Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 10, No. 02, June 2024.

SMART PROBABILISTIC ROAD MAP (SMART-PRM): FAST ASYMPTOTICALLY OPTIMAL PATH PLANNING USING SMART SAMPLING STRATEGIES

Muhammad Aria Rajasa Pohan and Jana Utama

(Received: 21-Dec.-2023, Revised: 27-Feb.-2024, Accepted: 14-Mar.-2024)

ABSTRACT

An asymptotically optimal path-planning guarantees an optimal solution if given sufficient running time. This research proposes a novel, fast, asymptotically optimal path-planning algorithm. The method uses five smart sampling strategies to improve the probabilistic road map (PRM). First, it generates samples using an informed search procedure. Second, it employs incremental search techniques on increasingly dense samples. Third, samples are generated around the best solution. Fourth, generated around obstacles. Fifth, it repairs the found route. This algorithm is called the Smart PRM (Smart-PRM). The Smart-PRM was compared to PRM, informed PRM and informed rapidly-exploring random tree-connect. Smart-PRM can generate the optimal path for any test case. The shortest distance between the start and goal nodes is the optimal path criterion. Smart-PRM finds the best path faster than competing algorithms. As a result, the Smart-PRM has the potential to be used in a wide variety of applications requiring the best path-planning algorithm.*

KEYWORDS

Probabilistic road map, Fast asymptotically optimal, Path planning, Intelligent sampling, Informed search.

1. INTRODUCTION

An algorithm for path planning is considered asymptotically optimal if it ensures that it will produce an optimal solution given a sufficient number of iterations or time [1]-[2]. The criteria for best solutions may be based on one or more conditions, such as the lowest fuel usage, lowest risk, comfort or shortest distance [3]. The shortest distance between the initial and goal nodes is used as the criterion for an optimal path in this study. Path-planning algorithms that provide optimal solutions are critical in a wide range of robotic applications [4], including automation processes in industries [5], robot navigation [6], driverless autonomous vehicles [7] and robotic surgery procedures [8]. These examples highlight the significance of optimal path-planning algorithms in addressing diverse robotic applications.

Several researchers have proposed asymptotically optimal path-planning algorithms; however, each algorithm exhibits distinct performance characteristics. One common parameter used to evaluate the performance of path-planning algorithms is the computational time required to generate an optimal path [9]-[11]. Karaman and Frazzoli introduced the Rapidly-exploring Random Tree (RRT*) algorithm, providing an asymptotically optimal solution [12]. Nonetheless, Qureshi et al. [13], J. Nasir et al. [14] and I. B. Jeong et al. [15] reported that the computational speed of RRT* in reaching optimal values still needs improvement. A factor contributing to the computational load of the RRT* algorithm is its necessity to sample throughout the entire search space.

To enhance the performance of the RRT* algorithm, Gammel et al. [16] proposed the Informed RRT* algorithm, which constrains the sampling area based on information from the currently known (yet non-optimal) paths. Wang et al. [17] modified the sampling method to enhance the search speed for an initial solution using a bio-inspired algorithm and an RRT algorithm. Mashayekhi et al. [18] combined the RRT-Connect and informed RRT* algorithms to develop a hybrid RRT approach. It is feasible to obtain the initial solution as rapidly as possible by combining the advantages of the two techniques. Informed RRT* has been coupled with the Dynamic Window Approach (DWA) by Dai et al. [19], while Ryu and Park [20] proposed using a grid-map structure in Informed RRT*. Meanwhile, Wu et al. [21] proposed that raising the APF-IRRT* algorithm's computational speed can assist in identifying the optimal solution faster than other algorithms. Aria [22] proposed updating the technique to become informed RRT*-Connect with local search to increase the informed RRT*'s convergence speed. Path-planning research based on informed sampling is still being developed.

M. A. R. Pohan and J. Utama are with Electrical Engineering Department, Universitas Komputer Indonesia, Bandung, Indonesia. Emails: muhammad.aria@email.unikom.ac.id and jana.utama@email.unikom.ac.id

Another asymptotically optimal path-planning algorithm is the Informed Probabilistic Road Map (PRM) algorithm proposed by the author in [23]. Aria reported that by combining informed searching with the PRM algorithm, the performance of the proposed algorithm can be enhanced by up to 25%. Ongoing research continues to improve the performance of the PRM algorithm. Chen et al. [24] proposed a new PRM sampling strategy to generate more suitable configurations for practical applications. Ravankar et al. [25] suggested the use of a Layered Hybrid PRM with an Artificial Potential Field (APF), while Liu et al. [26] proposed combining the PRM and D* algorithm.

This research proposes a new fast, asymptotically optimal path-planning algorithm called the Smart PRM (Smart-PRM) algorithm. The approach enhances the PRM algorithm through five smart sampling strategies. Test results demonstrate the Smart-PRM algorithm's ability to construct optimal paths across all scenarios. The computational time required for Smart-PRM to generate optimal paths surpasses that of PRM, informed RRT*-Connect and informed PRM algorithms. The Smart-PRM algorithm exhibits efficient convergence due to the incorporation of five smart sampling strategies. These include generating samples using an informed search procedure, employing incremental search techniques on increasingly dense samples, samples generated around the best solution, samples generated around obstacles and the algorithm repairing the found route using a wrapping procedure. The efficacy of each strategy is confirmed through testing, showcasing the Smart-PRM algorithm's potential for implementation in diverse robotic systems and autonomous vehicles.

While it is acknowledged that individual components of our proposed Smart-PRM algorithm draw upon existing techniques in motion planning, we contend that the integration and synergy of these strategies represent a novel and significant advancement in the field. Our approach synthesizes five distinct sampling strategies; namely an informed search procedure, incremental search techniques on increasingly dense samples, sample generation around the best solution, sample generation around obstacles and a route repair mechanism using the wrapping procedure. This amalgamation of strategies not only distinguishes our work, but also facilitates enhanced efficiency and performance compared to existing methods. Furthermore, our experimental results demonstrate a notable improvement in computational time and the ability to construct optimal paths across various scenarios when compared against traditional PRM, informed RRT*-Connect and informed PRM algorithms. The efficiency gains achieved by our Smart-PRM algorithm are particularly noteworthy, surpassing existing methods in terms of convergence speed and solution optimality.

This paper is organized as follows: Section 2 describes the design of the suggested Smart-PRM algorithm. This section describes the strategies used to improve PRM's performance. Section 3 contains the findings and discussion. Initially, the effects of each recommended technique on improving PRM performance are investigated. After that, the suggested Smart-PRM algorithm is compared to PRM, informed RRT*-Connect and informed PRM. Finally, Section 4 includes closing remarks.

2. Proposed Algorithm: Smart-PRM

The proposed algorithm enhances the PRM algorithm through five strategies. First, it generates samples using an informed search procedure. Second, it employs incremental search techniques on increasingly dense samples. Third, samples are generated around the best solution. Fourth, samples are generated around obstacles. Fifth, it repairs the found route using the wrapping procedure. Thus, the PRM algorithm will be repeated for several iterations. In iterations, before a path solution is found, the second and fourth strategies will be employed. However, after finding a path solution, the fifth, first, third and fourth strategies will be used. Sub-section from 2.1 to 2.5 will discuss each of those strategies. Sub-section 2.6 will discuss the complete algorithm of the proposed Smart-PRM.

