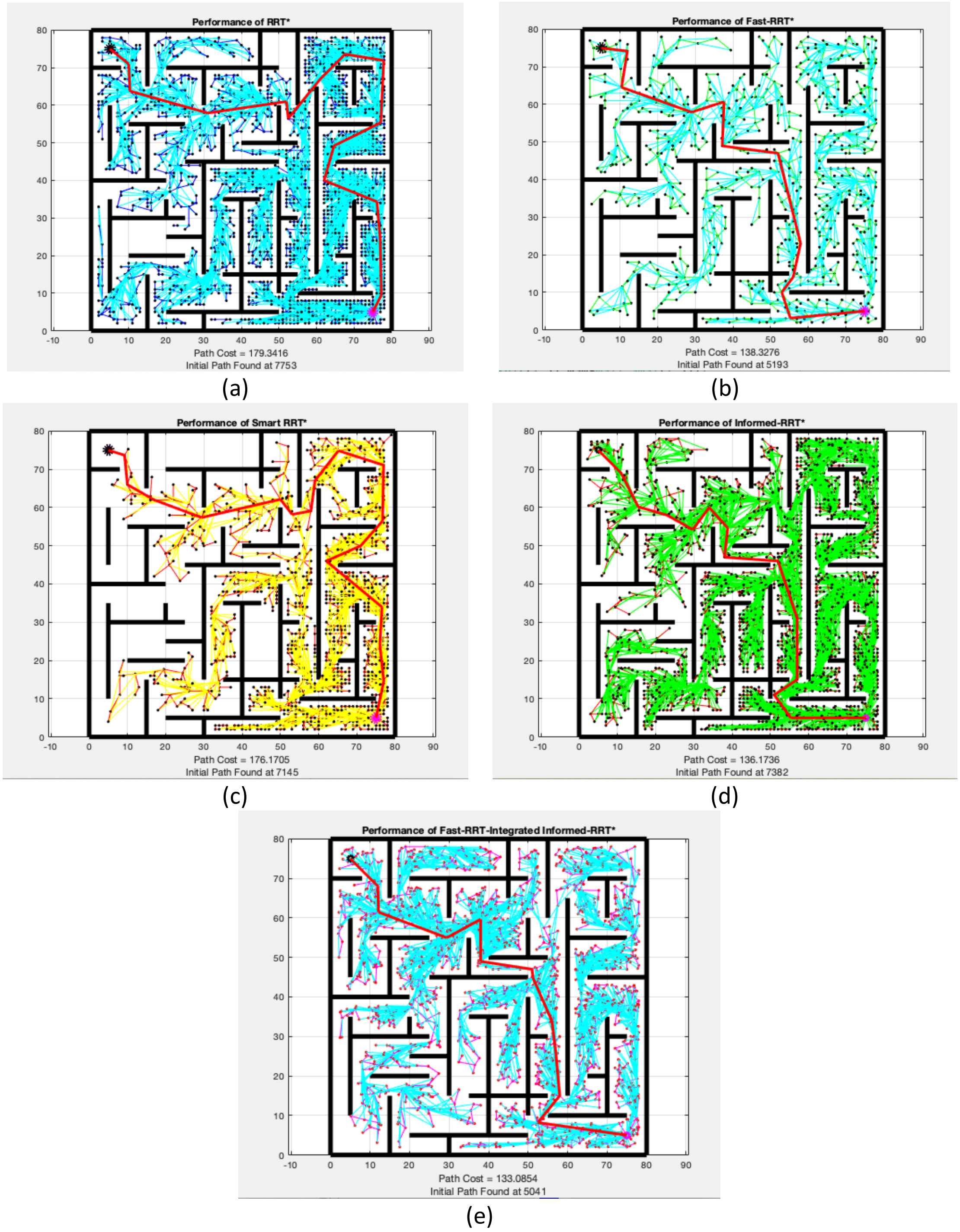


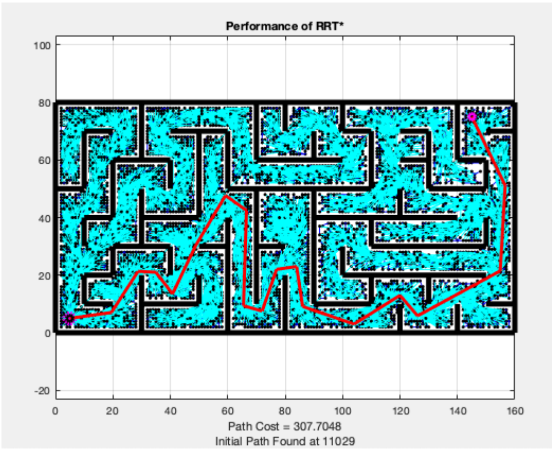
Fig. 8. Performance of Different Algorithm in Solving A Path Planning Problem in Environment 3 (a) RRT*, (b) Fast-RRT*, (c) RRT*-Smart, (d) Informed-RRT*, and (e) Proposed Method

As seen in Fig. 8 (a), 2000 random sampling repetitions relying on process rewiring, make RRT* able to produce a path with an optimality value of 179.3416. While Smart-RRT* actually produces a path cost value of 176.1705. Although slightly different, this incident shows that in limited sampling conditions, and the width of the space variant has an impact on the optimization process that cannot be maximized. While in a limited time of around 2000 repetitions, informed-RRT* can do better optimization and produce a shorter cost path, which is 136.1736. Thus, in the 3rd environment, optimization utilizing the bounding ellipse is more ideal than utilizing the sampling technique in the area around the beacon. In addition to strengthening the advantages of informed-RRT*, this incident also underlies that the proposed method with a wide duration has the potential to provide optimal results. This statement can be emphasized by the results shown by the proposed method. With around 4000 samplings, the proposed method can optimize the optimization process using ellipse restrictions. Although this value is considered sufficient to