计算机科学
COMPUTER SCIENCE

基于群智能体深度强化学习的模块化机器人自重构算法

王翰墨, 郑世杰, 徐若楠, 郭斌, 吴磊

引用本文

王翰墨, 郑世杰, 徐若楠, 郭斌, 吴磊基于群智能体深度强化学习的模块化机器人自重构算法[J]. 计算机科学, 2023, 50(6): 266-273.

WANG Hanmo, ZHENG Shijie, XU Ruonan, GUO Bin, WU Lei. Self Reconfiguration Algorithm of Modular Robot Based on Swarm Agent Deep Reinforcement Learning [J]. Computer Science, 2023, 50(6): 266-273.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230300044

基于群智能体深度强化学习的模块化机器人自重构算法

王翰墨　郑世杰　徐若楠　郭　斌　吴　磊
西北工业大学计算机学院　西安 710072
(whm2001@mail.nwpu.edu.cn)

摘　要　模块化机器人是由一定数量、具有独立功能的标准模块组合而成的。自重构问题是目前模块化机器人研究领域的热点与难点。传统的图论算法或者搜索算法在模块数量较多、复杂度较大时,无法在多项式时间内寻找到通用最优解。文中从群智能体深度强化学习的角度出发,将每个同构模块视为具有学习与感知能力的单智能体,提出了基于QMIX的模块化机器人自重构算法。针对该算法,设计了一种新型的奖励函数,并在限制智能体的动作空间的基础上,实现了智能体并行化移动,在一定程度上解决了多智能体之间的协调合作问题,从而实现了从初始构型向目标构型的转变。实验以9个模块为例,对比了该算法与基于A*的传统搜索算法在成功率以及平均步数上的差异。实验结果表明,在时间步数限制合理的情况下,基于QMIX的模块化机器人自重构算法的成功率能够达到95%以上,两种算法的平均步数大约在12步左右,QMIX自重构算法能够逼近传统算法的效果。

关键词:模块化机器人;自重构;群智能体协作;深度强化学习;构型空间与运动空间
中图法分类号　TP242.6

Self Reconfiguration Algorithm of Modular Robot Based on Swarm Agent Deep Reinforcement Learning

WANG Hanmo,ZHENG Shijie,XU Ruonan,GUO Bin and WU Lei
School of Computer Science,Northwestern Polytechnical University,Xi'an 710072,China

Abstract　Modular robots are composed of a certain number of standard modules with independent functions. At present, self reconfiguration is a hot and difficult problem in the field of modular robot research. For complex problems, the traditional graph theory algorithm or search algorithm cannot find its optimal solution in polynomial time, and the complexity increases exponentially with the increase of the number of modules. From the perspective of deep reinforcement learning of swarm agents, the research regards each isomorphic module as a single agent with learning and perception ability, and proposes a modular robot self reconfiguration algorithm based on QMIX. For this algorithm, a new type of reward function is designed and the parallel movement of the agent on the basis of limiting the action space of the agents is realized, which solves the problem of coordination and cooperation between multiple agents to a certain extent, thereby realizing the transition from the initial configuration to the target configuration. In addition, in experiments, 9 modules are taken as examples to compare the success rate and average steps between this algorithm and the traditional search algorithm based on A*. Experimental results show that when the time step limit is reasonable, the success rate of the modular robot self-reconfiguration algorithm based on QMIX can reach more than 95%, and the average number of steps of the two algorithms is about 12 steps. The QMIX self-reconfiguration algorithm can approach the effect of the traditional algorithm.

Keywords　Modular robot, Self reconfiguration, Swarm agent collaboration, Deep reinforcement learning, Configuration space and action space

1　引言

在以智能制造、互联网加为标志的第四次工业革命的浪潮驱动下,机器人相关技术与产业受到了广泛关注。在规模化工业生产过程中,专用机器人的高效率优势得到了充分体现,但是专用机器人的设计周期长、制造成本高,这是行业

到稿日期:2023-03-04　返修日期:2023-04-13
基金项目:国家杰出青年科学基金(62025205);国家自然科学基金(62032020,62102317)
This work was supported by the National Science Fund for Distinguished Young Scholars(62025205) and National Natural Science Foundation of China(62032020,62102317).
通信作者:郭斌(guob@nwpu.edu.cn)

内亟待解决的难题。近年来出现的模块化的概念致力于将机器人整体分解为一定数量、具备独立功能的标准模块进行研究，这是缩短机器人设计周期、减少制造与维护成本的有效途径。自重构模块化机器人是模块化机器人中的一种类型，能够根据所处环境的变化或者任务的不同自适应调整构型。随着模块数量的增大，构型空间的数量呈指数级增长，采用何种策略在初始构型与目标构型之间找一条快速且低耗能的优化路径，是目前自重构模块化机器人研究的热点与难点问题。

对于自重构模块化机器人的模块构型，可以将其分类为同构模块与异构模块。同构模块之间易于连接运动，而异构模块种类较多，每种模块有各自的功能，通用性较差，因此目前大多数自重构机器人采用同构模块设计[1]。针对同构模块的自重构问题，传统搜索算法可能存在速度慢、收敛过早等问题，而强化学习是通过与环境的交互来学习优化策略，它不仅能学会旧构型的重构方式，还能在新构型需求出现时进行适应。近年来，随着深度学习与强化学习的结合发展，群智能体深度强化学习为模块化机器人自重构问题提供了新的思路与方向。受此启发，本文提出了一种基于深度强化学习的模块化机器人自重构算法，旨在将每个模块看作一个具有一定感知与学习能力的单智能体，训练实现从初始构型到目标构型的快速转换。因为强化学习本身的特性，在利用基于MADRL对模块化机器人进行建模时，仍存在以下几项挑战。