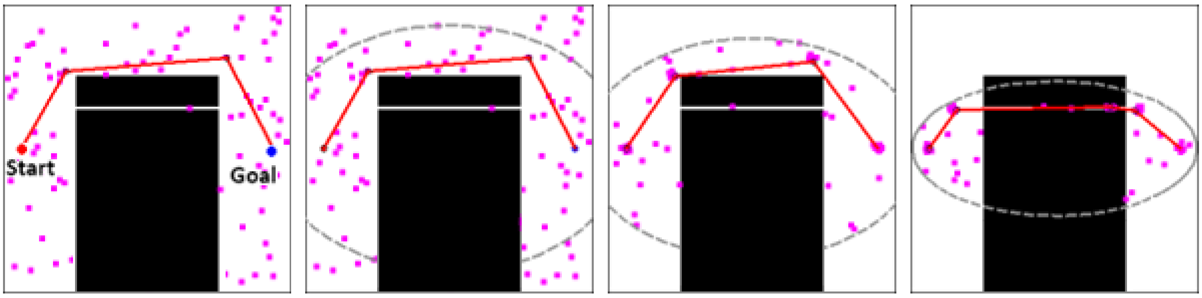
2.1 First Strategy: Informed Search Procedure for Sample Generation

This informed search procedure for sample generation emulates the informed search procedure in the informed RRT* algorithm proposed by Gammel et al. [16]. If a path solution connecting the start and goal nodes is successfully found during an iteration, an area is formed to restrict sample generation. This area takes the shape of an ellipsoid and its eccentricity depends on the length of the shortest-path solution found in that iteration. With the presence of this ellipsoidal area, the sample-generation process in the next iteration will only be carried out within this area. This area enhances the search concentration on regions with the potential to improve the quality of the path solution. Gammel et al. have demonstrated that once this ellipsoidal area is established, generating samples outside this area does not improve the quality of the path solution.

If a shorter-path solution is found in the next iteration, the size of this ellipsoidal area will decrease and the concentration of the path search will become more focused. Gammel et al. [27] claimed that using this method, the informed RRT* algorithm may obtain an optimal solution approximately 3.4 times faster than the RRT* algorithm.

An illustration of the informed search procedure for sample generation in the PRM algorithm is shown in Figure 1. In the first iteration, sample generation is randomly conducted throughout the area (Figure 1a). Then, using the created-sample nodes, Dijkstra's method [28] is used to find a path connecting the start and finish nodes. An example path successfully created by Dijkstra's algorithm is indicated by the red line in Figure 1a.

Once a path solution is found, an area is established to constrain the sample-generation area, represented by the grey ellipsoid in Figure 1b. Subsequently, the sample generation procedure is applied only within this ellipsoidal area in the next iterations, as shown in Figure 1c. Suppose that a shorter-path solution is found in the following iteration. In that case, the size of this ellipsoidal area will decrease further and the path search will be more concentrated, as depicted in Figure 1d. In the illustration of Figure 1, it can be observed that the optimal solution must pass through a narrow path. Using this first strategy, a solution approaching this optimal path can be achieved by the 10th iteration, as seen in Figure 1d. Therefore, a second strategy for enhancing the PRM algorithm is required to improve the convergence speed, where the search area begins with a small-sized ellipsoidal sub-set.



Iteration: 1	Iteration: 1	Iteration: 3	Iteration: 10
Cost: 182.7	Cost: 182.7	Cost: 173.34	Cost: 147.9
(a)	(b)	(c)	(d)

Figure 1. Illustration of the information-based sample generation process in the Smart-PRM algorithm: (a) Initial random sample generation, (b) Establishment of constraint area based on initial path solution, (c) Subsequent sample generation within the constrained area and (d) Decrease in constraint-area size with successive iterations, leading to a concentrated path search.

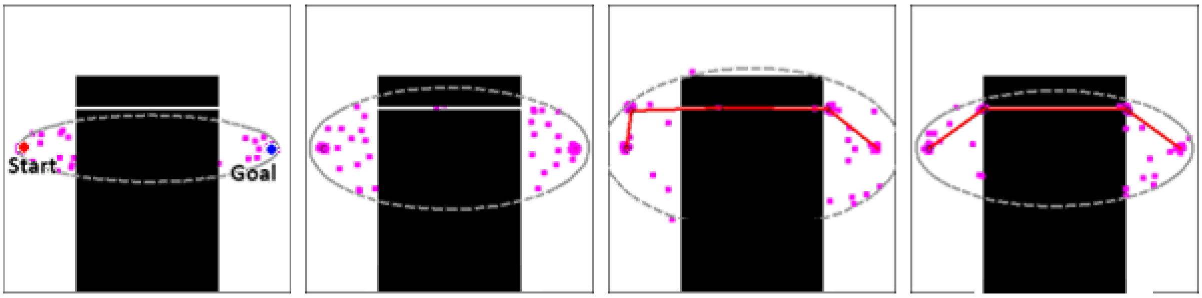
2.2 Second Strategy: Incremental Search Techniques on Increasingly Dense Samples

These incremental search techniques on increasingly dense samples emulate the strategies employed in initiating the incremental search techniques on increasingly dense samples within the Batch Informed Tree Star (BIT*) algorithm proposed by Gammell et al. in [27]. This second strategy is distinct from the standard informed RRT* algorithm. During the first iteration of the basic informed RRT* algorithm, no ellipsoidal area constrains the sample-generation area (as illustrated in Figure 1a). However, for the incremental search techniques on increasingly dense samples, initially, sample generation is randomly conducted throughout the entire area. Then, during the first iteration, a small-sized ellipsoidal area is created to restrict only the samples within that ellipsoidal area, which the Dijkstra algorithm will use to find a path connecting the start node with the goal node. If a path solution cannot be obtained by connecting the samples within that small ellipsoidal area, then the ellipsoidal area will be iteratively increased. With the ellipsoidal area growing larger, more dense samples will be within the ellipsoidal area and the Dijkstra algorithm will use more samples to find a path connecting the start node with the goal node.

Once a path solution is found, a new ellipsoidal area, the eccentricity of which depends on the length of that path solution, will be formed. The samples outside this new ellipsoidal area will be removed and transferred into this new ellipsoidal area, making the number of samples within the new ellipsoidal area denser. This ellipsoidal area will be reduced if a shorter-path solution is obtained and the samples outside the ellipsoidal area will be condensed into the new ellipsoidal area when a shorter-path solution is obtained. Gammell et al. reported that by employing these incremental search techniques on increasingly dense samples, the BIT* algorithm could achieve an optimal solution approximately 6.8 times faster than the RRT* algorithm.

An illustration of this second strategy is depicted in Figure 2. In the first iteration, sample generation is randomly conducted throughout the area. Following that, a small ellipsoidal area is created, as depicted in Figure 2a. The eccentricity of the ellipsoidal area constraining the sample-generation area is determined by a line connecting the start and goal nodes. Since the length of the path connecting the start and goal nodes is unknown in the first iteration, the line determining the eccentricity of the ellipsoid is based on an assumption. An assumption of a straight line connecting the start and goal nodes is used and then, a certain length tolerance is added to that line. This ellipsoidal area will restrict only the samples within it, which the Dijkstra algorithm will use to find a path connecting the start node with the goal node. If a path solution cannot be obtained by connecting the samples within this small ellipsoidal area, then the ellipsoidal area will be iteratively increased, as demonstrated in Figure 2b. With the growing ellipsoidal area, denser samples will be within the ellipsoidal area and the Dijkstra algorithm will utilize more samples to find a path connecting the start node with the goal node.

The procedure of gradually increasing the eccentricity of this ellipsoidal area is repeated until a path connecting the start and target nodes is obtained, as shown in Figure 2c. Once this path solution is discovered, the ellipsoidal area will not be extended in subsequent iterations. Instead, it will be lowered if a shorter-path solution is obtained, as shown in Figure 2d.



Iteration: 1	Iteration: 2	Iteration: 3	Iteration: 5
Cost: -	Cost: -	Cost: 154.8	Cost: 143.57
(a)	(b)	(c)	(d)

Figure 2. Illustration of incremental search techniques on increasingly dense samples in Smart-PRM algorithm: (a) Initial sample generation with a small ellipsoidal area, (b) Iterative expansion of the ellipsoidal area to include denser samples, (c) Finalization of the ellipsoidal area with a path solution and (d) Adjustment of the ellipsoidal area based on path optimization.

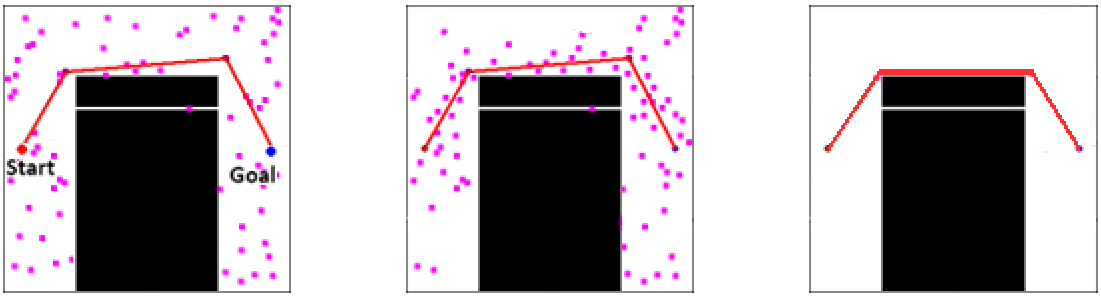
2.3 Third Strategy: Sample Generation around the Best Solution

The Smart-PRM algorithm's third strategy focuses on strategically generating samples around the identified best solution during algorithm iterations. This approach aims to refine the obtained path further and leverage the knowledge gained from the informed search.

