基于深度Q网络的改进RRT路径规划算法

李昭莹[1], 欧一鸣[2], 石若凌[1]

(1. 北京航空航天大学 宇航学院,北京 100191; 2. 哈尔滨工业大学(深圳) 机电工程与自动化学院, 广东 深圳 518055)

摘 要: 针对快速搜索随机树(rapidly-exploring random tree,RRT)路径规划算法存在的随机性大、搜索效率低等问题,结合强化学习可根据先验知识选择策略的特点,提出了一种基于深度Q网络(deep Q-network, DQN)的改进RRT优化算法。首先设计复数域变步长的避障策略,并建立RRT算法中随机树生长的马尔科夫决策过程(Markov decision process, MDP)模型;然后将避障策略和MDP模型接入RRT-Connect算法的接口,并设计训练和路径规划的具体流程;最后在MATLAB软件平台上进行仿真实验。仿真结果表明,改进后的基于深度Q网络的RRT-Connect算法(DQN-RRT-C)在快速性和搜索效率上有显著提高。

关键词: 快速搜索随机树; 深度Q网络; 路径规划; 马尔科夫决策过程

中图分类号: TP18 **文献标志码:** A **文章编号:** 2096-4641(2021)03-0017-07

Improved RRT Path Planning Algorithm Based on Deep Q-network

LI Zhaoying[1], OU Yiming[2], SHI Ruoling[1]

(1. School of Astronautics, Beihang University, Beijing 100191, China; 2. Department of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, Guangdong, China)

Abstract: Aiming at the problems of large randomness and low search efficiency of rapid exploring random tree (RRT) path planning algorithm, combined with the characteristics that reinforcement learning can select strategies according to prior knowledge, an improved RRT optimization algorithm based on deep Q-network (DQN) is proposed. Firstly, the obstacle avoidance strategy with variable step in complex domain is designed, and the Markov decision process (MDP) model of random tree growth in RRT algorithm is established. Then, the obstacle avoidance strategy and MDP model are connected to the interface of RRT-Connect algorithm, and the specific process of training and path planning is designed. Finally, the simulation experiment is carried out on the MATLAB software platform. The simulation results show that the improved RRT-Connect algorithm based on deep Q-network (DQN-RRT-C) has a significant improvement in rapidity and search efficiency.

Keywords: rapidly-exploring random tree(RRT); deep Q-network; path planning; Markov decision process(MDP)

0 引 言

在无人车智能化的相关领域中,路径规划是一个关键环节,也是当前的研究热点。从大型无人驾驶客车,到小型无人配送车、无人扫地车等,均需要进行路径规划。路径规划是在模型空间中找到一条从起始点到目标点的路径解,其路径解需满足一定的约束条件,并根据实际需要满足一定的性能指标(路径长短、时间、能耗等)。通常情况下,周围的环境由一些障碍物与威胁区域组成,得到的路径不仅要满足车辆的各

收稿日期: 2021-07-13; **修订日期:** 2021-08-02

基金项目: 分布式电推进飞行器控制技术湖南省重点实验室(2020TP1017)

作者简介: 李昭莹(1983—),女,博士,讲师,主要研究方向为飞行器姿态控制与航迹规划技术。Email:lizhaoying@buaa.edu.cn

种约束，同时要保证车辆沿该路径行驶时不与障碍物发生任何碰撞。

经过多年发展，各种各样的路径规划算法相继出现，如 Dijkstra 算法、A*算法、人工势场法、概率路图（probabilistic roadmap，PRM）算法、RRT 算法等。Dijkstra 算法和 A*算法均需对地图离散化，存在算法效率低的问题[1-2]；人工势场法采用虚拟力的思想，保证了实时性但易陷入局部零势能点[3]；PRM 算法结合了随机采样方法和 A*算法，使其效率较 A*算法效率有了很大提高，但有可能无法得到路径解[4]；LaValle 于 1998 年提出的 RRT 算法，具有无需对状态空间进行建模、简单高效等优点[5-6]，但也具有随机性大、路径质量非最优等问题。许多学者提出了一系列 RRT 算法的改进算法：LaValle 和东京大学的 Kuffner 提出了 RRT-Connect 算法，搜索效率相比于 RRT 算法搜索效率有了显著提高，但采样仍具有很强随机性[7]；Sertac 等提出了渐进最优的 RRT*算法，可以得到渐进最优解，但这也导致算法的效率较 RRT 算法效率有所下降，并且这种现象随着随机树的拓展愈发明显[8-9]；刘成菊等将人工势场法中引力的思想引入 RRT 算法，增强了随机树延伸的导向性，但可能会陷入局部极小值，因此如何因地制宜地选择合适的引力系数 K_p 是关键[10]。