(1) 建模的复杂性。强化学习是通过智能体与环境交互，来获取环境的反馈奖励以不断改进智能体动作的执行策略，这种驱动模式十分依赖环境的奖励信号。此外，在训练过程中也可能出现智能体动作空间过大的情况，因此智能体对象、奖励函数以及动作空间的定义对于强化学习来说极其重要。

(2) 多智能体之间的协调困难。每个智能体的策略随时间不断变化，不仅要学习自身的策略与动作价值函数，同时还要兼顾整体与目标构型的相似程度，很难协调多个智能体共同进步。

(3) 每个智能体的策略动作受到了一定的约束。例如，模块在实际移动过程中必须要有支撑以及所有模块移动后必须连通等。强化学习在处理类似的严格约束的优化问题时，计算成本较大。

本文工作的贡献主要包括以下3个方面：

(1)从群智能体深度强化学习的角度对问题进行分析建模，在强化学习训练过程中借鉴了QMIX算法的主要思想，奖励函数考虑了模块构型的重合度，在生成构型的重合度增大时给予了正向的奖励。

(2)采用针对构型的合法性，建立了每个模块的运动空间限制机制，同时实现了模块的并行化移动，帮助训练群智能体之间的协调配合，使所有模块能够从初始构型变换为目标构型。

(3)针对基于QMIX的模块化机器人自重构算法，对比了该算法在不同时间限制步数下与基于A^*的传统搜索算法在平均步数以及成功率方面的差异，结果显示QMIX自重构算法能够逼近传统算法的效果，验证了深度强化学习在模块化机器人自重构问题中的优势与不足。

本文第1节论述了本文的研究背景；第2节介绍了已有的模块化机器人相关的研究工作、强化学习相关概念与研究；第3节介绍了同构模块的组织形式，包括模块的构型空间以及运动空间；第4节介绍了基于QMIX的强化学习自重构算法；第5节设计了实验验证对比；最后总结全文并展望未来。

2 相关工作

本文相关的研究领域主要有模块化机器人自重构、强化学习和深度强化学习。

2.1 模块化机器人自重构

模块化机器人自重构问题指模块化机器人如何根据任务的不同，自主感知环境和控制模块，在两种构型之间寻找到最优路径并且完成构型重组的问题。目前，国内外学者对模块化机器人自重构问题的研究已经取得一定的成果。

Sun等[2]提出了一种使用辅助模块的移动模块化自重构机器人(M2SBot)的新重构方法，机器人可以通过该方法自主地将任意空间配置重新配置为另一个空间配置。然而，该方法的重新配置需要在太空中完成，因此有一定的局限性。Ahmadzadeh等[3]提出了一种基于流体动力学的模块化机器人自重构规划方法，该方法将自重构规划问题模拟成虚拟铸造模型，同时利用不可压缩流体引擎计算生成自重构规划路径。因其独立于构型空间且对模块连通性没有要求，该规划方法是高度可扩展的。Parhami等[4]受到地图路径规划的启发，提出了一种基于重构图的模块化机器人自重构规划方法，以图中节点代表机器人模块，最终使得规划搜索效率提升了5～8倍。

Liu等[5]指出模块化机器人自重构过程的成本可以用模块移动的总步数来衡量，进而自重构路径规划问题成为了一个优化问题。该优化问题属于NP难问题，目前无法在多项式时间内寻找到最优解。常用的传统自重构路径规划算法有图遍历与最短路径搜索方法，使用连通图和Dijkstra算法求解，但其复杂度是指数级的，面对模块数量较多的问题时难以使用。有研究人员采用了启发式的搜索算法，其中一种典型的算法就是A^*搜索，该算法通过对路径剩余距离的度量提高了搜索效率，可用的度量有重叠性度量、最小移动步数度量、最优分配度量等。然而，无论是运用图论定义自重构路径，还是运用启发式算法进行搜索，在模块数量较大时依然存在时间复杂度高以及串行移动等问题。

Tarek等[6]提出了一种基于遗传编程的变形模块化机器人重构规划器，详细介绍了如何将遗传编程用作处理重新配置规划问题的自动编程工具。Walter[7]通过使用每个处理器上恒定数量的信息，并依赖光与接触传感器收集到的感官信息来完成自重构，提出了基于传感器驱动的模块化机器人自重构算法。Whitzer等[8]为SMORES形式的模块化机器人设计了一种新颖的重新配置规划算法，实现了通过将初始配置与目标配置进行比较，每个模块分布式执行重新配置动作，从而导致系统的全局重新配置。

特定构型的模块化机器人由于形状、模块自由度、运动方式等方面的差异，自重构方法也会存在差异。Naz等[9]针对圆柱点阵类型的模块化机器人研究出了一种并行、异步和完全分散式的分布式自重构算法，在模块通信、模块移动和自重构

时间方面的性能高度可预测。Luo 等[10]提出了一种基于球形模块化机器人的自重构算法，用于帮助跨越各类障碍物。Gerbl 等[11]展示了扩展的二叉树如何作为自重构规划的有效工具，用于解决三角构型的模块化机器人自重构问题。Bassil 等[12]基于三维多孔结构提出了元模块模型，并设计了一种自重构算法，实验结果表明该算法使得模块通信数量与模块运动数量成比例，同时自重构时间与重构直径成线性关系。Buchi 等[13]针对 6 格栅的模块化机器人，提出了一种分布式异步自重构算法，用于平衡每个模块到达目标构型所需要的步数。另外，还利用 VisibleSim 模拟器对比了 C2SR 算法，结果表明该算法在自重构所需步数以及平衡每个模块的移动步数上都有提升。