perform the optimization process even with the fusion path technique, the results shown by the proposed method again explain that the optimization technique in the proposed method is indeed better and ideal. The node position of the initial path solution has the potential to be in wide and narrow areas, so the fusion process will have a good effect only in wide spaces, while in narrow areas it will be normal. This statement makes Fast-RRT* have a value that is still lacking compared to the proposed method and informed-RRT*, which is 138.3276.

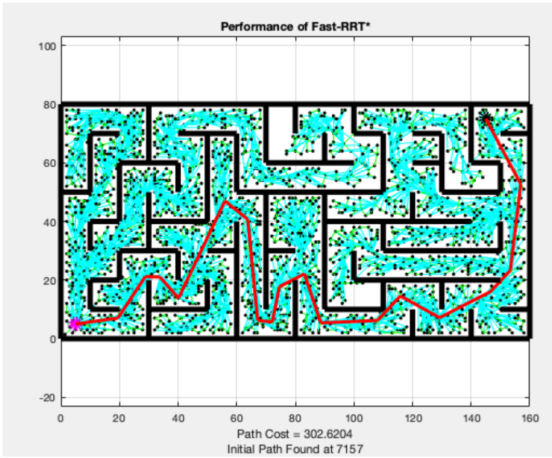
Table 3 - Path Cost of Different Algorithm's Solution in Environment 3

Algorithm	N-th Sampling Used to Obtain Initial Path	Path Cost (Units of Length)
RRT*	7753	179.3416
Fast-RRT*	5193	138.3276
Smart-RRT*	7145	176.1705
Informed-RRT*	7383	136.1736
Proposed Method	5041	133.0854

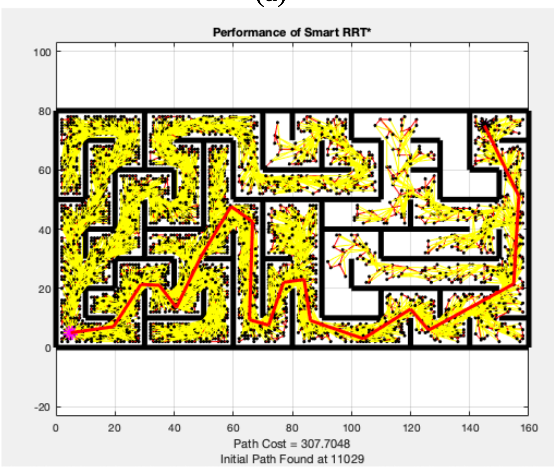
Furthermore, to re-test the consistency, the proposed method and other methods are compared again in solving the problem of environment 4. The sampling allowed for solving this problem is 15,000 because the span of the starting and goal points tends to be further with the parameter settings than with the solution in environments 1, 2, and 3. In contrast to the previous environmental conditions, in environment 4 there is only 1 type of road that can connect the starting and goal points.



