结合进化算法的深度强化学习方法研究综述

吕 帅[1),2)] 龚晓宇[1),2)] 张正昊[1),2)] 韩 帅[1),2),3)] 张峻伟[1),2)]

[1)](吉林大学计算机科学与技术学院 长春 130012)

[2)](符号计算与知识工程教育部重点实验室(吉林大学) 长春 130012)

[3)](Department of Information and Computing Sciences, Utrecht University, Utrecht 3584 CC, The Netherlands)

摘 要 深度强化学习是目前机器学习领域中重要的研究分支之一,它可以通过直接与环境进行交互实现端到端的学习,对高维度和大规模的问题有着很好的解决能力.虽然深度强化学习已经取得了瞩目的成果,但其仍面临着对环境探索能力不足、鲁棒性差、容易受到由欺骗性奖励导致的欺骗性梯度影响等问题.进化算法普遍具有较好的全局搜索能力、良好的鲁棒性和并行性等优点,因此将进化算法与深度强化学习结合用于弥补深度强化学习不足的方法成为了当前研究的热点.该文主要关注进化算法在无模型的深度强化学习方法中的应用,首先简单介绍了进化算法和强化学习基本方法,之后详细阐述了两类结合进化算法的强化学习方法,分别是进化算法引导策略搜索的强化学习和结合进化算法的深度强化学习,同时对这些方法进行了对比与分析,最后对该领域的研究重点和发展趋势进行了探究.

关键词 强化学习;深度强化学习;进化算法;遗传算法;进化策略

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2022.01478

Survey of Deep Reinforcement Learning Methods with Evolutionary Algorithms

LÜ Shuai[1),2)] GONG Xiao-Yu[1),2)] ZHANG Zheng-Hao[1),2)] HAN Shuai[1),2),3)] ZHANG Jun-Wei[1),2)]

[1)](*College of Computer Science and Technology, Jilin University, Changchun* 130012)

[2)](*Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun* 130012)

[3)](*Department of Information and Computing Sciences, Utrecht University, Utrecht* 3584 *CC, The Netherlands*)

Abstract Deep reinforcement learning is one of the most important branches in the field of machine learning, which can achieve end-to-end learning through direct interaction with the environment and is capable of solving high-dimensional and large-scale problems. Although deep reinforcement learning has achieved remarkable results, it still faces problems such as insufficient exploration of the environment, poor robustness, and susceptibility of gradients caused by deceptive rewards. In general, evolutionary algorithms have good global search ability, robustness, parallelism and other advantages. Therefore, the methods combining evolutionary algorithms with deep reinforcement learning to compensate the inadequacy of deep reinforcement learning methods have become a research hotspot recently. This paper focuses on the applications of evolutionary algorithms in model-free deep reinforcement learning methods. We introduce evolutionary algorithms and basic methods of reinforcement learning firstly. After that, we introduce the characteristics, advantages, disadvantages, and applicable tasks of evolutionary algorithms, deep reinforcement learning algorithms, and combined methods of evolutionary algorithms and deep reinforcement learning, showing the necessity of combined methods from a different aspect.

收稿日期:2021-07-05;在线发布日期:2022-01-19.本课题得到国家重点研发计划(2017YFB1003103)、国家自然科学基金(61763003)、吉林省自然科学基金(20180101053JC)资助.**吕 帅**,博士,副教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为人工智能、机器学习与自动推理.E-mail:lus@jlu.edu.cn.**龚晓宇**,硕士研究生,中国计算机学会(CCF)会员,主要研究领域为人工智能、机器学习.**张正昊**,硕士研究生,主要研究领域为人工智能、机器学习.**韩 帅**,博士研究生,主要研究领域为人工智能、机器学习.**张峻伟**,硕士研究生,主要研究领域为人工智能、机器学习.

Then, two types of reinforcement learning methods with evolutionary algorithms are elaborated, which are reinforcement learning with evolutionary algorithms guided policy search and combination of evolutionary algorithms and deep reinforcement learning. In reinforcement learning with evolutionary algorithms guided policy search methods, we categorize the different policy search methods into parameter distribution search methods, policy gradient approximation methods, and policy population search methods. Parameter distribution search methods regard the parameters of a policy as a distribution and sample the parameters from this distribution to form a new policy. Policy gradient approximation methods use the fitness of the policy as an approximation of the gradient to update the parameters. Policy population search methods search directly from individuals in the policy population and select the individual with higher fitness. Then, we focus on the combined methods of evolutionary algorithms and deep reinforcement learning which attracts the interest of scholars currently, including evolutionary algorithm experience-guided deep reinforcement learning methods and evolutionary algorithm modules-embedded deep reinforcement learning methods. The evolutionary algorithm experience-guided deep reinforcement learning methods use experience obtained from individuals by continually interacting with the environment to guide the value network of reinforcement learning, while the evolutionary algorithm module-embedded deep reinforcement learning methods embed the evolutionary algorithm as an auxiliary module in the learning process of reinforcement learning. Furthermore, we compare and analyze these methods in detail. In particular, we compare the characteristics of various algorithms in the methods combing evolutionary algorithms and deep reinforcement learning, including without-feedback guidance methods and with-feedback guidance methods. We also compare the performance of various widely-used algorithms in with-feedback guidance methods on the continuous control tasks of MuJoCo and give a detailed analysis and future directions for improvement and research. Finally, we summarize all the combined methods of evolutionary algorithms and deep reinforcement learning mentioned in the paper, and we study the research emphasis and development trend of this field. Although evolutionary deep reinforcement learning frameworks have been proposed, we think these methods still require further theoretical study to balance the issues of exploration and exploitation.

Keywords　reinforcement learning; deep reinforcement learning; evolutionary algorithms; genetic algorithms; evolution strategies

1　引　言

长期以来，强化学习都是机器学习方法中不可或缺的一部分，在国际上也一直是机器学习领域中炙手可热的研究分支. 在强化学习中，智能体首先根据环境状态进行决策从而产生动作，之后通过产生的动作与环境进行交互获得强化信号，调整产生决策的函数映射，使得智能体能够选择获得环境最大奖励的决策方案. 智能体经过长期与环境的交互，不断向累积回报最大的方向优化策略，最终使累积回报尽可能地最大化.