The Smart-PRM algorithm commences the third strategy once a path solution connecting the start and goal nodes is successfully found. In this strategy, the algorithm utilizes 50% of the sampling points for exploiting the area around this best solution, while the remaining 50% of the sampling points explore the area based on the informed search procedure described in the first strategy.

By concentrating sampling efforts around the best solution, the Smart-PRM algorithm aims to identify alternative paths or variations that may contribute to a more optimal solution. This exploration has the potential to uncover paths that were initially not considered. The approach for exploiting the area around the optimum solution highlights the exploitation process in the RRT-ACS algorithm presented by Pohan et al. in [29]-[30].

An illustration of this third strategy can be seen in Figure 3. Initially, sample generation is conducted randomly throughout the area. Then, as shown in Figure 3a, Dijkstra's algorithm is used to find a path connecting the start and end nodes using the generated sample nodes. After the path is obtained, some sampling nodes are relocated around the best path. As shown in Figure 3b, there are more sampling nodes around the obtained best path compared to Figure 3a. Therefore, using sampling nodes around the best path has the potential to obtain a more optimal route, as demonstrated in Figure 3c.



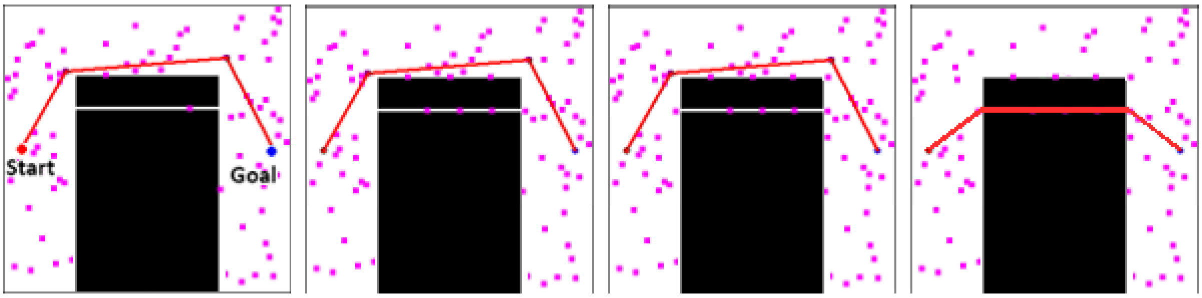
(a) (b) (c)

Figure 3. Illustration of sample generation around the best solution in Smart-PRM: (a) Pathfinding using Dijkstra's algorithm and initial sample generation, (b) Relocation of sampling nodes around the best path and (c) Potential optimization of route with sampling nodes around the best path.

2.4 Fourth Strategy: Sample Generation around Obstacles

The fourth strategy in the Smart-PRM algorithm focuses on strategically generating samples around obstacles encountered in the environment. After encountering newly identified obstacles during iterations, the Smart-PRM algorithm initiates the fourth strategy to systematically use several sampling points to explore and understand the areas around these obstacles. This strategy contributes to creating an optimal path, as optimal paths are often found around obstacles [31].

Strategic sampling around obstacles enhances the algorithm's flexibility and robustness, especially in scenarios where conventional approaches may face difficulties, such as in environments with narrow passages. An illustration of this fourth strategy can be seen in Figure 4. When the algorithm detects samples near an obstacle (purple points in the white gap in Figure 4a), the sides of the obstacle will be explored by more samples (as indicated by three purple points in the white gap in Figure 4b). When a sufficient number of areas on the sides of obstacles are explored by sample points (Figure 4c), there is the potential to discover a better path, as depicted in Figure 4d.



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

Figure 4. Illustration of sample generation around the obstacles in Smart-PRM: (a) Detection of samples near obstacles and initial exploration, (b) Increased exploration of obstacle sides by additional samples, (c) Sufficient exploration of areas around obstacles by sample points and (d) Potential discovery of better paths around obstacles.

2.5 Fifth Strategy: Route Repair Using the Wrapping Procedure

The path-correction strategy using the wrapping process emulates the wrapping-based Informed RRT* algorithm discussed in [32]. This wrapping process aims to find a shorter path by creating new nodes close to obstacles. An illustration of this fifth strategy is shown in Figure 5.

In the example case depicted in Figure 5, there is an initial red path consisting of four nodes. The wrapping process begins by creating a temporary node (X_{temp}) at node X_{i+1} or node X_2. Node X_{temp} is connected to node X_1 with a blue line, as shown in Figure 5a. Then, the position of node X_{temp} is advanced along the path connecting node X_{i+1} to node X_{i+2}, as in Figure 5b. The light blue area indicates the path covered by the blue line connecting X_1 to X_{temp}. The position of node X_{temp} continues to advance until an obstacle obstructs the blue line connecting X_1 to X_{temp}, as shown in Figure 5c. The position where the blue line meets the obstacle is marked as a new node for X_2 (denoted as X_2'). In the next iteration, the position of X_{temp} is advanced again, but because a new node X_2' has been found, the blue line now connects X_{temp} to X_2', as depicted in Figure 5d. The position of X_{temp} continues to advance until it reaches node X_{i+2} or node X_3. Once node X_3 is reached, the position of X_{temp} is further advanced along the path connecting node X_{i+2} to X_{i+3} (or node X_3 to X_4). This process is shown in Figure 5e. If the blue line connecting node X_2' to X_{temp} encounters an obstacle, the position where the blue line meets the obstacle is marked as a new node for X_3 (denoted as X_3'). This iteration continues until node X_{temp} reaches the destination node X_{goal}, as shown in Figure 5f. Figure 5g depicts a comparison of the initial path and the path produced by the wrapping operation. The red line is the original path and the blue line is the corrected/improved path as a result of the wrapping process. Green nodes represent new nodes created during the wrapping process.