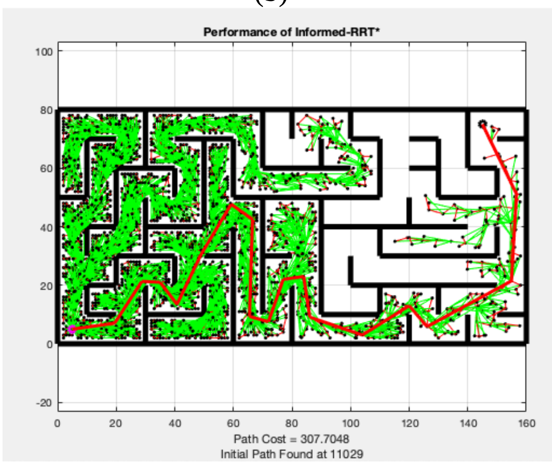
(a)



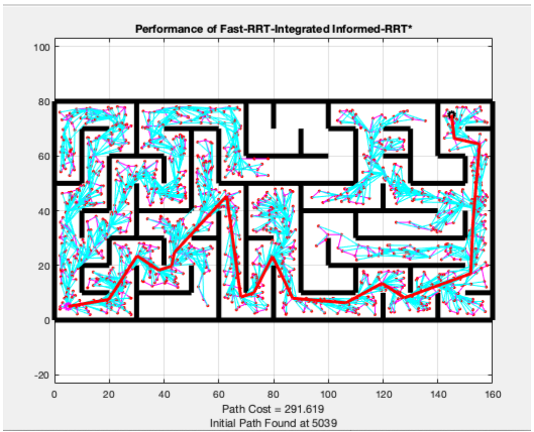
(b)



(c)



(d)



(e)

Fig. 9. Performance of Different Algorithm in Solving A Path Planning Problem in Environment 4 (a) RRT*, (b) Fast-RRT*, (c) RRT*-Smart, (d) Informed-RRT*, and (e) Proposed Method

Based on Fig. 9, it can be seen that fast-sampling in fast-RRT* and the proposed method can accelerate the initial path acquisition. Bias technique provides an influence to be able to reach unexplored areas while constraint sampling reduces complexity when expanding in areas with narrower widths. Based on this approach, fast-RRT* and the proposed method can find the initial path quickly with a sufficient level of optimality. In contrast to only utilizing random sampling and process rewiring. RRT*, Smart-RRT*, and informed-RRT* take longer and only leave a narrow search duration for the optimization process. Thus, it can be reaffirmed that providing acceleration in the initial path search process can increase the ideal level of optimization. As seen in Fig. 9 (a), Fig. 9 (c) and Fig. 9 (d) the cost path values obtained show the same results. So, it can be said that the short duration cannot be utilized by informed-RRT* and Smart-RRT* to perform optimization. This can be proven by comparing it with the cost path generated by fast-RRT*, see Fig. 9 (b). With sufficient duration to perform optimization, Fig. 9 (b) explains that complex optimization techniques, such as path fusion, still provide potential for improvement. It is said to be complex because path fusion does not only depend on the previous path found, but also on the availability of different new paths. While the new path is unlikely to be obtained in cases where there is only one type of solution. For this reason, it can be stated again that the technique of utilizing the prolate hyper-spheroid and a centralized optimization strategy will be more effective and efficient when the possibility of a new path being generated is low. Furthermore, to support these statements, Table 4 is presented.