随着人工智能技术的发展，深度强化学习近年来取得了可喜的进展。2017 年，DeepMind 团队推出了 AlphaGo，击败了围棋世界冠军李世石，展现了深度强化学习的巨大潜力。作为深度强化学习算法之一的 DQN 算法，将深度学习和传统 Q-learning 结合在了一起，很好地解决了 Q-learning 的维度灾难问题，可以处理复杂的、高维的环境特征，并与环境进行交互，完成决策过程。

针对 RRT 算法及其变体存在效率不高的问题，本文首先提出了一种复数域变步长避障策略，使随机树延伸时具有更强的避障能力；根据所设计的避障策略，设计 DQN 的动作空间，并结合 RRT 算法的特点，进一步设计 DQN 的状态空间和奖励函数，完成 MDP 的建模。RRT 的算法结构有很多接口，因此可以将设计好的 MDP 与 RRT-Connect 的接口相结合[10]；最后设计训练和路径规划相关流程，得到基于深度 Q 网络的 RRT-Connect 算法（DQN-RRT-C），通过在 MATLAB 软件平台上进行仿真实验，验证了算法的优势。

1 原理简介

1.1 RRT 算法

RRT 算法中起始点作为根节点，然后通过随机采样增加叶子节点的方式，生成随机树并不断拓展，最终随机树到达目标点或目标区域，得到路径可行解，如图 1 所示[11]。

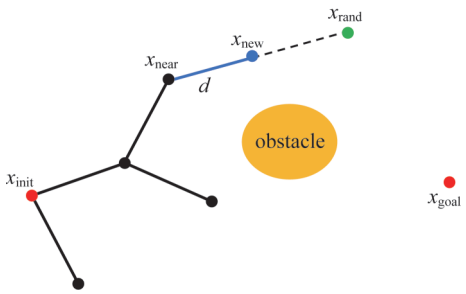


图 1 RRT 算法示意图

Fig. 1 Illustration of RRT algorithm

图 1 所示 RRT 算法随机树拓展过程可表达为

$$x_{new}=x_{near}+d\times\frac{x_{rand}-x_{near}}{\left\|x_{rand}-x_{near}\right\|} \tag{1}$$

式中：d 为常数，称为步长（Step），在传统 RRT 算法中为正实数；x_{rand} 为随机采样点；x_{near} 为随机树上距离 x_{rand} 最近的叶子节点；x_{new} 为新节点；若新树枝遭遇障碍物，则放弃此次搜索拓展，重新采样。为了控制随机树树枝密度、减少无效延伸，还可以对采样点增加限制，使

$$\left\|x_{rand}-x_{near}\right\|>r_s \tag{2}$$

式中：r_s 为常数，若不满足条件则重新采样。这样，随机树就更倾向于搜索那些还未涉足的区域，从而提高了搜索效率[12]。

为了提高 RRT 算法效率，RRT-Connect 算法被提出。RRT-Connect 算法又被称为双向 RRT 算法（bidirectional RRT，Bi-RRT）[13]。其基本思想是从初始点和目标点同时生长两棵随机树并使之最终相连接，达到快速构建随机树、提高路径规划效率的目的，如图 2 所示[7]。