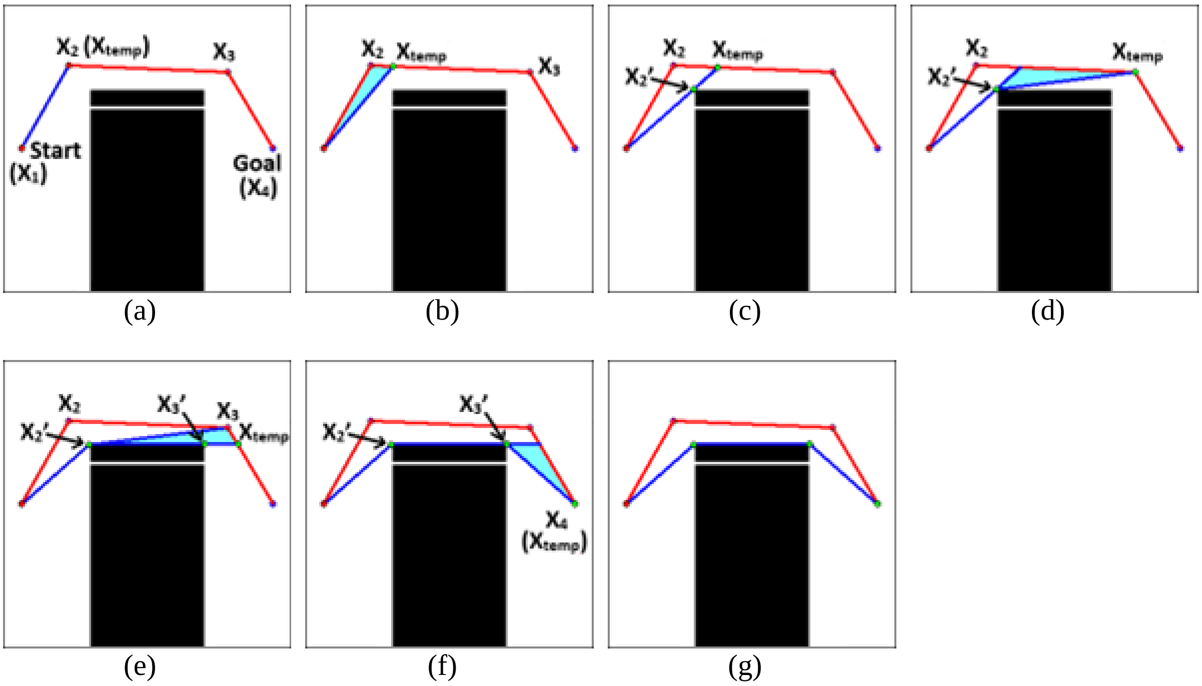


Figure 5. Illustration of the wrapping process to optimize the generated path. The red line represents the initial path, while the blue line represents the repairing/improved path: (a) Creation of temporary node (X_{temp}) and connection to X_1, (b) Advancement of X_{temp} along the path between nodes X_{i+1} and X_{i+2}, (c) Identification of obstacle obstruction and creation of new node X_2', (d) Continued advancement of X_{temp} towards node X_{i+2} or X_3, with connection to X_2', (e) Further advancement of X_{temp} along the path towards node X_{i+2} or X_3, with potential creation of new node X_3', (f) Completion of wrapping process when X_{temp} reaches destination node X_{goal} and (g) Comparison of initial and improved paths resulting from the wrapping operation.

2.6 Comprehensive Overview of the Smart-PRM Algorithm

The complete algorithm proposed is illustrated in Figures 6 and 7. The PRM algorithm consists of sample generation (lines 1-26 in Algorithm 1), roadmap construction (lines 30-37 in Algorithm 1) and path planning (the proposed algorithm uses Dijkstra's algorithm) connecting the start node to the goal node through the generated sample nodes (lines 38-39 in Algorithm 1).

The second strategy of the Smart-PRM algorithm is implemented in lines 3-16 of Algorithm 1. Setting the value of c_{max} to the minimum will create a small-sized ellipsoid subset area. If a path solution in this small area cannot be found, the ellipsoid area will be iteratively enlarged until a path solution connecting the start node to the goal node is found. The expansion process of the ellipsoid area during the path not being found is shown in line 44 of Algorithm 1.

The first strategy of the Smart-PRM algorithm is implemented in lines 17-25 of Algorithm 1 and Algorithm 2. In Algorithm 2, the generation of samples x_{rand} will only be done in the ellipsoid area surrounding x_{start} and x_{goal} with eccentricity depending on the length of c_{max}. Each time the algorithm finds a shorter path, the value of c_{max} will be updated (line 42 of Algorithm 1), therefore, the concentration of path search will increase.

Line 11 of Algorithm 2 implements the Smart-PRM algorithm's third strategy. Lines 27-29 of Algorithm 1 execute the Smart-PRM algorithm's fourth strategy. The fifth strategy of the Smart-PRM algorithm is implemented in Algorithm 1 (line 41).

Algorithm 1. $X_{sol} = \left(map, x_{start}, x_{goal}\right)$

1. $c_{max} \leftarrow \left\| x_{goal} - x_{start} \right\|_2$
2. $X_{sol} \leftarrow \emptyset$
3. **while** $|V_{init}| < n$ **do**
4. **repeat**
5. $x_{rand} \leftarrow$ RandomSampling(map)
6. $q \leftarrow x_{rand}$
7. **until** q **is** collision-free
8. $V_{init} \leftarrow V_{init} \cup \{q\}$
9. **end**
10. **while** $termination\_condition\_not\_meet$ **do**
11. **if** $X_{sol} = \emptyset$ **then**
12. **while** $q \in V_{init}$ **do**
13. **if** q $inside\_ellipsoid\_area$ $\left(x_{start}, x_{goal}, c_{max}\right)$
14. $V \leftarrow V \cup \{q\}$
15. **end**
16. **else**
17. $V \leftarrow \emptyset$
18. $E \leftarrow \emptyset$
19. **while** $|V| < n$ **do**
20. **repeat**
21. $x_{rand} \leftarrow$ Sample $\left(x_{start}, x_{goal}, c_{max}\right)$
22. $q \leftarrow x_{rand}$
23. **until** q **is** free of collisions
24. $V \leftarrow V \cup \{q\}$
25. **end**
26. **end**
27. **if** $new\_obstcale\_found$
28. $V \leftarrow V \cup \left\{q_{around\_obstacle}\right\}$
29. **end** **for all** $q \in V$ **do**
30. $N_q \leftarrow$ the neighbors of q chosen from V based on $dist$
31. **for all** $q' \in N_q$ **do**
32. **if** (q, q') is free of collisions **then**
33. $E \leftarrow E \cup \{(q, q')\}$
34. **end**
35. **end**
36. **end**
37. $T = (V, E)$
38. $X_{sol} \leftarrow Djikstra\left(q_{init}, q_{goal}, T\right)$
39. **if** $X_{sol} \neq \emptyset$ **then**
40. $X_{sol} \leftarrow Wrapping(X_{sol})$
41. $c_{max} \leftarrow min(x_{sol} \in X_{sol})\{Cost(x_{sol})\}$
42. **else**
43. $c_{max} \leftarrow c_{max} \times expansion\_coefficient$
44. **end**
45. **end**

Figure 6. Smart-PRM algorithm.

Algorithm 2. Sample $(x_{start}, x_{goal}, c_{max})$

if $|V| < n/2$ **then**
  $c_{min} \leftarrow \|x_{goal} - x_{start}\|_2$
  $x_{centre} \leftarrow (x_{goal} + x_{start})/2$
  $C \leftarrow \text{RotationToWorldFrame}(x_{start}, x_{goal})$
  $r_1 \leftarrow c_{max}/2$
  $\{r_i\}_{i=2,\ldots,n} \leftarrow \left(\sqrt{c_{max}^2 - c_{min}^2}\right)/2$
  $L \leftarrow \text{diag}\{r_1, r_2, \ldots, r_n\}$
  $x_{ball} \leftarrow \text{SampleUnitBall}$
  $x_{rand} \leftarrow (CLx_{ball} + x_{centre}) \cap X$
else
  $x_{rand} \leftarrow Sampling\_Near\_Best\_Path(X_{sol}, d)$
return x_{rand}

Figure 7. Sample-generation strategy in the smart-PRM algorithm.

3. Results and Discussion

Several tests were performed to validate the performance of the suggested path-planning algorithm. The first test aimed to verify the effectiveness of the first strategy of Smart-PRM, which generates samples using an informed search procedure. The second test was conducted to confirm the effectiveness of the second strategy of Smart-PRM, which employs incremental search techniques on increasingly dense samples. The third test aimed to verify the effectiveness of the third strategy of Smart-PRM, where samples are generated around the best solution. The fourth test was carried out to confirm the effectiveness of the fourth strategy of Smart-PRM, which generates samples around obstacles. The fifth test was conducted to verify the effectiveness of the fifth strategy of Smart-PRM, which repairs the found route using the wrapping procedure.

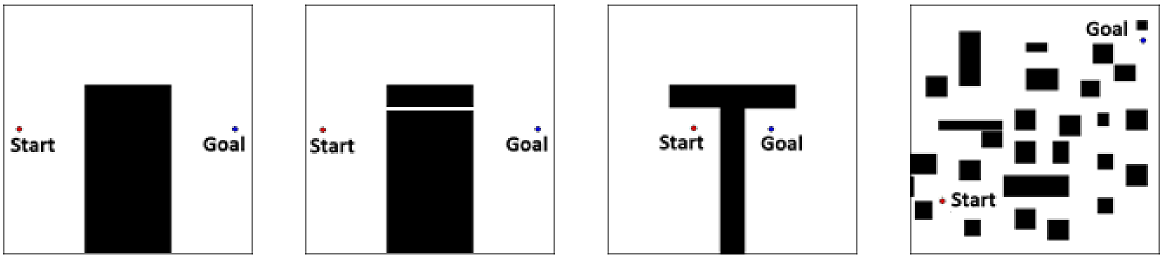
Meanwhile, the sixth test was developed to compare the Smart-PRM algorithm to the PRM algorithm [33], informed RRT*-Connect [18] and informed PRM [23]. The computational time for each approach to attain the optimal result was measured as a performance metric. All tests were done 40 times independently with the identical settings. The comparison was based on each algorithm's average performance across the 40 tests. All tests were carried out on a PC with a Core i5 3.20 GHz CPU and 4 GB RAM running Windows 10 64-bit. The Smart-PRM algorithm and the comparative algorithms were built in LabVIEW 7.1 using the Robotic Path-planning LabVIEW Libraries [34].