ZHANG 等[14]的发明专利使用基于强化学习的算法进行模块化机器人自重构规划，建立蒙特卡洛树搜索，将搜索样本输入神经网络进行训练，得到步数最少的自重构路径。该方法集传统的搜索与神经网络于一体，但是网络性能的提升需要反复搜索样本并训练，收集数据与训练无法同步进行。

Witz 等[15]提出了一种混合集中式/分布式的模块化机器人自重构方法。在该方法中，卷积神经网络根据模块化机器人初始构型与目标构型评估适应性最好的分布式自重构算法 C2SR 或者 TBSR，此外，该系统也可以扩展到任何数量的自重构算法。

2.2 强化学习

强化学习是机器学习的一类方法论，构成要素包括智能体、环境、状态、动作与奖励。常见的强化学习模型可以看作一个马尔可夫决策过程，智能体执行了某个动作后，环境将会转换到一个新的状态，对于该新的状态环境会给出奖励信号。智能体根据新的状态和环境反馈的奖励，按照一定的策略执行新的动作。上述过程为智能体和环境通过状态、动作、奖励进行交互的方式，强化学习正是用于描述和解决智能体在与环境的交互过程中通过学习策略以达成回报最大化或实现特定目标的问题。

强化学习问题所使用的算法主要分为策略搜索以及值函数两类。常见的强化学习算法包括了基于价值 Q 表格的 Q-Learning 算法、基于神经网络的 Deep Q-Learning 算法、基于策略梯度的 Policy-Gradient 算法等。无论采用何种算法，强化学习的目标都是学习更好的策略，即根据经验或者值函数估计采取行动，以达成回报最大化。

传统的强化学习算法在训练初期缺乏环境的先验知识，容易随机选择动作，这容易导致算法的迭代次数长、收敛速度慢。针对此问题，Li 等[16]提出了两阶段强化学习算法，用于模块化机器人编队，在第一阶段利用基于群体与知识共享的 Q-Learning 训练机制来获取最优共享 Q 表，而在第二阶段机器人根据共享 Q 表以及当前位置寻找目的地的最优路径。该算法与对比算法相比减少了近 50%的探索步数，编队运行时间也大幅缩短。

2.3 深度强化学习

深度强化学习是一种将深度学习的感知能力和强化学习的决策能力相结合的人工智能方法，通过智能体在环境中不断采取动作并获得反馈，深度神经网络对当前状态选择动作，从而获取达到问题的最优解的策略。深度强化学习可按智能体数量再分为单智能体和多智能体的深度强化学习，许多学者已经在该领域提出许多有效的算法。

Volodymyr 等[17]提出了深度强化学习算法，结合深度学习中的卷积神经网络和强化学习中基于 Q 值的决策方式，构建出了深度 Q 网络(DQN)并在 atari 等游戏上达到了与顶级人类玩家相近的水平。

Peter 等[18]提出了基于值分解的合作式多智能体学习算法(VDN)，将团队整体 Q 函数分解为各个智能体子 Q 函数之和，使单智能体通过局部的贪心法来最大化团队奖励，改善了多智能体强化学习中的虚假奖励和惰性智能体的状况。

Tabish 等[19]提出了 QMIX 多智能体强化学习算法，在值分解的基础上使用混合网络，将各个智能体的 Q 函数融合得到多智能体联合动作的 Q 函数，比 VDN 具有更优的效果。

3 同构模块的组织形式

3.1 构型空间

本文以二维模块化机器人为研究对象，即模块存在于笛卡尔平面，每个模块的位置可以用坐标(x,y)来表示。

每个模块的上、下、左、右 4 个方向均可以与其他任意模块相互连接，自重构机器人的所有模块必须以该方式连接成为一个整体。自重构机器人的各模块绝对位置和连接关系不同，形成了不同的构型。

在区分不同构型时，本文以模块的绝对坐标为准则，即不仅要求构型组成的形状相同，还要求机器人在二维平面中的所处位置相同。对于每一种构型的组成模块不作区分，即每个模块都是同构的，交换任意两个模块的位置不会认为是一种新的构型。

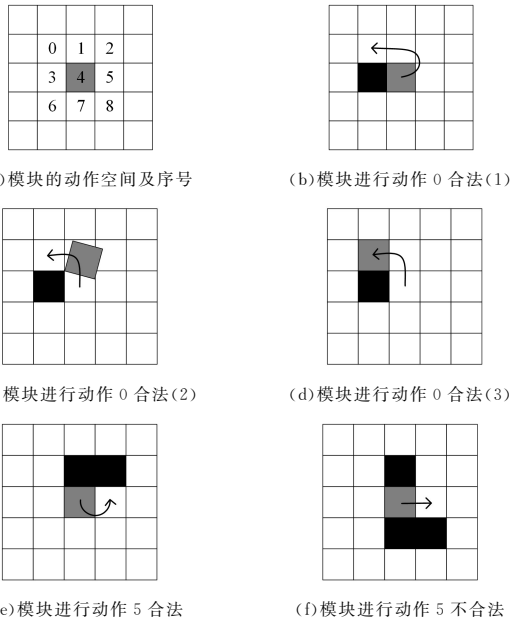
3.2 运动空间

以单一型模块为研究对象，每一个模块的形状为圆形，以绕边旋转的方式进行移动。在理想情况下，每个模块都能向周围 8 个方向进行移动，也可以保持不动，形成一个大小为 9 的动作空间，即如果模块的位置在(x,y)，移动后模块的位置可能是$(x,y-1)$，$(x,y+1)$，$(x-1,y)$，$(x+1,y)$，$(x-1,y-1)$，$(x-1,y+1)$，$(x+1,y-1)$，$(x+1,y+1)$或者(x,y)。动作空间如图 1(a)所示。在图 1 中，颜色表示模块类型，浅色模块为待移动的模块，深色模块为周围模块；箭头表示模块的运动，从起始位置指向目标位置。模块的移动需要满足 3 个基本要求：