2013 年，DeepMind 团队的 Mnih 等人首先将传统强化学习中的 Q-learning 算法[1]与深度神经网络相结合，并提出了深度 Q 网络(Deep Q-Network，DQN)算法[2-3]，使用 DQN 算法训练的智能体在 Atari 游戏中取得了超过人类得分的惊人表现. 这一成果开拓了深度强化学习(Deep Reinforcement Learning，DRL)这一新的方向，并成为了当今人工智能领域新的研究热点. 深度强化学习是一种端到端的学习方法，它不需要标记的数据作为输入，而是通过与环境进行交互获取原始输入信息，从而学习动作策略，通过不断的试错形成具有强大学习能力的智能体[4]. 2016 年，DeepMind 团队使用深度强化学习训练的 AlphaGo 智能体[5]击败了人类最顶尖的围棋选手，是机器学习领域的重大标志性事件，使得深度

强化学习成为研究者们关注的焦点.目前深度强化学习在机器博弈[5-7]、机器人控制[8]、自然语言处理[9]、最优控制[10]和计算机视觉[11]等领域中取得了广泛的应用,被认为是通向通用人工智能的重要方法之一[12].

虽然深度强化学习目前取得了瞩目的成果,但其依旧存在着很多问题需要解决.首先,在深度强化学习中需要平衡对环境的探索和对环境反馈信息的利用,目前研究者们已经围绕这一问题提出了基于状态伪计数[13]、好奇心预测模型[14]、噪声网络[15]等一系列增强深度强化学习探索能力的改进方法,但这些方法能解决探索问题的能力较为有限且部分方法开销较大.其次,一直被学者们诟病的深度强化学习的稳定性和鲁棒性问题也十分突出[16],例如深度确定性策略梯度(Deep Deterministic Policy Gradient,DDPG)算法[17]对超参数非常敏感,超参数必须固定在一个很小的范围内才能有较好的表现,虽然目前置信域策略优化(Trust Region Policy Optimization,TRPO)[18]、近端策略优化(Proximal Policy Optimization,PPO)[19]和双延迟深度确定性策略梯度(Twin Delayed Deep Deterministic Policy Gradient,TD3)[20]等方法较 DDPG 更为稳定,但在实际应用过程中也需要较为细致的超参数调整.再次,很多深度强化学习算法甚至在不同的随机数种子下运行的结果也相差甚远,这对深度强化学习在现实世界中的应用造成了巨大的阻碍.深度强化学习还有容易陷入局部最优解、容易对环境过拟合以及无法保证收敛等问题[21-23].

进化算法(Evolutionary Algorithm,EA)是模拟生物进化机制的一类搜索算法,它具有很好的全局搜索能力,不容易陷入局部最优解,同时具有良好的并行性.此外,进化算法具有很强的通用性,应用范围广泛,几乎所有科学和工程领域中都有所涉及,如非线性优化、机器人控制、生产调度、数据挖掘、计算社会科学等[24].进化算法(也称演化算法)作为一大类启发式无梯度优化算法,受优化问题性质约束极少,只需能够评估解的好坏即可运行求解,适用于求解复杂的优化问题[25].因此,研究者们已经将进化算法的优势应用到机器学习中,比如将进化算法用于神经网络架构的搜索[26]和深度学习超参数的选取[27]就是当前研究的热点问题.在强化学习中,也可以使用进化算法选择其超参数或优势个体,目前也已经发展成为了较成熟的框架[28-29].

表 1 总结了进化算法、强化学习算法以及结合进化算法的强化学习算法的特点、优缺点和适用的任务.虽然进化算法本身已经可以解决部分简单的强化学习问题,但其效率和泛化能力较差.为了利用进化算法的优点来弥补深度强化学习中的不足,结合进化算法与深度强化学习的方法应运而生.其既可以在动作空间进行探索,又可以在参数空间进行探索,在防止算法的过早收敛的同时使得算法比大多数深度强化学习的效果更好,上述众多优势使得结合进化算法的深度强化学习成为当前的研究热点.

表 1 进化算法、强化学习算法和结合进化算法的强化学习算法对比

方法	特点及优势	缺点	适用的任务
进化算法	不需要进行反向传播,高度可并行,适用范围广,对环境奖励设定要求较低,在参数空间进行探索,鲁棒性高	样本效率低,探索方式单一,没有学习以及泛化能力	广泛的全局优化任务
深度强化学习算法	可以通过学习和泛化得到复杂的行为,在动作空间中进行探索	需要进行反向传播,并行能力较差,存在置信分配、欺骗性奖励、稀疏奖励的问题,算法鲁棒性较差,容易过早收敛	符合 Markov 性的决策任务
结合进化算法的深度强化学习算法	既可以在动作空间进行探索,又可以在参数空间进行探索,可以在一定程度上防止算法过早收敛	相对深度强化学习样本效率较低,但随着训练步数的增加,效果好于一般的强化学习算法;算法所占用的资源较大	强化学习算法能够适用的各种任务;相对一般的深度强化学习更擅长欺骗性奖励、稀疏奖励任务

本文按照进化算法与强化学习结合的不同方式将结合进化算法的强化学习方法分为了进化算法引导策略搜索的强化学习和结合进化算法的深度强化学习两类,如图 1.在进化算法引导策略搜索的强化学习方法中,本文按照不同的策略搜索方法将其又分为了参数分布搜索方法、策略梯度近似方法和策略种群搜索方法.需要说明的是,虽然梯度近似方法并没有与强化学习算法进行结合,但其是此领域发展过程中较为重要的一类方法,此类方法不仅能够替代强化学习解决决策问题,还能够达到与深度强化学习相同的表现,同时也为之后将进化算法与深度强化学习进行结合提供了思路,因此本文将这类方法视为结合进化算法与强化学习思想的方法.最后本文将目前学者们最为关心与最为重视的与深度强化学习结合的方法进行了详细的归纳,由于此类方法多是直接使用了当前较为成熟的深度强化学习方法与框架,因此本文按照两种算法不同的耦合方式将此类方法又分为了经验指导方式与模块嵌入方

式，在前者中进化算法与深度强化学习是以并列的方式进行组合，并通过经验池和梯度信息相互联系，而后者则是将进化算法作为一个模块嵌入到深度强化学习的过程之中，从而对深度强化学习进行辅助.这两者最为明显的区别是，在进化算法部分，前者会以回合为单位进行评估，并直接使用累积回报作为进化算法的适应度函数，而后者则以时间步为单位进行评估，其适应度函数不能直接使用累积回报，通常需要根据嵌入的位置与方式进行特定的设计.