3.1 Experimental Scenarios

The proposed Smart-PRM method is compared to existing algorithms to validate its convergence speed and optimality performance. The performance of path-planning algorithms is evaluated using four common scenario cases. There are four scenarios: one with a single obstacle, one with narrow passages, one with a T-shaped obstacle and one with many randomly-scattered obstacles.

The testing scenario with a single obstacle is illustrated in Figure 8a. This scenario assesses whether an algorithm can produce an optimally convergent path. Mashayekhi et al. [18] utilized a testing scenario like this to evaluate their proposed path-planning algorithm. The testing scenario in an environment with narrow passages is depicted in Figure 8b. This scenario is employed to evaluate the effectiveness of path-planning algorithms when the goal node is hidden behind narrow passages. Gammel et al. [16] and Mashayekhi et al. [18] used testing scenarios like this.

The testing scenario in an environment with a T-shaped obstacle is shown in Figure 8c. This scenario assesses the algorithm's effectiveness in handling environments where the generated path needs to navigate turns. Islam et al. [35] used testing scenarios like this. The testing scenario in an environment with multiple randomly-scattered obstacles is illustrated in Figure 8d. This scenario is employed to evaluate the convergence speed of the path-planning algorithm. Gammel et al. [16] used testing scenarios like this.



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

Figure 8. Testing scenarios: (a) environment with a single obstacle, (b) environment with narrow passages, (c) environment with T-shaped obstacle, (d) environment with multiple randomly-scattered obstacles.

3.2 Verification of the First-strategy Effectiveness: Informed Search Procedure for Sample Generation

The first test aims to verify the effectiveness of the first strategy; namely, sample generation based on information. The test compares the basic PRM algorithm with the improved PRM algorithm using the first Smart-PRM strategy, which involves generating samples based on information. Testing is performed on the four scenarios mentioned in sub-section 3.1. The measured performance is the computation time of each algorithm to achieve the optimal path. The test results can be seen in Table 1. Furthermore, an analysis of the average-percentage comparison of convergence time to reach the optimal path for both algorithms can be found in Table 2.

Based on the data in Table 2, it can be observed that the average time of the improved PRM algorithm using the first Smart-PRM strategy is 5.49 times faster than the basic PRM algorithm. This result is consistent with the performance measurements of the informed RRT* algorithm (which employs the same algorithm-enhancement strategy) reported by Gammel et al. in [16]. Gammel et al. said that by limiting the sample-acquisition area to the subset ellipsoid area with eccentricity matching the length of the path solution in that iteration, the informed RRT* algorithm becomes 3.4 times faster than the RRT* algorithm in achieving the optimal path. This result verifies the effectiveness of the first strategy, which involves generating samples based on information, in improving the performance of the PRM algorithm.

Table 1. Comparison of improved PRM algorithm using the first strategy against the basic PRM algorithm (in seconds).

Scenario	Convergence time to achieve the optimal path	Improved PRM algorithm using the first strategy	Basic PRM
Scenario I: Single Obstacle	Best	0.13	12.90
	Average	2.20	13.10
	Worst	4.92	13.56
Scenario II: Narrow Passages	Best	0.42	2.79
	Average	1.60	8.61
	Worst	3.71	13.42
Scenario III: T-shaped Obstacle	Best	0.76	10.47
	Average	2.10	14.01
	Worst	6.07	30.97
Scenario IV: Multiple Obstacles	Best	0.49	4.61
	Average	1.55	6.14
	Worst	3.95	13.51

Table 2. Comparison of average convergence time of the improved PRM algorithm using the first strategy against the basic PRM algorithm.

Scenario	Comparison of convergence time (how many times faster)
Scenario I: Single Obstacle	5.95
Scenario II: Narrow Passages	5.38
Scenario III: T-shaped Obstacle	6.67
Scenario IV: Multiple Obstacles	3.96
Average	**5.49**

3.3 Verification of the Second-strategy Effectiveness: Incremental Search Techniques on Increasingly Dense Samples

The second test aims to verify the effectiveness of the second strategy. In this second test, the first strategy is not included; so, the enhancement of the PRM algorithm in this test is solely derived from the second strategy. The test compares the basic PRM algorithm with the improved PRM algorithm using the second S-PRM strategy. Testing is performed on the four scenarios mentioned in sub-section 3.1. The measured performance is the computation time of each algorithm to achieve the optimal path. The test results can be seen in Table 3. Furthermore, an analysis of the average-percentage comparison of convergence time to reach the optimal path for both algorithms can be found in Table 4.

Based on the data in Table 4, it can be observed that the average time of the improved PRM algorithm using the second Smart-PRM strategy is 7.48 times faster than the basic PRM algorithm. This result is consistent with what was reported by Gammel et al. [27] regarding the performance measurements of the BIT* algorithm (which employs a similar strategy to enhance the RRT* algorithm). Gammel et al. reported that by sampling in a small-sized sub-set ellipsoid area first, the BIT* algorithm can achieve an optimal solution 6.8 times faster than the RRT* algorithm. This result verifies the effectiveness of the second strategy; namely, using incremental search techniques on increasingly dense samples.

Table 3. Comparison of improved PRM algorithm using the second strategy against the basic PRM algorithm (in seconds).

Scenario	Convergence time to achieve the optimal path	Improved PRM algorithm using the second strategy	Basic PRM
Scenario I: Single Obstacle	Best	0.08	12.90
	Average	1.36	13.10
	Worst	3.05	13.56
Scenario II: Narrow Passages	Best	0.27	2.79
	Average	1.05	8.61
	Worst	2.41	13.42
Scenario III: T-shape Obstacle	Best	0.89	10.47
	Average	2.45	14.01
	Worst	7.12	30.97
Scenario IV: Multiple Obstacles	Best	0.31	4.61
	Average	0.97	6.14
	Worst	2.45	13.51

Table 4. Comparison of average convergence time of the improved PRM algorithm using the second strategy against the basic PRM algorithm.

Scenario	Comparison of convergence time (how many times faster)
Scenario I: Single Obstacle	9.66
Scenario II: Narrow Passages	8.20
Scenario III: T-shaped Obstacle	5.73
Scenario IV: Multiple Obstacles	6.33
Average	**7.48**

3.4 Verification of the Third-strategy Effectiveness: Sample Generation around the Best Solution

The third test aims to verify the effectiveness of the third strategy. In this third test, neither the first nor the second strategy is included; so, the enhancement of the PRM algorithm in this test is solely derived from the third strategy. The test compares the basic PRM algorithm with the improved PRM algorithm, which is enhanced only by adding the third Smart-PRM strategy. Testing is performed on the four scenarios mentioned in sub-section 3.1. The measured performance is the computation time of each algorithm to achieve the optimal path. The test results can be seen in Table 5. Furthermore, an analysis of the average-percentage comparison of convergence time to reach the optimal path for both algorithms can be found in Table 6.

Based on the data in Table 6, it can be observed that the average time of the PRM algorithm, when adding the third strategy, is 8.94 times faster than the basic PRM algorithm. This result verifies the effectiveness of the third strategy, which generates a sample around the best solution for improving the performance of the PRM algorithm.

Table 5. Comparison of improved PRM algorithm using the third strategy against the basic PRM algorithm (in seconds).

Scenario	Convergence time to achieve the optimal path	Improved PRM algorithm using the third strategy	Basic PRM
Scenario I: Single Obstacle	Best	0.05	12.90
	Average	1.20	13.10
	Worst	3.51	13.56
Scenario II: Narrow Passages	Best	0.18	2.79
	Average	0.91	8.61
	Worst	2.71	13.42
Scenario III: T-shaped Obstacle	Best	0.52	10.47
	Average	1.72	14.01
	Worst	6.02	30.97
Scenario IV: Multiple Obstacles	Best	0.20	4.61
	Average	0.86	6.14
	Worst	2.82	13.51

Table 6. Comparison of average convergence time of the improved PRM algorithm using the third strategy against the basic PRM algorithm.