(1)模块的移动必须有其余模块的足够支撑，以保障其从起始位置的某一边开始绕某一支点翻转，翻转至目标位置的某一边停止并形成新的连接。示意效果如图 1(b)－图 1(d)所示。

(2)模块在移动过程中，必须保留足够的几何空间，使模块翻转时不会与其他的模块相碰撞。

(3)模块移动的目标位置不能与其他模块的位置重合，模块的移动也不能破坏模块化机器人的连接整体性。为满足以上要求，当模块向周围 8 个方向移动时，必须保证有一侧存在模块以提供支撑，而另一侧留出翻转的空间。示意效果如图 1(e)、图 1(f)所示。



(a)模块的动作空间及序号 (b)模块进行动作 0 合法(1)

(c)模块进行动作 0 合法(2) (d)模块进行动作 0 合法(3)

(e)模块进行动作 5 合法 (f)模块进行动作 5 不合法

图 1 模块运动空间示意图

Fig. 1 Diagram of module movement space

另外，如果以串行的方式重构，则每一个时间步有且仅有一个模块进行移动。如果以并行的方式重构，则每个时间步可以有多个模块同时进行移动，但这些同时移动的模块不能相互碰撞或者存在冲突。

4 基于 QMIX 强化学习的自重构算法

4.1 传统算法及其不足

从数学角度来看，自重构问题是一类带有约束的优化问题，可以通过搜索策略来寻找可能的自重构路径。传统算法主要有基于图论的最优路径算法和启发式的算法，其中启发式的 A* 搜索算法是一种效果良好且实用的算法。它将从初始状态到目标状态的最小代价估计转化为从初始状态到某状态的最小代价与该状态到目标状态的路径的最小估计代价之和，每次选取最小代价估计的构型，并在此构型上进行一次移动，得到一个新的构型和新的代价估计，这样反复迭代直到搜索到目标构型，即得到了一条代价最小的重构路径。

传统搜索算法的核心是确定当前预估最优的节点，预估函数的好坏直接影响了算法的效率与输出结果。通常情况下，随着模块数量的增加，搜索的时间复杂度几乎是呈指数级上升。此外，重合度与剩余距离的估计往往是不准确的，这使得搜索算法的稳定性与可靠性下降。

鉴于传统搜索算法的弊端，下文提出了另一种深度强化学习的自重构算法，该算法以 QMIX 算法为主要核心思想。

4.2 多智能体马尔可夫决策过程

传统的强化学习指针对单个智能体进行训练与测试，也就是采用集中式策略执行动作，而针对模块化机器人自重构问题，由于每个模块都可以有不同的运动方式，整体学习将比单一代理的效率更高，因此这里采用分布式策略，将所有模块看作 N 个智能体的马尔可夫决策过程，具体定义为：

$$F=(N,S,A,R,\pi,T) \tag{1}$$

其中，N 为智能体的数量，S 为环境状态的有限集合，A 为 N 个智能体的联合动作空间，R 为奖励函数，π 为智能体观测到动作空间的映射，T 为当前环境状态下采取的动作到下一个环境状态的映射。

4.3 基于 QMIX 的自重构算法

QMIX 是一种多智能体强化学习的算法，其神经网络结构如图 2 所示。在多智能体强化学习中一个关键性的问题就是如何学习联合动作值函数，因为该函数的参数会随着智能体数量的增多而呈指数增长，如果动作值函数的输入空间过大，则很难拟合出一个合适的函数来表示真实的联合动作值函数。而 QMIX 在训练学习过程中加入全局状态信息辅助，通过联合动作-状态的总奖励值来学习每个智能体的分布式策略，这里的分布式策略是基于对联合动作值函数取最优参数值等价于对所有局部动作值函数取最优参数值的，即：

$$\underset{u}{\operatorname{argmax}}\, Q_{\text{tot}}(\tau,u)=\begin{pmatrix}\underset{u_1}{\operatorname{argmax}}\, Q_1(\tau_1,u_1)\\ \cdots\\ \underset{u_n}{\operatorname{argmax}}\, Q_n(\tau_n,u_n)\end{pmatrix} \tag{2}$$

我们可以发现，最优参数值就是通过贪心局部函数得到的，为了满足上述约束，需要使得联合动作值函数对每一个局部动作值步数的偏导都大于或等于 0，即它们的单调性相同，因此 QMIX 采用了一个混合网络对单智能体局部值函数进行合并，网络中的所有权值都是非负数，且对偏移量不加以限制，这样就可以确保满足单调性约束。在满足单调性约束的情况下，计算量随智能体数量线性增长，极大地提高了算法效率。另外，该算法能达到边收集数据边训练的效果，因此收敛的时间并不会太长。

4.3.1 基于模块重合度的奖励函数

由于 QMIX 算法的奖励是对于整体而言的，因此只能适用于合作环境，而不能用于竞争对抗环境。如果模块在到达目标构型时才给予奖励是不合适的，因为构型数量庞大，在很多情况下智能体无法达到目标构型，从而无法得到相应的奖励。本文提出了一种基于模块重合度的奖励函数：

$$r_{T+1}=\frac{O_{T+1}-O_T}{N_{\text{agent}}-O_0} \tag{3}$$

其中，O_{T+1} 表示第 T 时间步的当前构型经过一步扩展出的新构型与目标构型的重合度，O_T 表示在第 T 时间步的当前构型与目标构型的重合度，而 N_{agent} 表示智能体的数量，O_0 表示起始构型与目标构型的重合度。

