基于DEVS原子模型的智能体离散仿真构建方法

王霄汉[1,2]，张霖[1,2*]，赖李媛君[1,2]，谢堃钰[1,2]，胡听春[1]
(1. 北京航空航天大学，北京 100191；2. 复杂产品先进制造系统教育部工程研究中心，北京 100191)

摘要：智能体由于自身交互行为与学习行为的复杂性，难以直接被建模和仿真。*针对智能体离散仿真中的常见问题，借助DEVS（discrete event system specification）原子模型的事件转移机制表示智能体的交互与学习过程，通过对智能体交互模式、多状态外部事件转移控制、端口连接模式、以及强化学习事件转移表示等原子模型下智能体建模技术的介绍，给出了基于DEVS原子模型的智能体离散仿真构建方法。*在网格世界与倒立摆2个环境中进行仿真验证，实验结果证明了提出方法在构建智能体交互行为和学习行为的可行性和有效性。

关键词：智能体；DEVS；离散仿真；强化学习；状态转移；原子模型
中图分类号：TP391.9 文献标志码：A 文章编号：1004-731X(2022)02-0191-10
DOI: 10.16182/j.issn1004731x.joss.21-0263

Constructing the Agent Discrete Simulation Based on DEVS Atomic Model

Wang Xiaohan[1,2], *Zhang Lin*[1,2*], *Laili Yuanjun*[1,2], *Xie Kunyu*[1,2], *Hu Tingchun*[1]
(1. Beihang University, Beijing 100191, China; 2. Engineering Research Center of Complex Product Advanced Manufacturing Systems, Ministry of Education, Beijing 100191, China)

Abstract: Agents are difficult to be directly modeled and simulated due to the complexity of their own interaction and learning behaviors. *Aiming at the common problems in the discrete simulation of the agent, the event transfer mechanism of the discrete event system specification (DEVS) atomic model is applied to express the interaction and learning of an agent. Through the interaction mode of the agent, the transfer control of multi-state external events, the port connection mode, as well as the introduction of reinforcement learning event transfer representation, a discrete simulation construction method of the agent based on the DEVS atomic model is provided.* The simulation verification is carried out in the grid world and the cart-pole environment. The experimental results prove the feasibility and effectiveness of the proposed method in constructing the interactive and learning behaviors of the agent.

Keywords: agent; discrete event system specification(DEVS); discrete simulation; reinforcement learning; state transition; atomic model

引言

智能体离散仿真将复杂系统问题分解为基于智能体的子问题，并通过集中式或分布式的策略控制单个智能体，利用交互、感知、决策、以及涌现等特性实现对复杂系统问题的解决，并在过去几十年中在军事、调度、社会市场等各领域起到重要作用[1]。构建智能体离散仿真模型的关键在于对行为的定义，而交互和学习2种行为，是行为定义的重点和难点。DEVS(discrete event system specification)基于层次化的模块方式来构建

收稿日期：2021-03-29 修回日期：2021-04-01
基金项目：国家重点研发计划(2018YFB1701600)
第一作者：王霄汉(1998-)，男，博士生，研究方向为离散仿真、多智能体系统和强化学习。E-mail：by1903042@buaa.edu.cn
通讯作者：张霖(1966-)，男，博士，教授，研究方向为复杂系统建模仿真、智能制造、云制造、模型工程等。E-mail：johnlin9999@163.com

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

离散事件仿真模型，通过时间步长的推进和事件状态的转移过程完成对于离散模型的仿真，被广泛应用于各类仿真模型的搭建[2]。基于DEVS搭建智能体离散仿真模型的优势有2点。首先，智能体离散仿真模型属于离散模型，其与环境或其他智能体的交互过程可以看作DEVS的事件转移过程。其次，DEVS可以很好地对马尔可夫决策模型进行建模分析[3]，使得智能体的学习过程也能够以DEVS的事件转移过程进行描述。因此，DEVS十分适用于智能体离散仿真模型的搭建。

目前基于DEVS搭建智能体离散仿真模型的相关工作主要分为2种，代表了一个智能体的2种表示方式，分别是由单个原子模型表示，以及由多个原子模型组成的耦合模型表示。在利用原子模型表示智能体方面，文献[4]综合考虑了各类适用于智能体建模的方法，使用对单个原子模型进行描述扩充的方式构建智能体描述规范，以使得搭建的DEVS模型具有更好的建模灵活性。Müller[5]从多智能体的角度出发，基于DEVS搭建了一套系统模型，并且为了适应多智能体的特点，修改了原始DEVS原子模型的仿真过程。Barbieri[6]利用DEVS与强化学习实现财务杠杆效应的建模与仿真过程，将Q-learning算法集成到了一个原子模型之中，以此来模拟金融决策过程。在文献[7]中，针对机器学习与DEVS的融合问题给出了解释，认为将目前机器学习的算法与离散仿真中的动态环境结合起来可以发挥更大的作用，并给出智能体与环境的显式表示、DEVS时间在强化学习中的处理、以及多智能体环境下的DEVS模型表示方法。这些文献从不同的角度分析了基于DEVS原子模型搭建智能体模型的特点，但是并没有提供原子模型搭建一般智能体的普遍化方法，而只是利用了DEVS的离散事件调度能力，解决智能体仿真上的应用问题。在利用耦合模型表示智能体方面，Kessler[8]使用多个原子模型组成的耦合模型对层次马尔可夫模型进行了描述，给出了整个层次马尔可夫模型的DEVS架构，并使用了一个基于强化学习算法的网格世界作为仿真案例。文献[9]基于多种类型的原子模型构建了一种复杂的智能体感知架构，使得智能体在与环境的交互过程中能够充分利用感知过程以获取相关环境信息。Akplogan[10]利用DEVS耦合模型搭建智能体模型，使用原子模型表示智能体的信念，用包含几种原子模型的耦合模型来表示智能体的执行计划，并使用农业应用中的案例证明了整体架构的可行性。通过上述文献可以发现，耦合模型具备表示复杂智能体的能力，通过智能体内部各个功能化组件的详细建模，实现单智能体的复杂功能。然而这种建模方式相对单个原子模型的表示方法较为冗余。

