Proceedings of the 42nd Chinese Control Conference
July 24-26, 2023, Tianjin, China

HNSRRT*:A Path Planning Algorithm Based On Heuristic Non-uniform Sampling Method In Complex Obstacle Environment

Zhiwen Xu[1], Hui Zhang[2], Bo Chen[2], Xidong Zhou[2], Songtao Yin[1], Lian Yang[2],

1. College of Electrical and Information Engineering, Changsha University of Science and Technology, Hunan, China
E-mail: 237971195@qq.com, 741170053@qq.com; 531535977@qq.com

2. College of Robotics, and Robot Visual Perception and Control Technology National Engineering Research Center, Hunan University, Hunan, China
E-mail: zhanghuihby@126.com; cb233cb@163.com; 13699648817@163.com

Abstract: The traditional sampling-based algorithm such as Rapidly Random-exploring Tree (RRT) and various varieties have achieved tremendous success in the area of path planning. However, their excessive exploration in the state space leads to long time to find the optimal solution, large memory usage and cannot guarantee the quality of the planned path(generally evaluated by the cost of search time and the length of path) in sophisticated space. In this article, we propose an optimal path planning algorithm based on heuristic non-uniform sampling, namely the HNSRRT*, which successfully plans path in complex obstacle environments with optimal length and minimum time cost. The HNSRRT* utilizes heuristic function to generate non-uniform sampling distribution by Gaussian distribution,and constraints on sampling points can reduce the time wasted and path length increase caused by excessive exploration. We test the proposed HNSRRT* in 2D and 3D complex obstacle environment,comparing it with the three traditional sampling-base algorithms. The simulation results indicated that the effectiveness and efficiency of HNSRRT* and have an obvious improvement in term of time cost, path length compared with the existing algorithms.

Key Words: Robot, Path planning, Non-uniform sampling, RRT, Complex obstacle environment

1 INTRODUCTION

In recent decades, with the in-depth research of many researchers in the field of robot, robot path planning becomes one of the basic topics in the research of robot technology [1]. Path planning is not only applicable to the field of mobile robot, but also to the scope of unmanned aerial vehicle, manipulator etc. [2], which aims to plan a secure minimum length path in complex environment. A great number of algorithms have been proposed to deal with this problem, for instance, Artificial Potential Field(APF) [3] method, A* algorithm [4]based on graph search, sampling-based rapidly random-exploring tree algorithm(RRT) [5] and its variants, and Probabilistic RoadMap [6] method(PRM). The artificial potential field method is easy to fall into the trap of local minimum value in complex environment, and the A* algorithm has poor performance when the area of grid map increases. Sampling-based algorithms is widely used because of its high success rate in searching effective paths in complex environment. RRT is a sampling-based single-query path planning method [7], which has the advantages of probability completeness and prefect scalability [8]-[9]. However, the sampling-based algorithm is inseparable from the low search efficiency [10], which leads to a longer time to obtain the optimal solution. Therefore, in our daily applications like drones, intelligent vehicles, and underwater vehicles, it is critical to rapidly calculate an optimal path to save limited energy.

There is a plentiful literature on methods to solve the sampling efficiency problem of sample-based algorithm.Karaman et al. proposed RRT* [11], who introduced path cost between new node and parent node, but RRT* need a great deal of iterations to plan a path from start position to goal position. Kuffner et al.proposed RRT-Connect [12]to improve the efficiency of path search by simultaneously expanding the random tree with the start point and target point as root nodes. However, RRT-Connect has problem such as long path detour due to abundant turns of extended tree. Ahmed et al. proposed P-RRT* [13], which incorporates APF(Artificial potential fields) into RRT* to solve convergence rate in iteration processing. Wang et al.proposed Neural RRT* [14], who used a CNN model to train the probability distribution of optimal paths in the search space to conduct the sampling process. Wang et al.proposed GMR-RRT* [15], which learns navigational behavior and plan high-quality paths for robots from human demonstrations go through using GMR to capture key features of human demonstrations to form a probability density distribution of human trajectories in the current environment. However, the RRT* algorithm combined with deep learning or machine learning requires preprocessing and need a great deal of training samples, its poor generalization is not convenient in practical application.

