Keepaway抢球任务中基于策略重用的迁移学习算法

李学俊 陈士洋 张以文 李龙澍

(安徽大学计算机科学与技术学院 合肥 230601)

摘 要 在RoboCup Keepaway中,球员使用强化学习能获得很好的高层策略。然而由于Keepaway任务的状态空间巨大,强化学习需要探索很多步才能收敛,学习过程十分耗时。针对这一问题,对于5v4规模的Keepaway任务,将策略重用技术应用于抢球球员高层决策的强化学习中,以实现迁移学习。首先合理设计了球员在4v3和5v4任务间的迁移学习方案及状态与动作空间的映射,然后提出了基于策略重用的迁移学习算法。实验表明,对于5v4任务,在训练时间约束下,迁移学习比强化学习获得了更短的任务完成时间和更高的抢断成功率,从而学习到了较优的高层策略。因此,为达到相同策略水平,迁移学习所需的训练时间明显比强化学习少。

关键词 机器人足球,Keepaway,抢球策略,策略重用,迁移学习

中图法分类号 TP242 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.4.038

Transfer Learning Algorithm between Keepaway Tasks Based on Policy Reuse

LI Xue-jun CHEN Shi-yang ZHANG Yi-wen LI Long-shu

(School of Computer Science and Technology, Anhui University, Hefei 230601, China)

Abstract In RoboCup Keepaway task, players can gain good high-level strategy with reinforcement learning. However, as Keepaway tasks have very huge state space, normal reinforcement learning requires a great many searching steps to converge, and needs very long time. To solve this problem, for 5v4 scale Keepaway task, policy reuse technique is applied to the reinforcement learning procedure of takers' high-level decision to achieve transfer learning. The transferring plan along with the map of state and action space between 4v3 and 5v4 task were rationally designed. Then a policy reuse based algorithm was stated. Experiments show that after the same training time for 5v4 scale task, takers get shorter task finish time and higher stealing success rate during transfer learning than in normal reinforcement learning. So there are better policies learned by transfer learning. Transfer learning needs much less training time than normal reinforcement learning to get the same policy level.

Keywords RoboCup soccer, Keepaway, Stealing police, Policy reuse, Transfer learning

1 引言

RoboCup 2D(机器人足球世界杯仿真2D项目)通过模拟人类足球比赛,提出了一个复杂的实时多智能体决策问题,其是人工智能领域的一个前沿标准问题,具有重要的研究意义[1]。RoboCup 2D中有一个Keepaway任务,该任务中双方球员在一块小场地内进行控球或抢球对抗,以维持或争夺控球权为目标[2]。Keepaway任务的关键问题是高层动作决策,抢球球员需要从对方所有传球路线中选择一条封堵。

对Keepaway任务,Peter Stone[3]和左国玉[4]将Sarsa强化学习算法应用于持球球员的高层动作决策,使得高层持球决策得到优化。在强化学习过程中,智能体通过不断地进行动作尝试并观察动作的回报,逐渐学会在各种情形下选择对其有利的动作,以使自身在与环境交互过程中获得高的累积回报值[5]。Keepaway任务的规模很大,强化学习需要很多步才能收敛,学习十分耗时,一般需要10个小时以上才能学到较好的策略。在学习过程中,抢球球员完成任务时间长,抢球效率很低。针对这一问题,Taylor[6]和Fernández[7]对Keepaway中高层持球决策的普通强化学习进行延伸,通过使用策略重用技术,优化了高层持球策略的学习效率。

迁移学习通过找到相同领域内不同规模问题间的相似之处,利用已解决的较小规模问题策略来帮助解决较大规模问题的学习。迁移学习中的问题虽然规模不同,但是往往有很多相似之处,例如对于4v3和5v4规模的Keepaway任务,在为这两个规模的抢球策略设计强化学习模型时,可以看到它们的状态空间、动作空间存在自然延伸的关系。

策略重用[7]是一项实现迁移学习的技术,指进行强化学习的智能体利用过去在类似任务中学到的策略,来帮助加速当前任务学习进程的技术。策略重用适用于分段任务的迁移学习。策略重用前需要智能体事先通过学习获得类似任务的策略,以作为待重用的旧策略。策略重用时智能体既有对新任务的强化学习,又有对旧任务中学到的策略的重用,要求智