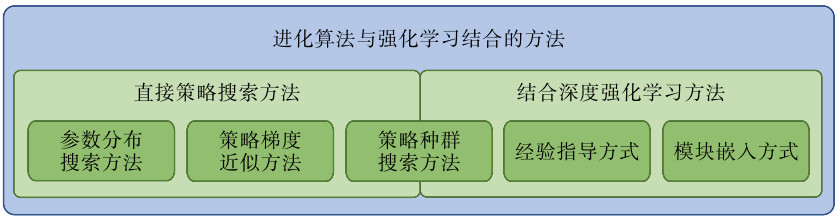


图 1　进化算法与强化学习结合的方法分类

本文将上述两类结合进化算法的强化学习方法分为 3 节进行综合评述，详细介绍了结合进化算法的强化学习的主要方法以及研究进展情况.在组织结构上，首先在第 2、3 节简要介绍了进化算法和强化学习的相关概念和主要方法，之后在第 4 节介绍了进化算法引导策略搜索的强化学习，第 5 节介绍了进化算法经验指导的深度强化学习方法并对它们进行了详细的对比，第 6 节介绍了进化算法模块嵌入的深度强化学习方法.最后，本文对上述方法进行了总结并阐述了该领域中存在的问题与未来的发展趋势.

2　进化算法

进化算法的本质是一种全局搜索算法，其中主要的进化思想来自于生物进化理论.进化算法为复杂的优化问题提供了一种通用的求解框架，如算法 1 所示.在求解的过程中，进化算法首先随机生成一个种群，并对种群中的个体进行适应度计算，之后通过选择操作选择出适应度高的个体作为父代个体，然后对其进行交叉和变异等操作并生成一个新的种群（即一组新的近似解），最后通过循环上述过程，不断逼近问题近似最优解.

当进化算法结合到深度强化学习算法之中时，进化算法中的个体通常作为深度强化学习中的策略网络，种群则是由多个策略网络所组成的网络集合，而进化算法中对个体的交叉与变异操作则对应策略网络参数的交叉与变异.在深度强化学习中评估某个个体的策略网络的方法是让其与环境进行交互，并将交互一回合所获得的累计回报看作此个体对环境的适应度.通过上述的映射与对应关系，进化算法可以很好地结合到深度强化学习算法中，并通过进化算法的流程逼近强化学习问题的近似最优解.

算法 1.　进化算法基本框架.

初始化：种群 P，其中包含 N 个个体

REPEAT

　估种群 P 中每个个体的适应度

　根据适应度从 P 中以一定方式选择出 n 个个体集合 P'

　对 P' 中的个体通过交叉和变异产生新个体从而形成新种群 P''

　$P \leftarrow P''$

UNTIL 终止条件

RETURN 种群中的最优个体

由于进化算法是一种无梯度算法，因此不会像基于梯度下降的算法一样容易陷入局部最优解.因为进化算法通常采用了多个个体进行搜索，即使单个个体遇到了局部最优解，算法依然会保留其它个体用于全局搜索，经过多次的选择和变异后，算法就可能跳出局部最优解.此外，算法的操纵对象是整个种群而非单个个体，每个个体是相对独立的，且不需要类似于梯度下降算法将梯度信息一层层的向前反向传播，这使得进化算法的可并行性非常好.进化算法也存在一些问题，如对于变异算子和适应度的设计有时较为困难、容易过早收敛到局部最优解、对于大规模问题开销较大且样本效率低下.

进化算法实际上是一类算法的统称，其中包括最早出现的遗传算法（Genetic Algorithm，GA）[30]、进化策略（Evolutionary Strategy，ES）[31]和进化规划（Evolutionary Programming，EP）[32]三大类.后来又出现了包括遗传规划（Genetic Programming，GP）[33]、差分进化算法（Differential Evolution，DE）[34]、分布估计算法（Estimation of Distribution Algorithm，EDA）[35]等一系列更为先进的进化算法.虽然当今对进化算法的分类和不同算法之间的定义存在较多的争议，但这些算法都具有共同的特

性且本质都是基于进化思想的.本文选择了进化算法中较为典型且与深度强化学习关系紧密的遗传算法、进化策略与分布式估计算法进行了更为详细的介绍,并对它们在深度强化学习下的效果与应用进行了讨论.

2.1 遗传算法

遗传算法被认为是最接近生物进化原理的一种进化算法,同时也是最早提出的一种进化算法.在遗传算法中使用了与遗传学中相同的术语.其中,种群表示可行解的一个集合,个体表示某一个可行解,每个个体被表示为一个变量序列,这个变量序列被称为染色体.染色体通常由一组简单的数字或字符串表示,可行解描述为染色体的过程被称为遗传编码,其中可行解编码的分量被称为基因,最常见的编码形式为二进制编码,即使用 0 与 1 的序列来描述可行解.

遗传算法的基本流程与进化算法的基本框架基本相同,主要包括选择、交叉、变异三种算子,算子分类如图 2 所示,其中:

选择算子会根据个体的适应度按照一定的规则选择一部分个体遗传到下一代种群,通常使用的选择算子都是基于比例的选择,如轮盘法(Roulette Wheel Selection),即个体被保留的概率与其适应度大小成正比,个体的适应度越大越有可能被保留下来.根据不同的问题,常用的选择算子还有竞争法(Tournament Selection)和等级轮盘法(Rank-based Wheel Selection).