In this paper, we propose a path planning algorithm based on heuristic non-uniform sampling method to solve in view of the above problems. The main contributions of this paper are as follows:

1.A noval choose the minimal cost of node method based

This work was supported in part by the National Key RD Program of China under Grant 2021ZD0114503, the Major Research plan of the National Natural Science Foundation of China (Grant No. 92148204), the National Natural Science Foundation of China under Grants 62027810, 61971071, 62133005, Hunan Leading Talent of Technological Innovation under Grant 2022RC3063, Hunan Science Fund for Distinguished Young Scholars under Grant 2021JJ10025, Hunan Key Research and Development Program under Grants 2021GK4011,2022GK2011, Changsha Science and Technology Major Project under Grant kh2003026, Joint Open Foundation of state Key Laboratory of Robotics under Grant 2021-KF-22-17, China University industry-University-research Innovation Fund under Grant 2020HYA06006.

on the heuristic function, which can optimize path cost.

2.A spatial sampling from optimal node to goal node bias non-uniform sampling method with Gaussian distribution is proposed to improve the sampling efficiency.

3.A proof of improvements on the cost of time and path by simulation.

The remainder of this paper is organized as follows. Section 2 introduces the background, and Section 3 presents the details of the HNSRRT* algorithm. Experiments and results show the performance comparison with existing RRT related algorithms in Section 4. The conclusions of this article in Section 5.

2 BACKGROUND

In this part, we formulate the problem in path planning and provide detail of RRT* algorithm.

2.1 Problem Formulation

The notations for fundamental path planning problem are denoted as follows. Let $\mathcal{X} \in \mathbb{R}^{\mathrm{n}}$ to be the state space,the free space and obstacle region are defined as $\mathcal{X}_{\mathrm{free}}$ and $\mathcal{X}_{\mathrm{obs}}$, respectively. Let $\mathrm{x}_{\mathrm{start}} \in \mathcal{X}$ and $\mathrm{x}_{\mathrm{goal}} \in \mathcal{X}$ are denoted as the start state and goal state, respectively.The collision-free path is denoted as σ:[0,1]$\rightarrow \mathcal{X}_{\mathrm{free}}$, which is defined as a collision-free path with σ (0)=$\mathrm{x}_{\mathrm{start}}$ and σ (1)=$\mathrm{x}_{\mathrm{goal}}$. Let $\sum$ represent the set of feasible paths. If $\sum \notin \emptyset$, the optimal planning problem is to search for a path in the set of all feasible paths by a cost function $c(\sigma)$, the optimal path planning problem [16] is defined as:

$$\sigma^{*}=\arg\min_{\sigma \in \sum}\{c(\sigma)|\sigma(0)=\mathrm{x}_{\mathrm{start}}, \sigma(1)=\mathrm{x}_{\mathrm{goal}},$$
$$\forall \mathcal{W} \in [0,1], \sigma(\mathcal{W}) \in \mathcal{X}_{\mathrm{free}}\} \tag{1}$$

For an arbitrary path σ, a shorter path length between start position and goal position represent the smaller value of cost function.

Algorithm 1: RRT^{*}

```
Input: x_start, x_goal, Stepsize and Map
Output: P
N ← x_start, E ← ∅, P = (N, E);
for i=1...N do
    x_rand ← UniformSample();
    x_nearest ← Nearest(P, x_rand);
    x_new ← Steer(x_nearest, x_rand);
    if ObstacleFree(x_new, x_nearest) then
        P ← Extend(P, x_new);
        X_near ← NearestNeighbors(P, x_new, r);
        x_parent ← ChooseParent(x_new, X_near);
        P ← Rewire(P, x_new, x_parent);
    end
    if |x_nearest − X_goal| < Stepsize then
        Return P;
    end
end
Return failure;
```

2.2 RRT*Algorithm

The RRT algorithm utilizes tree extension to find a start point to the target point. At each iteration, $\mathrm{x}_{\mathrm{rand}}$ is obtained by random sampling in the state space. The node nearest to the sampling point in the tree is sought for connection, and $\mathrm{x}_{\mathrm{new}}$ is obtained by a steering function and the stepsize. If the edge connecting $\mathrm{x}_{\mathrm{new}}$ to $\mathrm{x}_{\mathrm{nearest}}$ does not collide with the obstacle then $\mathrm{x}_{\mathrm{new}}$ is added to the tree. The algorithm will end if a path from the starting point to the target point is planned or the number of iterations reaches a set threshold.

