

Genetic RRT: Asymptotically Optimal Sampling-based Path Planning via Optimization of Genetic Algorithm

Xiyuan Wang *

School of Mechanical Engineering, Hefei University of Technology, Hefei, China

* Corresponding author: xiyuanwang@mail.hfut.edu.cn

Abstract. As one of the most frequently used stochastic planners, Rapidly-exploring random trees (RRT) is commonly applied to the path-planning problem in the continuous state space. For single-query problems, RRTs demonstrate nice properties in computation speed and universality. However, the probabilistically complete results of the planning problems hinder their further use. To date, some heuristic-biased sampling searches like hRRT and Informed RRT* addressed the optimality problem of RRTs in the path-planning. However, using the heuristic subsets to bias sampling improved the performance of the algorithms while eliminating the probability of finding other equal optimal solutions simultaneously. To minimize the cost of solutions in the path-planning problems, this paper presents a novel method to focus on the optimization of paths planned by RRTs in terms of the genetic algorithm. The demonstrated planning technique's advantages can be seen with a new algorithm, Genetic RRT, which retains the multi-results nature and the same effectiveness toward high-dimension path-planning problems as RRTs while significantly increasing the probability of finding an asymptotically optimal path via optimization of genetic algorithm, as the number of iterations increases.

Keywords: Rapidly-exploring random trees, Stochastic incremental searches, Genetic algorithm, shortest path.

1. Introduction

Over the past decades, several path-planning approaches have been proposed and some of them demonstrated nice properties in solving the general path-planning problem in 2D and other high-dimension spaces; among them, the algorithms can be roughly classified in graph-based searches or sampling-based searches. The former planning results are often optimal and complete and return failure if a solution doesn't exist, like the Dijkstra algorithm, D* and, A*. The latter can effectively avoid discretizing state spaces and, thus, scale with the problem's size better than the former; however, the result is probabilistically complete and becomes nontrivial as the sampling domain grows.

Rapidly-exploring Random Trees (RRTs) [1], as well as other stochastic planners like Expensive Space Trees (ESTs) [2] and, Probabilistic Roadmaps (PRMs) [3], perform well in the calculation speed concerning the continuous state spaces, demonstrating fascinating results in single-query problems solving process. Among these randomized path planners, the space-filling nature of RRTs tends to lead to better performance and consistency in path-planning results and, thus, expand standard path planning effectively to general nonholonomic planning problems. This allows RRTs to be more indifferent to problem size, scale better in high-dimension path-planning and focus more on kinodynamic constraints. However, sampling-based path-planning algorithms are probabilistically complete, rendering the less-strict completeness results and darkening their future practical use.

To date, some modified sampling-based planners have made claims about the optimality or suboptimality of the shortest path in a continuum based on RRTs. Urmson and Simmons [4] used a heuristic to bias RRT sampling behavior and improved its results, but with no words, quantifying the improvements. Karaman and Frazzoli [5] then demonstrated that all RRT-based algorithms could almost surely be enhanced by optimizing the sampling process and returning suboptimal results. Jonathan, Siddhartha, and Timothy [6] recognized the geometric relation between the distance-cost heuristic function and the prolate ellipsoidal informed subset and also realized that the bounds of the heuristic subsets significantly affect the number of iterations, computation speed as well as the solution cost of the algorithm and thus, demonstrated a direct method to sample in an ellipsoidal

subset iteratively which improved RRT* significantly. These algorithms are asymptotically optimal: the probability of finding an optimal or suboptimal solution gradually approaches one as the number of iterations keeps increasing.

Considering the existing paths will bias the expansion of a new state from state space and also, unable to rewire past path, the RRTs are not asymptotically optimal to any extent; every solution introduced by RRTs is randomly and waiting for further optimization. Through ushering the method of incremental rewiring of the path in RRTs, RRT* effectively solves this problem; by doing so, new states are not only considered to be the candidate for the optimal path in the tree but also supposed to be a potential replacement of the existing nearby states. Although with broad sampling in the global domain, the result in RRT* iteratively asymptotically approaches optimal, the space-filling nature forces RRT* to almost try every destination in the problem domain and evaluate the cost with regard to the initial state; in this sense, it is inconsistent with RRTs' single-query character and becomes low-efficient as the state space grows. To solve the problem of direct sampling in the heuristic subset while retaining RRTs' single-query nature, Informed RRT* [6] claimed to complete this daunting task by constructing the admissible ellipsoidal heuristic in the state space near the current state. However, as Informed RRT* shrinks the candidate planning states to subsets of the current domain by introducing heuristics. However, it is unavoidably dependent on the current cost, and thus, when the spheroid-shape states set is larger than the state space itself, Informed RRT* becomes invalid.