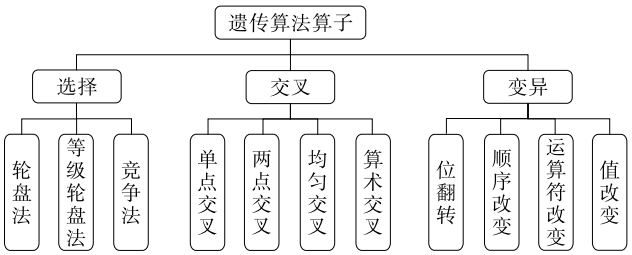


图 2 遗传算法算子分类

交叉算子根据编码方式的不同有多种不同的交叉方式,但通常都会选择种群中的两个个体作为操作对象,对它们的染色体进行对等交换或者异位交叉从而形成一个新的个体.

变异算子是针对某一个体,将其染色体上的某一基因按照一定的概率改变为其它的基因值,例如以二进制编码的基因的变异方式是简单地以一定概率将 0 反转为 1 或将 1 翻转为 0.

遗传算法的效果取决于编码方式、适应度函数、变异率等因素,因此在合适的问题上对遗传算法进行特定的设计是十分重要的.在将遗传算法与深度强化学习算法结合时,遗传算法通常以智能体神经网络的全部参数作为个体的染色体,并通常使用累计奖励作为环境适应度,而变异操作一般通过对神经网络参数加入高斯噪声实现.

2.2 进化策略

进化策略是一种模仿生物进化的黑盒参数优化方法,它使用特定的策略引导变异算子作用在决策变量上.进化策略的算法思想与遗传算法相同,但其不同的是进化策略采用实数值作为基因,并且其变异通过在实数值上添加一个高斯分布来进行.在进化策略中可以将高斯分布看作一种策略,它根据种群的适应度不断更新,并最终使个体趋于最优解,因此进化策略中的个体是按照确定的策略进行进化的,而遗传算法和进化规划则是以一种更为随机的方式进行.此外,在进化策略中取代遗传算法中交叉算子的是重组算子,它与交叉算子类似,也是利用两个父代个体的信息生成子代个体,但与交叉算子不同的是,重组算子不会保留任意一个父代个体的信息,而是每个基因都发生结合.

以标准的简单高斯进化策略为例,进化策略的具体做法是首先均匀的初始化一组个体作为初始种群,经过个体的重组和在个体参数上增加一个高斯分布的方式来进行个体的变异,评估这些个体的适应度,并选择适应度最高的前 μ 个个体作为子代个体.高斯分布更新的步长可以设定为固定值或以自适应的方式实现,其中自适应协方差进化策略(Covariance Matrix Adaptation Evolution Strategies,CMA-ES)[36]是最常见的动态步长更新方法,CMA-ES 使用了高斯分布中的协方差矩阵来动态地调节更新步长,同时它可以自适应地更新协方差矩阵.

在进化策略算法中,由于 CMA-ES 具有突出的表现,它被广泛运用在非线性优化、非凸优化以及连续优化问题上,并且也经常与监督学习进行结合用于解决强化学习问题.

2.3 分布估计算法

分布估计算法与进化策略类似,不同的是它通过一个概率模型来表示优势个体的分布,而不是像进化策略一样使用高斯分布描述当前种群向下一个种群移动的方向和步长.分布估计算法通常使用协方差矩阵定义一个多变量高斯分布作为其概率模型,在进行种群进化时,分布估计算法首先会对种群中的个体进行适应度评估,并选择出适应度最

高的前 n 个个体作为优势个体；之后利用这些优势个体的参数计算协方差矩阵，从而建立多变量高斯分布模型，并通过这一模型刻画优势个体的参数分布情况，从而表示这些优势个体；最后分布估计算法会在此分布中进行随机采样，并将采样出的个体作为新种群. 常见的分布式估计算法有PBIL(Population Based Incremental Learning)[37]、cGA(compact GA)[38]、BOA(Bayesian Optimization Algorithm)[39]等，与强化学习结合紧密的分布式估计算法有交叉熵方法(Cross-entropy Methods，CEM)[40-41]与PI2-CMA[42]等. 虽然前文将 CMA-ES 归为进化策略方法，但其也可以看作一种分布式估计算法，它与 CEM 的主要区别在于其使用了进化轨迹(Evolutionary Paths)[36]累积历代的搜索方向，而与 CEM 在更新协方差矩阵时只有细微的区别，前者使用新计算出的均值更新协方差矩阵，而后者使用当前种群的均值更新协方差矩阵[48].

分布估计算法仍然保留了进化算法的通用性，它同样可以简单地与其他算法进行结合构成新的算法，已有学者将分布估计算法与粒子群算法、模拟退火算法等进行了结合[44]，同时分布估计算法也可以用于直接解决一部分较为简单的强化学习问题[45-46]，因此将分布估计算法用于强化学习任务的指导与辅助上也成为了一个研究方向.

3　强化学习

强化学习是一种通过与环境交互并根据环境给出的状态和奖励进行决策的学习方法，经过数十年的发展，当今深度强化学习方法已经能够解决很多使用传统方法无法解决的问题[47]. 强化学习基于是否对环境进行建模可以分为无模型的强化学习(Model-free Reinforcement Learning)与基于模型的强化学习(Model-based Reinforcement Learning)两大类，本文主要针对无模型的强化学习进行讨论. 本节首先介绍强化学习的基本流程和符号定义，之后对无模型的强化学习中基于价值的方法和基于策略的方法的基本原理分别进行简单的介绍.