On the basis of RRT, RRT* has the following improvement, as shown in Algorithm 1. When RRT* connect $\mathrm{x}_{\mathrm{new}}$ to $\mathrm{x}_{\mathrm{nearest}}$, it will search for a parent node that is optimal in terms of cost at a certain radius on circle centered on $\mathrm{x}_{\mathrm{new}}$. If $\mathrm{x}_{\mathrm{new}}$ is added to tree, RRT* will rewire the neighbor node to check weather has a lower cost path that through $\mathrm{x}_{\mathrm{new}}$ than current path. Therefore, RRT* can obtain an optimal path as the number of samples goes to infinity. However, it takes a lot of time, memory usage and is a huge challenge in terms of computation. In practical applications, especially for mobile or flying robots, efficient computation of high-quality paths is required. [14].

3 HNSRRT*Algorithm

3.1 Gaussian Distribution

RRT and RRT* are randomly sampled by $\mathrm{x}_{\mathrm{rand}}$ to connect the value of cost function is least to reach the least optimal path. Due to the randomness of sampling, If the obstacle area or non-target area has a large number of sampling nodes, it will reduce search convergence rate and increase memory usage [17].

Therefore, we introduce Gaussian distribution function to change the state space of the sample, instead of the original uniform random sampling to improve the sampling effectiveness, where Gaussian Distribution $N(x, \mu, \sigma)$ with the mean value μ and standard deviation σ. The probability density function is defined as:

$$f(x, \mu, \sigma) = (\frac{1}{\sigma\sqrt{2\pi}})e^{\frac{-(x-\mu)^{2}}{2\sigma^{2}}} \tag{2}$$

According to probability density curve with show in Fig.1, the probability density around the mean is steeper when the standard deviation is small, so the sampling probability around mean is larger. Due to the peculiarity of Gaussian distribution function, the different intervals have different probabilities, the anticipative probability was obtained by altering interval range. Inspired by this property, we greatly improve the sampling efficiency by introducing Gaussian function to change the sampling state space.

3.2 Heuristic Function

RRT* is based on random sampling that leads to low search efficiency and requires multiple iterations to plan an optimal path. Therefore, We define the heuristic function is inspired by A*algorithm [4] as:

$$f(n) = \lambda g(n) + (1-\lambda)h(n), \quad n \in \mathbb{N}^{+} \tag{3}$$

$$g(n) = \begin{cases} 0 & n = 1 \\ \sum_{i=1}^{n-1} ||\mathrm{x}_{i+1} - \mathrm{x}_{i}|| & n \neq 1 \end{cases} \tag{4}$$

$$h(n) = ||\mathrm{x}_{goal} - \mathrm{x}_{n}|| \tag{5}$$

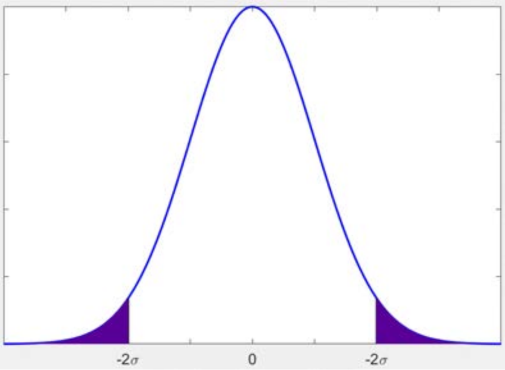
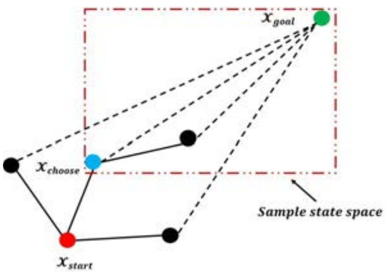
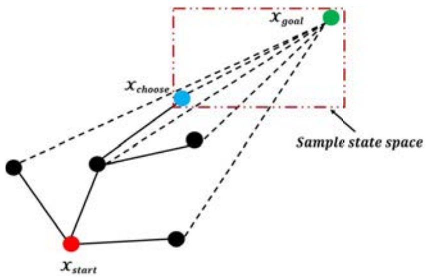