到稿日期:2014-05-25 返修日期:2014-08-28 本文受安徽省自然科学基金项目(1408085MF132),安徽大学青年骨干教师培养(02303301)资助。

李学俊(1976-),男,博士,副教授,主要研究方向为智能软件、云工作流,E-mail:xjli@ahu.edu.cn;**陈士洋**(1989-),男,硕士,主要研究方向为智能软件;**张以文**(1976-),男,博士,副教授,主要研究方向为智能计算;**李龙澍**(1956-),男,博士,教授,主要研究方向为机器学习。

能体具有在学习当前问题和利用过去策略之间平衡的机制[7]。其可以通过在学习过程中维护并更新代表每个策略重要性的权值，以权值作为依据概率性地选取每个策略的方法来实现这个平衡机制。在具体利用旧策略时，新旧策略的问题空间规模不同，包括状态空间和动作空间规模上的差别，需要分别对新旧问题的状态空间和动作空间进行映射[8]。因此可以通过分析新旧问题的状态空间和动作空间的特点，找到合理的映射方案。

然而目前尚无将策略重用应用于 Keepaway 任务中抢球动作决策迁移学习的文献研究。在 Keepaway 任务中，控球和抢球的任务目标相反，任务特点也有所不同，因而球队策略也存在区别。控球的特点是要求无球球员进行合理的无球跑动，同时持球球员选择合理的传球路线；抢球的特点则是要求抢球球员分工对控球球员进行压迫和逼抢。控球任务对无球球员的跑动要求相对较低，研究重点是持球球员的传球决策；而对于抢球，离球最近的抢球球员的决策比较固定，例如必须上前逼抢持球球员，否则球队很难抢下球，其他负责拦截传球路线的抢球球员的决策具有研究价值。本文针对 Keepaway 中抢球任务的上述特点，以高层抢球决策的强化学习为基础，研究基于策略重用的抢球决策的迁移学习，以获得较短的任务完成时间和较高的抢断成功率，从而学习到较优的高层策略。本文第 2 节描述了 Keepaway 平台的相关定义以及抢球球员与控球球员的高层动作和总体策略；第 3 节设计了高层抢球策略的任务间迁移学习方案以及状态与动作空间的映射；第 4 节提出了基于策略重用的任务间迁移学习算法；第 5 节实验验证了迁移学习算法；最后进行了总结与展望。

2 Keepaway 平台

2.1 相关定义

Keepaway 是两支球队在一定大小的场地内进行控球或抢球对抗的训练。Keepaway 在场地大小、控球和抢球球员人数等方面有不同规模。由于控球相对较难，一般控球球队比抢球球队多 1 名球员；球员总数一般介于 5 到 9，常见的为 3v2(3 名控球球员、2 名抢球球员之意，后面类推)、4v3 和 5v4 规模；训练区域一般比整个比赛的场地小得多，例如 20m×20m、30m×30m 的正方形区域[2]。

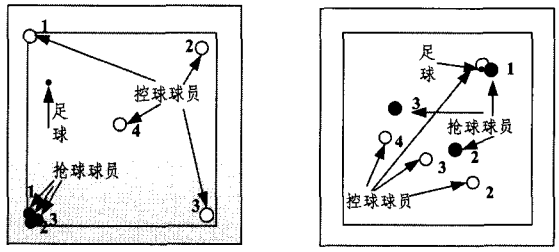
定义 1(训练段) 从训练开始状态到抢球球员抢下球或球离开训练区域为止的整个过程。

在一个训练段开始时，球在一名控球球员身边，其他球员距离该球员有一定距离；训练段开始后，抢球球队要去争夺足球，控球球队需要维持控球权。当抢球球员踢到球或者球在抢球球队的逼抢下离开训练区域，则抢球球队完成任务，当前训练段结束，开始下一个训练段，Keepaway 训练就这样不断地进行下去。