考虑一个标准的强化学习过程，在每一个时间步 t，智能体根据当前的环境状态 s_t，通过策略 π 选择出一个动作 a_t，在执行此动作后，环境返回下一个环境状态 s_{t+1} 和这一步的立即奖励 r_t，如图 3 所示[48]. 上述过程不断的重复进行，直到智能体达到最终状态或设定的最大步数后重置环境. 第 t 个时间步的 $\gamma\in[0,1]$ 折扣总回报为 $R_t=\sum_{k=0}^{\infty}\gamma^k r_{t+k}$，而智能体的目标就是最大化每一个状态 s_t 的期望回报，从而使整个过程的总回报期望最大化.

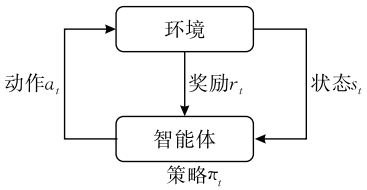


图 3　强化学习算法模型[48]

3.1　基于值函数的方法

在基于值函数的强化学习方法中，智能体会对环境给出的每个状态进行评估并根据状态的价值做出动作选择，经过多次的价值更新后智能体最终可以得到一个最优的与环境交互的决策策略.

具体来说，智能体处于状态 s 下且根据其策略 π 来决定后续动作时的折扣期望回报可以用价值函数 $V^{\pi}(s)$ 来表示：

$$V^{\pi}(s)=\mathbb{E}[R_t\mid s_t=s] \tag{1}$$

而动作价值函数 $Q^{\pi}(s,a)$ 是指在当前环境 s 下选择动作 a，即智能体处于状态动作对 (s,a) 时，后续动作根据智能体当前的策略 π 继续执行得到的期望回报：

$$Q^{\pi}(s,a)=\mathbb{E}[R_t\mid s_t=s,a_t=a] \tag{2}$$

以 Q-learning 为代表的基于值函数的强化学习方法会根据时间差分(TD-error)对 Q 函数在每次迭代后进行更新，其更新方式如下：

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha(r+\gamma\max_{a}Q(s_{t+1},a_{t+1})-Q(s_t,a_t)) \tag{3}$$

其中：α 为学习率. 经过上述公式的不断迭代与更新，在 Robbins-Monro 随机收敛条件下[49]，Q 函数最终会收敛于最优动作价值函数 $Q^{*}(s,a)=\max_{\pi}Q^{\pi}(s,a)$.

在基于值函数的深度强化学习中会使用神经网络表示 Q 函数并且根据时间差分对网络进行更新，其 Q 函数可以表示为 $Q^{\pi}(s,a;\theta)$，其中 θ 为神经网络的参数. 以 DQN 为代表，基于值函数的深度强化学习方法还有 DDQN[50] 和 Dueling DQN[51] 等.

3.2　基于策略梯度的方法

基于策略梯度的强化学习比基于值函数的强化学习方法更为直接，它不需要借助价值函数对状态价值进行估计来确定动作，而是通过直接对策略进行建模并使用策略直接决定动作，最后通过总回报对策略进行更新，其更新方式如下：

$$\theta\leftarrow\theta+\alpha\nabla_{\theta}\log\pi_{\theta}(a_t\mid s_t)R_t \tag{4}$$

其中，θ 为策略网络的参数. 基于策略梯度的强化学

习通常需要在每一回合结束后，通过蒙特卡罗方法计算该回合每一个时间步的折扣累积回报，之后根据回合轨迹的累积回报更新其策略，但此方法计算的折扣累计回报方差较大. 因此，为了降低策略梯度方法的方差，Williams等人[52]在此基础上进行了改进，将累计回报项 R 减去基线 b，提出的著名的 REINFORCE 算法，其参数更新方式为

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_{i,t} \mid s_{i,t})\right)\left(\sum_{t=1}^{T}R_t - b\right) \tag{5}$$

其中：b 为基线，θ 为策略 π 的参数，N 为采样个数，T 为回合长度.

4 进化算法引导策略搜索的强化学习

进化算法的本质是对解空间进行搜索从而得到近似的最优解，因此对于强化学习问题，可以将智能体的策略看作解空间，并将最优策略看作最优解，之后采取类似进化算法的搜索方法进行策略搜索. 分布估计算法[45-46]、粒子群算法[53]和模拟退火算法[54]等很多进化算法[55-56]均可以直接用于解决强化学习问题的工作，但其能解决的问题和任务较为有限且与强化学习算法并没有产生直接或间接的关联. 本节主要讨论与强化学习算法有一定关联的进化算法引导策略搜索的强化学习，并将这类方法按照策略搜索方式的不同分为参数分布搜索方法和策略梯度近似方法，并将这些方法进行了详细的对比与分析.

4.1 参数分布搜索方法

解决强化学习问题的方法通常有两大类，这两类方法均将强化学习问题构建为一个 Markov 决策过程(Markov Decision Process, MDP)，一类是对每个状态使用价值函数进行评估通过迭代不断逼近最优价值函数，即基于值函数的强化学习. 对于这类算法可以使用进化算法对其价值网络进行近似或对其结构进行搜索[57-58]，但通常只能解决维度较低的问题域. 而另一类是将强化学习看作一种策略搜索问题，通过各种黑盒优化方法来解决，即基于策略梯度的强化学习. 策略梯度方法的思想是增加累计奖励较高的策略所出现的概率，这与进化策略的思想类似. 我们可以将策略看作一个分布，通过将累积奖励作为适应度函数，对策略的参数空间进行搜索，从而获得更优的策略.