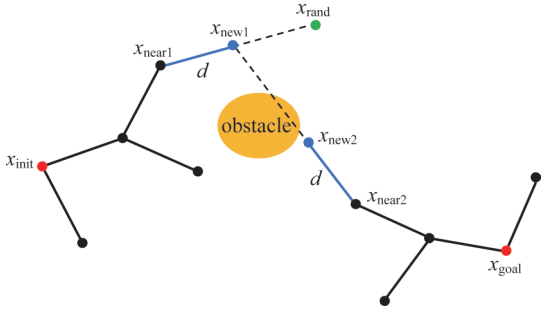


图 2 RRT-Connect 算法示意图

Fig. 2 Illustration of RRT-Connect algorithm

图2中：x_{rand}为随机采样点；x_{near1}为随机树1（图2左侧）上距离x_{rand}最近的叶子节点；x_{new1}为随机树1上的新节点；x_{near2}为随机树2（图2右侧）上距离x_{new1}最近的叶子节点；x_{new2}为随机树2上的新节点。一次搜索拓展包括随机拓展和贪婪拓展两大步骤，随机树1进行随机拓展，随机树2进行贪婪拓展。随机拓展的过程与传统的RRT算法的拓展过程相同；贪婪拓展则使用贪婪函数作为启发函数，使x_{new1}与x_{near2}之间反复迭代产生新树枝，直到遭遇障碍物或两棵随机树成功连接[5]。若贪婪拓展后两棵随机树未成功连接，则更换两棵随机树的地位，进行下一次的搜索拓展。

与RRT算法同理，在RRT-Connect随机拓展中也可以对采样点增加限制，使

$$\left\| x_{rand1} - x_{near1} \right\| > r_s \tag{3}$$

1.2　DQN算法

在实际问题中，智能体的状态值数量可能会非常庞大，传统的查表式Q-learning算法将会耗费大量内存，甚至可能在现实中无法实现，因此需要构建神经网络来近似得到所有Q值。

Q值可以用一个近似的价值函数$f(s,a;w)$来拟合，即

$$Q(s,a) \approx f(s,a;w) \tag{4}$$

式中：s为状态；a为动作；w为函数$f(s,a;w)$的参数，函数的构建则通过训练神经网络来完成。为了便于描述，我们用$Q(s,a;w)$来表示这个近似的价值函数，即

$$Q(s,a;w) = f(s,a;w) \tag{5}$$

在DQN算法中，目标Q值可表示为

$$Q(s_t,a_t) = R_{t+1} + \lambda \max Q(s_{t+1},a;w^-) \tag{6}$$

式中：t表示当前时刻；$t+1$表示下一时刻；R_{t+1}为奖励；$\max Q(s_{t+1},a;w^-)$为s_{t+1}下所有动作的Q值的最大者；λ为折扣因子，为常量。以目标值和预测值之间的均方误差作为损失函数L，即

$$L = [R_{t+1} + \lambda \max Q(s_{t+1},a;w^-) - Q(s_{t+1},a;w)]^2 \tag{7}$$

式中：$Q(s_{t+1},a;w)$为预测网络，$Q(s_{t+1},a;w^-)$为目标网络。w和w^-为神经网络的相关参数。简而言之，DQN利用神经网络作为函数逼近器来逼近$Q(s,a)$，并通过梯度下降来最小化误差[14]。

2　基于深度Q网络的RRT-Connect算法

2.1　避障策略

在传统RRT和RRT-Connect算法中，若延伸遭遇障碍物，则放弃本次拓展重新采样，因此，与RRT算法相比较，RRT-Connect算法的导向性有所提高，但其避障能力却有所下降。针对RRT-Connect避障能力下降的问题，引入了复数步长的概念，即

$$d = d_0 e^{j\theta} \tag{8}$$

$$z_{new} = z_{near} + d \times \frac{z_{ref} - z_{near}}{\left| z_{ref} - z_{near} \right|} \tag{9}$$

式中：d_0为步长长度，为正的实常数；θ为旋转角度，范围$(-\pi,\pi]$；z_{new}、z_{near}、z_{ref}分别为x_{new}、x_{near}、x_{ref}坐标对应的复数，x_{ref}为某个参考点。根据复数步长的概念，设计了复数域变步长的避障策略，如图3所示。

