Journal of System Simulation

Volume 31 | Issue 1 Article 3

4-16-2019

DP-Q(λ): Real-time Path Planning for Multi-agent in Large-scale Web3D Scene

Fengting Yan
School of Software Engineering, Shanghai 201804, China;

Jinyuan Jia
School of Software Engineering, Shanghai 201804, China;

DP-Q(λ): Real-time Path Planning for Multi-agent in Large-scale Web3D Scene

Abstract

Abstract: The path planning of multi-agent in an unknown large-scale scene needs an efficient and stable algorithm, and needs to solve multi-agent collision avoidance problem, and then completes a real-time path planning in Web3D. To solve above problems, *the DP-Q(λ) algorithm is proposed; and the direction constraints, high reward or punishment weight training methods are used to adjust the values of reward or punishment by using a probability p (0-1 random number). The value from reward or punishment determines its next step path planning strategy.* If the next position is free, the agent could walk to it. *The above strategy is extended to multi-agent path planning, and is used in Web3D.* The experiment shows that the DP-Q(λ) algorithm is efficient and stable in the Web3D real-time multi-agent path planning.

Keywords

Web3D, large-scale unknown environment, multi-agent, reinforcement learning, dynamic rewards p, path planning

Recommended Citation

Yan Fengting, Jia Jinyuan. DP-Q(λ): Real-time Path Planning for Multi-agent in Large-scale Web3D Scene[J]. Journal of System Simulation, 2019, 31(1): 16-26.

第 31 卷第 1 期 系统仿真学报© Vol. 31 No. 1
2019 年 1 月 Journal of System Simulation Jan., 2019

DP-Q(λ):大规模 Web3D 场景中 Multi-agent 实时路径规划算法

闫丰亭，贾金原
(同济大学，上海 201804)

摘要：大规模场景中 Multi-agent 可视化路径规划算法，需要在 Web3D 上实现实时、稳定的碰撞避让。提出了*动态概率单链收敛回溯 DP-Q(λ)算法，采用方向启发约束，使用高奖赏或重惩罚训练方法，在单智能体上采用概率 p(0-1 随机数)调节奖罚值，决定下一步的寻路策略*，同时感知下一位置是否空闲，完成行走过程的避碰行为，*将单智能体的路径规划方案扩展到多智能体路径规划方案中，并进一步在 Web3D 上实现了这一方案*。实验结果表明：该算法实现的多智能体实时路径规划具备了在 Web3D 上自主学习的高效性和稳定性的要求。

关键词：Web3D；大规模未知环境；多智能体；强化学习；动态奖赏 p；路径规划
中图分类号：TP391 文献标识码：A 文章编号：1004-731X (2019) 01-0016-11
DOI: 10.16182/j.issn1004731x.joss.16PQS-003

DP-Q(λ): Real-time Path Planning for Multi-agent in Large-scale Web3D Scene

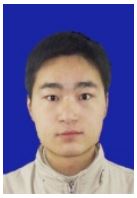
Yan Fengting, *Jia Jinyuan*
(School of Software Engineering, Shanghai 201804, China)

Abstract: The path planning of multi-agent in an unknown large-scale scene needs an efficient and stable algorithm, and needs to solve multi-agent collision avoidance problem, and then completes a real-time path planning in Web3D. To solve above problems, *the DP-Q(λ) algorithm is proposed; and the direction constraints, high reward or punishment weight training methods are used to adjust the values of reward or punishment by using a probability p (0-1 random number). The value from reward or punishment determines its next step path planning strategy.* If the next position is free, the agent could walk to it. *The above strategy is extended to multi-agent path planning, and is used in Web3D.* The experiment shows that the DP-Q(λ) algorithm is efficient and stable in the Web3D real-time multi-agent path planning.

Keywords: Web3D; large-scale unknown environment; multi-agent; reinforcement learning; dynamic rewards p; path planning

引言

在虚拟现实领域中，Web3D 上大规模复杂环境中的多智能体通过强化学习进行路径规划，向来是一个具有重要研究价值的课题，该课题也伴随着巨大的挑战。Agent 通过与环境的交互，获得奖罚，进行最优路径规划，避免单智能体与障碍物的碰撞，制定多智能体间碰撞避让策略。目前该领域研究的热点主要集中在如何采用智能算法进行路径规划[1-3]。智能算法各具特点，但普遍存在一些缺点：要么是智能体不具备自主学习能力，要么就是



收稿日期：2016-05-31 修回日期：2016-08-04;
基金项目：国家自然科学基金面上项目(61272270);
作者简介：闫丰亭(1980-)，男，山东，博士生，研究方向为虚拟现实与机器学习；贾金原(1963-)，男，内蒙古，博士，教授，研究方向为分布式虚拟现实、Web3D、游戏引擎。

算法复杂度太高，导致了最优路径规划收敛速度过慢[4-5]。这些缺陷制约了智能路径规划在 Web3D 虚拟现实场景中的实际应用。对于多个智能体要实现的路径规划、避让和避碰的功能，使得路径规划决策的计算复杂度进一步提高。

在障碍物未知的动态复杂环境中，智能体需要通过自主学习机制，计算出到达预定目标位置的最优路径[6]。近年来，人们借助强化学习方法本身所具有的自主学习能力，尝试在未知环境下为智能体进行自主路径规划[7]。

就强化学习而言，目前该研究领域中普遍采用的学习算法有 Q-learning 算法和 Q(λ)算法[8]，但这些算法也存在学习时间长且收敛速度慢的缺陷。尤其是在解决复杂环境下路径规划时，这些算法的适应性不够强。为解决这些缺陷，本文改进了现有强化学习算法，提出了动态概率单链收敛回溯 DP-Q(λ)算法，智能体可以边学习边建立状态链，基于回溯思想改善 Q(λ) 学习中数据传递的滞后性，根据奖赏或惩罚的次数增大奖赏或惩罚的权重，并给予调节参数 p，防止过度调节，使得当前状态的动作决策能够快速并适度地受到后续动作决策的影响，然后将该算法应用到单个智能体和多个智能体的路径规划中，从而在未知环境下进行路径规划，最终提高了智能体的学习速度，也加速了路径规划中智能体的避障和避让，使得智能体在网页上可以实时寻得最优路径并被可视化呈现。本文首先通过仿真实验验证了该算法的有效性，随后使用 AS3.0 强面向对象语言，实现了单个智能体、多个智能体和多个智能群在 2D 网页的虚拟场景中进行路径规划，并做了展示。最后在 Web3D 上实现了大规模复杂场景中多智能体的路径规划的应用。