定义 2(抢球任务完成时间) 一个训练段持续的时间长度，一般以一个服务器仿真周期为单位。

定义 3(抢断成功率) 对于待统计的 N 个训练段，在给定时间限制下抢球成功的训练段数占总的训练段数 N 的百分比。

图 1 展示了 30m×30m 场地下 4v3 规模 Keepaway 的训练段开始和中间场景，其中有 4 名控球球员和 3 名抢球球员。



(a)4v3 Keepaway 训练段开始场景　(b)4v3 Keepaway 训练段中间场景

图 1　4v3 规模 Keepaway 训练场景

2.2 抢球球员的高层动作和总体策略

Peter Stone[2]根据人类足球的知识，通过封装球员的底层原子动作，为 Keepaway 任务定义了一系列的高层动作。本文针对高层策略进行研究，所涉及的策略均基于这些现有的高层动作。抢球球员使用的高层动作包括：

(1)“踢球”：试图踢到球；

(2)“跑向球”：直接跑向球，以达到截球或控球的目的；

(3)“拦截路线(i)”：移动到对方当前控球球员 $K1$ 和对方其他球员 Ki 连线上的一个位置，以期截获对方传球，$2\leqslant i\leqslant m$，其中 m 为控球球员数，下同。

在 Keepaway 中，如引言中所述，考虑到抢球任务的特点，本文使用如下总体策略：一名抢球球员离球很近可以获得控球权时，使用“踢球”动作，这样就可以完成抢球任务；当控球球员在控球时，抢球球员中距离控球球员最近的那名抢球球员应该上前逼抢，即使用“跑向球”动作。对于以上这两种情况，总是采用这样的固定策略；而在其他情况下，动作决策分为传统手工策略和强化学习策略。抢球球员的总体策略用伪代码表示为：

Step 1　如果球在我的控制范围内，返回“控球”动作；

Step 2　如果我是离球最近的那名抢球队员，返回“跑向球”动作；

Step 3　利用手工策略或进行强化学习，在{拦截路线(2)，拦截路线(3)，拦截路线(4)}中选择动作。

在 Step 3 中，传统手工抢球策略的一般思路是：预测抢球球员最有可能的传球路线进行拦截，一般会选取具有最大安全角度的路线。

2.3 控球球员的高层动作和总体策略

控球球员是 Keepaway 中与抢球球员对抗的另一方，控球球员的高层动作[2]包括：

(1)“控球”：保持对球的控制并不让对方靠近球；

(2)“传球(i)”：将球传给第 i 名队友，$2\leqslant i\leqslant m$；

(3)“跑位”：跑到一个不受对方紧逼的安全区域，给队友提供传球路线；

(4)“跑向球”：跑向球，以达到截球或控球的目的。

控球球员的总体策略用伪代码表示为：

Step 1　如果存在一个正在持球或可以更快踢到球的队友，返回“跑位”动作；

Step 2　如果球不在自身控制范围，返回“跑向球” 动作；

Step 3　如果 4m 内没有抢球球员，返回“控球”动作；

Step 4　选择具有最大传球角度的传球路线 i，返回“传球(i)” 动作。

3 任务间的迁移学习方案与空间映射

本节针对 5v4 规模的 Keepaway 问题，利用策略重用技

术实现高层抢球策略的迁移学习方案与映射的设计。

3.1 迁移学习方案

为了在5v4抢球训练中进行迁移学习，选择4v3抢球策略作为旧策略，这个旧策略可以通过进行4v3抢球训练的强化学习得到。对5v4任务中的4名抢球球员设置迁移学习方案，如表1所列。3名抢球球员T1、T2和T3先进行4v3任务的普通强化学习，学到4v3任务下的抢球策略，然后在5v4任务中基于学到的4v3任务下的抢球策略进行迁移学习；第4名抢球球员T4从零开始进行5v4任务的强化学习。