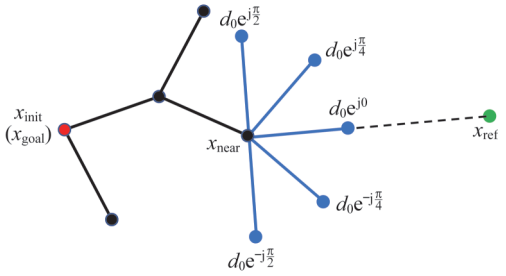


图3　参考点导向的避障策略

Fig. 3　Reference point-oriented obstacle avoidance strategy

当采样点为参考点时，便包含5个可选择的步长。当树枝延伸时，只能选择其中一个步长，而具体如何选择步长，则要结合DQN算法。

2.2　MDP模型

根据2.1节的避障策略，可以设计出DQN算法相应的MDP模型。

1）状态空间S

RRT算法一般是基于一张二值像素地图来进行路径规划的，灰度值为255的区域是障碍物空间，为0的区域则是自由空间。图中的每一个像素都作为地图上的一个已知点，并有其对应的二维坐标。因此，模型的状态空间S可表达为

$$S = \{(x,y) | 0 \leqslant x \leqslant W, 0 \leqslant y \leqslant H\} \tag{10}$$

式中：W为地图的宽度，H为地图的高度，由地图的宽高像素数量决定。

2）动作空间A

动作空间即图 3 中 5 个复数步长的集合，用一个列向量来表示

$$\boldsymbol{A}=[\,d_0\mathrm{e}^{\mathrm{j}0}\ \ d_0\mathrm{e}^{\mathrm{j}\frac{\pi}{4}}\ \ d_0\mathrm{e}^{-\mathrm{j}\frac{\pi}{4}}\ \ d_0\mathrm{e}^{\mathrm{j}\frac{\pi}{2}}\ \ d_0\mathrm{e}^{-\mathrm{j}\frac{\pi}{2}}\,]^{\mathrm{T}} \tag{11}$$

3) 奖励函数

由于以不同步长延伸，所得到的树节点与目标点的接近程度不同，因此奖励 R 是步长 d 的函数。

$$R=\begin{cases}-5 & \text{树枝遭遇障碍物}\\ 5-\dfrac{8\cdot\left|\mathrm{Arg}(d)\right|}{\pi} & \text{树枝未遭遇障碍物}\\ -2 & \text{满足条件P}\end{cases} \tag{12}$$

式中：$\mathrm{Arg}(d)$ 表示复数步长 d 的辐角主值，条件 P 的表述为：当树枝以 $d_0\mathrm{e}^{\mathrm{j}0}$ 动作到达最优动作为 $d_0\mathrm{e}^{\mathrm{j}\frac{\pi}{2}}$ 或 $d_0\mathrm{e}^{-\mathrm{j}\frac{\pi}{2}}$ 的区域时。最优动作的定义会在 2.3.2 小节中介绍，该区域通常为十分接近障碍物的区域。

2.3 算法流程

2.3.1 训练 Q 网络

DQN 算法中 Q 值的更新，可以概括为“当前状态下选择动作→执行动作→下一状态→奖励→更新 Q 值”几步，而 Q 值则使用神经网络来拟合[14-15]。根据 DQN 算法的神经网络训练方法，可以设计 DQN-RRT-C 更新一次神经网络的流程图，如图 4 所示。

图 4 中：Batchsize 表示一次训练的样本数。实际上往往需要进行多次训练更新，在 DeepMind 于 2015 年提出的 DQN 算法中，每隔 C 次训练就要把预测网络的权值赋给目标网络[16]。