For the optimal path-finding problem, also in want of getting rid of the limitation of the heuristic subset and focusing on the optimization of searched paths themselves, a novel combination is presented: some necessary transformations using the genetic algorithm are implemented on coarse paths resulting from RRT, leading to better solutions. This paper shows that the probability of improving such results becomes appreciable as the size of the global domain and the state space increases. It is also demonstrated that the efficiency of finding different suboptimal solutions can be improved, and the algorithm can be expected in future practical use.

The organization of the remainder of this paper is as follows. Section 2 introduces the definition of the path-planning problem and compares two main categories of planning algorithms while pointing out some shortcomings. Section 3 demonstrates some modifications of GA when applying to optimize the path-planning solutions. Section 4 presents an example of Genetic RRT and articulates the main body of the algorithm. Some simulations are shown in section 5. Some conclusions and discussion are made in section 6.

2. Background

2.1. Problem definition

The path-planning problem in this paper is defined similarly to [6]. Path planning will be viewed as a search for a sequence of states on a continuous path in the state space $X \subseteq \mathbb{R}^n$ from a start state, x_{start} , to the desired final state, x_{final} . Some obstacles, $X_{obs} \subsetneq X$, are fixed in the space, and an exact representation of the location and range of X_{obs} is not available, the planner can only check whether a new state is in collision with obstacles by checking whether it lies in the X_{obs} . Let $X_{free} \subsetneq X$ be the complement of X_{obs} , and $x_{start}, x_{final} \in X_{free}$. For a given start state, let $\mathcal{T}$ denotes the sequence of states on a continuous path. Let ρ represents the cost of a $\mathcal{T}$ with respect to Euclidean distance.

The path-planning problem of finding the most efficient path in the state space can be expressed by minimizing a cost function, ρ , and $\mathcal{T}^*$ is the candidate of suboptimal sequence.

$$\mathcal{T}^* = \operatorname{argmin}\{\rho(\mathcal{T}) \mid \mathcal{T}(0)=x_{start}, \mathcal{T}(1)=x_{final}, \forall s \in [0,1], \mathcal{T}(s) \in X_{free}\} \quad (1)$$

2.2. Prior work

Prior work regarding the path-planning planners can be divided into two different categories: sampling-based, like RRTs, and graph-based, like A* and D*.

1) *Sampling-based*: Sampling-based algorithms are probabilistically complete and not optimal like RRTs and PRMs, but also demonstrated a significant probability of potential improvement considering their kernel thought of randomness. When applying RRTs to 2-D problems, setting a bias toward the final state artificially, the results are far from optimal and expensive. Fig. 1 shows some simulations in which planners are required to find a path passed through the start state, the final state, and the other two states. The probability of randomly selecting a new state grows as the artificial bias increases.

