A Dynamic Random Way to Test Path Planning Algorithms

Xiao-Yi Zhang[1] and Chao Huang[1]

***Abstract*— This paper adapts the idea of dynamic random testing to path planing algorithms, commonly used in CPS systems. We construct the test profile by partitioning the task region and dynamically adjust it by changing the probability of generating obstacles. Experimental results show that our approach can improve the test efficiency and find meaningful test cases.**

***Index Terms*— Software testing, path planning, dynamic random testing, rapidly random tree**

I. Introduction

As the development of cybernetic and information science, applications of Cyber-physical systems (CPS), such as self-driving cars [1], Unmanned Aerial Vehicle [2], and automatic robotic arms [3], tend to become self-adaptive and have been widely applied to finish complex tasks, such as rescue, reconnaissance, shipment, etc [2]. However, a failure in such systems may cause disruptive catastrophes. Both the higher safety requirement and the complex system structure will bring more challenges to the task of testing and verification.

This paper focuses on the path planning algorithm, which acts as the decision making part in the robotic agents [4]. Current path planning algorithms, such as Rapidly Random Tree (RRT) [4], tend to have non-deterministic elements to adapt to the complexity and uncertainties of real-life scenarios. As a result, the path generated in the same scenario could also be various. Additional concern is the complexity of real-life scenarios. The threats such as obstacles could have various distributions. Thus how to introduce reasonable test profiles that can generate meaningful test cases are valuable to be discussed. At last, considering the path planning algorithm, its performance should be measured by continuous values such as path length. Thus, to evaluate a test case, how to deal with continuous values instead of just passed or failed should also be concerned.

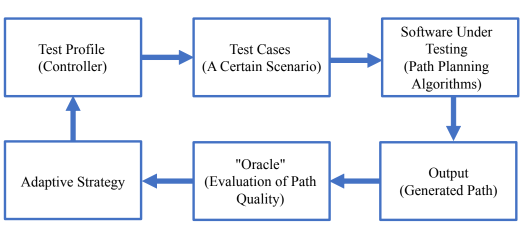


Fig. 1. Framework of applying DRT to path planning algorithms

The authors are supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST. Funding Reference number: 10.13039/501100009024 ERATO

[1] Xiao-Yi Zhang and Chao Huang are working as researchers in JST ERATO HASUO Projects, National Institute of Informatics, Tokyo, Japan. xiaoyi@nii.ac.jp, chao_huang@nii.ac.jp

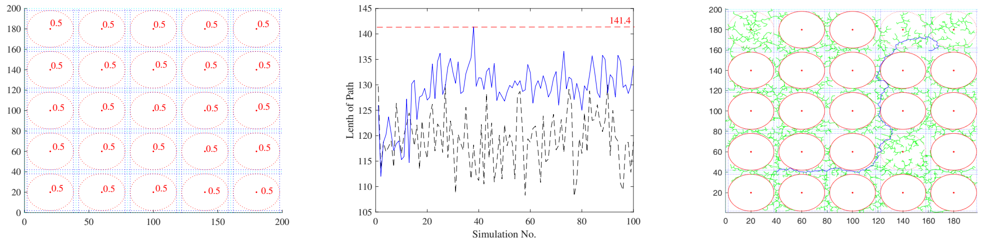
To address the above issues and improve the testing process, we propose a search-based testing approach, aiming to efficiently reaching the scenarios that are meaningful for the analysis or falsification of the considered path planning algorithm. Specifically, the idea of Dynamic Random Testing (DRT) is adopted. DRT belongs to the domain of software cybernetics, proposed by Cai et al. [5], [6], whose basic framework is shown in Fig. 1. It uses the term "test profile" as a controller for the testing process. Specifically, a test profile generates test cases following certain distributions involving both heuristics and randomness. In this way, we can not only have a direction of searching for meaningful test cases but also keep the randomness in deal with the uncertainties. Then, the adaptive strategy is designed as a feedback mechanism to dynamically adjust the current test profile according to the derived output path, so that the probability of generating better test cases can increase. Although the idea of DRT is natural suitable to the CPS-oriented path planning algorithms, specific techniques should also be delicately designed.

II. Modeling the Test Profile

Suppose the basic task region is R, in which a robotic agent starts from a point s_0, to the target point s_f. Then a test case is interpreted as a set of obstacles in R. Here, to make the test profile model an available mechanism to generate diverse obstacle scenarios, we introduce the idea of discretization and randomness as shown in Fig. 2(a). First we divide R into $m \times n$ subregions with identical sizes and "put" an obstacle creator at the center of each subregion. Then a test profile can be expressed as a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, in which each $X_{i,j} \in [0, 1]$ at row i and column j indicates the probability that the associated subregion $R_{i,j}$ contains an obstacle.