Fig. 1: Gaussian probability curve

Where $f(n)$ is the evaluation cost of each node, $g(n)$ represents the actual cost of each node to the start point, and $h(n)$ denoted the cost of the Euclidean distance. At the same time, we introduce λ that represents a parameter to weigh the cost of reconstruction. We can get a node x_{choose} that has less evaluation cost than other nodes to create a sample space as shown in Fig.2(a).



(a) Sample state space I



(b) Sample state space II

Fig. 2: Diagram of the sample constraint strategy

3.3 HNSRRT*

In the HNSRRT*, the heuristic function is utilized to find the minimum evaluation cost of node in the tree at first, where λ=1 means that the path cost is compensated by the time cost, and λ=0 represent a decrease in time cost and an increase in path cost. Therefore, we denote the fraction of parameter λ as 0.48, because the simulation results show that λ=0.48 provides the best performance in terms of balancing the cost of time and path. According to the identified nodes and goal point, a new sampling state space is formed to guide the sampling points to converge to the target point continuously, which reduces the excessive state space exploration, as shown in Fig2(b) . At the same time, the σ and μ are set to 1 and 0, respectively, to debias the probability of sampling state space.

Algorithm 2: $HNSRRT*$

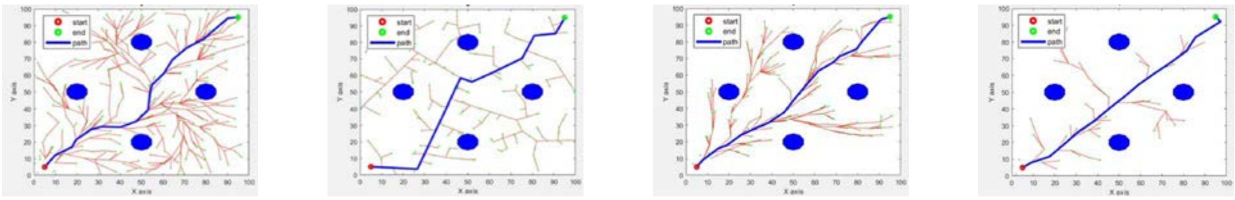
Input: $\mathrm{x_{start}}, \mathrm{x_{goal}}, \mathcal{X}$,Stepsize and Map
Output: $\mathcal{P}$

$N \leftarrow \mathrm{x_{start}}, E \leftarrow \emptyset, \mathcal{P} = (N, E)$;
for i=*1*...N **do**
  // Find a optimal cost node in tree
  $\mathrm{x}_{choose} = Find(\mathcal{P})$;
  //The subspace of x_{choose} *and* x_{goal}
  $\mathcal{X}_{sample} \leftarrow Space(\mathrm{x}_{choose}, \mathrm{x}_{goal})$;
  if $|Randn()|$ <0.5 **then**
    $//\mathrm{x}_{rand}$ *sample in subspace*
    $\mathrm{x}_{rand} \leftarrow UniformSample(\mathcal{X}_{smaple})$;
  else
    $//\mathrm{x}_{rand}$ *sample in global space*
    $\mathrm{x}_{rand} \leftarrow UniformSample(\mathcal{X})$;
  end
  $\mathrm{x_{nearest}} \leftarrow Nearest(\mathcal{P}, \mathrm{x_{rand}})$;
  $\mathrm{x_{new}} \leftarrow Steer(\mathrm{x_{nearest}}, \mathrm{x_{rand}})$;
  if *ObstacleFree*($\mathrm{x_{new}}, \mathrm{x_{nearest}}$) **then**
    $\mathcal{P} \leftarrow Extend(\mathcal{P}, \mathrm{x_{new}})$;
    $\mathrm{X}_{near} \leftarrow NearestNeighbors(\mathcal{P}, \mathrm{x}_{new}, r)$;
    $\mathrm{x}_{parent} \leftarrow ChooseParent(\mathrm{x}_{new}, \mathrm{X}_{near})$;
    $\mathcal{P} \leftarrow Rewire(\mathcal{P}, \mathrm{x}_{new}, \mathrm{x}_{parent})$;
  end
  if $|\mathrm{x}_{nearest} - \mathrm{X}_{goal}| < Stepsize$ **then**
    Return $\mathcal{P}$;
  end
end
Return failure;