本文提出的动态概率单链收敛回溯 DP-Q(λ)算法的基本思路，是基于回溯思想，根据不同状态具有不同收敛优先级的特点，采取距离目标状态越近的状态区域先收敛，为后续状态的收敛提供必要条件，使用回溯的技巧改进 Q(λ)算法中数据传递滞后的缺陷[9]，累计奖赏或惩罚的次数，相应的增大奖赏或惩罚的权值，并给予调节参数 p，防止过度调节，使得当前状态的动作决策能够快速并适度的受到后续动作决策的影响，提速当前状态的动作决策受到后续动作决策的影响，从而加速算法的学习速度，减少了自主学习时间。

对于多个智能体的路径规划，采用 leader 思想，减少每个智能体都需要自主进行路径规划的冗余计算，同时，又切近现实场景中的实际情景，这是因为智能体是在社会心理约束的条件下，具有社会属性的人群运动的特点[10]。多个智能体的路径规划，是以多个运动智能体为研究对象，需要研究动态不确定环境中各个智能体自主学习的策略，根据社会心理学的原理，解决每个智能体的最优路径计算，每个智能体和环境的避碰，以及智能体间的碰撞问题，在最短的时间内，到达最终目标位置。该方法提高了规划效率，使得整个系统具有了良好的学习能力和规划效率。最后在 Web3D 场景中，将该算法实现的路径规划，给予了展示。

1 背景

多智能群是一种在复杂环境下具有自行组织、自主运动和自主规划的多个智能体群体。多智能体的路径规划是基于单个智能体在一定约束规则下，为达到某一个目标进行路径规划，能够自主学习并实现最优路径规划[11]。

单智能体要研究的问题可以归纳为 3 个方面的内容：1) 确定单智能体所处的位置；2) 确定单智能体的目标位置；3) 智能体在如何避障的前提下完成路径规划。首先是通过地图构建，建立环境模型[12]，然后智能体通过定位自身所在的具体位置，进行路径规划决策，这个决策过程就是智能体按照一定的标准(如行走路径最短、行走时间最少或费用最少等)，计算出一条从起点到目标位置的无碰路径[13]。本文研究的重点在于网页 3D 上的单智能体和多智能体能够在突发事件情况下，从未知复杂环境中，自主导航，自行碰撞避让，规划出一条最短路径[14]。

第 31 卷第 1 期 系统仿真学报 Vol. 31 No. 1
2019 年 1 月 Journal of System Simulation Jan., 2019

路径规划基本可分为如下几类[15]：1) 已知环境下的静态障碍物路径规划；2) 未知环境下的静态障碍物路径规划；3) 已知环境下的动态障碍物路径规划；4) 未知环境下的动态障碍物的路径规划。可以根据智能体对环境掌握情况的不同，将智能体路径规划分为两类：基于先验知识且环境信息完全已知的全局路径规划和基于试错学习方式的局部路径规划[16]。

局部路径规划算法(Local path planning)[17]是在场景信息完全未知或部分未知情况下，通过和环境信息不断沟通，进行识别场景，从而确定出智能体在地图中的当前位置及其周围局部范围内障碍物的分布情况，基于此情况，规划出一条从当前位置到某一目标位置的最优路径。局部路径规划算法主要有：人工势场法、模糊逻辑算法、遗传算法和人工神经网络算法等。

全局路径规划算法(Global path planning)[18]是指在空间环境完全已知条件下，智能体根据先验知识自出发点到目标点，找到一条符合一定性能的可行的或者最优的路径,目前主要的全局路径规划算法有 4 种：可视图法、自由空间法、栅格法、拓扑法[19]。全局规划简单易行，计算实时性强，便于路径规划的实现，但在动态规划中，计算量大，实时性不强，不能保证寻到最优路径，不适合动态环境下的路径规划[20]。

智能寻路系统的特征是适应未知环境，学习能力是智能系统的核心。就机器学习而言，根据学习方式的不同，学习方法分为监督学习(Supervised learning)、非监督学习(Unsupervised learning)和强化学习(Reinforcement learning)三大类[21]，其中强化学习是依据环境反馈作为输入的、特殊的、适应环境的机器学习方法。

强化学习通过智能体与环境交互进行在线学习，适用于智能体在环境未知情况下的路径规划控制[22]。最早提出强化学习算法的是 Watkins,该算法是在对环境无任何先验知识情况下，通过不断试错获得最大化奖赏值的一种控制策略[23]。传统强化学习收敛速度慢，学习时间长，针对该缺陷，国内外学者提出了很多的改进算法，主要有以下 4 类[24]：

Q 值更新策略：Xin Ma 等根据传 Q 学习算法提出了 Q(λ)学习算法解决奖赏延时问题，Q(λ)学习是一种增量式多步 Q 学习算法，通过不断修正前面状态增强对后面状态的观测，比如 Q(λ)学习方法每次可对前面多步的状态数据进行快速的信息优化；Hwang 提出通过加快强化学习的学习效率应用于复杂未知环境中。

动作选择策略：贪心策略每次根据当前状态-动作对选择最大化的值函数对应的动作值，可以理解为先达到局部最优，然后达到全局最优，但贪心策略往往容易陷入局部最优解。

Q 值初始化策略：Dearden 提出选择合适的 Q 初始化值可以加快强化学习算法的学习速度；Hong-yang 等提出模糊规则用于 Q 值的初始化来提高学习效率；势能场函数和神经网络同样也可以用于 Q 值的初始化以提高强化学习的学习效率。