目前并没有研究针对智能体的离散仿真给出更具通用性的建模方法，本文给出基于DEVS原子模型的智能体离散仿真构建方法，主要解决以下2个挑战：

(1) 智能体之间的交互过程可能十分复杂，难以通过DEVS直接定义。虽然DEVS为不同原子模型间的交互提供了外部事件转移和输出函数，但是其模式较为固化，要想实现不同智能体间的灵活交互，还需要对事件转移过程进一步控制。

(2) 智能体的学习模式和算法多样，导致一些复杂过程难以在DEVS规范下实现。强化学习为智能体的适应性学习提供了有效的算法，但是强化学习本身种类各异，使得智能体与环境之间的交互模式不固定，加大了DEVS的建模难度。

针对这2个主要挑战和一些构建智能体模型的一般性问题，本文总结了基于DEVS原子模型搭建智能体的方法，对基于DEVS的事件转移实现智能体交互和学习行为的控制方法进行了介绍，并基于经典的智能体网格世界问题、倒立摆控制问题对单智能体仿真、多智能体并行仿真、智能体强化学习等代表性过程进行了建模分析。本文旨在提供一般性的DEVS原子模型表示智能体的方法，也为智能体离散仿真提供基于DEVS的建模依据。

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

1 背景

1.1 DEVS规范

作为基于离散事件仿真的框架，DEVS能够将复杂的离散事件模型分解为不同原子模型来表示。作为DEVS耦合模型的组成最小单元，一个原子模型可以由如下七元组表示[11]：

$$AtomicDEVS=<S,t_a,\delta_{\text{int}},X,\delta_{\text{ext}},Y,\lambda>$$

式中：X为外部输入事件集合；S为系统的状态集合；Y为输出事件集合；模型的时间T连续且$T\in R$；δ_{int}为内部事件转移函数；δ_{ext}为外部事件转移函数；t_a为事件推进函数；$t_a(s)$表示在没有外部事件到达时系统保持状态s的时间；λ为输入函数。

系统的总状态可以用集合

$$Q=\{(s,\ e)\,|s\in S,0\leqslant e\leqslant t_a(a)\}$$

式中：e为系统在状态s停留的时间；(s,e)为系统当前的状态。

如无外部事件到达，系统在经过$t_a(s)$时间后，状态s将会转移到δ_{int}，同时将e重置为0。若有外部事件$x\in X$到达系统，设系统在状态s的停留时间为e，则其立即转移到新状态，并将持续时间e重置为0。系统的输出将会伴随着内部转移发生，且状态转移前的状态s用于产生输出。

定义 闲置状态：在不触发外部事件的情况下，原子模型一直保持当前状态直到无穷时刻，即不发生任何内部或外部事件转移的状态，即$t_a(\infty)$。

1.2 强化学习

强化学习是智能体完成学习过程的重要方法，并随着人工智能的发展被应用于自动驾驶、股票机器人、游戏对抗等领域。在强化学习下，智能体通过与环境的交互过程学习到最优策略，以最大化智能体的回报值或者完成某个固定的目标[12]。在强化学习中，智能体以马尔可夫决策过程(MDP)进行描述，可以被描述为一个元组：$\langle S, A, \pi, P, R\rangle$，其中$S$代表动作空间，$A$代表动作空间，$\pi: S\to A$代表智能体的策略，$P: S\times A\to P$为状态转移概率，$R: S\times A\times S\to R$为奖励函数[13]。强化学习一般通过优化两个值函数实现通过与环境的交互进行适应性学习的过程，分别是值函数$V(s)\leftarrow E\{G_t|S_t=s\}$和动作状态值函数$Q(s,a)\leftarrow E\{G_t|S_t=s,A_t=a\}$，其中$G_t=\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}$为用于估计未来累计奖励值的回报函数。一般将通过优化值函数实现学习过程的方法称为基于值的强化学习算法，如Q-learning，SARSA；而将直接优化策略的过程称为基于策略的强化学习算法，如REINFORCE算法[12,18]。