Scenario	Comparison of convergence time (how many times faster)
Scenario I: Single Obstacle	10.96
Scenario II: Narrow Passages	9.48
Scenario III: T-shaped Obstacle	8.15
Scenario IV: Multiple Obstacles	7.18
Average	**8.94**

3.5 Verification of the Fourth-strategy Effectiveness: Sample Generation around Obstacles

The fourth test aims to verify the effectiveness of the fourth Smart-PRM strategy. The first, second and third strategies are not included in this fourth test. Therefore, this test's enhancement of the PRM algorithm is solely derived from the fourth strategy. The test compares the basic PRM algorithm with the improved PRM algorithm using the fourth Smart-PRM strategy. Testing is performed on the four scenarios mentioned in sub-section 3.1. The measured performance is the computation time of each algorithm to achieve the optimal path. The test results can be seen in Table 7. Furthermore, an analysis of the average percentage comparison of convergence time to reach the optimal path for both algorithms can be found in Table 8.

Based on the data in Table 8, it can be observed that the average time of the improved PRM algorithm, when using the fourth strategy, is 6.22 times faster than the basic PRM algorithm. This result verifies the effectiveness of the fourth strategy, which involves generating samples around obstacles, in improving the performance of the PRM algorithm.

3.6 Verification of the Fifth-strategy Effectiveness: Route Repair Using the Wrapping Procedure

The fifth test is aimed at verifying the effectiveness of the fifth Smart-PRM strategy. The first, second, third and fourth strategies are not included in this fifth test. Therefore, this test's enhancement of the PRM algorithm is solely derived from the fifth Smart-PRM strategy. The test compares the basic PRM algorithm with the improved PRM algorithm using the fifth strategy. Testing is performed on the four scenarios mentioned in sub-section 3.1. The measured performance is the computation time of each algorithm to achieve the optimal path. The test results can be seen in Table 9. Furthermore, an analysis of the average-percentage comparison of convergence time to reach the optimal path for both algorithms can be found in Table 10.

Table 7. Comparison of improved PRM algorithm using the fourth strategy against the basic PRM algorithm (in seconds).

Scenario	Convergence time to achieve the optimal path	Improved PRM algorithm using the fourth strategy	Basic PRM
Scenario I: Single Obstacle	Best	0.21	25.79
	Average	3.56	26.19
	Worst	7.96	27.12
Scenario II: Narrow Passages	Best	0.69	5.58
	Average	2.65	17.21
	Worst	6.12	26.84
Scenario III: T-shape Obstacle	Best	1.65	20.93
	Average	4.55	28.01
	Worst	13.19	61.93
Scenario IV: Multiple Obstacles	Best	0.79	9.22
	Average	2.52	12.28
	Worst	6.39	27.02

Table 8. Comparison of average convergence time of the improved PRM algorithm using the fourth strategy against the basic PRM algorithm.

Scenario	Comparison of convergence time (how many times faster)
Scenario I: Single Obstacle	7.37
Scenario II: Narrow Passages	6.49
Scenario III: T-shape Obstacle	6.16
Scenario IV: Multiple Obstacles	4.87
Average	**6.22**

Table 9. Comparison of improved PRM algorithm using the fifth strategy against the basic PRM algorithm (in seconds).

Scenario	Convergence time to achieve the optimal path	Improved PRM algorithm using the fifth strategy	Basic PRM
Scenario I: Single Obstacle	Best	0.03	12.90
	Average	1.01	13.10
	Worst	3.98	13.56
Scenario II: Narrow Passages	Best	0.09	2.79
	Average	0.79	8.61
	Worst	3.01	13.42
Scenario III: T-shaped Obstacle	Best	0.15	10.47
	Average	0.90	14.01
	Worst	4.92	30.97
Scenario IV: Multiple Obstacles	Best	0.10	4.61
	Average	0.78	6.14
	Worst	3.20	13.51

Table 10. Comparison of average convergence time of the improved PRM algorithm using the fifth strategy against the basic PRM algorithm.

Scenario	Comparison of convergence time (how many times faster)
Scenario I: Single Obstacle	12.97
Scenario II: Narrow Passages	10.89
Scenario III: T-shaped Obstacle	15.56
Scenario IV: Multiple Obstacles	7.87
Average	**11.82**

Based on the data in Table 10, it can be observed that the average time of the improved PRM algorithm, when using the fifth strategy, is 11.82 times faster than the basic PRM algorithm. This result verifies the effectiveness of the fifth strategy, which involves path refinement using the wrapping process, in improving the performance of the PRM algorithm.

3.7 Analyzing the Contribution of Each Sampling Strategy

Based on Tables 2, 4, 6, 8 and 10, a table illustrating the contribution of each sampling strategy, as demonstrated in Table 11, can be constructed. Table 11 presents a comparison of convergence time using each strategy against the basic PRM algorithm across various scenarios.

Table 11. Comparison of convergence time using each strategy against the basic PRM algorithm across various scenarios.

Scenario	**Comparison of convergence time (how many times faster) using each strategy against basic PRM**				
	1st Strategy	**2nd Strategy**	**3rd Strategy**	**4th Strategy**	**5th Strategy**
Scenario I	5.95	9.66	10.96	7.37	**12.97**
Scenario II	5.38	8.20	9.48	6.49	**10.89**
Scenario III	6.67	5.73	8.15	6.16	**15.56**
Scenario IV	3.96	6.33	7.18	4.87	**7.87**
Average	5.49	7.48	8.94	6.22	**11.82**

As depicted in Table 11, which compares the convergence time using each sampling strategy with the basic PRM algorithm across various scenarios, we can evaluate the relative contributions of each strategy to the overall algorithm performance. Upon examining the data, it is evident that based on the test results, the fifth strategy, Route Repair Using the Wrapping Procedure, demonstrates the most significant contribution to achieving superior performance across different scenarios.

3.8 Performance Comparison between the Smart-PRM Algorithm and Other Algorithms

The sixth test compares the Smart-PRM algorithm (which implements all five proposed techniques) to the informed RRT*-Connect and informed PRM algorithms. The test is run on the four scenarios described in sub-section 3.1. The calculation time of each algorithm to find the best path is assessed as performance. Table 12 displays the test results. Table 13 also contains a study of the average-percentage comparison of convergence time to reach the optimal path for both techniques.

Table 12. Comparison of the Smart-PRM algorithm against the informed RRT*-Connect and informed PRM algorithms (in seconds).

Scenario	**Convergence time to achieve the optimal path**	**Smart-PRM**	**Informed RRT*-connect**	**Informed PRM**
Scenario I: Single Obstacle	Best	0.02	0.56	0.13
	Average	0.60	3.24	2.19
	Worst	1.35	7.16	4.92
Scenario II: Narrow Passages	Best	0.06	1.87	0.42
	Average	0.47	11.63	1.62
	Worst	1.07	31.63	3.71
Scenario III: T-shaped Obstacle	Best	0.10	3.90	0.76
	Average	0.66	6.73	2.09
	Worst	2.70	19.35	6.07
Scenario IV: Multiple Obstacles	Best	0.06	3.52	0.49
	Average	0.43	13.67	1.57
	Worst	1.09	28.79	3.95

Table 13. Comparison of average convergence time of the Smart-PRM algorithm against the informed RRT*-Connect and informed PRM algorithms.

Scenario	**Comparison of convergence time (how many times faster)**	
	Informed RRT*-Connect	**Informed PRM**
Scenario I: Single Obstacle	5.36	3.62
Scenario II: Narrow Passages	24.92	3.46
Scenario III: T-shaped Obstacle	10.20	3.16
Scenario IV: Multiple Obstacles	31.78	3.64
Average	**18.06**	**3.47**

According to the statistics in Table 13, the Smart-PRM algorithm has an average time that is 18.06 times faster than the informed RRT* algorithm and 3.47 times faster than the informed PRM algorithm. Therefore, the Smart-PRM algorithm requires less computational time to design an optimal path than the informed RRT* and informed PRM algorithms. The results of the tests show that the Smart-PRM algorithm can create an optimal path in all test scenarios.