表1 4v3规模到5v4规模的迁移学习方案

任务＼球员	抢球球员 T_1	抢球球员 T_2	抢球球员 T_3	抢球球员 T_4
4v3规模	从零开始学习策略 $\Pi_{4v3}^{T_1}$	从零开始学习策略 $\Pi_{4v3}^{T_2}$	从零开始学习策略 $\Pi_{4v3}^{T_3}$	不参与
5v4规模	通过重用策略 $\Pi_{4v3}^{T_1}$ 学习策略 $\Pi_{5v4}^{T_1}$	通过重用策略 $\Pi_{4v3}^{T_2}$ 学习策略 $\Pi_{5v4}^{T_2}$	通过重用策略 $\Pi_{4v3}^{T_3}$ 学习策略 $\Pi_{5v4}^{T_3}$	从零开始学习策略 $\Pi_{5v4}^{T_4}$

3.2 状态与动作空间的映射

在5v4任务中进行策略重用时，旧策略是4v3规模的，而当前问题是5v4规模的。当球员更新获得一个5v4任务空间 S_{5v4} 的环境状态 s_{5v4} 时，为了利用旧策略，需要将它映射为一个4v3问题空间 S_{4v3} 的世界状态 s_{4v3}，这样可以利用旧策略得到在旧问题空间下的解，假设得到对应动作决策 a_{4v3}，该解属于4v3问题的动作空间 A_{4v3}；还要把 a_{4v3} 映射回当前5v4规模问题的动作空间 A_{5v4} 下，得到 a_{5v4} 以作为重用策略得到的动作决策。可以看到，在策略重用时，要进行任务间的两次映射，第一次是新状态空间到旧状态空间的映射 $\rho_S: S_{5v4} \rightarrow S_{4v3}$，第二次是旧动作空间到新动作空间的映射 $\rho_A: A_{4v3} \rightarrow A_{5v4}$。

对于 ρ_S，由于抢球训练的状态空间在定义时是选取一些关键的相对量，状态维度随着任务规模的扩大而自然延伸。根据文献[3]中状态空间的定义，S_{5v4} 表示为 $\{dist(K1,C), dist(K2,C), \cdots, dist(K5,C)\}$，$S_{4v3}$ 表示为 $\{dist(K1,C), dist(K2,C), \cdots, dist(K4,C)\}$，所以4v3规模任务的状态向量总包含在5v4规模任务的状态向量之中，映射是只需进行投影，把多余的 $dist(K5,C)$ 分量丢弃。同样，对于 ρ_A，A_{4v3} 表示为{拦截路线(2)，拦截路线(3)，拦截路线(4)}，A_{5v4} 表示为{拦截路线(2)，拦截路线(3)，拦截路线(4)，拦截路线(5)}。因此只需进行恒等映射，即依旧策略选择的动作作为5v4问题最终的决策动作。

4 基于策略重用的迁移学习算法

在4v3任务中通过强化学习得到策略 Π_{4v3} 后，抢球球员在5v4任务中进行迁移学习。迁移学习算法在本质上属于强化学习算法，它与普通强化学习的不同在于，学习过程中动作决策采用当前学到的策略，或者重用旧策略。Sarsa算法是由Rummery和Niranjan[9]提出的一种基于模型的强化学习算法。Peter Stone[3]在将强化学习应用于Keepaway中控球球员的决策时，使用Sarsa算法得到了很好的效果。本文在普通强化学习算法Sarsa的基础上，给出抢球球员的迁移学习算法，即PR-Sarsa算法。

在RoboCup 2D中，服务器和球员都是按照每100ms的周期离散处理的[1]。球员在一个周期选择一个高层动作并最终执行后，并不能立即得到下一个周期的状态，而是要等到下一个周期开始；同样，等到下一次进行迁移学习的动作选择时才能得到对应上次迁移学习所选动作的回报值[2]。所以本文在应用PR-Sarsa算法时，将迁移学习过程分为“训练段开始”、“训练段中”和“训练段结束”3个阶段，如图2所示。“训练段开始”只进行动作的选择，“训练段中”进行动作选择并更新上一个动作的Q值，“训练段结束”只进行上一个动作的Q值更新。

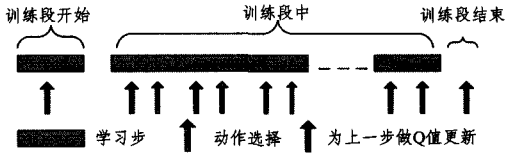


图2 5v4 Keepaway中抢球球员的迁移学习过程

抢球球员的强化学习PR-Sarsa算法的伪代码如下：

Step 1 初始化Q值表，使表中的所有值为接近0的随机值；初始化权重值表，$W_\Omega = W_{\Pi_{4v3}} = 0$，$\Omega$ 代表当前学习的策略；初始化训练段的编号 k 为0。