Table 4 - Path Cost of Different Algorithm's Solution in Environment 4

Algorithm	N-th Sampling Used to Obtain Initial Path	Path Cost (Units of Length)
RRT*	11029	307.7048
Fast-RRT*	7157	302.6204
Smart-RRT*	11029	307.7048
Informed-RRT*	11029	307.7048
Proposed Method	5039	291.619

By observing Table 4, it is shown that with a duration of around 4000 is not enough to make optimization techniques on RRT*, smart-RRT*, and that informed-RRT* is enough to improve the path value of the formed path. In addition to being influenced by the range of the starting and goal points, this is also factored by the winding initial path that makes the ellipse wide and the centralized sampling has the same characteristics as RRT*. This also applies to smart-RRT*, the formed path stretches from the starting to the goal point winding with a large number of beacons. So that the optimization technique centered around the beacon cannot work well in a narrow time. Thus, in cases like this, the optimization time becomes a dominant factor in the aim of increasing the optimality of the formed path. Referring to the results in environments 1, 2, and 3, it can be said that the proposed method can effectively solve problems both in terms of the speed of obtaining the initial path and the optimality of the resulting path.

5. Conclusion

The optimization process of informed-RRT* depends on when the initial path is found. Technically, informed-RRT* utilizes the way RRT* finds the initial path. Because of this method, the duration to get the path becomes long because there is no directional sampling in RRT*. In addition, the exploration that jumps and random makes this method inefficient in large-scale and complex environments. The path optimization provided by informed-RRT* is only achieved when the duration is sufficient. The long duration to find the initial path reduces the duration of path optimization in limited actions. For this reason, bias and constraint sampling are involved in this study to ensure that the exploration process does not repeat itself in areas that have been explored. Based on the results that have been presented previously, an average decrease in the path cost value of 3.94% is obtained, which is an increase in the optimality of informed-RRT* to the proposed method. This achievement is supported by the work of the proposed method which is faster with an average acceleration value of 47.90% compared to the base method, informed-RRT. This shows that the proposed method has a better convergence rate and optimality than the previous method.

Acknowledgement

This research was funded by the Directorate of Research, Technology, and Community Service, Directorate General of Higher Education, Research and Technology (Ditjen Diktiristek), Ministry of Education, Culture, Research and Technology (Kemdikbudristek), 2024 and supported by Universitas Mercu Buana.

References

Adriansyah, A., Suwoyo, H., Tian, Y., & Deng, C. (2019). IMPROVING THE WALL-FOLLOWING ROBOT PERFORMANCE USING PID-PSO CONTROLLER. *Jurnal Teknologi (Sciences & Engineering)*, *81*(3), 119–126. https://doi.org/10.11113/JT.V81.13098

Agrawal, S., & Shrivastava, V. (2017). Particle swarm optimization of BLDC motor with fuzzy logic controller for speed improvement. *8th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2017*. https://doi.org/10.1109/ICCCNT.2017.8204006

Al-Ansarry, S., & Al-Darraji, S. (2021). Hybrid RRT-A*: An Improved Path Planning Method for an Autonomous Mobile Robots. *Iraqi Journal for Electrical and Electronic Engineering*, *17*(1), 1–9. https://doi.org/10.37917/ijeee.17.1.13

Bailey, T., Nieto, J., Guivant, J., Stevens, M., & Nebot, E. (2006). Consistency of the EKF-SLAM algorithm. *IEEE International Conference on Intelligent Robots and Systems*, *1*, 3562–3568. https://doi.org/10.1109/IROS.2006.281644

Bastapure, A. A., Chiddarwar, S. S., & Goyal, A. (2023). A Comparative Study of A*, RRT, and RRT* Algorithm for Path Planning in 2D Warehouse Configuration Space. *Lecture Notes in Electrical Engineering*, *1066 LNEE*, 95–107. https://doi.org/10.1007/978-981-99-4634-1_8

Bingul, Z., Ertunc, H. M., & Oysu, C. (2005). Applying Neural Network to Inverse Kinematic Problem for 6R Robot Manipulator with Offset Wrist. *Adaptive and Natural Computing Algorithms*, 112–115. https://doi.org/10.1007/3-211-27389-1_27

Bozorg-Haddad, O., Solgi, M., & Loáiciga, H. A. (2017). Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization. In *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*. https://doi.org/10.1002/9781119387053