减小状态空间策略：针对较为复杂的环境，智能体在运动过程中会有很多状态，这就需要很大的 Q 值表存储相应的 Q 值。Martin 等提出通过坐标排除法减小状态空间；Wang 等提出了采用自适应的方法调整状态的数量；S.K.Pradhan 根据状态的临界值将大的状态空间划分为多个小的状态空间；Barraquand 通过对离散的状态空间重新定义来限制状态空间的大小。

常见的强化学习算法有瞬时差分(TD(0)和 TD(λ))算法，Q 学习算法，SARSA 算法，Monte-Carlo 方法，R 学习算法，Q-learning 算法和 Q(λ)算法。

经典强化学习算法 Q-learning 算法和 Q(λ)算法在较大规模和复杂环境下路径规划中很难达到理想的效果，其最大缺陷就是学习时间长，获得最优值的收敛速度慢。为解决该问题，本文提出了动态概率单链收敛回溯 DP-Q(λ)算法，该算法是的智能体在学习中建立状态链，通过回溯奖罚反馈值改善学习中数据传递的滞后性，使当前状态的动作决策能够根据奖赏值次数以及概率参数 p，快速地受到

后续动作决策的影响，并被应用到单个智能体、多个智能体在未知环境下的路径规划中，从而解决了学习速度慢，以及多个智能体的避障和避碰问题，使得每个智能体都能够快速正确地找到一条最优路径。最后进行了实验数据对照，证明了该算法的高效性，通过编程实现了 Web3D 多智能体寻路系统软件(使用强面向对象语言 AS3.0 实现)。

2 动态概率单链收敛回溯 DP-Q(λ)算法

高奖赏，重惩罚的信息反馈，可使得跟踪迹具有更强的作用效果，从而加速收敛的速度。本文提出的奖惩收敛记忆表，使用记忆体以及抽样的方法，在数据存储结构上使用向量数组存储查询方式，为多智能体规划路径在路径相交时，后续智能体就不再进行路径查询，直接利用前面智能从该位置处的路径规划方案，该方法大大节省了智能体寻路的规划时间。

2.1 算法的提出

使用迭代反馈概率数据方式，加速单链路径上的迭代收敛速度。将当前获得的 q 值逐步向后反馈，更新走过的数据 Q 值表。Q 值表中的数据是迭代排序的，可从最小 Q 值表中得到最小代价值的状态，作为下一步动作的当前状态，确保最终得到一条最短路径。

使用高奖赏，重惩罚的收敛记忆表记录数据，通过记忆矩阵可以获得记忆表中的数据，定义该记忆矩阵：

$$M_t \leftarrow [(s_t, a_t), r_t, \lambda_t] \tag{1}$$

用到的索引向量数组：$(s_t, a_t) \leftarrow [i_t, m_j]$，这里 i_t 是对应的 Q 值表中，对更新链中 s_t 状态在 Q 值表中的索引，m_j 记录的是第 d_j 个群体有无路径规划通过的情况，其初始值 m_j=0，若有规划的路径通过该位置，那么 m_j = 1，则后续智能体只需要规划到该位置的最短路径即可。

动态概率单链收敛回溯 DP-Q(λ)算法，使用记忆矩阵获取到状态链，当路径上没有智能体通过时，采用高奖赏重惩罚概率回溯的思想，依次迭代计算获得并更新 q 值：

Initalize $Q(s, a)$ = int(math.random()) and M= $[(s,a),r,\lambda]$

Repeat (for each episode)

Initialize s_t

Repeat (for each step of episode)

Select and perform an action

$a_t = \arg\max_{a \in A} Q(s_t, a)$

Observe the subsequent state s_{t+1}

Receive an immediate r_t reward

Initiate m_j =0

Initiate $(s_t, a_t) \leftarrow [i_t, m_j]$

Add a new row for memory matrix

$Mt \leftarrow [(s_t, a_t), r_t, \lambda_t]$

If (agent = Multi-agents)

then m_j = 1

for k = t-1 to 1 do

$$Q_{t+1}(s_k, a_k) \leftarrow (1-\lambda_k) Q_t(s_k, a_k) + \lambda_k (1+p)\{r_k + \gamma \max_{a_t \in A} Q_{t+1}(s_{k+1}, a_k)\}$$

Update Q-value along the state chained in M_t

Until k is 1

Until s is terminal

2.2 数据更新图

在通过不断迭代中，$t+1$时刻更新状态 s_t 对应的 Q 值的同时，单链中状态 s_t 前面的状态 s_{t-1}，s_{t-2}.....s_2, s_1，根据惩罚次数以及随机概率 p，也同时被一步更新，从而完成单步重复迭代。

动态概率单链收敛回溯 DP-Q(λ)算法的迭代过程步骤：

第一步训练 $s_0 \leftarrow s_1$

第二步训练 $s_0 \leftarrow s_1 \leftarrow s_2$

第三步训练 $s_0 \leftarrow s_1 \leftarrow s_2 \leftarrow s_3$

.....

第 n 步训练 $s_0 \leftarrow s_1 \leftarrow s_2 \leftarrow s_3 \cdots\cdots s_{n-1} \leftarrow s_n$

https://dc-china-simulation.researchcommons.org/journal/vol31/iss1/3
DOI: 10.16182/j.issn1004731x.joss.16PQS-003

第 31 卷第 1 期 系统仿真学报 Vol. 31 No. 1
2019 年 1 月 Journal of System Simulation Jan., 2019

其中：s_0 表示起始点；s_1 表示经历的第一个状态；s_2 表示经历的第二个状态；s_n 表示经历的第 n 个状态，箭头表示数据传递方向，所传递的数据包括奖励值 r_t 和某状态所占有的 Q 值。

与单步的 Q-learning 算法相比，Q 值的更新速度变快，那么学习收敛需要的步数相应的减少了，所以收敛速度变快。

如果在某状态下，可以不断得到下一步的可行状态，假设已有 i 个状态均可行，则已有状态得到奖赏值为 $0.0i$ 奖赏；如果传递回来的不可行状态次数为 j，则得到 $0.0j$，的惩罚值。为了防止过度奖赏，给予一个随机数 p，使得奖赏值被调整到 $0.0i\times p$ 和 $0.0j\times p$。