Step 2 训练段的编号加1，s 初始化为当前环境状态。

Step 3 依照当前权重表和soft-max概率选择公式 $P(\Pi_j) = \frac{e^{rW_j}}{\sum_{p=0}^{1} e^{rW_p}}$ 选出当前任务段 k 将要使用的策略 Π_k，如果 Π_k 是 Π_Ω，则进行正常的强化学习，否则重用策略 Π_k，选择动作 a。

Step 4 执行动作 a，观察回报值 r 和新的环境状态 s'。

Step 5 当前权重表为依据随机选择一个策略 Π_k，如果 Π_k 是 Π_Ω，则进行正常的强化学习，否则重用策略 Π_k，选择动作 a'。

Step 6 更新Q值：$Q_t(s,a) := (1-\alpha)Q_{t-1}(s,a) + \alpha[r + \gamma Q_{t-1}(s',a')]$。根据 r 更新权重值表。

Step 7 将 s 更新为 s'，a 更新为 a'。

Step 8 如 s 是任务结束状态，转到Step 2；否则转到Step 5。

Step 1是初始化操作。Step 3首先依据权值表 W_0、W_1 和soft-max概率选择公式，概率性地进行当前策略的学习和旧策略的重用，并做出动作决策。Step 4执行动作，观察新的环境状态。Step 5为新的状态选择动作，暂不执行动作。Step 6为上一步更新Q值表，并且根据回报值 r 更新权值表 W_0、W_1。Step 7进入下一周期，更新环境状态。Step 8根据当前环境状态决定继续当前训练段还是开始新的训练段。

5 实验分析

5.1 实验设置

为了分析基于策略重用的迁移学习的训练效果，实验对象采用最典型的30m×30m场地下的5v4规模Keepaway任务。本文让5v4任务中高层抢球策略的迁移学习与普通强化学习过程进行比较。实验分为两步：第一步，进行4v3任务中高层抢球策略的普通强化学习；第二步，利用第一步学到的策略 Π_{4v3} 进行5v4任务中基于策略重用的高层抢球策略的迁移学习。

实验环境为Ubuntu Linux3.5.0-17-generic，Intel x86_32 3.20GHz，2.00GB RAM。设定场地大小设定为30m×30m。

服务器设置开启 360 度球员视角和无噪声视觉模式。在抢球球员的强化学习过程中，根据实验优化，设定折扣因子 γ 为 1.0，ε-greedy 动作选择策略的参数 ε 为 0.01。为了进一步验证不同学习率下的强化学习的训练效果和收敛性，分别选取 0.125、0.250、0.375 的学习率进行实验。

5.2 实验结果和分析

图 3 和图 4 分别展示了 5v4 规模的 Keepaway 中任务完成时间和给定周期内的抢球成功率随着训练时间变化的情况；图 5 和图 6 分别展示了 5v4 规模 Keepaway 手工策略、学习率为0.125的普通强化学习和迁移学习的任务完成时间和抢断成功率随着训练时间变化的情况。