重合度 O 描述了两个模块化机器人构型之间的重合数量，其数值为两个模块化机器人绝对位置相同的模块数量。其最小值为 0，表示两个机器人没有任何模块在同一位置；最大值为模块化机器人的模块总数 N_{agent}，表示两个机器人所有的模块都重合。

这个公式体现了多智能体每一步的奖励与新构型、旧构型的重合度都有关系，这能鼓励多智能体尽可能地学习增大重合度的策略。另外，我们还考虑了时间延迟，即每一步多智能体都会得到一个绝对值非常小的负奖励，这能在一定程度上促使模块群体尽快变为目标构型。

QMIX 的损失函数用于反向传播修正网络参数，具体表示为：

$$L(\theta)=\sum_{i=1}^{h}(y_i^{\text{tot}}-Q_{\text{tot}}(s,a,\tau;\theta))^2 \tag{4}$$

其中，h 为从经验回放池中采样的数量。而更新过程中用到了传统的深度神经网络的思想，可以利用式(5)来完成：

$$y^{\text{tot}}=r+\gamma\max_{a'}\widetilde{Q}(s',a',\tau';\widetilde{\theta}) \tag{5}$$

其中，$\widetilde{Q}$ 是目标网络。

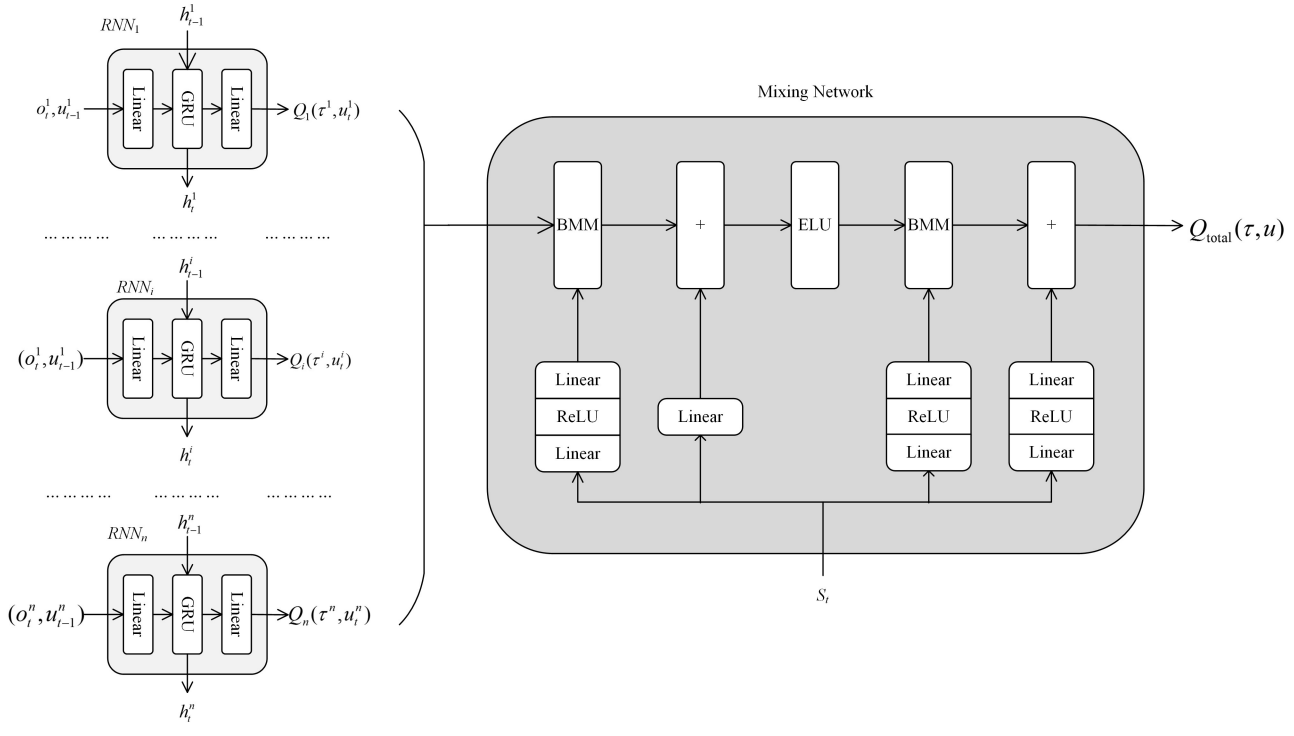


图 2 QMIX 网络结构

Fig. 2 QMIX network structure

4.3.2 模块运动空间限制机制

由于每个智能体原则上都存在 9 种可以选择的动作(包括原地不动)，这使得构型的变化数量非常庞大，在不加以限制的情况下到达目标构型的概率很低，智能体难以进行学习，因此采用何种动作执行策略加以限制值得关注。

为了使得目标构型出现的概率尽可能大，以便智能体更好地收集数据进行学习，我们让到达目标构型对应的智能体不再移动。此外，翻转不合法的模块也将会留在原来的位置。但是，在变为某些构型时，部分模块的停留使得其他模块由于构型的连通性或者翻转合法性而无法移动，在这种情况下必须判断可移动的智能体数量，若为零则说明所有智能体都无法移动，无法重构为目标构型，此时到达目标构型对应的智能体不加以限制，可以自由移动。