Caceres, C., Rosario, J. M., & Amaya, D. (2017). Approach of Kinematic Control for a Nonholonomic Wheeled Robot using Artificial Neural Networks and Genetic Algorithms. *2017 International Work Conference on Bio-Inspired Intelligence: Intelligent Systems for Biodiversity Conservation, IWOBI 2017 - Proceedings*. https://doi.org/10.1109/IWOBI.2017.7985533

Chen, C., Du, H., & Lin, S. (2017). Mobile robot wall-following control by improved artificial bee colony algorithm to design a compensatory fuzzy logic controller. *ECTI-CON 2017 - 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 856–859. https://doi.org/10.1109/ECTICon.2017.8096373

Chen, J., Zhao, Y., & Xu, X. (2021). Improved RRT-Connect Based Path Planning Algorithm for Mobile Robots. *IEEE Access*, *9*, 145988–145999. https://doi.org/10.1109/ACCESS.2021.3123622

Chen, W., & Qin, H. (2011). Path planning of mobile robot based on hybrid cascaded genetic algorithm. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 501–504. https://doi.org/10.1109/WCICA.2011.5970564

Dain, R. A. (1998). Developing Mobile Robot Wall-Following Algorithms Using Genetic Programming. *Applied Intelligence*, *8*(1), 33–41. https://doi.org/10.1023/A:1008216530547

El-Ghoul, S., Hussein, A. S., Wahab, M. S. A., Witkowski, U., & Ruckert, U. (2008). A Modified Multiple Depth First Search Algorithm for Grid Mapping Using Mini-Robots Khepera. *Journal of Computing Science and Engineering*, *2*(4), 321–338. https://doi.org/10.5626/jcse.2008.2.4.321

Ginting, S. E. (Sakaria), & Sembiring, A. S. (Abdul). (2019). Comparison of Breadth First Search (BFS) and Depth-First Search (DFS) Methods on File Search in Structure Directory Windows. *Login, 13*(1), 26–31. https://doi.org/10.24224/LOGIN.V13I1.23

Ginting, S. E., & Sembiring, A. S. (2019). *Jurnal Teknologi Komputer Comparison of Breadth First Search (BFS) and Depth-First Search (DFS) Methods on File Search in Structure Directory Windows*. *13*(1), 26–31.

Ibarra-Pérez, T., Ortiz-Rodríguez, J. M., Olivera-Domingo, F., Guerrero-Osuna, H. A., Gamboa-Rosales, H., & Martínez-Blanco, M. del R. (2022). A Novel Inverse Kinematic Solution of a Six-DOF Robot Using Neural Networks Based on the Taguchi Optimization Technique. *Applied Sciences (Switzerland)*, *12*(19). https://doi.org/10.3390/app12199512

Islam, F., Nasir, J., Malik, U., Ayaz, Y., & Hasan, O. (2012). RRT*-Smart: Rapid convergence implementation of RRT* towards optimal solution. *2012 IEEE International Conference on Mechatronics and Automation, ICMA 2012*, 1651–1656. https://doi.org/10.1109/ICMA.2012.6284384

Jeong, I. B., Lee, S. J., & Kim, J. H. (2015). RRT*-Quick: A motion planning algorithm with faster convergence rate. *Advances in Intelligent Systems and Computing*, *345*, 67–76. https://doi.org/10.1007/978-3-319-16841-8_7

Jiang, G., Luo, M., Bai, K., & Chen, S. (2017). A precise positioning method for a puncture robot based on a PSO-optimized BP neural network algorithm. *Applied Sciences (Switzerland)*, *7*(10). https://doi.org/10.3390/app7100969

Juang, C. F., Jhan, Y. H., Chen, Y. M., & Hsu, C. M. (2018). Evolutionary Wall-Following Hexapod Robot Using Advanced Multiobjective Continuous Ant Colony Optimized Fuzzy Controller. *IEEE Transactions on Cognitive and Developmental Systems*, *10*(3), 585–594. https://doi.org/10.1109/TCDS.2017.2681181

Kuffner, J. J., & La Valle, S. M. (2000). RRT-connect: an efficient approach to single-query path planning. *Proceedings - IEEE International Conference on Robotics and Automation*, *2*, 995–1001. https://doi.org/10.1109/ROBOT.2000.844730

Li, J., Liu, S., Zhang, B., & Zhao, X. (2014). RRT-A∗ Motion planning algorithm for non-holonomic mobile robot. *Proceedings of the SICE Annual Conference*, 1833–1838. https://doi.org/10.1109/SICE.2014.6935304

Li, L., Li, A., Tian, Y., Wang, W., Chen, W., Fan, Y., & Xi, F. (2018). A Robust Visual SLAM System Based on RGB-D Camera Used in Various Indoor Scenes. *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1632–1637.