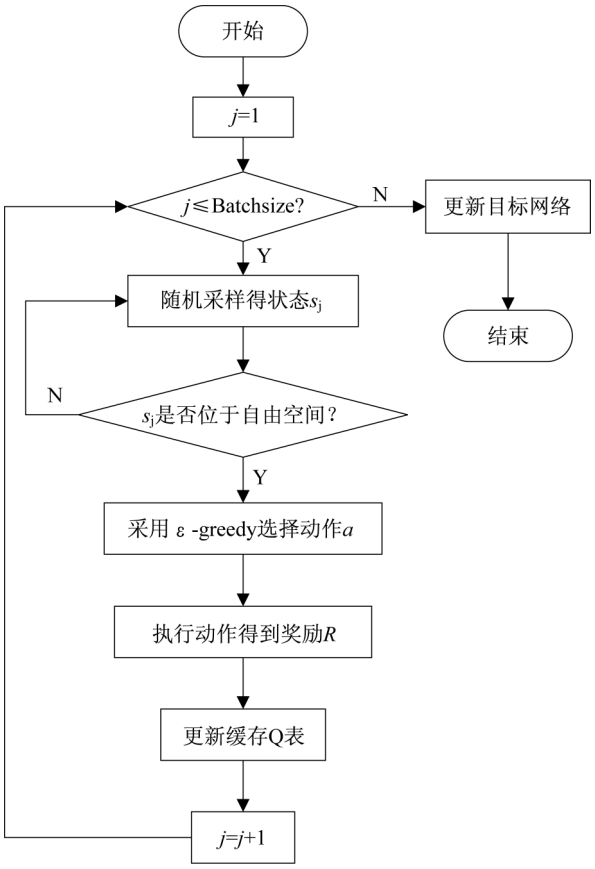


图 4 一次训练更新的流程图

Fig. 4 Flow chart of training updating

2.3.2 提取最优动作表

最优动作即动作空间中 Q 值最大的动作，一个状态对应一个最优动作。最优动作表就是将自由区域中所有状态下的最优动作用表格储存。在拥有 Q 网络的情况下，可以使用 Q 网络分别计算所有状态下的所有动作对应的 Q 值，然后将各状态下的最优动作提取出来制成最优动作表。提取出来的最优动作表，可以直接用于路径规划的动作选择。

2.3.3 路径规划

根据得到的最优动作表，在地图环境固定的情况下，可使用 DQN-RRT-C 算法进行路径规划。

与传统 RRT-Connect 算法相同，DQN-RRT-C 算法也包括随机拓展和贪婪拓展 2 大步骤。随机拓展具体方法与 RRT-Connect 算法相同，贪婪拓展则增加了参考点导向避障的环节，如图 5 所示。