Williams 等人[52]在 REINFORCE 一文中也提到可以将 REINFORCE 方法使用在多参数的分布控制上，例如可以使用 REINFORCE 算法控制进化策略中高斯分布的均值与方差. 因此，Sehnke 等人[59]就运用这一想法提出了一种使用参数探索的策略梯度算法(Parameter-exploring Policy Gradients，PEPG)，这一算法将策略用高斯分布进行描述，并直接对策略的参数空间进行采样，最后使用与 REINFORCE 相同的方式计算梯度，但不同的是它使用这一梯度对策略分布的参数进行更新，如式(6)和式(7).

$$\nabla_\rho J(\rho) \approx \frac{1}{N}\sum_{i=1}^{N} f(h^i)\nabla_\rho \log p(\theta \mid \rho) \tag{6}$$

$$\nabla_{\mu_i} \log p(\theta \mid \rho) = \frac{(\theta_i - \mu_i)}{\sigma_i^2}$$

$$\nabla_{\sigma_i} \log p(\theta \mid \rho) = \frac{(\theta_i - \mu_i)^2 - \sigma_i^2}{\sigma_i^3} \tag{7}$$

其中：p 为高斯分布，ρ 为其参数，参数 ρ 包括均值 μ 与方差 σ，h 为智能体与环境交互的序列或轨迹，f 为适应度函数，在此处为累计回报函数，N 为采样个数，θ 为策略参数. 此外，Sehnke 等人[59]根据带有基线的采样方式又提出了一种基于对称采样的方法，解决了基线所带来的误差，提供了更精确的梯度估计.

如果说 PEPG 算法是进化策略运用在强化学习算法中的一次尝试，那么 Wierstra 等人[60]提出的自然进化策略(Natural Evolution Strategies，NES)则可以看作将策略梯度思想运用在进化策略算法的一种方法. 与 PEPG 类似，NES 使用概率分布来描述策略个体的参数分布，并将其作为种群，之后从这一分布采样得到个体，即策略参数 θ. NES 使用个体的适应度作为梯度更新的权重，并通过自然梯度来指引概率分布的参数向具有更高适应度的方向进行更新，如式(8)～式(10).

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)\nabla_\theta \log p_\theta(x_i) \tag{8}$$

$$\boldsymbol{F}_\theta \leftarrow \frac{1}{N}\sum_{i=1}^{N}\nabla_\theta \log p_\theta(x_i)\nabla_\theta \log p_\theta(x_i) \tag{9}$$

$$\theta \leftarrow \theta + \alpha \boldsymbol{F}^{-1}\nabla_\theta J \tag{10}$$

其中：x 为从种群概率分布 p 中采样得到的样本，$\boldsymbol{F}$ 为 Fisher 信息矩阵(Fisher Information Matrix)，N 为采样个数，θ 为策略参数. 此外 NES 还采用了基于等级的适应度重塑和自适应采样，这使得 NES 具

有了更强的鲁棒性.

当前大多数强化学习方法都需要利用 Markov 性(Markov Property)，即使用时间差分进行价值迭代. 虽然在智能体与环境交互的过程中，每一步都包含着大量的信息，但在深度强化学习中使用时间差分进行迭代来获得的最优策略不一定能使得智能体在整个回合获得最大的奖励，而且这类基于梯度的方法经常会使得智能体陷入局部最优解. 所以从这一角度来看，以进化策略为代表的进化算法在解决强化学习问题上具有一定的优势，从侧面证明了两者进行结合也是合理的. 而更重要的一点是，强化学习的梯度方向是固定的，缺乏随机性与探索度，这与其使用 Markov 性进行迭代有着密切的关系，虽然出现了很多增加强化学习探索度的方法，但并没有从根本上解决强化学习探索度低下的问题. 而进化算法只以最终的结果为标准进行参数空间上的探索，这为智能体带来了更多的可能性，也为强化学习方法带来了更大的探索度. 进化算法只考虑一个回合结束后的最终总回报而不考虑中间的过程，这与 REINFORCE 等策略梯度算法的思想具有一定的相似性，所以能很好地与类似的算法进行结合. 因此越来越多的学者将研究重心转移到将进化算法与策略梯度方法结合的方法上，这也为之后使用梯度近似的策略搜索的出现奠定了基础.

4.2　策略梯度近似方法

很长一段时间，由于计算能力低下的问题导致进化算法无法用于解决高维度与大规模的问题，而随着当前计算能力的不断提高，学者们开始重新考虑使用进化策略进行直接的策略梯度近似从而替代深度强化学习方法来解决强化学习问题，以 OpenAI-ES[61] 为代表的一些进化算法在解决强化学习问题上也可以与深度强化学习达到相同或者更好的效果. 此类方法可以看作使用进化算法进行参数搜索方法的进一步延伸，其利用强化学习的思想，并使用进化算法得到的适应度对策略梯度进行近似，正是在此类方法的影响下，学者们才在之后提出了进化算法与深度强化学习结合的方法.

OpenAI 的 Salimans 等人率先提出了一种使用进化策略代替深度强化学习的方法 OpenAI-ES，该方法与 NES 类似，但其利用了进化算法可并行的特点，使得进化策略第一次可以快速的解决强化学习领域中的 MuJoCo[62] 与 Atari 游戏，并且可以达到甚至超过深度强化学习的效果. 具体来说，OpenAI-ES 与其他进化策略算法相同，是一个无梯度的黑盒优化算法. 它将深度强化学习的策略直接描述为一个高斯分布，使用策略网络参数 θ 作为高斯分布的均值，并使用了一个固定的协方差矩阵 $\sigma^2 I$，最后使用回合的总回报作为参数更新的方向，其策略参数更新方式如式(11).

$$\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n\sigma} \sum_{i=1}^{n} F(\theta_t + \sigma\varepsilon_i)\varepsilon_i \tag{11}$$

其中：ε 为从标准高斯分布中采样得到的一个随机扰动，F 为适应度函数，在此处为累计回报函数，n 为采样样本个数. 这种方法与 NES 一样同样使用了 REINFORCE 的思想，但与之不同的是 OpenAI-ES 直接对策略分布参数按照总回报大小进行更新，而不是像 NES 一样使用梯度上升对参数进行搜索，这使得 OpenAI-ES 可以进行无梯度的学习. 此外，除了使用并行化技术，OpenAI-ES 还使用了虚拟批量标准化和镜像采样等方法提高了算法的鲁棒性.