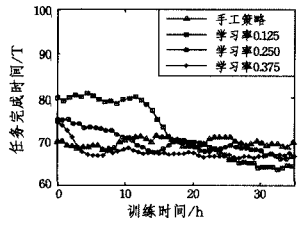


图 3 强化学习的任务完成时间

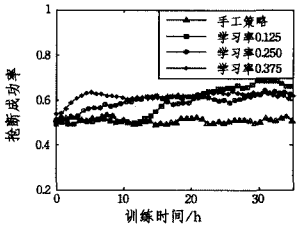


图 4 强化学习的抢断成功率

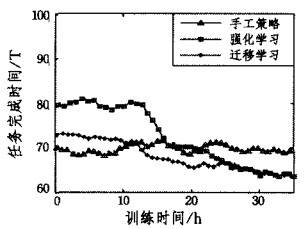


图 5 迁移学习任务完成时间

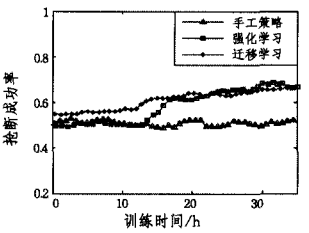


图 6 迁移学习抢断成功率

从图 3 看到，手工策略任务完成时间基本维持在 70 周期。手工策略下任务完成时间随着训练进行的变化不大，这是因为手工策略不具有学习和记忆能力，不能随着训练进行获取经验并提高决策。0.125 的学习率下，任务完成时间从 80 周期经学习最终稳定到 65 周期；0.250 的学习率下，任务完成时间从 75 周期经学习最终稳定到 68 周期；0.375 的学习率下，任务完成时间从 74 周期经学习最终稳定到 68 周期。以手工策略为基准，对于任务完成时间，0.125 的学习率下降 7.1%，0.250 的学习率下降 2.9%，0.375 的学习率下降 2.9%，可见 0.125 的学习率最好。从图 4 看到，在 65 周期的时间限制下，手工策略的抢断成功率为 52.0%；0.250 和 0.375的学习率的强化学习抢断成功率都提升到 63.0%左右，提升了 11.0%；0.125 的学习率的强化学习抢断成功率提升到 67.0%左右，提升了 15.0%，可见 0.125 的学习率最好。

图 5 为普通强化学习为达到较好的策略所需的训练时间，任务完成时间在开始的 13 个小时内维持在 80 个周期左右，到 13 小时后才出现显著下降；而迁移学习的任务完成时间在开始时就比普通强化学习少 8 个周期，相对少 8.8%，在学习 13 个小时后就已降低到 70 个周期，而普通强化学习完成任务时间降低到 70 个周期则需要 22 个小时。从图 6 看到，在训练开始的 18 个小时内，迁移学习的抢断成功率均比普通强化学习高 5.0%。可以看到，相对于普通强化学习，迁移学习在训练开始时就获得更优的决策，在相同训练时间的约束下，迁移学习比强化学习获得更优的高层策略，即达到同样训练效果所需的时间更短。这是因为迁移学习利用了有助于解决当前问题的相关策略，实现了经验的借鉴。

结束语 在 RoboCup 2D Keepaway 的高层抢球动作决策中，虽然普通强化学习方法最终能学得较优的策略，但为达到较好的策略需要花费大量的训练时间；并且在学习开始的很长一段时间内，抢球球员完成任务时间很长，抢断成功率低。针对普通强化学习的这一问题，本文进行了基于策略重用的迁移学习的研究，通过合理重用已经学得的 4v3 任务的策略，实现了 5v4 任务中的高层抢球策略的迁移学习。实验发现迁移学习在训练开始时就表现出较优的决策，并且比普通强化学习更快地收敛到理想的策略水平，可以缩短训练时间，表明了基于策略重用的迁移学习在高层抢球决策中的有效性。而如何优化策略重用时不同规模任务间状态空间和动作空间的映射，使智能体的动作选择更加合理，可以作为下一步的研究方向。

参 考 文 献

[1] Chen M, Klaus D, Ehsan F. User Manual: RoboCup Soccer Server Manual for Soccer Server Version 7.07 and Later[EB/OL]. http://sourceforge.net/projects/sserver/files

[2] Stone P, Kuhlmann G, Taylor M E, et al. Keepaway Soccer: from Machine Learning Testbed to Benchmark[M]. RoboCup 2005: Robot Soccer World Cup IX. Berlin: Springer Verlag, 2006: 93-105

[3] Stone P, Sutton R S, Kuhlmann G. Reinforcement Learning for RoboCup Soccer Keepaway [J]. Adaptive Behavior, 2005, 13(3): 165-188

[4] 左国玉，张红卫，韩光胜. 基于多智能体强化学习的新强化函数设计[J]. 控制工程，2009，16(2)：239-242

[5] Sutton R S, Barto A G. Reinforcement Learning: an Introduction [M]. Cambridge, MA: The MIT Press, 2012