2.3 环境模型

使用栅格法建立环境模型，规定$(x, 0)$、$(0, y)$、$(x_{\max}, y)$、$(x, y_{\max})$为墙壁，如图 1 所示。

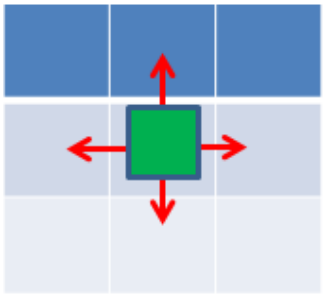


图 1 基于栅格的寻路算法
Fig. 1 Path planning based on grid

该实验采用栅格法，建立栅格的环境模型，等分网络，每个网络交汇处坐标格作为 agent 的一个状态。每一个状态都是环境矩阵 $\boldsymbol{EM}$ 的一个元素，环境中的障碍物用整齐方格表示，设在场景中占有的方格数为 20%，其位置随机分布。可以通过随机的障碍累积形成不同形状的障碍物，单元障碍物方框的边长为 1，相应的 EM 的元素值为 1。环境矩阵 $\boldsymbol{EM}$ 可以表示为：$\boldsymbol{EM}=\{em_{ij}\,|\,em_{ij}=0 \text{ or } 1, i, j\in N^+\}$，若 $em_{ij}=0$，表示该状态为自由区域，即该状态处没有障碍；若 $em_{ij}=1$，表示该状态为墙壁，是障碍物所在区。

我们实验中建立 72×52 的坐标，除去四边的墙壁，内部可用的空间为 70×50，单位间隔距离为 1，共计有 3 500 个状态。

设定智能体大小为一个 10×10 像素的对象，使用坐标 (x_0, y_0) 表示其所在位置。智能体可有 8 个方向的动作选择，可采用单步长分别在上方、下方、左方、右方、左上方、左下方、右上方和右下方[21]选择路径。智能体沿着栅格线行走，变量 $\boldsymbol{A}$ 代表离散动作的矩阵，将其定义为 $\boldsymbol{A}$ = [0,1; 0,–1; –1,0; 1,0; –1,1; –1,–1; 1,1; 1,–1]，A 用于表示 8 个动作值。

2.4 奖赏函数定义

奖赏函数是根据每个动作的执行，与外界环境作用后得到的反馈信息，这些信息就是奖罚值。奖赏函数的设置，直接影响到学习速度。本文采用非线性奖惩函数，表征立即奖惩值，该分段函数设计如下：

$$r_t=\begin{cases}0.1 & ws=0, \text{diagonal}=0\\ 0.14 & ws=0, \text{diagonal}=1\\ -0.2 & ws=1\\ 1 & aim\end{cases} \tag{2}$$

若智能体在运动过程中没有碰到障碍物，且动作选择了上下或者左右状态，则获得 0.1 的立即奖赏；当智能体在运动过程中没有碰到障碍物，且动作选择了对角状态，则获得 0.14 的立即奖赏；如果智能体下一步的行走会碰到障碍物，则获得–0.2 的奖赏；若智能体到达目标位置，会获得最大的奖赏值 1。

2.5 Q 值表的建立

Q 值表是 Q 值的 lookup 表，它的使用是强化学习算法的一项关键技术，存储的环境坐标也对应着动作数据。针对本文采用的栅格环境模型中的路径规划，智能体在每一个栅格处可到达的状态对应 8 个动作，这里给每一个动作分配一个 q 值，比如使用 72×52 的坐标系，内部有 3 500 个状态可到达，则拥有 3 500 个 q 值，我们初始化 Q 值表为 1×3 500 的零矩阵。

Q 值表建立后，让智能体一遍又一遍的从起点出发一直到达目标点，不停地完成训练循环，对应地修改 Q 值表，当智能体找到了一条最优路径后则结束探索。每一次的训练循环，智能体都从起点出发，不停的训练。当智能体在动作集合中选择某一动作后，环境接收该动作并发出状态转移，同时给出应得的奖赏 r。如果其动作获得了环境正的奖赏(+r)，则智能体此后产生该动作的趋势就会增强；反之，如果获得的是环境负的奖赏(–r)，则该动作趋势就会减弱。最终通过足够多次的训练，Q 值表会表现出相对收敛的情况，机器人将会在 Q 值表中找到一条最优的路径。

2.6 其它参数初始化

初始化折扣因子 $\gamma = 0.96$，及学习效率 $\lambda_1 = 0.25$；路径长度标准差期望 $\text{Convgoal} = 0.01$；最大训练循环次数 $\text{Trials} = 300$；每次尝试最大探索步数 $\text{Maxite} = 2000$；最低尝试次数 $\text{Avgtrials} = 20$。

路径规划的目标是找到一条无碰撞的最优路线，通过强化学习方法，收敛到一个最好的值，这个值是路径长度最短的一条路线。接着智能体会沿着该路径通过可视化技术，行进到目标位置。通过记录所有尝试次数中寻找到的路径，以路径长度的标准差衡量算法收敛标准。若路径长度的标准差小于路径总长度标准差期望 Convgoal，则可以认为该学习是收敛的。

根据 Q-learning 算法的特征可知，该算法的智能体在探索路径的前期尝试中，所探索的路径长度会呈现出很大的误差，需要通过人工方式，进行数据过滤，并需要设定一个最低的尝试次数 Avgtrials。如果智能体尝试的次数小于 Avgtrials，此时路径的长度将不被计入到标准差计算中。