We describe HNSRRT* in detail as shown in Algorithm 2. To begin with, we initialize a tree $\mathcal{P} = (N, E)$ is composed of a node set $N \subset \mathcal{X}_{free}$ and edge set $E \subseteq \mathcal{X}_{free}$ and initialize sampling state space $\mathcal{X} \in \mathbb{R}^{\mathrm{n}}$.In the light of environmental information and parameter settings, we can find that a node X_{choose} is obtained by heuristic function and a new sampling state space $\mathcal{X}_{sample} \subset \mathcal{X}$ is created by X_{choose} and X_{goal}. Then, according to Gaussian distribution function $Randn()$ to determine sampling state space, where $|Randn()| < 0.5$ is intended to bias the sampling point toward the sampling state space $\mathcal{X}_{sample}$. In order to intensify the rate of convergence, the $|Randn()|$ range is expected to aggrandize, but the larger the range, the easier it is to fall into local optima. Generally speaking, comprehensive convergence rate as well as the various factors we choose the $|Randn()|$ range less than 0.5. The following procedure is similar to RRT* , as shown in Algorithm 1.If the distance between the nearest node and the target node is less than the stepsize, which directly connects and returns a path connecting the start point to the target point, or return failure when given number of iterations is reached.

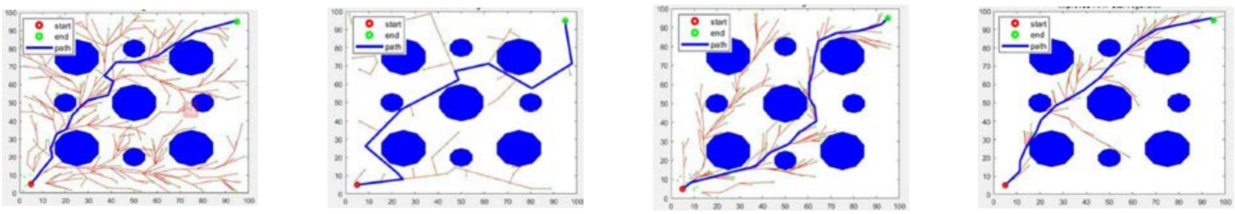
4 SIMULATION AND RESULTS

In order to verify the effectiveness of the proposed algorithm for simulation experiments, Matlab2019a is selected as the simulation platform, the main frequency of the computer is 2.40GHz and the memory is 4GB. The simulation is divided into 2D and 3D simple environment and complex environment.



(a) RRT* (b) RRT-Connect (c) P-RRT* (d) HNSRRT*

Fig. 3: Comparison of HNSRRT* and ohter algorithms in 2D simple environment



(a) RRT* (b) RRT-Connect (c) P-RRT* (d) HNSRRT*

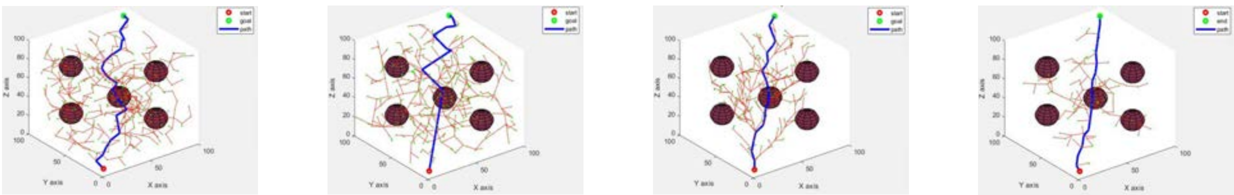
Fig. 4: Comparison of HNSRRT* and other algorithms in 2D complex environment