Q-learning算法应用了时序差分法进行动作状态值函数的估计。时序差分法是一种单步更新的值函数迭代算法，用于控制值函数收敛到最优解，表示为[14]

$$V(s)\leftarrow V(s)+\alpha\left(R_{t+1}+\gamma V(s')-V(s)\right)$$

式中：α为学习率；γ为衰减因子；R_{t+1}为状态从$s\to s'$的奖励值。

Q-learning基于时序差分法更新智能体的动作状态值函数，并在每一步采用选取最大值函数的方法估计智能体的最优策略，在每一次智能体与环境进行交互之后，采用的更新过程为

$$Q(s,\ a)\leftarrow Q(s,\ a)+\alpha\{r+\gamma\cdot max_{a'}Q(s',\ a')-Q(s,a)\}$$

式中：a'为下一步要采取的动作。

Q-learning代表了强化学习过程中智能体与环境进行交互并学习到适应性策略的过程，虽然不同强化学习算法之间差异性较大，但是智能体与环境的事件转移过程基本一致。

2 基于原子模型的智能体建模

基于原子模型构建智能体模型，主要是基于原子模型的范式描述智能体的交互和学习2个行为。其中交互是智能体与环境以及其他智能体进行信息交换的基础，学习是智能体对于环境适应性的过程。2种行为是智能体“智能”属性的重要体现。

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2.1 智能体交互行为的建模方法

智能体间的通讯过程是智能体离散仿真中的重要环节，智能体以通讯的方式完成与其他智能体之间消息的传递。针对智能体的交互过程，FIPA(foundation for intelligent physical agents)组织给出了智能体通讯过程的标准规范[15]。交互协议可以看作按照固定规则的智能体发送和接受动作的组合，均可以拆解为发送和接受消息的混合动作。原子模型需要利用状态转移过程，灵活定义不同智能体间信息的交互。利用DEVS的事件推进机制定义智能体的同步、异步交互模式，使用多状态转移过程控制不同协议下事件的推进，并基于DEVS端口的连接实现不同智能体间的连接。

2.1.1 同步和异步交互模式

同步交互模式指的是智能体在发送消息后处于闲置状态，直到接收到新的消息才会触发新的状态转移过程。而异步交互模式下智能体在消息发送后会继续完成其他事件转移，而不会以消息的到来作为事件触发的条件。

同步和异步交互过程中，消息的发送和接受通过原子模型的输出函数和外部事件转移过程实现，两种交互模式的发送过程并没有差异，区别在于执行输出函数之后到下一次外部事件到达的过程。为实现上述控制能力，定义原子模型的t_a函数为

$$t_a=\begin{cases} n, & \text{if 同步交互} \\ infinity, & \text{else 异步交互} \end{cases}$$

式中：n为大于0的数。同步交互一般用于智能体与环境进行交互的过程，代表着感知外部环境的过程；而异步交互过程则普遍存在于智能体之间的信息传递。

2.1.2 多状态外部事件转移控制

在智能体交互的过程中，可能会在n次外部事件转移后再进行一次决策，以完成整个交互过程。例如，机器人路径仿真中控制器需要连续与机器人交互2次，获得2次的位置，之后才能依据2次位置的差计算机器人运行的方位[16]，即完成多次状态s的外部事件转移后才进行一次状态s'的外部事件转移，称之为多状态外部事件转移。多状态外部事件转移函数的定义如下所示，以2个状态为例，通过收集各端口外部事件转移过程的信息，并利用变量控制状态的转换，实现2种状态下智能体不同的交互需求。

算法1 多状态外部事件转移控制算法：

初始化：$portlist\neq\Phi$，$index>0$，$savelist=\Phi$，$scheduler\neq\Phi$

function external_transition:

for each $port_i\in portlist$ **do**

收集产生外部事件的$port_i$端口的信息

$index\neq 0$

$index\leftarrow index-1$

*savelist*增加当前收集的信息(状态s)else处理*savelist*(状态s')

2.1.3 两种端口连接方法

实现智能体原子模型之间端口的连接主要有2种形式，如图1所示。由于DEVS的端口连接较为灵活，使得整个耦合模型中的任意两个原子模型之间可以建立专门的输入输出联系，如图1(a)所示的方式1。此外，还可以只保留每个原子模型的单个输入和输出端口，并将所有原子模型的端口连接起来，如图1(b)所示的方式2。

设在耦合模型中存在n个智能体原子模型，并且两两原子模型之间都需要建立专门的端口连接。则在方式1下每个原子模型内部需要定义的端口数量为$2(n-1)$，在整个耦合模型中需要连接的端口数为

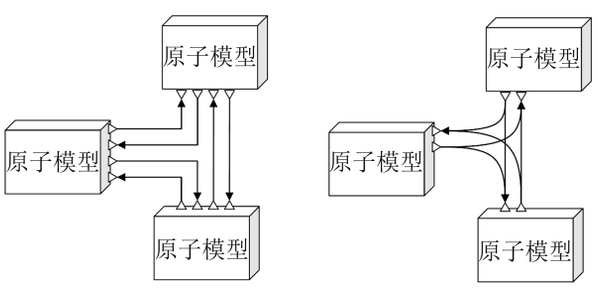
$$2C_n^2=n(n-1)$$

方式2下每个智能体内部的端口定义数为2，而耦合模型中需要连接的端口数也为$n(n-1)$。因此，方式2减少了原子模型内部的端口定义数，并且固定的端口数十分有利于基于面向对象的示例化方法来构建同一基类下属性不同的原子模型，