2.7 路径规划策略

针对动态未知环境下的路径规划，需要避免多个智能体间大冲突，并根据强化学习的原理，进行路径规划，规划步骤如下：

Step 1：初始化各项数据，开始从起点到终点进行循环训练。

Step 2：智能体从起点准备出发。

Step 3：获取当前坐标位置的 8 个动作状态对应的 8 个值，通过贪婪决策，找出最大 q 值对应的动作，记录原坐标。

Step 4：获取下一个坐标位置和获得的奖励值，如果下一个坐标位置上有障碍物，则智能体下一个坐标从另外 7 个坐标中选择一个，那么该步获得奖励值为–0.2；如果下一坐标上无障碍物，并且如果是上、下、左、右方向，则返回奖励值为 0.1，对角方向上则返回奖励值为 0.14，并进入到下一个坐标；如果下一坐标位置是终点，则返回奖励值为 1。接着智能体通过算法更新相应的 q 值；如果下一个位置是一个动态智能体，则该智能体切换到等待状态，直到前面智能体走掉。

Step 5：如果该智能体群中 m_j=0,则继续单智能体寻路；若 m_j=1,则直接读取 Q 值表数据取得后续路径坐标值。

Step 6：判断当前坐标位置是否是终点，若是，则进入到 step 7，否则，进入 step 3。

Step 7：判断智能体尝试次数是否超过 Avgtrials，如果超过，则进入 step8，否则进入 step2。

Step 8：判断获得的路径总长度的标准差的期望是否小于 Convgoal，若小于则进入 step9，否则从 step2 再次开始寻路；

Step 9：寻路过程结束。

动态环境中，群内智能体通过对 m_j 状态值的判断，若 m_j=0，执行单智能体寻路，若 m_j=1，则从该处索引$|(s_t, a_t)$读取记忆体中对应该位置存储的路径坐标，添加到自身后续路径序列中；智能体行走过程中，到达某位置，需要将该位置 EM 中的 n_j 数值置为 1；智能体行走过程中，需要判断下一位置中是否有动态智能体存在，即其 EM 中 n_j 状态值是否为 1，若为 1，则表明该位置被占用，需要等待智能体离开某位置，则该位置重新置 n_j=0。

第 31 卷第 1 期 系统仿真学报 Vol. 31 No. 1
2019 年 1 月 Journal of System Simulation Jan., 2019

2.8 实验结果的比较

在可移动空间为 70×50 网格中，每个网格 10 个像素。在网页屏幕上，我们设定演示屏幕尺寸是 800×600，比较适合人眼视觉的普遍要求；在移动终端设备上可根据屏幕尺寸或网页页面大小自适应调节界面大小。本实验中，我们设定场景中障碍物是随机分布的，障碍物的平均密度设定为 20%，通过随机分布的方式放置于不同的位置，可以通过本文实现的模型随时进行“重置地图”，根据需要重新随机分布障碍物和智能体，布局试验场景。使用该算法开发的原型系统，可设定智能体的数量、智能体的群数以及设定智能体的目标位置，在网页或移动终端设备上，进行单智能体、多智能体和智能群的路径规划。

使用本文提出的算法，与一些经典或者比较优秀的算法做了对比实验。本文提出的多智能体动态概率单链收敛回溯 DP-Q(λ)寻路算法与经典 Q(λ)算法、Q-learning 算法、TD(λ)算法，以及优秀的 potential field-based Q-learning 算法，就单智能体路径规划策略收敛速度和收敛值上进行了对比实验。实验结果如图 2 所示。

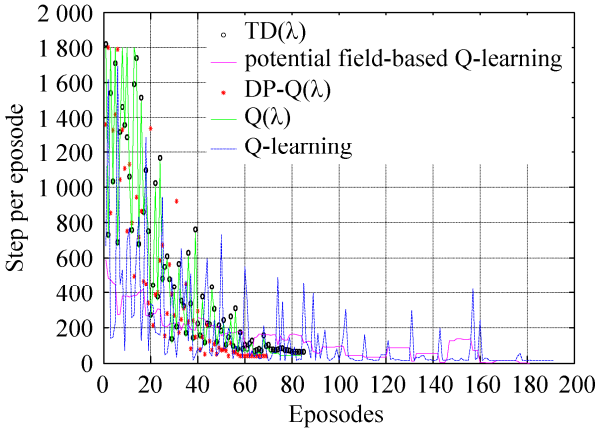


图 2 DP-Q(λ) 算法
Fig. 2 DP-Q(λ) algorithm

从图 2 可以看到，本文提出的 DP-Q(λ)算法就收敛速度看，远远快于 Q-learning 算法，也快于传统的 Q(λ)算法和 TD(λ)算法。对于改进的优秀算法 potential field-based Q-learning 算法，由于场景的复杂性，使得路径搜索过程中会频繁出现较大的震荡性。DP-Q(λ)算法初始搜索步数较多，是因为初始反馈回来的信号相对较少，没有体现出来 p 的优势，但随着迭代次数增多，优势越来越明显。该算法在 Eposodes=58 的时候达到了收敛，Q-learning 算法在 Eposodes=162 的时候达到了收敛，Q(λ)算法在 Eposodes=68 的时候达到了收敛，其它两种强化学习算法，收敛速度以及收敛值也略微差于 DP-Q(λ)算法的计算结果。

3 单智能体、多智能体和智能群寻路规划

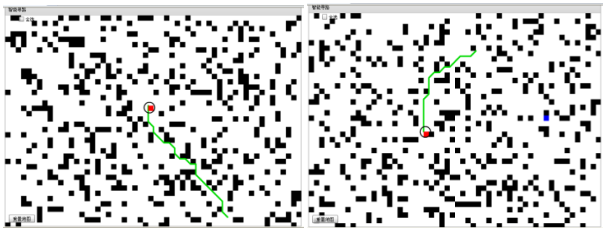
基于本文提出的 DP-Q(λ)算法，在网页和移动终端设备上实现了多智能体路径规划原型系统，该原型系统可以在网页和移动终端设备上进行单智能体、多智能体和智能群的路径规划工作。我们的工作完成了智能体和静态场景的避碰。多智能体路径规划在网页和移动终端设备 Web 的实现，满足当前互联网时代的需要，方便了人们对寻路功能的使用，为进一步的扩展路径规划在多领域的实际需要，提供了基础解决方案。