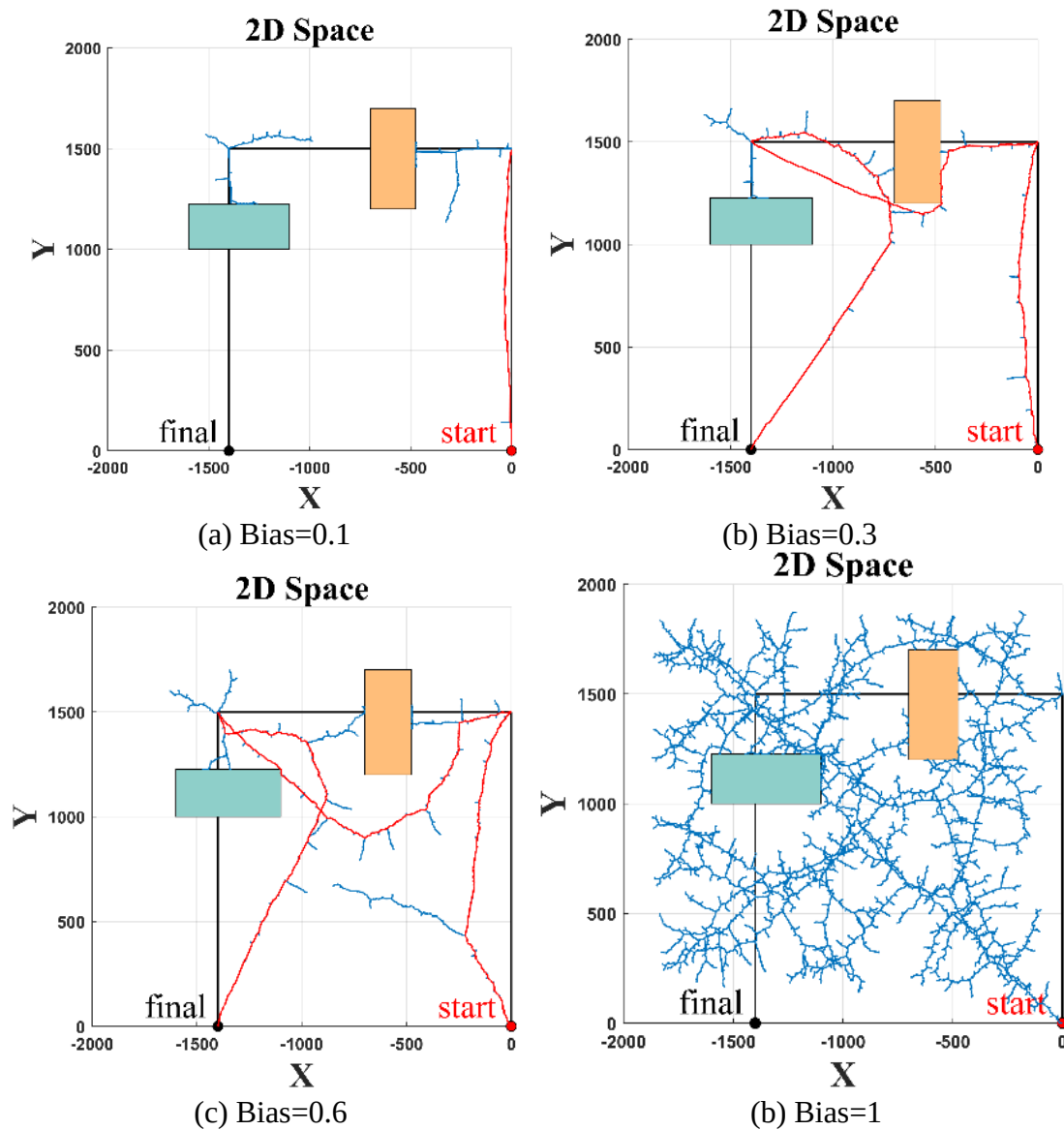


Figure 1. Paths planned by RRT with different biases

To date, heuristic-biased sampling like hRRT [4] and Informed RRT* [6] have successfully addressed the optimal problem of RRT by iteratively expanding states sampled from the heuristic subset, which bound is demarcated by biasing the weight of different states drawn from the global domain. However, each iteration of these algorithms requires refining the admissible heuristic that the nearby state will be biased by distributing different weights, and the next state will be selected according to the probability. The heuristic-biased sampling algorithms, while asymptotically optimal, also eliminate the probability of finding other equal optimal solutions in one iteration, rendering it

unsuitable in some practical situations that needed multi-results. To this extent, they betray the multi-results nature of RRTs.

2) *Graph-based*: Graph-based path-planning methods like A* and D* are complete and optimal. They allocate almost every state with cost (usually the distances) and select the minimum state as its next state. While performing well in 2-D cases and grid-like spaces, Graph-based methods scale worse than Sampling-based methods in high-dimension problems as they always fall into the expensive calculation of every possible state in the workspace and are high time-cost. Fig. 2 represents some simulations of the Graph-based algorithm (A*) in different cases.