此外，Salimans 等人[61] 还讨论了进化策略与传统策略梯度各自的优势和适用情景，他们基于两者探索度相同的角度思考了这一问题，认为在长时间步、某一时间步的动作具有深远影响和没有较好的奖励函数的情况下进化策略更具有优势. 实际上，进化策略由于是在参数范围上进行直接的探索，其探索效果通常要优于普通的强化学习算法. OpenAI-ES 可以看作结合进化算法与深度强化学习方法的开端，之后许多学者基于此文的思想提出了多种使用进化算法代替、指导与辅助深度强化学习的工作.

尽管 OpenAI-ES 使用了无梯度的学习方式，但是其方法也可以被看作一种基于梯度的学习，因为其本质是对梯度的一种估计. 因此，Such 等人[63] 进一步探究了完全无梯度的进化算法是否可以用于解决大规模的强化学习问题，他们的思路与 OpenAI-ES 类似，不同的是将进化策略替换为了简单的遗传算法，同时提出了一种引入新颖性搜索(Novelty Search，NS)[64] 的遗传算法. 从结果来看，遗传算法在强化学习任务上具有一定的速度和探索度优势. 这也进一步说明了以进化算法为代表的无梯度算法具有更好的探索能力，且在具有局部最优解的问题上的表现较基于梯度的方法更好.

Conti 等人[65] 将新颖性搜索算法和质量多样性算法(Quality Diversity，QD)[66] 与进化策略进行了结合，提出了 NS-ES、NSR-ES 和 NSRA-ES 三种使用进化策略解决具有稀疏奖励或欺骗性奖励的强化学习任务的方法. 其中 NS-ES 是将新颖性搜索算法与进化策略相结合的算法，它使用了一个行为特征

函数 $b(\pi_\theta)$来表示策略 π_θ的特征，并将其存储在集合$\mathcal{A}$中，而策略的新颖度 N 则使用$\mathcal{A}$中所有其它的$b(\pi_\theta)$之间的 k 近邻来衡量. 而 NS-ES 的策略参数更新公式与 OpenAI-ES 相似，其差别在于使用了新颖性作为进化策略中种群的适应度：

$$\theta_{t+1}^m \leftarrow \theta_t^m + \alpha \frac{1}{n\sigma}\sum_{i=1}^{n} N(\theta_t^m + \sigma\varepsilon_i, \mathcal{A})\varepsilon_i \quad (12)$$

其中：m 为智能体的个数，N 为新颖度函数，n 为采样样本个数. 而 NSR-ES 和 NSRA-ES 是将质量多样性算法与进化策略结合的算法，其中 NSR-ES 在 NS-ES 的基础上综合考虑了新颖度与总回报，将两者的均值作为进化策略中种群的适应度：

$$\theta_{t+1}^m \leftarrow \theta_t^m + \alpha \frac{1}{n\sigma}\sum_{i=1}^{n} \frac{F(\theta_t^{i,m}) + N(\theta_t^{i,m}, \mathcal{A})}{2}\varepsilon_i \quad (13)$$

而 NSRA-ES 则在 NSR-ES 的基础上使用调节因子 w 控制新颖度与总回报在适应度中所占的权重：

$$\theta_{t+1}^m \leftarrow \theta_t^m + \alpha \frac{1}{n\sigma}\sum_{i=1}^{n} wF(\theta_t^{i,m})\varepsilon_i + (1-w)N(\theta_t^{i,m}, \mathcal{A})\varepsilon_i \quad (14)$$

上述方法均直接使用进化算法替代深度强化学习并用于解决强化学习中的任务，这些方法在部分任务上展现出了等同或超过深度强化学习方法的表现. 这表明进化算法在强化学习领域上具有一定的优势，近期也有越来越多的学者致力于改进进化策略的方法来解决强化学习的问题，如置信域进化策略(Trust Region Evolution Strategies，TRES)[67]、指导进化策略(Guided Evolutionary Strategies)[68]等，这一方面是由于当前日益增长的算力，另一方面也可以看作是深度学习的思想推动了进化算法的发展与改进，这也为学者们将深度强化学习与进化算法的优势进行互补的方法提供了一部分思路.

4.3 策略种群搜索方法

除了上述两种进化算法引导策略搜索的强化学习外，还有一种更为直接的策略搜索方法. 与参数分布搜索方法不同，策略种群搜索方法不在策略的参数空间上进行搜索，而是直接将多个策略看作一个种群在策略空间上进行搜索，使用适应度函数评估并选择出最好的策略. 这类方法将强化学习与进化算法通过耦合度较低的方式结合起来，不仅可以在具有不同策略的种群中选择出具有更好策略的个体，同时还可以用于在具有不同超参数的策略的种群中进行超参数的搜索.

Jaderberg 等人[27]提出了一种简单的异步优化算法 PBT，PBT 可以在网络训练的过程中对超参数进行自动的选择，并通过进化算法对种群中的个体进行淘汰和选择. 其具体做法是，首先种群中所有个体随机初始化超参数与网络权重，并对每个个体周期性的进行异步评估. 如果个体表现较差，则会寻找种群中更好的个体替代自身，之后对超参数或权重进行一定的变异再继续训练. 这种训练方式十分简单，但是可以极大地提高训练速度并且降低种群搜索所需的开销.

PBT 不仅可以用于机器翻译任务，并且可以用于训练生成对抗网络等多种监督学习网络，最重要的是 PBT 可以和强化学习算法进行结合. PBT 可以对强化学习算法的学习率、熵、内在奖励等超参数进行优化，PBT 已经被用于强化学习中的多智能体竞争问题[69]和新颖性搜索问题[65]. 此外，Jung 等人[70]使用 PBT 利用强化学习智能体种群中的最优策略搜索更好的策略，但对于最优学习者并没有采用 PBT 中的直接拷贝的方式，而是使用了软更新的方式来引导种群在策略空间中搜索更好的策略. 随后，Parker-Holder 等人[71]在 PBT 的基础上提出了 PB2 算法，PB2 使用概率模型来引导策略搜索，相较于 PBT 不仅更高效而且更具有理论保障，PB2 可以在一系列的强化学习任务中用更少的开销达到更好的表现.