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

使得原子模型内部的端口定义不会随着实例化智能体数量的变化而变化。

然而，方式1具备更强的普适性，方式2则仅适用于串行的交互过程，在处理并行事件时可能会引起事件的紊乱。设第$i \in \{1,2,3,\ldots,n\}$原子模型在t时刻向第$j \in \{x|x \in \{1,2,3,\ldots,n\}, x \neq i\}$模型输出事件，而第$k \in \{x|x \in \{1,2,3,\ldots,n\}, x \neq i,j\}$个原子模型原本在第$t+m$时刻发生内部事件转移，则会由于$t$时刻收到了原子模型$i$的外部事件转移后，虽然可以使用条件语句控制模型$k$不发生内部事件转移，但是模型$k$的内部事件会被调整为$2t+m$，从而引起事件紊乱。因此，方式2的连接方式虽然节省了端口连接工作，但是由于任意输出函数都会驱动所有原子模型的外部事件转移，对于一些智能体交互的场景并不适应。在基于DEVS原子模型构建智能体模型中，需要根据任务的需求灵活选择2种端口连接方式。



(a) 方式1　　(b) 方式2

图1　2种DEVS的端口连接方式

Fig. 1　Two DEVS port connection methods

2.2 智能体强化学习建模方法

在DEVS规范下构建强化学习型智能体，主要是利用原子模型之间的外部事件转移传递智能体与环境之间交换的信息，从而实现智能体对环境的适应性学习过程。

2.2.1 智能体与环境的耦合模型

如图2所示，马尔可夫决策过程是智能体与环境的同步交互过程，可以看作是2个原子模型之间的事件转移[8]。智能体向环境发送动作和当前状态，然后环境反馈给智能体新的状态和其他感知信息，以此循环进行两者之间的同步交互过程。强化学习中智能体与环境交互的过程本身十分适合基于DEVS进行建模，是典型的离散事件转移过程。

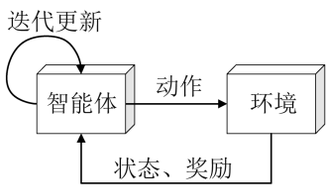


图2　智能体与环境的强化学习耦合模型

Fig. 2　Couple model of reinforcement learning between agent and environment

2.2.2 强化学习的事件转移表示

在智能体收到环境的反馈之后，需要根据反馈内容使用强化学习算法进行更新。基于DEVS原子模型实现强化学习的更新过程，主要有4个要点。

(1) 明确学习更新发生的时机。大多数更新发生在2次智能体与环境的交互之间，对应于原子模型执行完外部事件转移，执行输出函数之前，而并非对应于内部事件转移函数执行的期间。然而，不同强化学习算法的更新时机并不一致，需要在明确DEVS中事件转移顺序的基础上，按照算法的更新顺序进行定义。

(2) 确定更新的相对频率。相对频率指的是更新对交互过程的频率，虽然一些传统的算法其更新频率与交互频率一致，即一次交互一次更新，但是并不适用于所有算法的更新过程，因此需要利用算法1控制更新的相对频率。

(3) 确定智能体的训练状态。强化学习中，智能体在训练时并不一定处于单一的与环境交互的状态，而可能处于多种状态的组合。例如，部分算法需要提前收集与环境交互的信息，存储到缓存数据结构中，然后才开始训练过程。这些状态都需要提前在外部事件函数的定义中考虑。

(4) 定义智能体的初始化。智能体在与环境交互中存在初始化过程，在强化学习算法中可能只对应一个语句，但是智能体离散仿真需要重点考

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

虑初始化过程，通过控制输出函数和外部事件转移过程实现智能体与环境的初始化。

3 实验

通过实验展示使用DEVS原子模型构建一般智能体模型的过程和结果，并对结果从DEVS事件转移的角度进行分析。分别基于自行搭建的网格世界二维离散环境，以及OpenAI组织开发的倒立摆控制环境[17]，使用DEVS对2种环境中的智能体进行建模和仿真，验证文中给出的智能体交互和学习行为的建模方法。实验使用Van等[18]搭建的Python pypdevs库进行实现原子模型和耦合模型的搭建。

3.1 基于网格世界的智能体仿真

网格世界是一个经典的智能体离散仿真环境，具有空间离散，时间离散的特征。如图3所示，智能体以5×5大小的网格世界的左下角作为初始位置，网格中除了常规位置外，还存在2类特殊位置，分别是2个智能体会优先考虑的位置，以及一个最后考虑的位置。一个时间步长智能体跨越一个位置，并且无法斜向运动。此外，智能体的感知范围为当前位置的前后左右4个位置，见图4。在整个仿真中，智能体是随机行走的，但是当其感知范围内出现特殊位置时，会优先考虑或拒绝特殊位置。此外，智能体会尽可能前往未走过的位置。