Table 1: COMPARISON WITH DIFFERENT ALGORITHM IN 2D SIMPLE ENVIRONMENT

Algorithm	Avg.Time(s)	Avg.Number of sample	Avg.Cost of Path
RRT*[11]	0.108	304.79	139.58
RRT-Connect [12]	0.035	146.87	153.35
P-RRT* [13]	0.084	219.46	134.36
HNSRRT*	**0.053**	**92.56**	**135.31**

Table 2: COMPARISON WITH DIFFERENT ALGORITHM IN 2D COMPLAX ENVIRONMENT

Algorithm	Avg.Time(s)	Avg.Number of sample	Avg.Cost of Path
RRT*[11]	0.245	378.16	168.22
RRT-Connect [12]	0.057	203.82	234.22
P-RRT*[13]	0.142	273.34	148.75
HNSRRT*	**0.059**	**167.21**	**145.23**



(a) RRT* (b) RRT-Connect (c) P-RRT* (d) HNSRRT*

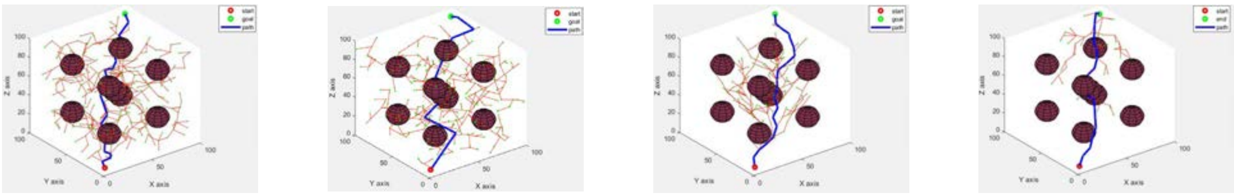
Fig. 5: Comparison of HNSRRT* and other algorithms in 3D simple environment

4.1 2D Simulation Experiment

The experimental parameters are designed as follows: the search area is a square of 100*100, the starting point(x_{start}) of the red circle is (5,5), the ending point(x_{goal}) of the green circle is (95,95), the stepsize set to 5, and the blue line is the path planned by different algorithms. The blue circle is the obstacle, the comparison of different algorithms in the environment of simple obstacles as shown in Fig.3 and the environment of complex obstacles as shown in Fig.4. We find that in a simple environment, our proposed algorithm is superior to other algorithms in number of sample, reduces memory usage, is superior to RRT* and P-RRT* algorithms in the cost of time, and is superior to RRT-Connect algorithm and RRT* algorithm in the cost of path. In the complex environment, we compare our proposed algorithm with the other three algorithms through multiple experiments, and there is a obvious improvement in path cost and sampling times, and it is close to the RRT-Connect algorithm in terms of time cost. Our proposed algorithm reduces the time cost by about 50% and the path cost by about 3% compared with the P-RRT* algorithm, as shown in Table 1 and Table 2.