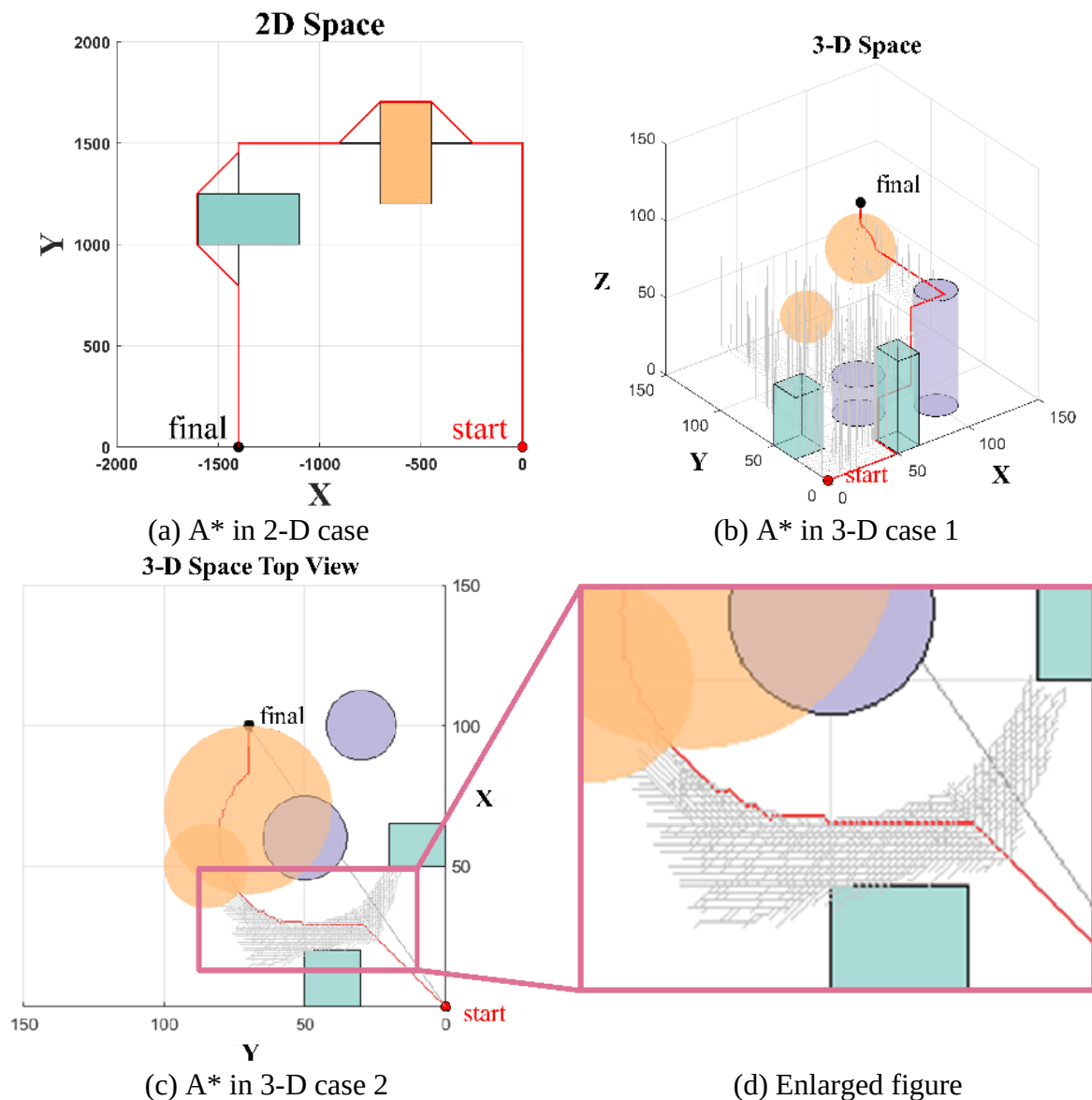


Figure 2. Paths planned by A* in different cases

Path-planning algorithms like A* and D* use graph-based methods which require a discretization of the continuum, although ensuring the optimal result in the grid-like space also reduces optimality and efficiency in high-dimension continuous areas simultaneously. The red line shown in **Fig. 2(a)** represents a suboptimal, not optimal, solution of A* in the continuum because of the discretization. The grey lines in Fig. 2(b) represent every path A* tried, and they cover 75% points of the global domain according to the data. Fig. 2(c) shows some meaningless calculation A* completed, and Fig. 2(d) shows the enlarged image.

Thus, a novel method that can retain the multi-results nature of RRT while improving its optimality and scale well when it comes to high-dimension problems is expected. A simple combination of RRT and GA (Genetic Algorithm), Genetic RRT, is clearly up to the task as Genetic RRT not only

demonstrates a clear improvement toward RRT but also multiple suboptimal results can be generated at the same time. The space-filling nature of RRT provides GA with a bountiful population, while GA optimizes the expansion of the modified path toward the result computed by RRTs.