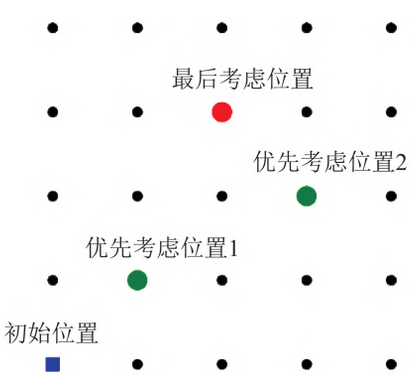


图3 网格世界环境
Fig. 3 Grid world environment

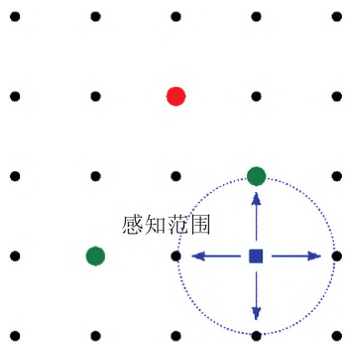


图4 网格世界中智能体的感知
Fig. 4 Perception of an agent in grid world

3.1.1 单智能体与环境的交互与学习过程

分别利用2个原子模型代表智能体与环境，并连接两者的端口进行仿真。智能体向环境提供当前的位置，然后由环境反馈给智能体下一步可运行的位置，并标记出特殊位置，以此推进智能体在网格世界中的仿真过程。设置智能体的t_a=0.8，代表决策时间，而设置环境t_a=0，代表环境的即时响应。在5 s的仿真运行后，智能体的行走路线如图5所示，在遇到了一次特殊位置后，智能体开始随机行走。图6为在仿真过程中发生的原子模型事件转移情况，其中向下的绿色箭头代表此刻发生了内部事件转移，而向上的蓝色箭头代表此刻发生了外部事件转移。图6表示智能体在0.8 s向环境发送信息，并立刻驱动环境在0.8 s的外部事件转移，然后环境将感知结果反馈给智能体。整个仿真时间内发生了6组事件转移过程，使得智能体在环境中初始化并前进了5步。

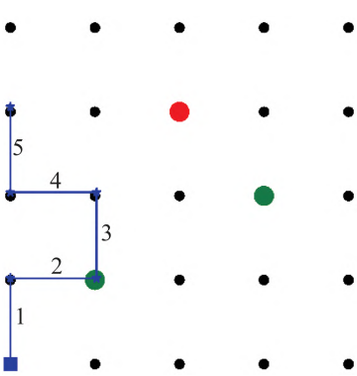


图5 单智能体在网格世界中的仿真路径
Fig. 5 Simulation path of a single agent in grid world

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

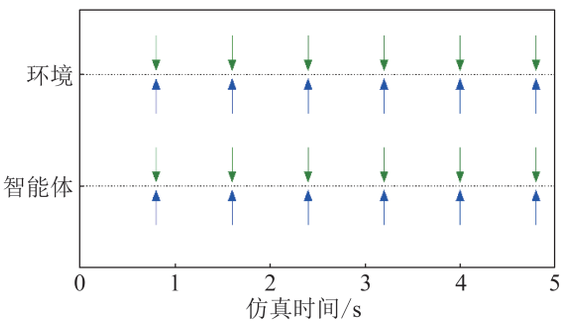


图6　智能体与环境交互的事件转移时刻图
Fig. 6　Event transition time diagram of interaction between agent and environment

基于上述搭建的耦合模型，利用强化学习算法替代智能体原子模型中的规则集，使得智能体自主适应环境。按照图3所示环境，设置奖励矩阵为

$$\boldsymbol{Reward}_{\text{gridword}}=\begin{bmatrix}-1 & -1 & -1 & -1 & -1\\ -1 & -1 & -5 & -1 & -1\\ -1 & -1 & -1 & 50 & -1\\ -1 & 2 & -1 & -1 & -1\\ -1 & -1 & -1 & -1 & -1\end{bmatrix}$$

为了鼓励智能体前往更远的优先考虑位置，将优先考虑位置2的奖励值设置极高，以获得更好的训练效果。使用强化学习算法Q-learning作为智能体的学习算法，取代原始智能体的规则集定义，各项参数的定义如表1所示。

表1　智能体Q-learning参数设置
Table 1　Q-learning parameter settings

超参数名	值
学习率α	0.7
衰减因子γ	0.5
探索率ε	0.5
算法迭代次数	200

将仿真时间从5 s扩充为1 000 s，并控制智能体最多前进5步后返回初始位置。当$t=100$ s时，智能体学习到反复经过优先考虑位置1可以获取一个不错的收益，如图7所示。当$t=1\,000$ s时，智能体学习到前往优先考虑位置2可以获得更高的收益，如图8所示。在进行了5次重复仿真实验后，智能体的累计收益随迭代次数的变化如图9所示。其中浅绿色阴影代表5次实验的波动范围，而实线表示了平均值。5次实验基本上都在110次迭代后收敛到了图8所示结果，表明基于DEVS构建的智能体强化学习依然具有很好的收敛性。