在实现时，每完成一个时间步后，环境会对每一个智能体下一时间步的可选择动作进行预先判定，排除明显不合法的动作，并且以重合度为标准引导智能体朝目标构型移动，同时又不会将智能体的所有动作限制住。可选择动作判断的具体规则如下。

规则 1 若智能体$agent_i$的动作act_j的执行违反了构型合法性或移动合法性，则将act_j排除出$agent_i$的可选择动作。

规则 2 若智能体位置与目标构型某一模块位置重合，则只保留原地不动，将其作为可选择动作。

规则 3 若根据规则 1 和规则 2，所有智能体都只有原地不做动作为其可选择动作，则无视规则 2 的限制，重新分配可选择动作。

4.3.3 基于空间预留的并行化移动

不同于传统搜索算法，QMIX 算法可以很容易地实现多个模块的同时协作，从而达到并行化自重构的效果。在模块之间协作得当的情况下，并行地自重构可以很好地节省时间成本，以更高的效率完成自重构的路径规划。

这要求我们不仅要考虑每个智能体的翻转是否合法，同时还需预留每个模块翻转移动需要的其他支撑模块以及翻转路径，当存在智能体需要借助别的模块所预留的空间进行移动时，该动作视为不具有移动合法性的动作而被排除。这样并行的模块各自拥有自己专属的位置空间进行移动，避免了同时并行的模块之前发生冲突。将该方法加入可选择动作判断的规则 1 中，能够确保模块并行的正确性，同时获得了并行化移动所带来的更快的速度。

4.3.4 基于 QMIX 的自重构算法的详细设计

本文在 QMIX 算法的基础上提出，奖励函数、模块运动空间的限制机制以及并行化移动方式的主要目标是降低强化学习训练的难度。然而，强化学习的不可预知性还是导致可能存在构型变换失败的重要因素。因此，我们的目标是在提高算法的成功率的同时，减少移动步数，并尽可能达到或者超过传统基于 A* 的搜索算法的效果。

基于 QMIX 的自重构算法在算法 1 中进行了详细说明。

该算法描述了一个多智能体强化学习的过程，一共进行 *max_step* 轮训练。在每一轮的训练中，设置最大时间步数 T，在每一时间步中，首先会获取当前环境的全局状态信息，再根据智能体编号依次获取每个智能体的观测信息和可选择动作信息；将这些数据输入 QMIX 网络并得到价值函数，根据 ε-*greedy* 策略选取每个智能体的动作；然后将每个智能体的动作在环境中实施，得到环境反馈的下一状态和基于重合度的奖励函数值，这些状态转移信息作为一条经验回放存储到经验池中；最后根据新的环境状态筛选每个智能体下一时间步的合法动作，生成每个智能体的可选择动作。当前构型达到目标构型时，即成功完成自重构，而 QMIX 网络会从存储的经验回放中采样并更新参数。

算法 1 基于 QMIX 的自重构算法

1. 初始化经验回放池 D，容量为 M，每轮最大步数 T，智能体数量 N，当前已走步数 step=0，最大训练步数 max_step
2. 初始化 QMIX 网络参数 θ
3. WHILE step<=max_stepDO
4. 重置构型所在环境信息，初始状态 s_0
5. FOR t=0 to T DO
6. 获取当前全局状态 s_t，设置动作向量 a_t
7. FOR i=1 to N DO
8. 获取智能体观测 o_i
9. 获取智能体 $avail\_act_i$
10. 依据智能体观测 o_i 与 QMIX 网络得到价值函数 q_value
11. 以 $1-\varepsilon$ 的概率选择动作 $act_i=\underset{i}{\arg\max}$ q_value
12. 否则在可选择动作 $avail\_act_i$ 中随机选取动作 act_i
13. 将动作 act_i 加入 a_t 中
14. ENDFOR
15. 利用动作向量 a_t 并行化更新智能体状态环境，并根据式(3)得到奖励 r_t，同时进入下一状态 s_{t+1}，并将所有智能体的状态转移信息存储到经验池中
16. FOR i=1 to N DO
17. 筛选合法的动作，加入 $avail\_act_i$ 列表
18. 进行重合判断，根据可选择动作判断规则 2 和 3 是否限制了重合模块的移动，并修改 $avail\_act_i$
19. 将 $avail\_act_i$ 加入 $avail\_actions_{t+1}$
20. ENDFOR
21. IF 当前构型到达目标构型
22. BREAK
23. ENDIF
24. 从经验池中采样训练，由式(4)更新网络参数
25. ENDIF
26. 根据 t 值更新当前已走步数 step
27. ENDWHILE

5 实验验证对比

5.1 实验设置

5.1.1 构型样本收集

我们围绕模块数量为 9 个的模块化机器人展开实验，按照上述构型空间和运动空间的规则进行自重构，进行软件模拟的二维平面空间大小为 11×11。为了统一标准，我们设置了一个构型样本空间，其中包含了 82 种较为常见和规则的构型，包括方形、一字形、T 字形、L 字形、蛇形等，如图 3 所示。