III. Test Case Generation

During the testing process, suppose the current test profile is $\mathbf{X}$, then we generate the test case by selecting whether to put an obstacle or not at each subregion $R_{i,j}$ following the probability of $X_{i,j}$. The generated test case is also a matrix $\mathbf{C} \in \{0, 1\}^{m\times n}$, in which element at the row i and column j is a binary value $C_{i,j}$. Here $C_{i,j} = 0$ and $C_{i,j} = 1$ represent that subregion $R_{i,j}$ has and does not have an obstacle, respectively. We assume that an obstacle is a circle with certain radius. Specifically, we leave a margin b in each subregion. Suppose the subregion is a box with length B, then the radius of each obstacle is $r = B/2 - b$. Particularly, we let $b = B/20$. In this setting, we can ensure that, for any test case, there are at least one paths from the start point to



(a) Structure of Test Profile (b) Experimental Result (c) An instance of the worst-case scenario

Fig. 2. Simulation profile and results.

the target point. Then we focus on examining the quality of the generated paths based on the concerned algorithm.

IV. Path Evaluation

According to different requirements, there are different criteria to evaluate a generated path. Here, we merely use the voyage, that is, the length of path. Given a path p, its quality degree $Q(p)$ can be calculated by $Q(p) = 1/Length(p)$, which means that the shorter p is, the higher quality it has.

V. Adaptive Strategy

Considering the dynamic strategy of adjusting the current test profile, we want to find the cases at which the path planning algorithm does not perform well, that is, finding some potential shortcomings. In addition, because $Q(p)$ is not binary, we should decide the direction of adjustment by comparing the current test case and the previous one. Specifically, let $\mathbf{C}_{k-1}$ and $\mathbf{C}_k$ denote the $k-1^{th}$ and k^{th} test cases generated during testing, respectively; and let p_{k-1} and p_k denote the paths generated under $\mathbf{C}_{k-1}$ and $\mathbf{C}_k$. Let $\mathbf{X}^k$ denote the current test profile. Then, for the next test profile $\mathbf{X}^{k+1}$, each $X_{i,j}^{k+1}$ can be calculated as follows:

$$X_{i,j}^{k+1} = \begin{cases} \text{sgn}(Q(p^{k-1}) - Q(p^k))(X_{i,j}^k + \delta) & \text{if } C_{i,j}^k = 1 \\ \text{sgn}(Q(p^{k-1}) - Q(p^k))(X_{i,j}^k - \delta') & \text{if } C_{i,j}^k = 0 \end{cases} \tag{1}$$

Eq. (1) indicates that if the quality of current path is worse (better) than the previous one, we increase (decrease) the probability of generating obstacles at the subregions that (do not) associated with the current obstacles. Here we set the values of δ and δ' to be 0.1 and 0.05, respectively.

VI. Experimental Study and Result Analysis

We conduct an experimental study based on the typical RRT algorithm [4]. The results are shown in Fig. 2(b) and Fig. 2(c). In Fig. 2(b), we record the derived path length at each test case. The blue solid and black dashed curves represent the performance of our approach and pure random testing, respectively. Compared with pure random testing, it can be observed that the paths generated through our approach are becoming longer along with the test execution process, which indicates that our approach tends to search for the paths with lower robustness to the voyage requirement. In addition, our approach can obtain the worst-case scenario associated with maximal path length (i.e., 141.4), which is much longer than the worst-case path obtained through pure random testing. In general, compared with pure random testing, our approach can improve the test efficiency in this situation. By further checking the worst-case scenario (See Fig. 2(c)), we found that the scenario that causes the RRT performs relatively worse is not simply the one that has obstacles at each subregion. Instead, several traps could be required (e.g., for the center in the position $(140, 60)$). This, to some extent, indicates that RRT may be influenced by the imbalance of threat information as what observed in [2].

VII. Conclusion

This paper adapts the idea of DRT to the testing of path planing algorithms, commonly used in CPS systems. To address specific characteristics of our problem, several strategies are proposed. We construct the test profile by partitioning the task region and dynamically adjust it by changing the probability of generating obstacles. Experimental results show that our approach can improve the test efficiency and find meaningful test cases. Our study shows the feasibility of devising dynamic random approaches in dealing with the challenges of testing path planning algorithms. Future work includes developing other kind of test profiles to fit various objectives and making theoretical analysis to facilitate the design of testing strategy.

References

[1] J. Kim, H. Kim, K. Lakshmanan, and R. R. Rajkumar, "Parallel scheduling for cyber-physical systems: Analysis and case study on a self-driving car," in *Proceedings of the ACM/IEEE 4th international conference on cyber-physical systems*. ACM, 2013, pp. 31–40.

[2] Z. Zheng, Y. Liu, and X. Y. Zhang, "The more obstacle information sharing, the more effective real-time path planning?" *Knowledge-Based Systems*, vol. 114, pp. 36–46, 2016.

[3] P. Lertkultanon and Q.-C. Pham, "A certified-complete bimanual manipulation planner," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1355–1368, 2018.

[4] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[5] K. Cai, H. Hu, C.-h. Jiang, and F. Ye, "Random testing with dynamically updated test profile," in *Proceedings of the 20th International Symposium On Software Reliability Engineering (ISSRE 2009)*, 2009, pp. 1–2.

[6] J. Ding and X. Y. Zhang, "Comparison analysis of two test case prioritization approaches with the core idea of adaptive," in *2017 29th Chinese Control And Decision Conference (CCDC)*. IEEE, 2017, pp. 1723–1730.