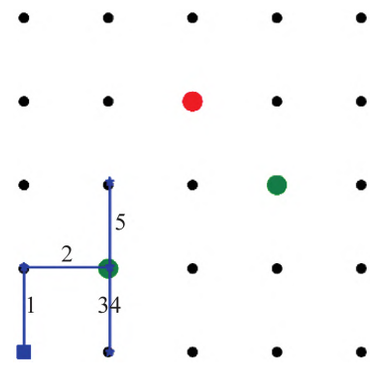


图7　仿真时间推进到100 s时智能体训练的路径
Fig. 7　Path of agent when simulation time advances to 100 s

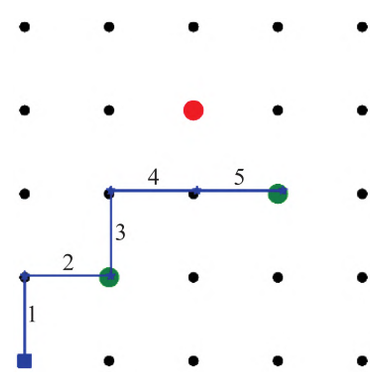


图8　仿真时间推进到1 000 s时智能体训练的路径
Fig. 8　Path of agent when simulation time advances to 1 000 s

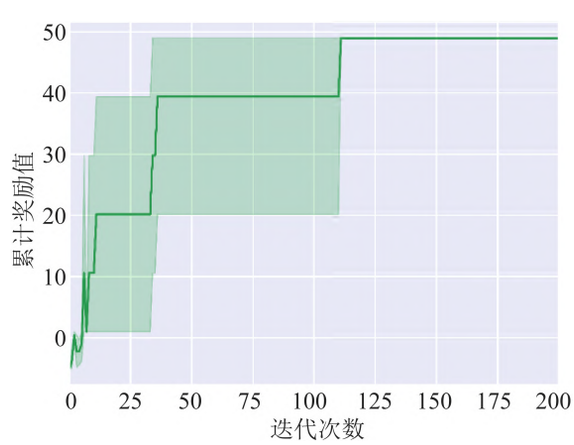


图9　智能体的Q-learning算法训练曲线
Fig. 9　Q-learning algorithm training curve

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

3.1.2 多智能体与环境的并行仿真

在基于DEVS搭建了单智能体与环境的耦合模型后，对多智能体与环境的交互模型进行构建。相比于单智能体，多智能体在与环境进行交互时需要考虑交互顺序的问题，本文以两智能体与环境的交互过程为例进行研究。

在网格世界中，环境作为多智能体共同交互的对象，具有及时反馈智能体感知信息的作用。对于多智能体，其与环境的交互顺序应以智能体的感知顺序为依据。基于原子模型搭建智能体1与智能体2，并与环境原子模型进行连接，初始化智能体1和2的位置分别为(0,0)，(4,4)。在基于图1(a)方式一的端口连接下，设置两个智能体的时间步长均为1，即$t_a=1$，总仿真时间为5 s。获得的仿真结果如图10所示，2个智能体在环境中运行了4步，并各自穿过了一个优先考虑位置。图11表示2个智能体以相同频率与环境进行交互过程，并同时获得环境的反馈。接着，调整智能体2的时间步长为0.7，其他设置保持不变，使得2个智能体异频开展仿真，结果如图12所示，智能体2在5 s的仿真时间中运行了6步，并到达了2个优先考虑位置。图13表示2个智能体各自与环境交互的事件转移过程，2个智能体在各自的时间步长下完成与环境的交互，实现多智能体在环境中的异频并行仿真。

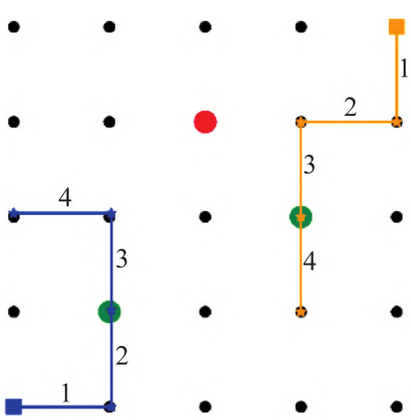


图10 多智能体同频并行仿真路径
Fig. 10 Multi-agent parallel simulation path at same frequency

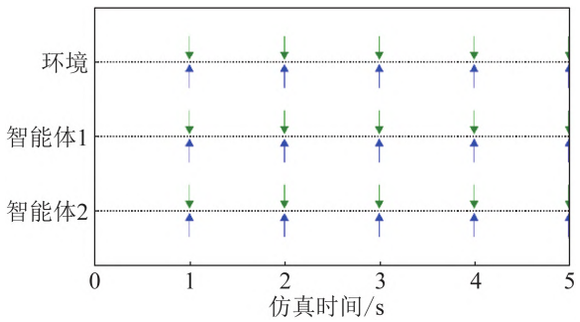


图11 多智能体同频并行仿真事件转移时刻图
Fig. 11 Multi-agent parallel simulation event transition timing diagram at same frequency

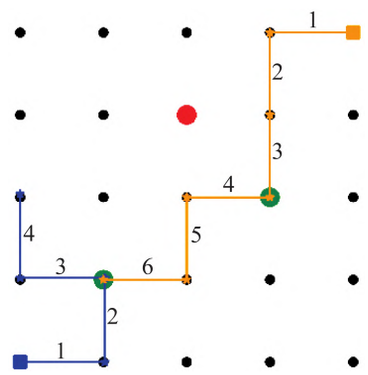


图12 多智能体异频并行仿真路径
Fig. 12 Multi-agent parallel simulation path at different frequency