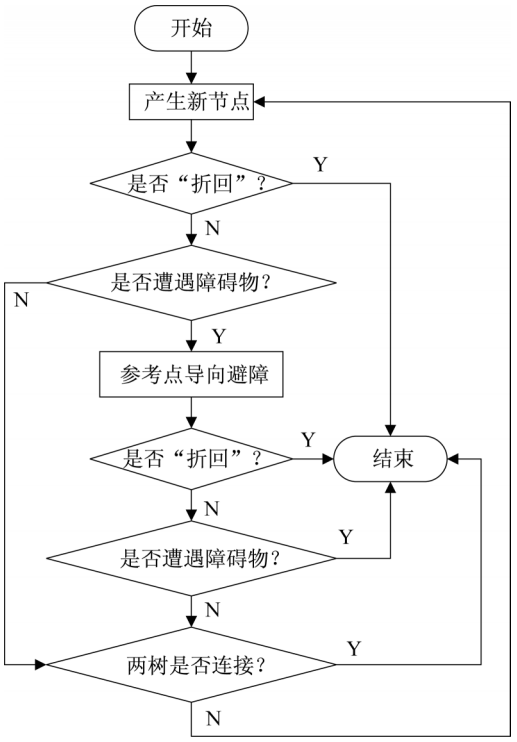


图 5 DQN-RRT-C 贪婪拓展流程

Fig. 5 Flow chart of greedy expansion for DQN-RRT-C

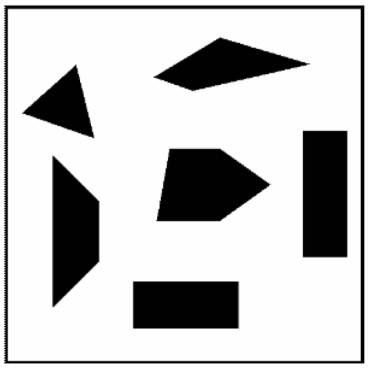
当贪婪拓展遭遇障碍物时，算法便会启用参考点导向避障策略，即：以参考点为基准，结合最优动作表，使树枝以5个复数步长中的一个延伸，达到避开障碍物的目的。由于参考点的坐标往往与起始点或者目标点并不相同，而在DQN-RRT-C算法中，随机树向参考点、起始点或目标点都有延伸的趋势，在某些情况下会产生矛盾，造成路径变长，折点变多，严重时甚至会出现局部振荡或者树枝缠绕的现象，从而导致路径质量下降，也不利于路径规划效率的提高。因此在图5中，当贪婪拓展出现“折回”现象，也就是相邻树枝夹角为锐角时，便停止贪婪拓展。

3　仿真实验及结果分析

3.1　地图模型建立

仿真实验的平台为MATLAB软件，首先建立仿真实验所使用的地图模型。二值位图在MATLAB中对应只有0和1的二值矩阵(0为自由空间，1为障碍物空间)，因此可以编写程序构建二值矩阵得到二值地图，或者根据灰度阈值将已有的彩色地图二值化。通过构建二值矩阵，得到两张仿真实验使用的500×500地图，如图6所示。

(a) 地图a



(b) 地图b

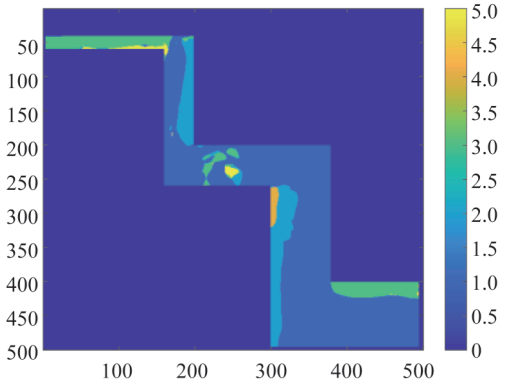
图6　地图模型

Fig. 6　Maps

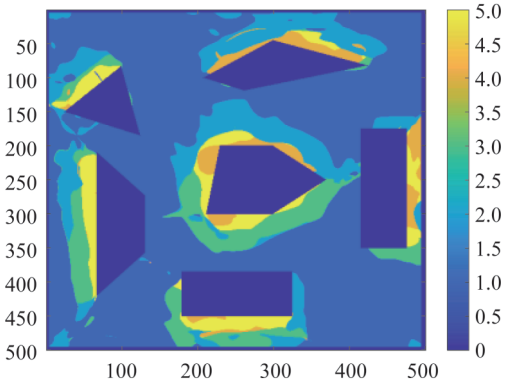
3.2　前期训练

用MATLAB编写相关程序，取目标点坐标为(450，450)，参考点坐标为(250，250)，步长长度$d_0=30$，对2张地图进行训练，并提取最优动作表。

提取出来的最优动作表，可以在地图上进行可视化。将式(11)中各动作分别用相应的列序号1～5来表示，并用颜色区分，可得到可视化图，如图7所示。



(a)



(b)

图7　DQN-RRT-C算法的最优动作可视化图

Fig. 7　Optimal action visualization diagram of DQN-RRT-C

3.3　路径规划仿真

记(50，50)→(450，450)为路线a，(450，50)→(450，450)为路线b，其中路线a起始点和目标点的连线经过中心的参考点，路线b起始点和目标点的连线则离参考点较远。取步长长度$d_0=30$，在2张地图对所有算法各进行1 000次实验，分别求取时间指标的样本均值$\bar{t}$和标准差σ，如表1所示。