[6] Taylor M, Stone P, Liu Y. Transfer Learning via Inter-task Mappings for Temporal Difference Learning [J]. Journal of Machine Learning Research, 2007, 8(1): 2125-2167

[7] Fernández F, García J, Veloso M. Probabilistic Policy Reuse for Inter-task Transfer Learning [J]. Robotics and Autonomous Systems, 2010, 58(7): 866-871

[8] Fernández F, Veloso M. Probabilistic Policy Reuse in a Reinforcement Learning Agent[C]// Nakashima H, Wellman M. AAMAS'06 Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-agent Systems. New York: ACM Press, 2006: 720-727

[9] Rummery G A, Niranjan M. On-Line Q-learning using Connectionist Systems[R]. Cambridge, England: Cambridge University Engineering Department, 1994

[10] Walsh T J, Li L, Littman M. Transferring State ions between MDPs[C]// Proceedings of the ICML'06 Workshop on Structural Knowledge Transfer for Machine Learning. 2006

[11] Taylor M E, Stone P. Behavior Transfer for Value-function-based Reinforcement Learning[C]// Pechoucek M. The Fourth International Joint Conference on Autonomous Agents and Multi-agent Systems. New York: ACM Press, 2005: 53-59

[12] Fernández F, Veloso M. Policy Reuse for Transfer Learning Across Tasks with Different State and Action Spaces[C]// ICML'06 Workshop on Structural Knowledge Transfer for Machine Learning. 2006

（下转第 225 页）

接度的排序更能在有限的列表长度内包含最终选择的站点。为此，基于案例测试了不同邻接表长度对最终求解结果的影响。测试时设定邻接表长度分别为站点数的15%、20%、30%、40%和100%，但至少保证不少于100个邻接站点。

实验结果如表4、表5所列。从表4可以看出，与不限定邻接表长度(100%)的情况相比，在40%、30%、20%的情况下，平均执行时间显著下降；但从20%向下再继续缩减邻接表长度，执行时间下降得并不明显，并且开始对求解结果产生影响(见表5)。因此根据实验结果，将邻接表长度限定在20%的范围内是比较合理的，既能保证求解质量，又能显著地减少求解时间。

表4 邻接表长度对求解时间的影响

案例	最大乘车时间/s	不同邻接表长度下多个案例的平均求解时间/s				
		15%	20%	30%	40%	100%
RSRB01-RSRB08	2700	1029.48	1050.26	1124.79	1205.99	2101.48
RSRB01-RSRB08	5400	936.36	995.09	1213.63	1405.72	2528.27
CSCB01-CSCB08	2700	1437.90	1464.81	1568.88	1656.23	2524.31
CSCB01-CSCB08	5400	1214.73	1274.39	1509.35	1748.30	3021.20

表5 邻接表长度对车辆数的影响

案例	最大乘车时间/s	不同邻接表长度下多个案例的平均车辆数				
		15%	20%	30%	40%	100%
RSRB01-RSRB08	2700	76.88	76.88	76.88	76.88	76.88
RSRB01-RSRB08	5400	64.63	64.75	64.63	64.63	64.63
CSCB01-CSCB08	2700	86.00	85.75	85.75	85.75	85.75
CSCB01-CSCB08	5400	68.75	69.00	69.00	69.13	69.13

表5中，当邻接表长度从100%下降到30%时车辆数都没有增加，甚至在CSCB(5400)的情况下还有下降。再继续缩小邻接表长度到20%时，RSRB(5400)出现了偶然性的增加。当邻接表长度下降到15%时，平均车辆数变化不大，但是具体到案例，车辆数有增有减，对结果产生了一定的影响。这验证了基于时空邻接度排序邻接表能够在有限的长度下，尽可能地包含更多最终真正执行移动的站点，但邻接表过短时，它会限制移动站点的选择。通过对邻域搜索过程的监视，发现个别案例出现邻接表缩短时车辆数更少的情况，这主要是因为限定了一些站点之后，导致邻域搜索的轨迹发生了变化，出现了更有利于缩减路径数的特例。