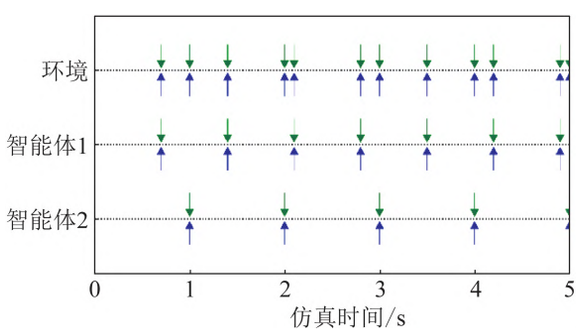


图13 多智能体异频并行仿真事件转移时刻图
Fig. 13 Multi-agent parallel simulation event transition timing diagram at different frequency

3.2 基于gym库的智能体仿真

在基于DEVS构建智能体仿真的过程中，往往需要与第三方环境库进行联合仿真，以获得更好的仿真效果。在联合外界环境进行仿真时，依然需要构建智能体与环境的耦合关系，并将第三

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

方库的接口封装在环境原子模型的内部。使用gym库中的倒立摆环境作为联合仿真环境，并封装在环境原子模型中。图14所示为倒立摆环境，需要控制底部黑色小车左右移动，以保持顶部长杆处于直立状态，每保持直立1 s即获得1奖励值[17]。

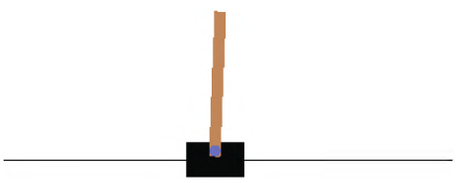


图14　gym倒立摆环境
Fig. 14　Cart-pole environment

在构建智能体与环境的耦合模型时，采用与3.1.1节中单智能体与环境的耦合模型相同的设置。智能体采用REINFORCE强化学习算法进行训练，目标是保持长杆直立持续100 s。不同于Q-learning，REINFORCE算法采用蒙特卡罗方法进行迭代，即智能体需要与环境交互100次之后，才能完成一次迭代更新[19]。REINFORCE算法使得智能体原子模型的外部事件转移发生后存在多种状态，因此采用算法1进行搭建。此外，还需要考虑每次迭代与环境的初始化过程。整个模型依然是智能体原子模型与环境原子模型的耦合模型，其中智能体采用的REINFORCE算法超参数如表2所示。

表2　智能体REINFORCE参数设置
Table 2　REINFORCE parameter settings

超参数名	值
学习率α	0.000 1
衰减因子γ	0.99
策略网络模型尺寸	4×64×128×1
算法迭代次数	700

每一轮的迭代更新智能体与环境都会进行最多100次交互，然后完成一次算法迭代更新。完整交互100次意味长杆保持直立100 s，否则这一轮的交互就会提前停止。经过700次的迭代后，智能体在REINFORCE算法下的训练曲线如图15所示，累计奖励在500轮迭代后逐渐接近最优值100。图9与图15证明了强化学习在DEVS原子模型下依然具有较好的收敛性，将学习过程融入到了智能体仿真之中。

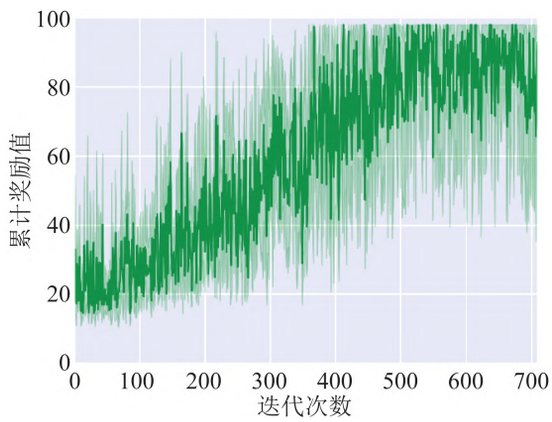


图15　智能体的REINFORCE算法训练曲线
Fig. 15　Agent's REINFORCE algorithm training curve

4　结论

本文给出了基于DEVS原子模型的智能体离散仿真构建方法，分析了基于原子模型表示智能体的优势与相关工作，指出重点在于通过原子模型的状态转移描述智能体的交互行为与学习行为。通过分析智能体交互与学习的特征，总结出了利用DEVS构建这两种行为的控制方法和表示模式。使用自行搭建的网格世界环境，验证了基于DEVS的智能体交互、学习、并行的仿真过程；使用第三方库gym，对联合仿真的构建方法进行了演示。实验结果表明DEVS可以很好地构建智能体离散仿真过程，并且不会影响算法本身的表现。

本文所述方法为智能体离散仿真提供了解决方案，从最具普适性的角度分析了基于DEVS原子模型构建智能体模型的要素，促进了DEVS与智能体仿真、强化学习的融合。发现单个原子模型可以满足多数智能体的建模需求，并且与环境原子模型组成的耦合模型可以实现绝大多数强化学习算法的建模和仿真。但是单原子模型难以实

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

现灵活的交互过程，在智能体内部行为复杂时就会难以描述，并且单原子模型构建的智能体解释性较差。但是单原子模型表示智能体的普适性和简洁性依然无法忽略，在面临大多数智能体仿真问题时都具有较强的建模能力。

参考文献

[1] Xie J, Liu C C. Multi-agent Systems and their Applications [J]. Journal of International Council on Electrical Engineering(S2234-8972), 2017, 7(1): 188-197.