对于单智能体、多智能体以及智能群，我们使用强面向对象语言 ActionScript3.0，在 DP-Q(λ)基础的上进行了实现。这在以下部分做了阐述，并展示了在网页上的寻路情况以及分析了对应的数据。

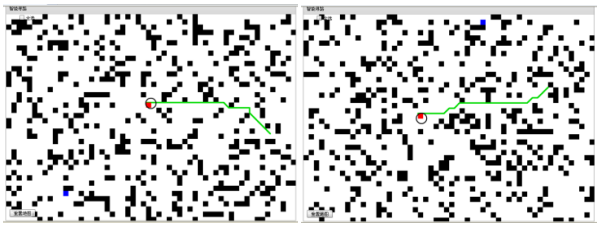
3.1 单智能体在网页上的路径规划

基于 ActionScript3.0 强面向对象语言，我们实现了单智能体在 Web 上和移动终端设备 Web 上路径规划。我们采用了 70×50 个网格的路径规划场景，每个网格设定为 10 像素，界面设计为 800×600 像素。

图 3 分别对单智能体从不同方向和不同场景中，进行了寻路实验。红色方块为单智能体，该智能体行走的路线通过绿色粗线表示。黑色方块部分为随机产生的障碍物，障碍物占空间的 20%，随机分布。白色区域为可行走空间。



(a) scene 1 (b) scene 2



(c) scene 3 (d) scene 4

图 3 单个智能体在网络平台上的寻路图
Fig. 3 Path planning of one agent on network platform

表 1 是该次寻路搜索的指标数据，我们通过 5 个指标说明这次寻路的效率情况。从数值上看，单智能体的平均寻路时间远远低于 1 s。在打开网页时，人们往往可以忍耐的时间为 2 s。在计算机和移动终端设备内存往往几个 G 的情况下，该算法寻路占用的内存非常的少，实际计算设备可以满足该需要。

表 1 单智能体在 Web 上寻路分析
Tab. 1 Path planning analysis of one agent on Web

图号	搜索步数	分析节点	排序次数	用时/s	内存占用/M
a	30	127	68	0.017	0.000 11
b	32	141	64	0.022	0.000 08
c	38	192	85	0.021	0.000 12
d	47	257	102	0.018	0.000 14

3.2 多智能体在网页上的路径规划

多智能体的路径规划是基于单智能体路径规划算法基础上，除了考虑智能体与静态障碍物间的碰撞检测外，还要考虑智能体间的动态碰撞避让的问题。

未知复杂的真实场景，使得多智能体的寻路任务远远重于单智能体，多智能体寻路要求多智能体系统中的每个智能体都要寻找到一条从开始位置到目标位置的路径，如何既能够和静态场景避碰，同时又能和动态智能体避碰，并寻找到最优路径，是要解决的一个难题。

多智能体路径规划由于计算复杂度高，收敛速度慢，以及智能体间的碰撞冲突问题，在使用 DP-Q(λ)基础上引入目标矢量方向，使移动智能体能够快速的收敛，增加了多移动智能体间防止冲突的强化信号。

多移动智能体寻路算法的主要任务是避障、避碰和到达目标。基于单智能体 DP-Q(λ)可以有效避障且寻到最优路径，多智能体需要解决路径寻优算法的复杂度和避碰问题。这就是多智能体与单智能体的最大区别之所在。如果多智能体中一个智能体寻找到了最优的路径，刚好其它智能的路径规划中与其路径有重合位置，那么可以避免可以借助之前智能体得到的 q 数据，免除该智能体进一步寻路的计算，减小群体寻路时间复杂度，这是本文提出的路径重用方案。

路径重用：如果后面的智能体和前面智能体的路径有重叠处，那么该智能体借助前面智能体寻得的路径，直接取值，更新其 Q 值表，从而可以直接从 Q 值表取得前面智能体寻到的路径，大大节省了寻路计算时间。

本文使用了 10 个智能体，100 个智能体和 200 个智能体在网页上基于 5 个指标数据的寻路情况，这些数据可从一般性上反映 DP-Q(λ)算法的性能。下面是针对 10 个智能体情况下，所做的 4 组实验(a 实验，b 实验，c 实验，d 实验)。

表 2 中，以序号为 a 的实验例，是 10 个智能体搜索的总步数为 385，分析节点总个数为 2 177，排序总次数为 964，用的总共时间是 0.021 s，所需总内存是 0.005 754。另外 3 个实验数据也如表 2 所示。表 3 是 100 个智能体寻路的数据情况。图 4 及表 4 是 200 个智能体的寻路图及数据情况。

https://dc-china-simulation.researchcommons.org/journal/vol31/iss1/3
DOI: 10.16182/j.issn1004731x.joss.16PQS-003

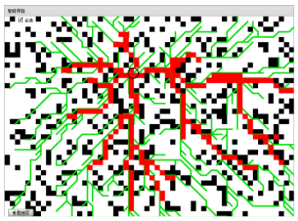
第 31 卷第 1 期 系统仿真学报 Vol. 31 No. 1
2019 年 1 月 Journal of System Simulation Jan., 2019

表 2 10 个智能体在 Web 上的寻路分析
Tab. 2 Path planning analysis of 10 agents on Web

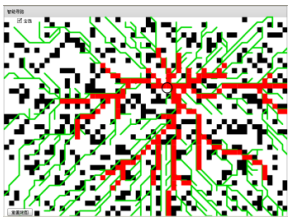
实验序号	搜索步数	分析节点	排序次数	用时/s	内存占用/M
a	385	2 177	964	0.021	0.005 754
b	643	3 765	1 323	0.030	0.005 600
c	1 066	5 840	1 979	0.071	0.008 721
d	470	2 674	1 043	0.032	0.006 325

表 3 100 个智能体在 Web 上的寻路分析
Tab. 3 Path planning analysis of 100 agents on Web

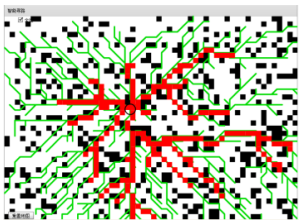
实验序号	搜索步数	分析节点	排序次数	用时/s	内存占用/M
a	15 500	81 123	22 973	0.523	0.641 474
b	18 083	97 401	28 542	0.704	0.704 786
c	4 377	24 093	9 374	0.2221	0.494 566
d	6 321	34 942	12 769	0.285	0.575 435