Li, Q., Wang, J., Li, H., Wang, B., & Feng, C. (2022). Fast-RRT*: An Improved Motion Planner for Mobile Robot in Two-Dimensional Space. *IEEJ Transactions on Electrical and Electronic Engineering*, *17*(2), 200–208. https://doi.org/10.1002/tee.23502

Mashayekhi, R., Idris, M. Y. I., Anisi, M. H., & Ahmedy, I. (2020). Hybrid RRT: A semi-dual-tree RRT-based motion planner. *IEEE Access*, *8*, 18658–18668. https://doi.org/10.1109/ACCESS.2020.2968471

Mashayekhi, R., Idris, M. Y. I., Anisi, M. H., Ahmedy, I., & Ali, I. (2020). Informed RRT∗-Connect: An Asymptotically Optimal Single-Query Path Planning Method. *IEEE Access*, *8*, 19842–19852. https://doi.org/10.1109/ACCESS.2020.2969316

Min, H., Lin, Y., Wang, S., Wu, F., & Shen, X. (2015). Path planning of mobile robot by mixing experience with modified artificial potential field method. *Advances in Mechanical Engineering*, *7*(11), 6–10. https://doi.org/10.1177/1687814015619276

Naderi, K., Rajamaki, J., & Hamalainen, P. (2015). RT-RRT∗: A real-time path planning algorithm based on RRT∗. *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, MIG 2015*, *November*, 113–118. https://doi.org/10.1145/2822013.2822036

Nasir, J., Islam, F., Malik, U., Ayaz, Y., Hasan, O., Khan, M., & Muhammad, M. S. (2013). RRT*-SMART: A Rapid Convergence Implementation of RRT*. *International Journal of Advanced Robotic Systems*, *10*. https://doi.org/10.5772/56718

Ni, J., Wang, K., Huang, H., Wu, L., & Luo, C. (2016). Robot path planning based on an improved genetic algorithm with variable length chromosome. *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016*, 145–149. https://doi.org/10.1109/FSKD.2016.7603165

Noreen, I., Khan, A., & Habib, Z. (2016). A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms. *IJCSNS International Journal of Computer Science and Network Security*, *16*(10), 20–27. http://cloud.politala.ac.id/politala/1. Jurusan/Teknik Informatika/19. e-journal/Jurnal Internasional TI/IJCSNS/2016 Vol. 16 No. 10/20161004_A Compar ison of RRT, RRT and RRT - Smart Path Planning Algorithms.pdf

Rachmawati, D., & Gustin, L. (2020). Analysis of Dijkstra's Algorithm and A∗ Algorithm in Shortest Path Problem. *Journal of Physics: Conference Series*, *1566*(1). https://doi.org/10.1088/1742-6596/1566/1/012061

Ran, D., Ma, J., Peng, F., & Li, H. (2021). SUMMARY OF RESEARCH ON PATH PLANNING BASED ON A* ALGORITHM. *IET Conference Proceedings*, *2020*(2), 84–89. https://doi.org/10.1049/ICP.2021.1291

Sun, Z., Wang, J., & Meng, M. Q. H. (2022). Multi-Tree Guided Efficient Robot Motion Planning. *Procedia Computer Science*, *209*, 31–39. https://doi.org/10.1016/J.PROCS.2022.10.096

Suwoyo, H., Adriansyah, A., Andika, J., Shamsudin, A. U., & Zakaria, M. F. (2023). an Integrated Rrt*Smart-a* Algorithm for Solving the Global Path Planning Problem in a Static Environment. *IIUM Engineering Journal*, *24*(1), 269–284. https://doi.org/10.31436/iiumej.v24i1.2529