结束语 与传统构造式启发算法相比，基于邻域算子优化混载SBRP能显著提高解的质量，但大规模的邻域搜索增加了问题的求解时间。为解决邻域搜索耗时的问题，本文设计了一种基于时空相关的邻域搜索算法。该算法基于站点的邻接表构造邻域搜索空间，综合考虑了站点间的空间距离、时间距离，同时进行了简单约束的预处理。基于时空相关度对邻接表排序使得最终接受的邻接解通常位于搜索空间中比较靠前的位置，因此适当限定邻接表长度并不会对求解质量产生大的影响。实验表明，将邻接表长度限定为站点数20%(同时不小于100)的情况下，能在基本保证求解质量的同时节省50%以上的求解时间。

参考文献

[1] Newton R M, Thomas W H. Design of school bus routes by computer[J]. Socio-Economic Planning Sciences, 1969, 3(1): 75-85

[2] Bodin L D, Berman L. Routing and scheduling of school buses by computer[J]. Transportation Science, 1979, 13(2): 113-129

[3] Park J, Kim B-I. The school bus routing problem: A review[J]. European Journal of Operational Research, 2010, 202(2): 311-319

[4] Bodin L D, Golden B, Assad A, et al. Routing and scheduling of vehicles and crews: the state of the art[J]. Computers and Operations Research, 1983, 10(2): 63-211

[5] Braca J, Bramel J, Posner B, et al. A computerized approach to the New York City school bus routing problem[J]. IIE Transactions, 1997, 29(8): 693-702

[6] Vidal d S L, Siqueira P H. Heuristic Methods Applied to the Optimization School Bus Transportation Routes-A Real Case[C]// IEA/AIE 2010, Part II. Berlin: Springer Verlag, 2010: 247-256

[7] Park J, Tae H, Kim B-I. A post-improvement procedure for the mixed load school bus routing problem[J]. European Journal of Operational Research, 2012, 217(1): 204-213

[8] Nanry W, Barnes J. Solving the pickup and delivery problem with time windows using reactive tabu search[J]. Transportation Research Part B, 2000, 34(2): 107-21

[9] 党兰学，王震，刘青松，等. 一种求解混载校车路径的启发式算法[J]. 计算机科学，2013，40(7)：248-253

[10] Fang H, Kilani Y, Lee J H M, et al. Reducing search space in local search for constraint satisfaction[C]// AAAI/IAAI. 2002: 28-33

[11] Chen S Y, Smith S F. Improving Genetic Algorithms by Search Space Reductions (with Applications to Flow Shop Scheduling)[C]//GECCO. 1999: 135-140

[12] 戚铭尧，丁国祥，周游，等. 一种基于时空距离的带时间窗车辆路径问题算法[J]. 交通运输系统工程与信息，2011，11(1)：85-89

(上接第193页)

[13] Riedmiller M, Gabel T, Hafner R. Reinforcement Learning for Robot Soccer[J]. Autonomous Robots, 2009, 27(1): 55-73

[14] Gabel T, Riedmiller M. On Progress in RoboCup: the Simulation League Showcase in RoboCup 2010: Robot Soccer World Cup XIV[M]. Berlin: Springer Verlag, 2011: 36-47

[15] Kalyanakrishnan S, Stone P. Characterizing Reinforcement Learning Methods through Parameterized Learning Problems[J]. Machine Learning, 2011, 84(1/2): 205-247

[16] Sherstov A A, Stone P. Function Approximation via Tile Coding: Automating Parameter Choice in Abstaction, Reformulation and Approximation [M]. Berlin: Springer Verlag, 2005: 194-205

[17] Stone P, Sutton R S. Scaling Reinforcement Learning toward RoboCup Soccer[C]// the Eighteenth International Conference on Machine Learning. Massachusetts: Williamstown, 2001: 537-544

[18] 程显毅，朱倩. 一种改进的强化学习方法在RoboCup中的应用[J]. 广西师范大学学报：自然科学版，2010，28(3)：99-102

[19] 刘春阳，谭应清，柳长安. 多智能体强化学习在足球机器人中的研究与应用[J]. 电子学报，2010，38(8)：1958-1962

[20] 李瑾，刘全，杨旭东. 一种改进的平均奖赏强化学习方法在RoboCup训练中的应用[J]. 苏州大学学报，2012，28(2)：21-26