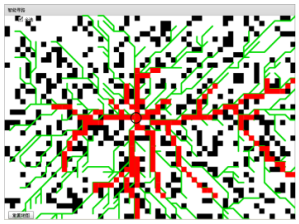
(a) scene 1



(b) scene2



(c) scene3



(d) scene 4

图 4 200 个智能体在网络平台上的寻路图
Fig. 4 Path planning of 200 agents on network platform

表 4 200 个智能体在 Web 上的寻路分析
Tab. 4 Path planning analysis of 200 agents on Web

实验序号	搜索步数	分析节点	排序次数	用时/s	内存占用/M
a	13 065	69 803	24 112	0.705	2.299 804
b	12 032	63 163	22 821	0.732	2.078 327
c	21 201	111 542	33 585	0.674	2.252 052
d	12 074	61 388	21 742	0.601	2.149 420

3.3 智能群在网页上的路径规划

基于 AS3.0 语言实现的智能群的路径规划，为了解决智能体行走过程中频繁出现的群体间的碰撞避让，本文使用了一维数组和多维数组进行了解决。在初始化的原始数组中，我们定义出所有智能体，由此方便了我们对所有智能体有区别的控制操作，随后，我们可以将智能体根据需要，编组成不同的智能群，图 5 是 2 群共计 200 个智能体的寻路图。

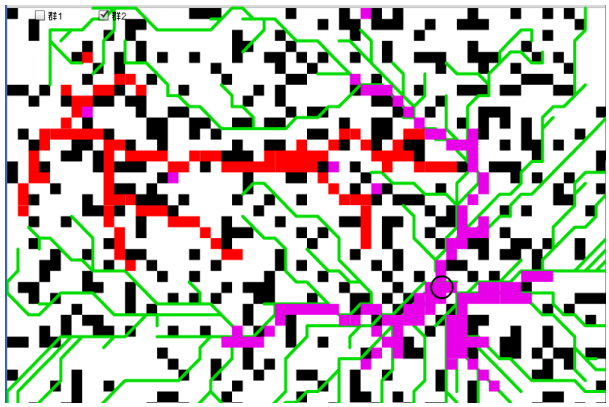


图 5 2 群(100 个智能体/群)寻路图
Fig. 5 Two group agents(100 agents)path planning

表 5 是 2 群共计 200 个智能体在 Web 上的寻路数据情况。

表 5 2 群 200 个智能 Web 上的寻路分析
Tab. 5 Path planning analysis of 2 groups (200 agents) on Web

实验序号	搜索步数	分析节点	排序次数	用时/s	内存占用/M
a	15 592	81 202	23 802	0.521	0.641 452
b	18 179	97 362	28 773	0.704	0.704 963

3.4 多智能体在 Web3D 上的路径规划

我们将该算法的路径规划应用到 Web3D 上，将 2D 空间升级为 3D 空间，以 200 个智能体路径规划为例。

在表 5 中，以序号为 a 的图为例，是其中一组智能体(100 个，两组共 200 个)路径搜索的总步数为 15 592，分析节点总个数为 81 202，排序总次数为 23 802，用的总共时间是 0.521 s，所需总内存是 0.641 452。同理可以知道另一群智能体的具体寻路情况。

图 6 是 Web3D 上 200 个智能体的路径规划。在这里可以由用户通过交互性的选择部分智能体，作为一个群进行移动，也可以将默认的整体作为一个群体，进行路径规划。

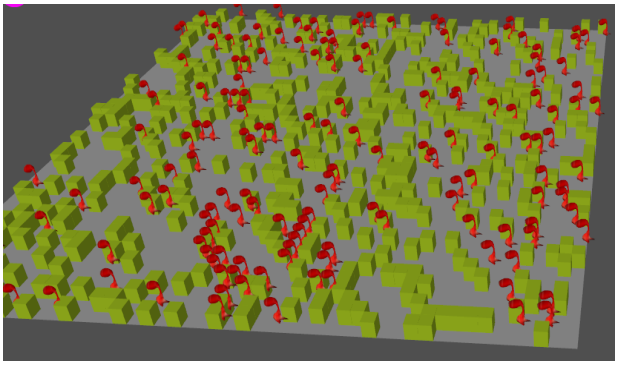


图 6 Web3D 上 200 个智能体行走情景
Fig. 6 200 agents walking scenario on Web3D

图 6 中路径规划的寻路步数、所耗时间以及内存占用率等数据与表 5 相同。

4 实验结果分析

本文提出的 DP-Q(λ)算法，就收敛速度看，远远快于 Q(λ)和 Q-learning 算法，在收敛值上也较 TD(λ)算法和 potential field-based Q-learning 算法更准确，这些信息可以从图 2 看到。

基于移动终端设备网页的单智能体、多智能体和智能群的路径规划，其实验数据在第 3 部分已经通过实验场景路径图及实验得出的 5 个指标的数据，给出了具体的阐述。

基于 DP-Q(λ)算法，本文使用强面向对象语言 ActionScript3.0，开发了面向移动终端设备网页上的多智能体强化学习的路径规划这一原型系统(如图 6 所示)。

5 结论

本文提出的 DP-Q(λ)在复杂环境下寻路的有效性、自动适应性和学习速度均高于 Q-learning 算法、Q(λ)算法，以及当前很多优秀的智能寻路算法。

实验中基于寻路规划中涉及到的 5 个核心指标，衡量了 DP-Q(λ)寻路算法的性能。从单个智能体寻路，到多个智能体，以及智能群的寻路所得到数据来看，该寻路算法收敛速度快，占用物理空间小。根据打开网页时，人们可以忍耐的时间为 2 s 来计算，本文提出的多智能寻路算法可以流畅的运行在 Web3D 和移动终端设备上，具备了实时性。

下一步的工作是，基于多智能体和智能群在目前未知复杂场景路径规划基础上，进一步实现多种复杂 Web3D 场景中，根据实际路径规划的需要，实现多种智能寻路算法的按需切换，形成一个 Web3D 路径规划决策系统，该系统同时基于社会心理学理论解决紧急突发事件中人群疏散的问题。

参考文献:

[1] Viet H H, An S H. Dyna-Q-based Vector Direction for Path Planning Problem of Autonomous Mobile Robots in Unknown Environments[J]. Advanced Robotics (S0169-1864), 2013, 27(3): 159-173.