Suwoyo, H., Deng, C., Tian, Y., & Adriansyah, A. (2018). Improving a wall-following robot performance with a PID-genetic algorithm controller. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, *2018-Octob*, 314–318. https://doi.org/10.1109/EECSI.2018.8752907

Suwoyo, H., Hajar, M. H. I., Indriyanti, P., & Febriandirza, A. (2024). The use of Fuzzy Logic Controller and Artificial Bee Colony for optimizing adaptive SVSF in robot localization algorithm. *Sinergi (Indonesia)*, *28*(2), 231–240. https://doi.org/10.22441/sinergi.2024.2.003

Tian, Y., Suwoyo, H., Wang, W., & Li, L. (2019). An ASVSF-SLAM Algorithm with Time-Varying Noise Statistics Based on MAP Creation and Weighted Exponent. *Mathematical Problems in Engineering*, *2019*, 28–34. https://doi.org/10.1155/2019/2765731

Tian, Y., Suwoyo, H., Wang, W., Mbemba, D., & Li, L. (2019). An AEKF-SLAM Algorithm with Recursive Noise Statistic Based on MLE and EM. *Journal of Intelligent and Robotic Systems: Theory and Applications*. https://doi.org/10.1007/s10846-019-01044-8

Wang, H., Qi, X., Lou, S., Jing, J., He, H., & Liu, W. (2021). An efficient and robust improved a* Algorithm for path planning. *Symmetry*, *13*(11). https://doi.org/10.3390/sym13112213

Wang, H., Yan, Y., & Wang, D. W. (2010). A GA based SLAM with range sensors only. *2010 11th International Conference on Control Automation Robotics \& Vision*, 1796–1802.

Wen, S., Chen, X., Ma, C., Lam, H.-K., & Hua, S. (2015). The Q-learning obstacle avoidance algorithm based on EKF-SLAM for NAO autonomous walking under unknown environments. *Robotics and Autonomous Systems*, *72*, 29–36.

Wu, D., Wei, L., Wang, G., Tian, L., & Dai, G. (2022). APF-IRRT*: An Improved Informed Rapidly-Exploring Random Trees-Star Algorithm by Introducing Artificial Potential Field Method for Mobile Robot Path Planning. *Applied Sciences (Switzerland)*, *12*(21). https://doi.org/10.3390/app122110905

Wu, Z., Meng, Z., Zhao, W., & Wu, Z. (2021a). Fast-RRT: A RRT-Based Optimal Path Finding Method. *Applied Sciences 2021, Vol. 11, Page 11777*, *11*(24), 11777. https://doi.org/10.3390/APP112411777

Wu, Z., Meng, Z., Zhao, W., & Wu, Z. (2021b). Fast-RRT: A RRT-based optimal path finding method. *Applied Sciences (Switzerland)*, *11*(24). https://doi.org/10.3390/app112411777

Xinying, X., Jun, X., & Keming, X. (2006). Path planning and obstacle-avoidance for soccer robot based on artificial potential field and genetic algorithm. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, *1*, 3494–3498. https://doi.org/10.1109/WCICA.2006.1713018

Yu, F., Sun, Q., Lv, C., Ben, Y., & Fu, Y. (2014). A SLAM algorithm based on adaptive cubature Kalman filter. *Mathematical Problems in Engineering*, *2014*. https://doi.org/10.1155/2014/171958

Yusuf, L. I., & Musdholifah, A. (2024). A new hybrid parallel genetic algorithm for multi-destination path planning problem. *Indonesian Journal of Electrical Engineering and Computer Science*, *34*(1), 584–591. https://doi.org/10.11591/ijeecs.v34.i1.pp584-591

Zhang, D., Xu, Y., & Yao, X. (2018). An Improved Path Planning Algorithm for Unmanned Aerial Vehicle Based on RRT-Connect. *Chinese Control Conference, CCC*, *2018-July*, 4854–4858. https://doi.org/10.23919/ChiCC.2018.8483405

Zhou, Y., Zheng, L., & Li, Y. (2012). An improved genetic algorithm for mobile robotic path planning. *Proceedings of the 2012 24th Chinese Control and Decision Conference, CCDC 2012*, 3255–3260. https://doi.org/10.1109/CCDC.2012.6244515