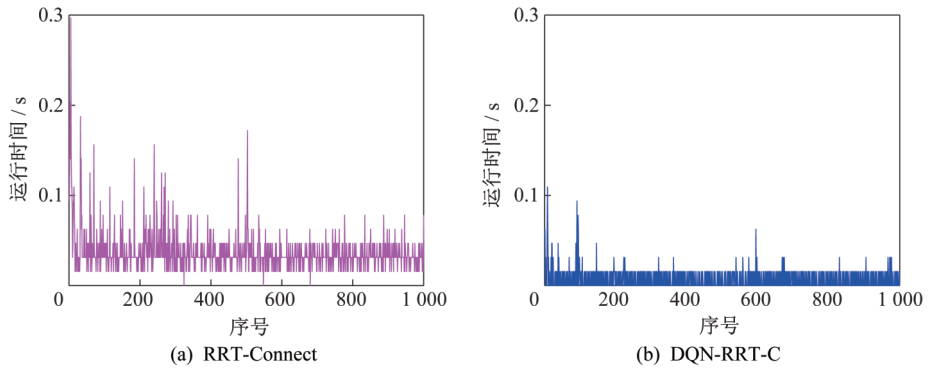
将1 000次实验的时间作图展示，如图8～10所示，图中纵轴表示每进行一次路径规划的运行时间，横轴为实验序号。

表 1 时间指标

Tab. 1 Time index

环境	算法	样本均值$\bar{t}$/ms	$\bar{t}$优化率	标准差σ/ms	σ优化率
地图 a	RRT-Connect	37.2	—	22.8	—
路线 a	DQN-RRT-C	6.8	81.7%	10.9	52.2%
地图 b	RRT-Connect	31.9	—	22.5	—
路线 a	DQN-RRT-C	15.2	52.4%	16.5	26.7%
地图 b	RRT-Connect	15.0	—	23.1	—
路线 b	DQN-RRT-C	10.0	33.3%	14.3	38.1%

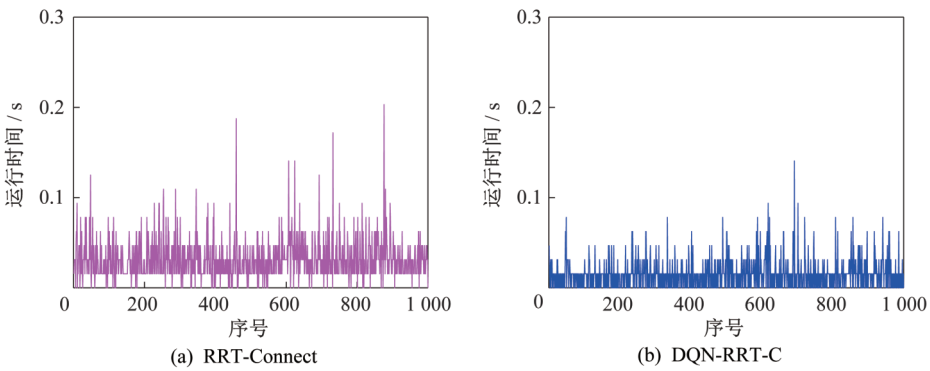
注:优化率 $=(1-$优化后的指标/优化前的指标$)\times 100\%$