3. Modification of genetic algorithm

Genetic algorithm-based planners are widely used in motion-planning problems of mobile robot in the static environment [7]-[11]. Two evaluating parameters are frequently used in scoring a candidate's fitness in the GA, the sum of the distance cost and the smoothness of the overall path.

1) *Distance cost*: the distance cost of a path can be calculated by summing up the distance between every two connected points. For $x, x+1 \in \mathcal{T}$, define p_1 as:

$$p_1 = \sum_{x=start\ state}^{final\ state-1} \rho(\mathcal{T}(x), \mathcal{T}(x+1)) \quad (2)$$

2) *Smoothness*: the smoothness of a path can be evaluated by the angle formed by any two adjacent lines on the path (Fig. 3); the larger the angle is, the smoother the path is. Comparison between different algorithms is shown in Fig. 4.

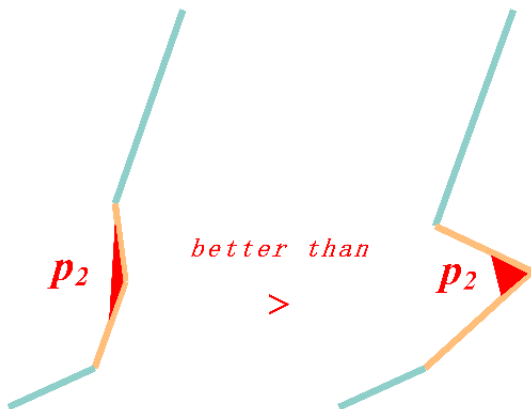


Figure 3. Smoothness parameter of path

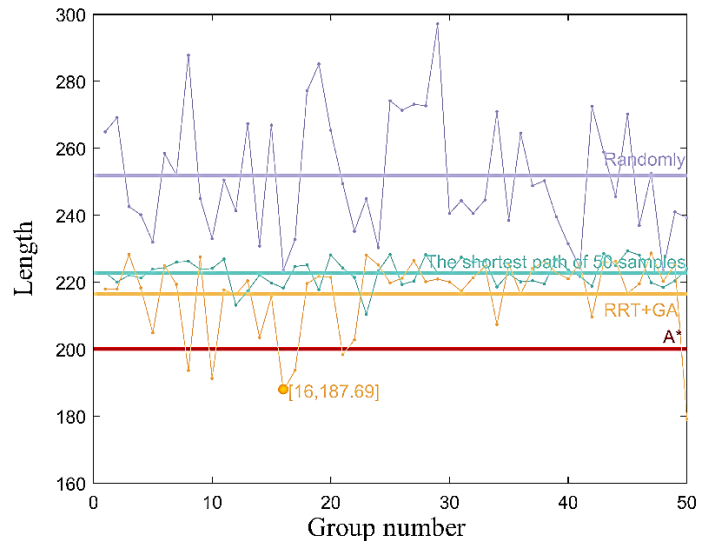


Figure 4. Comparison between different algorithms

The smoothness of each path can be calculated as follows

$$p_2 = \sum_{x=start\ state+1}^{final\ state-1} \arccos\left(\frac{\rho_1^2 + \rho_2^2 - \rho_3^2}{2\rho_1\rho_2}\right) \quad (3)$$

Where $\rho_1 = \rho(\mathcal{T}(x-1), \mathcal{T}(x))$, $\rho_2 = \rho(\mathcal{T}(x), \mathcal{T}(x+1))$, $\rho_3 = \rho(\mathcal{T}(x-1), \mathcal{T}(x+1))$.

3) *Fitness function*: after standardization and normalization of the evaluating parameters, the fitness of each offspring, f , can be expressed as $f = p_1 \cdot r + p_2 \cdot (1-r)$, where r represents the ratio of the importance of distance cost to smoothness, which can be adjusted artificially.