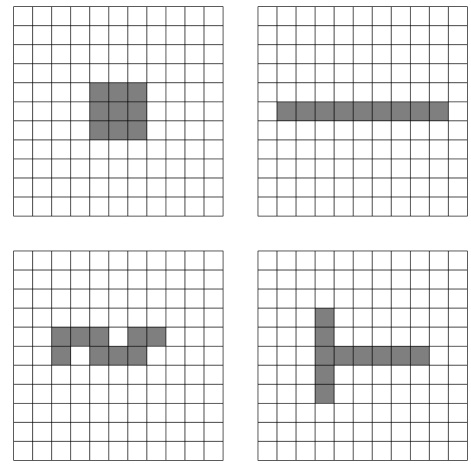


图 3 几种常见的构型

Fig. 3 Several common configurations

5.1.2 参数设置

在对比实验中，传统 A* 搜索算法没有设置相关参数，而在 QMIX 模型中，学习率设置为 0.001，经验回放池的容量为 1 000，批量大小都固定为 128，值从 1 线性下降到 0.1，共进行了大约 100 万步的训练。在强化学习的环境中，每个智能体所能观察到的状态以及全局状态都是当前的整个构型空间以及目标构型的空间状态。

5.1.3 实验对比流程

对于本文提出的基于 QMIX 的自重构算法，从上述构型样本空间中随机抽取起始构型和目标构型，形成强化学习的环境，并按照上述训练方法进行训练。在进行一定程度的训练后，模型会得到收敛。接下来将围绕训练好的模型和传统模型进行一系列测试。

自重构问题的起始构型和目标构型从该构型样本空间内统一抽取，记录基于 A* 传统算法和基于 QMIX 多智能体强化学习算法的自重构的所需步数，进而对比两种算法在解决模块化机器人自重构规划问题上的优劣。我们选取最为普通的方形作为起始构型，而将剩余的 81 种作为目标构型进行变形，得到了几组测试数据，测试项包括变形的成功与否和变形所花费的步数。

5.2 实验结果

本文得到了不同算法的成功率和平均步数，如表 1 所列，步数随构型变化的曲线如图 4 所示。

表 1 传统搜索算法与基于 QMIX 的自重构算法的对比

Table 1 Comparison between traditional search algorithms and self reconfiguration algorithms based on QMIX

算法类型	时间步数限制	成功率/%	平均步数
传统 A* 搜索算法	无	100	12.14
QMIX 自重构算法	200	97.53	12.24
QMIX 自重构算法	100	98.77	12.51
QMIX 自重构算法	30	95.06	12.12
QMIX 自重构算法	25	88.89	11.39

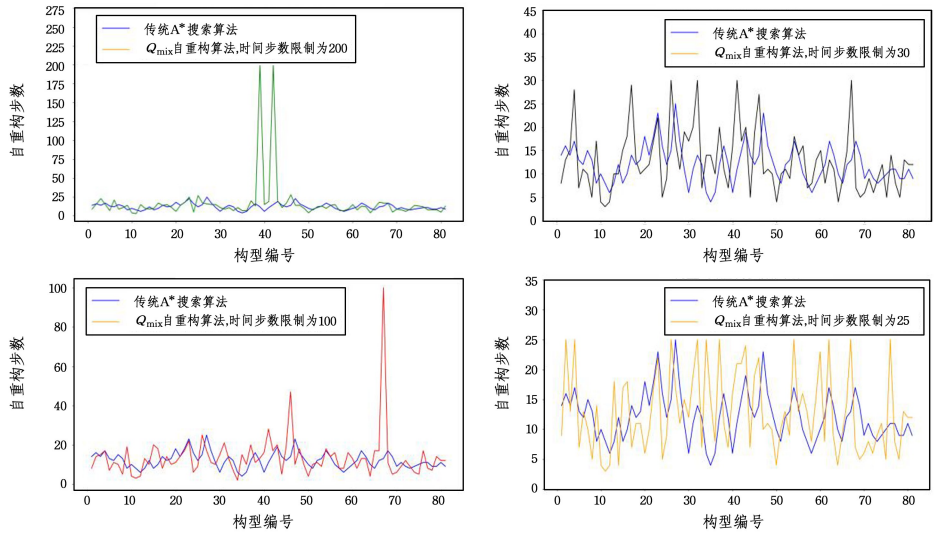


图 4 传统搜索算法与基于 QMIX 的自重构算法步数对比曲线图

Fig. 4 Comparison curves of the number of steps between traditional search algorithm and self reconfiguration algorithm based on QMIX

从对比表与步数对比曲线图可以看出，传统的 A* 算法能够保证 100% 的成功率，QMIX 强化学习算法的自重构成功率随着训练时间步数限制的增大无法达到 100%，但依旧维持着一个较高的成功率，这是因为强化学习训练过程中随机的动作空间非常庞大，训练难以收敛到准确率非常高的模型上，因此有部分构型编号在时间限制内未能完成自重构。在自重构的平均步数上，QMIX 自重构算法在时间步数限制较严时有较好的表现，这与模块并行化移动密不可分。在绝大多数构型上，基于 QMIX 的自重构算法都能取得与传统算法较为接近的步数，这说明基于 QMIX 的自重构算法训练的智能体能有效移动完成自重构过程。

在实际情况下，模块化机器人通常是用于完成特定任务的，需要的构型不会太多，因此基于 QMIX 的自重构算法完全可以满足任务要求。例如，空间在轨任务模块化机器人用于狭小空间内感应、收集、处理和传送信息，仿生机器人在抢险救灾中的侦查、检测以及物资输送等方面有显著优势[20]。相比传统算法，QMIX 多智能体强化学习的策略更容易实现模块之间的并行协作，使其能够在总体上达到逼近传统算法的效果，甚至在平均步数上实现超越。

5.3 实验结论

传统图论算法或者搜索算法在模块数量增多的情况下，复杂度显著提升。在这种背景下，本文从不同角度出发，讨论了基于 QMIX 的同构模块化机器人自重构算法，用于训练同构模块从初始构型变换为目标构型。我们在训练过程中引入了合适的奖励函数与动作限制机制，同时实现了模块并行化移动处理，有助于引导模块群体协作合作，在实现自重构的同时降低自重构路径的平均步数。另外，我们也在平均步数以及成功率方面对比了该算法与传统 A* 搜索算法的差异，验证了深度强化学习在模块化机器人自重构问题中的优势与不足。