3.9 Evaluating the Stability of the Smart-PRM Algorithm

According to Xue [36], a path-planning algorithm is considered stable if it consistently produces the same path when planning the same task. Therefore, we will evaluate the stability of the Smart-PRM algorithm using the data provided in Table 14. Table 14 summarizes the statistical results of performance measurements obtained by Smart-PRM and other algorithms in various benchmark scenarios. Performance measurements include the best-path length, worst-path length, average-path length and standard deviation. A decrease in standard deviation indicates that the cost values of paths generated in each iteration are more consistent. As shown in Table 14, the standard deviation of the Smart-PRM algorithm is the smallest or relatively small compared to the standard deviation of other algorithms in each benchmark scenario. This smaller standard deviation suggests that the Smart-PRM algorithm tends to be more stable compared to other available algorithms

Table 14. Comparison of algorithm stability across various benchmark scenarios. Best results are highlighted for each section.

Scenario	Algorithm	Best	Worst	Mean	Std
Scenario I: Single Obstacle	Smart-PRM	**285.73**	**285.73**	**285.73**	**0**
	Informed RRT*-Connect	**285.73**	286.00	**285.73**	**0**
	Informed PRM	**285.73**	286.00	**285.73**	**0**
Scenario II: Narrow Passages	Smart-PRM	**258.84**	**259.26**	**258.84**	**0.001**
	Informed RRT*-Connect	**258.84**	262.40	259.89	0.004
	Informed PRM	**258.84**	259.89	259.47	**0.001**
Scenario III: T-shaped Obstacle	Smart-PRM	**275.54**	**275.54**	**275.54**	**0**
	Informed RRT*-Connect	277.42	280.70	279.06	0.004
	Informed PRM	**275.54**	278.35	276.24	0.003
Scenario IV: Multiple Obstacles	Smart-PRM	**307.35**	**307.79**	**307.57**	**0.007**
	Informed RRT*-Connect	307.27	314.86	309.41	0.08
	Informed PRM	308.56	311.39	309.78	0.023

3.10 Example Application

As an example of an application requiring fast asymptotically optimal path planning, we find that our algorithm, with its fast convergence, would be highly beneficial in the implementation of autonomous vehicles. The need for algorithms with fast convergence is paramount in traffic-safety contexts, where optimal path planning and rapid response to unforeseen situations are crucial. For instance, in Figure 9, we illustrate a scenario where an autonomous vehicle encounters a curve on the road while pedestrians are crossing unexpectedly. In such situation, autonomous vehicles must be able to respond quickly to plan alternative safe routes and avoid potential accidents. This study can be used as a reference for the current issues in vehicle automation, as discussed in previous studies [37]-[40].

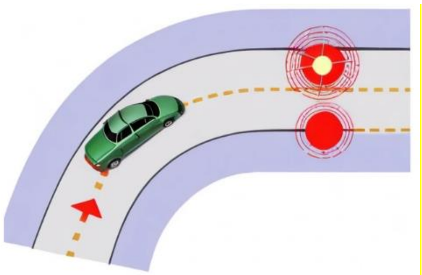


Figure 9. An illustration where autonomous vehicles (green car) must be able to quickly plan alternative routes when sudden changes in environmental conditions are encountered, such as sudden pedestrian crossings (illustrated by the red circle).

4. CONCLUSIONS

This research proposes a new fast, asymptotically optimal path-planning algorithm called the Smart-PRM algorithm. The method is improving the PRM algorithm. The results of the tests reveal that the Smart-PRM algorithm can provide optimal pathways in all test circumstances. The Smart-PRM algorithm takes less processing time to construct an optimal path than the PRM, informed PRM and informed RRT*-connect algorithms. The Smart-PRM algorithm can have good convergence speed, because it uses five smart sampling strategies. First, it generates samples using an informed search procedure. Second, it employs incremental search techniques on increasingly dense samples. Third, samples are generated around the best solution. Fourth, samples are generated around obstacles. Fifth, it repairs the found route using the wrapping procedure. The effectiveness of each strategy has been verified through test results. Thus, the smart-PRM algorithm has the potential to be implemented in various applications that need an optimal path-planning algorithm.

REFERENCES

[1] O. Salzman and D. Halperin, "Asymptotically Near-Optimal RRT for Fast, High-quality Motion Planning," IEEE Transactions on Robotics, vol. 32, no. 3, pp. 473-483, DOI: 10.1109/TRO.2016.2539377, June 2016.

[2] M. Larrañaga, M. Assaad, A. Destounis and G. S. Paschos, "Asymptotically Optimal Pilot Allocation over Markovian Fading Channels," IEEE Transactions on Information Theory, vol. 64, no. 7, pp. 5395-5418, DOI: 10.1109/TIT.2017.2772814, July 2018.

[3] I. Noreen, A. Khan and Z. Habib, "Optimal Path Planning Using RRT* Based Approaches: A Survey and Future Directions," Int. J. of Advanced Computer Science and Applications, vol. 7, no. 11, pp. 97-107, DOI: 10.14569/ijacsa.2016.071114, Jan. 2016

[4] M. A. Hossain and I. Ferdous, "Autonomous Robot Path Planning in Dynamic Environment Using a New Optimization Technique Inspired by Bacterial Foraging Technique," Robotics and Autonomous Systems, vol. 64, pp.137-141, 2015.

[5] C. Friedrich, A. Csiszar, A. Lechler and A. Verl, "Efficient Task and Path Planning for Maintenance Automation Using a Robot System," in IEEE Transactions on Automation Science and Engineering, vol. 15, no. 3, pp.1205-1215, DOI: 10.1109/TASE.2017.2759814, July 2018.

[6] H. Yang, J. Qi, Y. Miao, H. Sun and J. Li, "A New Robot Navigation Algorithm Based on a Double-layer Ant Algorithm and Trajectory Optimization," IEEE Transactions on Industrial Electronics, vol. 66, no. 11, pp. 8557-8566, 2018.

[7] Y. Rasekhipour, A. Khajepour, S. -K. Chen and B. Litkouhi, "A Potential Field-based Model Predictive Path-planning Controller for Autonomous Road Vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 5, pp.1255-1267, DOI: 10.1109/TITS.2016.2604240, May 2017.

[8] M. Diana and J. Marescaux, "Robotic Surgery," British Journal of Surgery, vol. 102, no. 2, pp. e15–e28, DOI: 10.1002/bjs.9711, January 2015

[9] M. Elbanhawi and M. Simic, "Sampling-based Robot Motion Planning: A Review," IEEE Access, vol. 2, pp. 56-77, DOI: 10.1109/ACCESS.2014.2302442, 2014.

[10] Y. Yang, J. Pan and W. Wan, "Survey of Optimal Motion Planning," IET Cyber-systems and Robotics, vol. 1, no. 1, pp.13-19, 2019.

[11] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi and I. Ahmedy, "Hybrid RRT: A Semi-dual-tree RRT-based Motion Planner," IEEE Access, vol. 8, pp.18658-18668, DOI: 10.1109/ACCESS.2020.2968471, 2020.

[12] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," Int. J. of Robotics Research, vol. 30, no. 7, pp.846-894, Jun. 2011.

[13] H. Qureshi et al., "Triangular Geometry Based Optimal Motion Planning Using RRT*-motion Planner," Proc. of 2014 IEEE 13th Int. Workshop Adv. Motion Control (AMC), pp.380-385, DOI: 10.1109/AMC.2014.6823312, Yokohama, Japan, 2014.

[14] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan and M. S. Muhammad, "RRT*-SMART: A Rapid Convergence Implementation of RRT," Int. Journal of Advanced Robotic Systems, vol. 10, no. 7, p. 299, 2013.

[15] I. B.Jeong, S. J. Lee and J. H. Kim, "Quick-RRT*: Triangular Inequality-based Implementation of RRT* with Improved Initial Solution and Convergence Rate," Expert Systems with Applications, vol. 123, pp. 82-90, 2019.

[16] J. D. Gammell, T. D. Barfoot and S. S. Srinivasa, "Informed Sampling for Asymptotically Optimal Path Planning," IEEE Transactions on Robotics, vol. 34, no. 4, pp. 966-984, Aug. 2018.