4. Genetic RRT

Algorithm 1: Genetic RRT ($x_{start}, x_{final}, step, K, M, N, I$)

```

1   $\mathcal{T}.init(x_{start});$ 
2   $c \leftarrow \emptyset;$ 
3  for  $k = 1$  to  $K$  do
4      for  $m = 1$  to  $Population$  do
5          for  $n = 1$  to  $N$  do
6               $\mathcal{T}.init(x_{start});$ 
7               $x_{rand} \leftarrow \text{samplePoint}(x_{final});$ 
8               $x_{near} \leftarrow \text{findNearPoint}(x_{rand}, \mathcal{T});$ 
9               $x_{new} \leftarrow \text{expandPoint}(x_{near}, x_{final}, step);$ 
10              $\mathcal{T}.addVertex(x_{new});$ 
11         end
12          $c.addPath(\mathcal{T});$ 
13     end
14     for  $i = 1$  to  $I$  do
15          $\mathcal{T}_{father} \leftarrow c.leastCost();$ 
16          $\mathcal{T}_{mother} \leftarrow c.randSelect();$ 
17          $\mathcal{T}_{child} \leftarrow GA(\mathcal{T}_{father}, \mathcal{T}_{mother});$ 
18          $c.replacePath(\mathcal{T}_{child});$ 
19     end
20 end
21  $\mathcal{T}_{best} \leftarrow c.leastCost();$ 
22 return  $\mathcal{T}_{best};$ 

```

In Alg. 1, for some initial given conditions, an example of Genetic RRT is presented. It is similar to the RRT algorithm in [1] in the main path-planning part, with the addition of lines 14 to 19 as the optimization part using GA. The child, $\mathcal{T}_{child}$, is not always better than its parents with respect to distance cost, and the Genetic RRT will only preserve those better children's genes and replace their mothers' genes.

As RRT finds solutions, $\mathcal{T}$ (line10), Genetic RRT adds the path to the candidate paths set (line 12). It uses the distance cost, p_1 , to calculate each path's cost and select the least one as the "father gene" in the GA (line15). The Genetic RRT allocates each candidate in the paths set, c , with a fitness, f , as demonstrated in Section 3, and determines the candidates' selected probability according to f . A random selection with regard to each candidate's probability will provide the "mother gene" (line 16). To optimize those coarse paths, mutation and crossover, included in the GA algorithm (line 17) [12], will be implemented on $\mathcal{T}_{father}$ and $\mathcal{T}_{mother}$, leading to a better offspring, $\mathcal{T}_{child}$, with respect to the distance cost, and $\mathcal{T}_{child}$ will replace $\mathcal{T}_{mother}$ in the candidate paths set, c (line 18) to renew the gene pool.

At each iteration, the optimization of GA ensures an improvement of existing candidate paths [13]. Thus, Genetic RRT is asymptotically optimal: as the number of iterations increases to infinite, the probability of finding the optimal path approaches one. At the same time, the multi-results nature of RRT isn't dampened by the GA algorithm, considering that every iteration of Genetic RRT conducts optimization on different initial gene pools. What's more, Genetic RRT could be further improved by combining other path-smoothing algorithms and incorporating them into a path-planning system as a module.

5. Simulations

Genetic RRT was compared to the graph-based method (A*) and sampling-based method (RRT) in a 3-D case in Fig. 4. Genetic RRT (the yellow line) finds some appreciable results even better than A* (the red line) does (like point [16, 187.69]) while showing a significant improvement toward basic RRT (the purple line). This also proves that the path planning results of A* and other graph-based methods in continuous space are not optimal, and Genetic RRT performs well in the continuum.

Genetic RRT also retains the multi-results nature of stochastic incremental searches. As shown in Fig. 5(b), A* (the red line) and other graph-based algorithms can only find one path given the start and final states, no matter how many times they iterate while Genetic RRT (the blue lines) can easily find more than one better path with the same initial conditions.