结束语 本文采用 QMIX 深度强化学习算法，对模块化机器人自重构路径规划问题进行求解。我们基于重合度的奖励函数，辅以运动空间限制和并行化移动，对 QMIX 网络进行训练，在模块数为 9 的模块化机器人自重构路径规划求解中达到了 95% 的成功率，平均消耗步数与传统 A* 搜索算法接近。

我们未来的工作主要聚焦于探索更优的奖励函数和神经网络模型，设计自重构成功率更高、平均消耗步数更少的算法。在验证方面，我们会增加模块化机器人的复杂度，增大机器人的模块数量，并将实验场景从 2D 转向 3D。此外，还将对比更多自重构算法，不仅限于传统搜索算法（如 A* 搜索），同时考虑其他类型的强化学习算法以寻找不同方法的优缺点。

参 考 文 献

[1] DAI Y, ZHANG Q H, GAO Y F, et al. Overview of self-reconfigurable modular robot module design[J]. Journal of Harbin University of Technology, 2021, 26(5): 34-43.

[2] SUN X, GE W, WANG X, et al. A reconfiguration approach for self-reconfigurable modular robot using assisted modules[C]// IEEE International Conference on Mechatronics & Automation. IEEE, 2015: 1436-1441.

[3] AHMADZADEH H, MASEHIAN E. A fluid dynamics approach for self-reconfiguration planning of modular robots [C]// RSI International Conference on Robotics & Mechatronics. IEEE, 2016: 139-145.

[4] PARHAMI P, MORADI H, ASADPOUR M, et al. Generating an efficient hub graph for self-reconfiguration planning in modular robots[C]// Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on. IEEE, 2015: 476-481.

[5] LIU Y J, YU M J, YE Z P, et al. Path planning for self-reconfigurable modular robots: a survey[J]. Scientia Sinica Informationis, 2018, 48(2): 143-176.

[6] TAREK A, NOUREDDINED, YVES D, et al. Genetic Programming-based Self-reconfiguration Planning for Metamorphic Robot[J]. International Journal of Automation and Computing, 2018, 15(4): 57-68.

[7] WALTER J E. Sensor-Driven Algorithm for Self-Reconfigura-

tion of Modular Robots[C]// 2018 International Conference on Reconfigurable Mechanisms and Robots. 2018:1-7.

[8] LIU C, WHITZER M, YIM M. A Distributed Reconfiguration Planning Algorithm for Modular Robots[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 4231-4238.

[9] NAZ A, PIRANDA B, GOLDSTEIN S C, et al. A distributed self-reconfiguration algorithm for cylindrical lattice-based modular robots[C]// IEEE International Symposium on Network Computing & Applications. IEEE, 2016.

[10] LUO H, LI M, LIANG G, et al. An Obstacle-crossing Strategy Based on the Fast Self-reconfiguration for Modular Sphere Robots[C]// IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2020.

[11] GERBL M, GERSTMAYR J. Self-reconfiguration planning of adaptive modular robots with triangular structure based on extended binary trees[C]// IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2020: 3312-3319.

[12] BASSIL J, PIRANDA B, MAKHOUL A, et al. RePoSt: Distributed Self-Reconfiguration Algorithm for Modular Robots Based on Porous Structure [C]// IEEE/RSJ International Conference on Intelligent Robots and Systems. 2022: 12651-12658.

[13] BUCHI B, MABED H, FRÉDÉRIC L, et al. Translation based Self Reconfiguration Algorithm for 6-lattice Modular Robots [C]// International Symposium on Parallel and Distributed Computing. IEEE, 2021: 49-56.

[14] ZHANG Y Z, WANG W H, HUANG P F, et al A Self Reconstruction Planning Method for Heterogeneous Modular Robots Based on Reinforcement Learning Algorithm: CN110297490A [P] 2019.

[15] WITZ F, BUCHI B, MABED H, et al. Deep Learning for the selection of the best modular robots self-reconfiguration algorithm [C]// 2022 IEEE Symposium on Computers and Communications. Rhodes, Greece, 2022: 1-6.

[16] LI W K, YUE H W, WANG H M, et al. Modular self-reconfigurable robot formation based on improved reinforcement learning [J]. Computing Technology and Automation, 2022, 41(3): 6-13.

[17] VOLODYMYR M, KORAY K, DAVID S, et al. Playing Atari with Deep Reinforcement Learning[J]. arXiv: 1312. 5602, 2013.

[18] SUNEHAG P, LEVER G, GRUSLYS A, et al. Value-Decomposition Networks For Cooperative Multi-Agent Learning [J]. arXiv: 1706. 05296, 2017.

[19] RASHID T, SAMVELYAN M, DE W, et al. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning [J]. Journal of Machine Learning Resarch, 2020, 21(1): 7234-7284.

[20] ZHANG Y, WANG Q, KANG Y L, et al. Summary of key technologies and research prospects of modular self-reconfigurable robots[J]. Journal of Hebei University of Science and Technology, 2022, 43(6): 602-612.

WANG Hanmo, born in 2001, undergraduate, is a member of China Computer Federation. His main research interest is modular robot.

GUO Bin, born in 1980, Ph. D, professor, Ph. D supervisor, is a member of China Computer Federation. His main research interests include ubiquitous computing and crowd intelligence with the deep fusion of human, machine and things.

(责任编辑:喻藜)