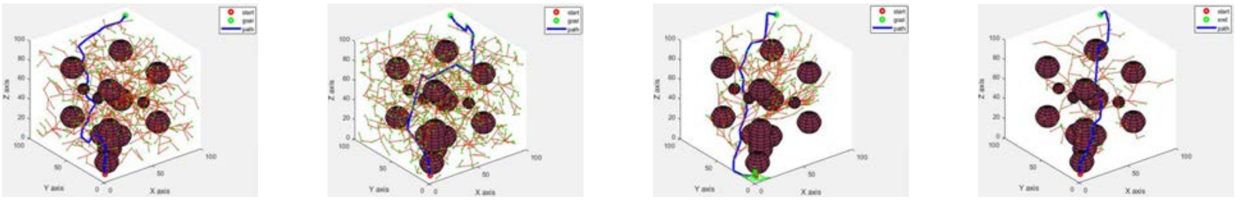
4.2 3D Simulation Experiment

In the 3D environment simulation experiment, we divided a simple environment and two intricate environments. The experimental parameters are set as follows:the search area is a cube of 100*100*100, the starting point(x_{start}) of the red circle is (5,5,5), the ending point(x_{goal}) of the green circle is (95,95,95), the stepsize set to 10, and the blue line is the path planned by different algorithms. The brown sphere is the obstacle,the comparison of different algorithms in the environment of simple obstacles as shown in Fig.5 and the environment of complex obstacles as shown in Fig.6 and Fig.7. Through the comparison of simulation experiments in 3D simple and complex environments, the sample number of our proposed algorithm is significantly less than the other three algorithms. In the simple environment, HNSRRT* reduces the time cost by about 40% compared to RRT* and improves the path cost by about 15%, as shown in Table 3, and has satisfied performance in terms of time-consuming as well as path length compared with other algorithms. In two different complex environments, the random sampling of RRT* algorithm in the sampling space leads to excessive search, which



(a) RRT* (b) RRT-Connect (c) P-RRT* (d) HNSRRT*

Fig. 6: Comparison of HNSRRT* and other algorithms in 3D complex environment I



(a) RRT* (b) RRT-Connect (c) P-RRT* (d) HNSRRT*

Fig. 7: Comparison of HNSRRT* and other algorithms in 3D complex environment II

Table 3: COMPARISON WITH DIFFERENT ALGORITHM IN 3D SIMPLE ENVIRONMENT

Algorithm	Avg.Time(s)	Avg.Number of sample	Avg.Cost of Path
RRT*[11]	0.102	431.83	219.38
RRT-Connect[12]	0.055	221.13	227.29
P-RRT*[13]	0.076	241.19	177.226
HNSRRT*	**0.059**	**102.32**	**183.96**

Table 4: COMPARISON WITH DIFFERENT ALGORITHM IN 3D COMPLAX ENVIRONMENT I

Algorithm	Avg.Time(s)	Avg.Number of sample	Avg.Cost of Path
RRT*[11]	0.158	473.41	220.14
RRT-Connect [12]	0.068	243.55	240.22
P-RRT* [13]	0.103	267.37	193.75
HNSRRT*	**0.062**	**124.54**	**194.21**

Table 5: COMPARISON WITH DIFFERENT PARAMETERS IN 3D COMPLAX ENVIRONMENT II

Parameter λ	Avg.Time(s)	Avg.Number of sample	Avg.Cost of Path
0	0.145	218.63	208.23
0.48	**0.096**	**175.25**	**204.96**
0.67	0.114	182.17	209.90
1	0.105	217.35	216.67

Table 6: COMPARISON WITH DIFFERENT ALGORITHM IN 3D COMPLAX ENVIRONMENT II

Algorithm	Avg.Time(s)	Avg.Number of sample	Avg.Cost of Path
RRT*[11]	0.230	696.33	244.38
RRT-Connect[12]	0.095	432.33	235.21
P-RRT*[13]	6.269	780.32	204.63
HNSRRT*	**0.098**	**172.35**	**202.48**

leads to high time cost and cannot find an optimal path in a short time. RRT-Connect algorithm speeds up the search rate through bidirectional sampling but cannot guarantee the path quality, while P-RRT* algorithm is prone to be affected by the resultant force of potential field in complex environment, which reduces the performance. In complex environment II, a local trap is caused by an obstacle in front of the starting point, P-RRT* is prone to fall into local minimal, leading to the planning time increase, the path length becomes longer, the memory consumption caused by the surge of sampling times and even the planning failure. However, our algorithm uses two samplers to achieve non-uniform sampling, which sampler is selected by Gaussian function, so that our algorithm does not fall into environmental traps. Meanwhile, we set λ equal to 0.48 by comparing the performance of different heuristic factor λ value in complex environment II, as shown in Table 5. Therefore, we find that our proposed algorithm has good performance in complicated environment. On the whole, our proposed algorithm is better than the other three algorithms, and has a significant improvement in the cost of time and path compared with RRT* as shown in Table 4 and Table 6.