[2] Van Tendeloo Y, Vangheluwe H. Extending the DEVS Formalism with Initialization Information[J]. arXiv preprint arXiv:1802.04527, 2018.

[3] Seo C, Zeigler B P, Kim D. DEVS Markov Modeling and Simulation: Formal Definition and Implementation[C]// Proceedings of the 4th ACM International Conference of Computing for Engineering and Sciences, 2018: 1-12.

[4] Bae J W, Lee G H, Moon I C. Formal Specification Supporting Incremental and Flexible Agent-based Modeling[C]//Proceedings of the 2012 Winter Simulation Conference (WSC). IEEE, 2012: 1-12.

[5] Müller J P. Towards a Formal Semantics of Event-based Multi-agent Simulations[C]//International Workshop on Multi-Agent Systems and Agent-Based Simulation. Springer, Berlin, Heidelberg, 2008: 110-126.

[6] Barbieri E, Capocchi L, Santucci J F. DEVS Modeling and Simulation of Financial Leverage Effect Based on Markov Decision Process[C]//2018 4th International Conference on Universal Village (UV). IEEE, 2018: 1-5.

[7] Capocchi L, Santucci J F, Zeigler B P. Discrete Event Modeling and Simulation Aspects to Improve Machine Learning Systems[C]//2018 4th International Conference on Universal Village (UV). IEEE, 2018: 1-6.

[8] Kessler C, Capocchi L, Santucci J F, et al. Hierarchical Markov Decision Process based on DEVS Formalism [C]//2017 Winter Simulation Conference (WSC). IEEE, 2017: 1001-1012.

[9] Zhang M. Constructing a Cognitive Agent Model using DEVS Framework for Multi-agent Simulation[C]// Proc. 15th Eur. Agent Syst. Summer School (EASSS), 2013: 1-5.

[10] Akplogan M, Quesnel G, Garcia F, et al. Towards a Deliberative Agent System based on DEVS Formalism for Application in Agriculture[C]//Proceedings of the 2010 Summer Computer Simulation Conference. 2010: 250-257.

[11] Chow A C H, Zeigler B P. Parallel DEVS: A Parallel, Hierarchical, Modular Modeling Formalism[C]// Proceedings of Winter Simulation Conference. IEEE, 1994: 716-722.

[12] 孙长银, 穆朝絜. 多智能体深度强化学习的若干关键科学问题[J]. 自动化学报, 2020, 45: 1-12.
Sun Changyin, Mu Chaoxu. Important Scientific Problems of Multi-agent deep Reinforcement Learning [J]. Acta Automatica Sinica, 2020, 45: 1-12.

[13] Haarnoja T, Zhou A, Abbeel P, et al. Soft Actor-critic: Off-policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor[C]//International Conference on Machine Learning. PMLR, 2018: 1861-1870.

[14] 张红旗, 杨峻楠, 张传富. 基于不完全信息随机博弈与 Q-learning 的防御决策方法[J]. 通信学报, 2018, 39(8): 56-68.
Zhang Hongqi, Yang Junnan, Zhang Chuanfu. Defense Decision-making Method based on Incomplete Information Stochastic Game and Q-learning[J]. Journal on Communications, 2018, 39(8): 56-68.

[15] 蒲玮, 李雄. 基于扩展 FIPA-ACL 的装备保障 Agent 通信语言[J]. 系统工程理论与实践, 2018, 38(1): 220-228.
Pu Wei, Li Xiong. Equipment Support Agent Communication Language based on Extended FIPA-ACL [J]. System Engineering Theory&Practice, 2018, 38(1): 220-228.

[16] 梁凯, 陈志军, 闫学勤. 移动机器人路径规划仿真研究[J]. 现代电子技术, 2018, 41(17): 6.
Liang Kai, Chen Zhijun, Yan Xueqin. Simulation Study on Effective Path Planning for Mobile Robot[J]. Modern Electronics Technique, 2018, 41(17): 6.

[17] 陈建平, 邹锋, 刘全, 等. 一种基于生成对抗网络的强化学习算法[J]. 计算机科学, 2019, 46(10): 265-272.
Chen Jianping, Zou Feng, Liu Quan, etc. Reinforcement Learning Algorithm Based on Generative Adversarial Networks[J]. Computer Science, ,2019, 46(10): 265-272.

[18] Van Tendeloo Y, Vangheluwe H. The Modular Architecture of the Python (P) DEVS Simulation Kernel [C]//Proceedings of the 2014 Symposium on Theory of Modeling and Simulation-DEVS. 2014: 387-392.

[19] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[J]. IEEE Transactions on Neural Networks (S1045-9227), 1998, 9(5): 1054.

(C)1994-2023 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net