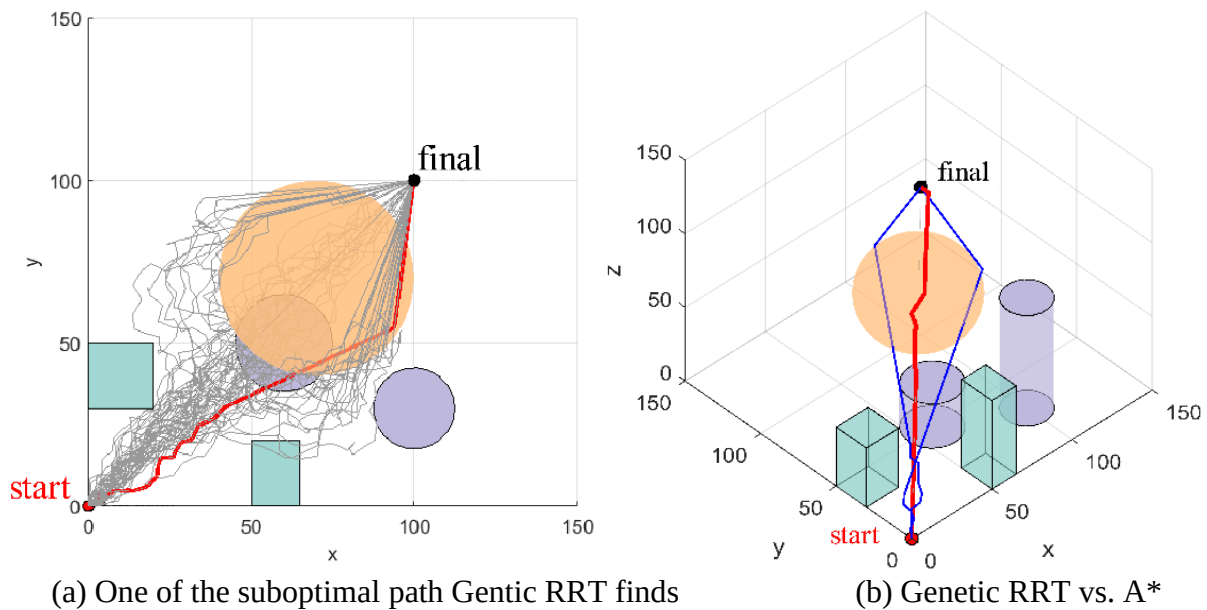


Figure 5. Paths planned by Genetic RRT

6. Conclusion

In this paper, an asymptotically optimal algorithm, Genetic RRT, is proposed while retaining some nice properties of RRT; when combined with GA, an appreciable improvement toward RRT can be concluded from the results. As a sampling-based method, the Genetic RRT also demonstrates even better optimality than graph-based methods like A* in continuous state space. However, there are still many problems regarding this method; for example, although Genetic RRT iteratively replans its path and can find the best solution, if one exists, as the number of iterations approaches infinity, it can only find the suboptimal path every time, and cannot ensure the resolution optimally. What's worse, the final results of Genetic RRT largely depend on the quality of the initial gene pool, considering the initial population influence the results of GA to some extent. Thus, further work is expected to improve each iteration's accuracy and ensure the final result's optimality.

References

- [1] LaValle S M. Rapidly-exploring random trees: A new tool for path planning. 1998.
- [2] Hsu D, Kindel R, Latombe J C, et al. Randomized kinodynamic motion planning with moving obstacles. *The International Journal of Robotics Research*, 2002, 21 (3): 233 - 255.
- [3] Kavraki L E, Svestka P, Latombe J C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 1996, 12 (4): 566 - 580.

- [4] Urmson C, Simmons R. Approaches for heuristically biasing RRT growth. Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453). IEEE, 2003, 2: 1178 - 1183.
- [5] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. The international journal of robotics research, 2011, 30 (7): 846 - 894.
- [6] Gammell J D, Srinivasa S S, Barfoot T D. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014: 2997 - 3004.
- [7] Ismail A T, Sheta A, Al-Weshah M. A mobile robot path planning using genetic algorithm in static environment. Journal of Computer Science, 2008, 4 (4): 341 - 344.
- [8] Nazarahari M, Khanmirza E, Doostie S. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. Expert Systems with Applications, 2019, 115: 106-120.
- [9] Lamini C, Benhlima S, Elbekri A. Genetic algorithm-based approach for autonomous mobile robot path planning. Procedia Computer Science, 2018, 127: 180 - 189.
- [10] Elhoseny M, Tharwat A, Hassanien A E. Bezier curve-based path planning in a dynamic field using modified genetic algorithm. Journal of Computational Science, 2018, 25: 339 - 350.
- [11] Xin J, Zhong J, Yang F, et al. An improved genetic algorithm for path-planning of unmanned surface vehicle. Sensors, 2019, 19 (11): 2640.
- [12] Katoch S, Chauhan S S, Kumar V. A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 2021, 80 (5): 8091 - 8126.
- [13] Mirjalili S. Genetic algorithm. Evolutionary algorithms and neural networks. Springer, Cham, 2019: 43 - 55.