[17] J. Wang, W. Chi, M. Shao and M. Meng, "Finding a High-quality Initial Solution for the RRTs Algorithms in 2D Environments", Robotica, vol. 37, no. 10, pp. 1677-1694, 2019.

[18] R. Mashayekhi, M. Y. I. Idris, M. H. Anisi, I. Ahmedy and I. Ali, "Informed RRT*-Connect: An Asymptotically Optimal Single-Query Path Planning Method," IEEE Access, vol. 8, pp. 19842-19852, DOI: 10.1109/ACCESS.2020.2969316, 2020.
[19] J. Dai, D. Li, J. Zhao and Y. Li, "Autonomous Navigation of Robots Based on the Improved Informed-RRT Algorithm and DWA," Journal of Robotics, vol. 2022, Article ID: 3477265, pp.1-9, 2022.
[20] H. Ryu and Y. Park, "Improved Informed RRT* Using Gridmap Skeletonization for Mobile Robot Path Planning," Int. J. of Precision Engineering and Manufacturing, vol. 20, no. 11, pp.2033-2039, 2019.
[21] 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," Applied Sciences, vol. 12, no. 21, p. 10905, 2022.
[22] M. Aria, "Path Planning Algorithm Using Informed Rapidly Exploring Random Tree*-Connect with Local Search," J. of Eng. Science and Technology, Special Issue on INCITEST, pp. 50-57, 2020.
[23] M. Aria, "Optimal Path Planning Using Informed Probabilistic Road Map Algorithm," Journal of Engineering Research, ASSEEE Special Issue, pp.1-15, DOI: 10.36909/jer.ASSEEE.16105, 2021.
[24] G. Chen, N. Luo, D. Liu, Z. Zhao and C. Liang, "Path Planning for Manipulators Based on an Improved Probabilistic Roadmap Method," Robotics and Computer-Integrated Manufacturing, vol. 72, pp.102196, DOI: 10.1016/j.rcim.2021.102196, 2021.
[25] A. Ravankar, A. Ravankar, T. Emaru and Y. Kobayashi, "HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots," IEEE Access, vol. 8, pp.221743-221766, DOI: 10.1109/ACCESS.2020.3043333, 2020.
[26] C. Liu, S. Xie, X. Sui, Y. Huang, X. Ma, N. Guo and F. Yang, "PRM-D* Method for Mobile Robot Path Planning," Sensors, vol. 23, no. 7, p. 3512, 2023.
[27] J. D. Gammell, T. D. Barfoot and S. S. Srinivasa, "Batch Informed Trees (BIT*): Informed Asymptotically Optimal Anytime Search," Int. J. of Robotics Research,, vol. 39, no. 5, pp. 543–567, DOI: 10.1177/0278364919890396, Jan. 2020.
[28] M. Iqbal, K. Zhang, S. Iqbal and I. Tariq, "A Fast and Reliable Dijkstra Algorithm for Online Shortest Path," SSRG Int. J. of Computer Science and Engineering, vol. 5, no. 12, pp. 24–27, DOI: 10.14445/23488387/ijcse-v5i12p106, Dec. 2018.
[29] M. A. R. Pohan, B. R. Trilaksono, S. P. Santosa and A. S. Rohman, "Path Planning Algorithm Using the Hybridization of the Rapidly-exploring Random Tree and Ant Colony systems," IEEE Access, vol. 9, pp. 153599–153615, DOI: 10.1109/access.2021.3127635, Jan. 2021.
[30] M. A. R. Pohan and J. Utama, "Efficient Sampling-based for Mobile Robot Path Planning in a Dynamic Environment Based on the Rapidly-exploring Random Tree and a Rule-template Sets," Int. J. of Engineering, vol. 36, no. 4, pp. 797-806, 2023.
[31] C. Scheffer and J. Vahrenhold, "Approximate Shortest Distances among Smooth Obstacles in 3D," J. of Computational Geometry, vol. 10, no. 1, pp. 389-422, 2019.
[32] M. A. R. Pohan, "Asymptoticcaly-Optimal Path Planning Using the Improved Probabilistic Road Map Algorith," Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali and Elektronika Terapan, vol. 10, no. 2 , pp. 116-127, 2022.
[33] M. I. Abdulakareem and F. A. Raheem, "Development of Path Planning Algorithm Using Probabilistic Roadmap Based on Ant Colony Optimization," Engineering and Technology J., vol. 38, no. 3, pp. 343-351, 2020.
[34] M. A. R. Pohan, "LabVIEW Libraries Untuk Algoritma Perencanaan Jalur Robotik," Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali and Elektronika Terapan, vol. 10, no. 1, pp. 47-62, 2022.
[35] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan and M. S. Muhammad, "RRT*-SMART: A Rapid Convergence Implementation of RRT," Int. J. of Advanced Robotic Systems, vol. 10, no. 7, p. 299, 2013.
[36] Y. Xue, X. Zhang, S. Jia, Y. Sun and C. Diao, "Hybrid Bidirectional Rapidly-exploring Random Trees Algorithm with Heuristic Target Graviton," Proc. of 2017 IEEE Chinese Automation Congress (CAC), pp. 4357-4361, DOI: 10.1109/CAC.2017.8243546, Jinan, China, 2017.
[37] M. Al-Khalil, R. Abu-Rhayyem, A. Hammoudeh and T. A. Edwan, "Unmanned Ground Vehicle with Virtual Reality Vision,"Jordanian Journal of Computers and Information Technology (JJCIT), vol. 4, no. 1, pp. 58-79, DOI: 10.5455/jjcit.71-1510942341, April 2018.
[38] M. Aria and J. Utama, "An Autonomous Parking System Using the Hybridization of the Rapidly-Exploring Random Trees Star and Ant Colony System," Journal of Advanced Research in Applied Sciences and Engineering Technology, vol. 31, no. 1, pp. 291-297, 2023.
[39] M. A. R. Pohan and J. Utama, "Efficient Autonomous Road Vehicles Local Path Planning Strategy in Dynamic Urban Environment Using RRT-ACS, Bi-directional Rule Templates and Configuration Time-Space," Journal of Engineering Science and Technology, vol. 18, no. 5, pp. 2388-2397, 2023.
[40] M. Aria, "New Sampling Based Planning Algorithm for Local Path Planning for Autonomous Vehicles," J. of Engineering Science and Technology, Special Issue on INCITEST, pp. 66-76, 2019.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 10, No. 02, June 2024.

ملخص البحث:

يؤدّي التّخطيط لإيجاد المسار الأمثل إلى ضمان الحصول على الحلّ الأمثل إذا ما أعطي النّظام الوقت الكافي للعمل.

يهدف هذا البحث إلى اقتراح خوارزميةٍ جديدةٍ وسريعةٍ للتّخطيط للمسار الأمثل. وتستخدم هذه الطريقة خمس استراتيجيات ذكية لأخذ العينات من أجل تحسين خريطة الطّريق لإيجاد المسار الأمثل. فأولاً، يتمّ إنتاجُ العينات باستخدام إجراءات بحثٍ قائمة على المعلومات. وثانياً، يتمّ استخدام تقنيات بحثٍ متزايدة على العينات متزايدة الكثافة. وثالثاً، يجري إنتاج العينات حوْل الحلّ الأمثل. أمّا الاستراتيجية الرابعة فتتمثل في أخذ العينات حوْل العوائق، بينما تقوم الاستراتيجية الخامسة على إصْلاح المسارِ الذي تمّ إيجاده.

تسمّى الخوارزمية المقترحة في هذا البحث خوارزمية (PRM) الذّكية، وقد جرت مقارنتها بعددٍ من الخوارزميات الأخرى، وتبين أنّ بإمكان الحوارزمية المقترحة إيجاد المسار الامثل لأيّ حالةٍ من حالات الاختبار. وكان معيار الأداء هو أقصر مسافةٍ بين عقدة البداية وعقدة الهدف. وباستطاعة الخوارزمية المقترحة إيجاد المسار الأمثل على نحوٍ أسرع من الخوارزميات الأخرى، وذلك يجعلها مرشّحةً للاستخدام في مدى واسعٍ من التّطبيقات التي تحتاج الى خوارزمية لإيجاد المسار الأمثل، ومنها المركبات ذاتية القيادة على سبيل المثال.