(a) RRT-Connect (b) DQN-RRT-C

图 8 地图 a 路线 a 的时间指标

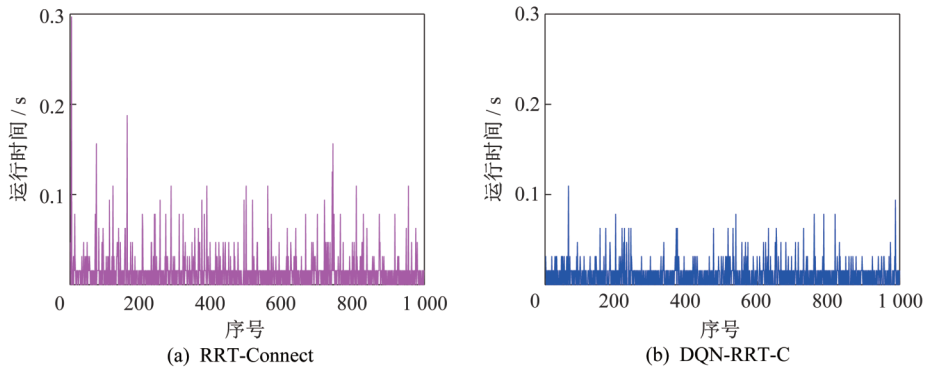
Fig. 8 Time of path a in map a



(a) RRT-Connect (b) DQN-RRT-C

图 9 地图 b 路线 a 的时间指标

Fig. 9 Time of path a in map b



(a) RRT-Connect (b) DQN-RRT-C

图 10 地图 b 路线 b 的时间指标

Fig. 10 Time of path b in map b

4　结束语

DQN-RRT-C算法只需要依赖固定地图的信息，运算时间的样本均值越小，表明算法的平均运行时间越少，算法效率越高；运算时间的标准差越小，表明算法运行时间随机性越小，算法时间性能越稳定，可靠性越好。由时间指标的实验结果可知，无论是在效率上还是在时间性能稳定性上，DQN-RRT-C算法的时间性能均优于RRT-Connect算法的时间性能，但受路线的影响较大，且仍存在一定的随机性。

参　考　文　献

[1] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths [J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.

[2] DIJKSTRA E W. A note on two problems in connexion with graphs[J]. Numerische Mathematik, 1959, 1(1): 269-271.

[3] KHATIB O. Real-time obstacle avoidance for manipulators and mobile robots[C/OL]//Proceedings. 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA. IEEE, 1985: 500-505 [2021 - 06 - 14]. http://ieeexplore. ieee. org/document/1087247/. DOI: 10. 1109/ROBOT. 1985. 1087247.

[4] 宋宇，王志明. 基于角点PRM的移动机器人路径规划[J]. 长春工业大学学报，2019，40(4)：344-348.

[5] 莫栋成，刘国栋. 改进的RRT-Connect双足机器人路径规划算法[J]. 计算机应用，2013，33(8)：2289-2292.

[6] 李永丹，马天力，陈超波，等. 无人驾驶车辆路径规划算法综述[J]. 国外电子测量技术，2019，38(6)：72-79.

[7] KUFFNER J J, LAVALLE S M. RRT-Connect: An efficient approach to single-query path planning[C]//Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, April 24-28, 2000, San Francisco, CA, USA. IEEE, 2000.

[8] KARAMAN S, FRAZZOLI E. Incremental sampling-based algorithms for optimal motion planning[C]//Robotics: Science and Systems VI. Robotics: Science and Systems Foundation, 2010.

[9] 杨也，倪建军，陈一楠，等. 改进RRT*的室内机器人路径规划算法[J]. 计算机测量与控制，2020，28(1)：241-245.

[10] 刘成菊，韩俊强，安康. 基于改进RRT算法的RoboCup机器人动态路径规划[J]. 机器人，2017，39(1)：8-15.

[11] LAVALLE S M, KUFFNER J J Jr. Randomized kinodynamic planning[J]. The International Journal of Robotics Research, 2001, 20(5): 378-400. DOI:10. 1177/02783640122067453.

[12] NADERI K, RAJAMÄKI J, HÄMÄLÄINEN P. RT-RRT*: A Real-Time Path Planning Algorithm Based on RRT*[C/OL]//Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games. Paris France: ACM, 2015: 113-118 [2021 - 04 - 21]. https://dl. acm. org/doi/10. 1145/2822013. 2822036. DOI:10. 1145/2822013. 2822036.

[13] 张鹏宇. BiRRT-ACO融合算法在机器人路径规划中的应用研究[D]. 开封：河南大学，2019.[知网]

[14] 苏达桑·拉维尚迪兰. Python强化学习实战：应用OpenAI Gym和TensorFlow精通强化学习和深度强化学习[M]. 连晓峰，译. 北京：机械工业出版社，2018：119 - 122.
苏达桑·拉维尚迪兰. Python强化学习实战 应用OpenAI Gym和TensorFlow精通强化学习和深度强化学习[M]. 北京：机械工业出版社，2019.

[15] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J/OL]. arXiv preprint, 2013: ArXiv: 1312. 5602 [Cs], [2021 - 04 - 30]. http://arxiv. org/abs/1312. 5602.

[16] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.