5 CONCLUSIONS

This paper studies the problem of low search efficiency of RRT* and the need for multiple sampling iterations to find a progressive optimal path, and proposes a path planning algorithm based on heuristic non-uniform sampling method.We introduce a heuristic function to select the node with the minimum evaluation cost and the target node to form a new sampling state space, and then select the sampling space according to the probability distribution of Gaussian function to ensure the quality of path cost and time cost,and the simulation results show that the proposed algorithm has good performance in time cost, path cost and memory usage compared with the existing algorithms. However,in terms of time cost, HNSRRT* is similar to RRT-connect algorithm. In the next step, it will be considered to combine bidirectional extension with non-uniform sampling and constrained sampling points to achieve further improvement.

References

[1] L. Chen, Y. Shan, W. Tian, B. Li and D. Cao, "A Fast and Efficient Double-Tree RRT*-Like Sampling-Based Planner Applying on Mobile Robotic Systems," in IEEE/ASME Transactions on Mechatronics, vol. 23, no. 6, pp. 2568-2578, Dec. 2018, doi: 10.1109/TMECH.2018.2821767.

[2] X. Ruan, J. Zhou, J. Zhang, and X. Zhu, "Robot goal guide

RRT path planning based on sub-target search," Control Decis., vol. 35, pp. 25432548,Jun. 2020.
[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," Proceedings. 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 1985, pp. 500-505, doi: 10.1109/ROBOT.1985.1087247.
[4] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, July 1968, doi: 10.1109/TSSC.1968.300136.
[5] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," 1998.
[6] L. E. Kavraki, P. Svestka, J. . -C. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566-580, Aug. 1996, doi: 10.1109/70.508439.
[7] L. Jiang, S. Liu, Y. Cui and H. Jiang, "Path Planning for Robotic Manipulator in Complex Multi-Obstacle Environment Based on Improved-RRT," in IEEE/ASME Transactions on Mechatronics, vol. 27, no. 6, pp. 4774-4785, Dec. 2022, doi: 10.1109/TMECH.2022.3165845.
[8] L. Jaillet, J. Corts and T. Simon, "Sampling-Based Path Planning on Configuration-Space Costmaps," in IEEE Transactions on Robotics, vol. 26, no. 4, pp. 635-646, Aug. 2010, doi: 10.1109/TRO.2010.2049527.
[9] R. Pepy and A. Lambert, "Safe Path Planning in an Uncertain-Configuration Space using RRT," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 5376-5381, doi: 10.1109/IROS.2006.282101.
[10] J. Qi, H. Yang and H. Sun, "MOD-RRT*: A Sampling-Based Algorithm for Robot Path Planning in Dynamic Environment," in IEEE Transactions on Industrial Electronics, vol. 68, no. 8, pp. 7244-7251, Aug. 2021, doi: 10.1109/TIE.2020.2998740.
[11] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," Int. J. Robot. Res., vol. 30, no. 7, pp. 846894, 2011.
[12] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 2000, pp. 995-1001 vol.2, doi: 10.1109/ROBOT.2000.844730.
[13] A. H. Qureshi and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning," in Autonomous Robots. Springer, 2017,pp. 10791093.
[14] J. Wang, W. Chi, C. Li, C. Wang and M. Q. . -H. Meng, "Neural RRT*: Learning-Based Optimal Path Planning," in IEEE Transactions on Automation Science and Engineering, vol. 17, no. 4, pp. 1748-1758, Oct. 2020, doi: 10.1109/TASE.2020.2976560.
[15] J. Wang, T. Li, B. Li and M. Q. . -H. Meng, "GMR-RRT*: Sampling-Based Path Planning Using Gaussian Mixture Regression," in IEEE Transactions on Intelligent Vehicles, vol. 7, no. 3, pp. 690-700, Sept. 2022, doi: 10.1109/TIV.2022.3150748.
[16] J. Wang, M. Q. . -H. Meng and O. Khatib, "EB-RRT: Optimal Motion Planning for Mobile Robots," in IEEE Transactions on Automation Science and Engineering, vol. 17, no. 4, pp. 2063-2073, Oct. 2020, doi: 10.1109/TASE.2020.2987397.
[17] Li Yuncheng and Shao Jie, "A revised Gaussian distribution sampling scheme based on RRT* algorithms in robot motion planning," 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, 2017, pp. 22-26, doi: 10.1109/ICCAR.2017.7942654.