[2] Bayili S, Polat F. Limited-damage A*: A Path Search Algorithm That Considers Damage As A Fesibility Criterion[J]. Konwledge-Based Systems (S0950-7051), 2011, 24: 501-502.

[3] MA Tsai, C C Huang, H C Chan, et al. Parallel Elite Genetic Algorithm and Its Application to Global Path Planning for Autonomous Robot Navigation[J]. IEEE Transactions on Industrial Electronics (S0278-0046), 2011, 58(10): 4813-4821.

[4] Xin Ma, Ya Xu, Guoqing Sun, et al. State-chain Sequential Feedback Reinforcement Learning for Path Planning of Autonomous Mobile Robots[J]. Journal of Zhejiang University-SCIENCE (Computer & Electronics) (S1869-1951), 2013, 14(3): 167-178.

[5] Yong Song, Yi-bin Li, Cai-hong Li, et al. An Efficient Initialization Approach of Q-learning for Mobile Robots[J]. International Journal of Control Automation and System (S1598-6446), 2012, 10(1): 166-172.

[6] Fatemeh Fatyhinezhad, Vali Derhami, Mehdi Rezaeian. Supervised Fuzzy Reinforcement Learning for Robot Navigation[J]. Applied Soft Computing (S1568-4946), 2016, 40(C): 33-41.

[7] Francisco Martinez-Gil, Miguel Lozano, Fernando Fernandez. Strategies for Simulating Pedestrian Navigation with Multiple Reinforcement Learning Agents[J]. Autonomous Agents and Multi-Agent

https://dc-china-simulation.researchcommons.org/journal/vol31/iss1/3
DOI: 10.16182/j.issn1004731x.joss.16PQS-003

Systems (S1387-2532), 2015, 29(1): 98-130.

[8] Fard M, Pineau J. Non-deterministic Policies in Markovian Decision Process[J]. Journal of Articificial Intelligence Research (S1076-9757), 2011, 11(2): 1-24.

[9] Desouky S F, Schwartz H M. Q(λ) - learning Adaptive Fuzzy Logic Controllers for Pursuit–evasion Differential Games[J]. International Journal of Adaptive Control & Signal Processing (S0890-6327), 2011, 25(10): 910-927.

[10] Lglesias A, Martinez P, Aler R, et al. Reinforcement Learning of Pedagogical Policies in Adaptive and Intelligent Educational Systems[J]. Knowledge-Based Systems (S0950-7051), 2009, 22(4): 266-270.

[11] Al-Taharwa I, Sheta A, Al-Weshah M. A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment[J]. Journal of Computational Science (S1877-7503), 2008, 4(4): 341-344.

[12] Remolina E, Kuipers B. Towards A General Theory of Topological Maps[J]. Artificial Intelligence (S0004-3702), 2004, 15(2): 47-104.

[13] Barraquand J, Langlois B, Latombe J C. Numerical Potential Field Techniques for Robot Path Planning[J]. IEEE Transactions on Systems Man and Cybernetics Part A-systems and Humans (S1083-442), 1992, 22(2): 224-241.

[14] Dolgov D, Thrun S, Montemerlo M, et al. Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments[J]. International Journal of Robotics Research (S0278-3649), 2010, 29(5): 485-501.

[15] Jaradat M A K, Al-Rousan M, Quadan L. Reinforcement Based Mobile Robot Navigation in Dynamic Environment [J]. Robotics & Computer Integrated Manufacturing (S0736-5845), 2011, 27(1):135-149.

[16] Kala R, Shukla A, Tiwari R. Fusion of Probabilistic A* Algorithm and Fuzzy Inference System for Robotic Path Planning[J]. Artificial Intelligence Review (S0004-3702), 2010, 33(4): 307-327.

[17] Hwang H J, Viet H H, Chung T. Q(λ) Based Vector Direction for Path Planning Problem of Autonomous Mobile Robots[J]. Lecture Notes in Electrical Engineering (S1876-1100), 2011, 10(4): 433-442.

[18] Still S. An Information-theoretic Approach to Curiosity-driven Reinforcement Learning[J]. Theory in Biosciences (S1431-7613), 2012, 131(3): 139-148.

[19] Guo M, Liu Y, Malec J. A New Q-learning Algorithm Based on The Metro Poliscriterion[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics (S2168-2216), 2004, 34(5): 2140-2143.

[20] Alvarez C, Santons M, Lopez V. Reinforcement Learning vs. A* in A Role Playing Game Benchmark Scenario [J]. Computational Intelligence, Foundations and Applications (S14100-8640), 2010, 13(6): 644-650.

[21] Bkassiny M, Li Y, Jayaweera S K. A Survey on Machine-Learning Techniques in Cognitive Radios[J]. IEEE Communications Surveys & Tutorials (S1553-877X), 2013, 15(3): 1136-1159.

[22] Martin J A, De Lope J, Maravall D. Analysis and Solution of A Predator-protector-prey Multi-robot System by A High-level Reinforcement Learning Architecture and Adaptive Systems Theory[J]. Neuron Computing (S0941-0643), 2010, 58(12): 1266-1272.

[23] Suo Tan, Simon X Yang. A Fuzzy Inference Controller with Accelerate/Brake Module for Mobile Robots[C]. Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September, Fuzzy Information and Engineering, 2008: 810-815.

[24] Yang H Y, Zhang F Z . Autonomous Mobile Intelligent Robots on Fuzzy System with Optimal Theories[J]. Fuzzy Information and Engineering (S1616-8658), 2009. 12(4): 24-32.