近年学者们基于 PBT 等算法的思想又提出了一系列将进化算法用于强化学习策略种群搜索的算法框架. Franke 等人[28]提出了一种适用于离策略(Off-policy)的自动强化学习(Auto RL)框架 SEARL. SEARL 对于强化学习策略的超参数的搜索不仅包括网络参数还包括神经网络的结构. SEARL 会对种群中的个体顺序进行 5 个基本的进化过程，其中包括初始化、评估、选择、变异和训练，此外由于是针对离策略的强化学习算法，SEARL 还加入了一个所有个体共享的经验池.

Gupta 等人[29]提出了深度进化强化学习框架 DERL，DERL 可以进化出具有不同形态的智能体从而在复杂的环境中学习到具有挑战性的运动和操作任务. DERL 首先会在种群初始化具有不同拓扑结构的个体，并通过 PPO 算法并行的对个体进行训练和评估，在选择的过程中对所有个体 4 个为一组进行轮盘选择，之后会将一个经过变异后的子代加入到种群中. 与之前算法最大的不同之处在于，DERL 从以代为周期的进化改变为了异步并行的进化，因此无需种群中所有个体完成训练后再进行评估和进化，只需要种群中任意一个个体完成训练就

可以立即进行选择、变异等操作. DERL 实现了迄今最大规模的同步形态进化和强化学习模拟.

5　进化算法经验指导的深度强化学习

进化算法与深度强化学习各有优缺点，其中进化算法所具有的探索能力是深度强化学习一直以来所缺乏的，因此将两者进行结合并通过进化算法所获得的经验来指导深度强化学习进行探索成为了提升深度强化学习表现的一个重要的方向. 虽然进化算法经验指导的深度强化学习是最近才被学者提出的，但已有较多学者围绕这一方向进行了研究. 进化算法经验指导的深度强化学习指的是将进化算法中的种群个体与环境交互的经验存储到强化学习的经验池中并供其使用的一类方法，本节将此类方法分为无反馈的指导与有反馈的指导，并将这些方法进行了详细的对比与分析.

5.1　无反馈的指导

为了不影响正常深度强化学习的梯度下降更新过程，无反馈的指导通常的做法是首先单独使用进化算法先对环境进行探索，并将探索得到的经验存储下来，之后对这些经验进行处理和特征提取或直接作为深度强化学习的经验，最后再使用深度强化学习方法对这些特征或经验和自身与环境交互得到的经验一并进行利用与学习. 无反馈的进化算法经验指导的深度强化学习算法框架如图 4 所示.

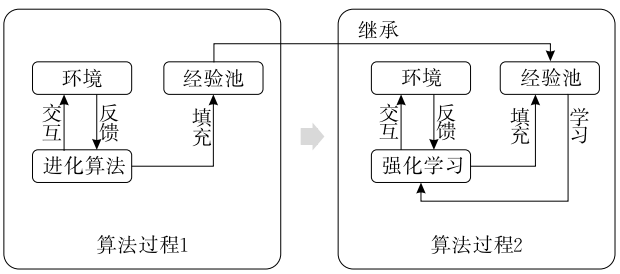


图 4　无反馈的进化算法经验指导的深度强化学习算法流程

5.1.1　神经进化自助 Q 学习算法

Zimmer 等人[72]为了使机器人更快和更好地适应未知的环境和状况提出了这种首先使用进化算法对环境进行探索而之后使用深度强化学习方法对数据进行利用的神经进化自助 Q 学习算法. 他们认为需要两种不同的算法进行协作，其中第一种算法可以直接使用底层传感器信息并对策略进行直接的搜索，即进化算法，而第二种算法可以使用高层表征快速的学习行为与动作，即深度强化学习方法. 具体来说，他们首先在源域使用了一种非支配排序遗传算法(Nondominated Sorting Genetic Algorithm II, NSGA-II)[73]进行与任务无关的探索，并使用了与机器人当前位置坐标相关的一种行为多样性作为进化算法的评估标准. 此外，他们除了使用进化算法对网络参数进行探索，还使用了神经进化[74]的方法对网络结构进行调整. 之后对使用进化算法生成的状态与动作轨迹进行特征提取，其中包括使用线性模型和聚类算法提取动作集合、辨识与状态相关的传感器输入，最后使用这些提取到的特征运用 Q 学习先后在源域与目标域中进行训练.

Zimmer 等人[72]提出的方法在机器人领域中的应用显得较为复杂与繁琐，因为其需要对传感器数据进行复杂的特征提取，但其为在模拟环境进行训练与测试的深度强化学习方法提供了一种思路. 这种在开始使用与任务无关的进化算法之后再使用深度强化学习的方法可以加快深度强化学习算法的收敛，这在一定程度说明了通过这种耦合方式进化算法对深度强化学习起到了一定的指导作用. 此外，在深度强化学习中探索和利用过程通常是高度耦合且无法分离的，Zimmer 等人[72]的做法可以看作将探索部分与利用部分进行了分离，即使用进化算法进行探索，使用深度强化学习进行利用. 虽然这种分离是不彻底的，但一方面说明了以进化算法为基础的探索方法的有效性，另一方面为专注提高深度强化学习的探索度或利用率提供了一种更为便捷的方法.

5.1.2　目标探索过程策略梯度算法

Colas 等人[75]基于上述思想与目标探索过程算法(Goal Exploration Processes, GEPs)[76]提出了目标探索过程策略梯度(GEP-PG)算法，并将该方法用于解决深度强化学习中的稀疏奖励和欺骗性奖励问题. Colas 等人[75]指出深度强化学习如果经过细致的超参数调整虽然可以达到很好的效果且相较于进化算法具有更高的样本效率，但是深度强化学习的探索度不足难以应对环境中存在的欺骗性奖励.

GEP-PG 与 Zimmer 等人[72]的方法一样具有两个算法过程，前一个为进化算法，后一个为深度强化学习算法. 在基于目标的探索算法中，它首先将策略网络参数 θ 与其对应的策略结果 o 作为一个元组 $\langle\theta, o\rangle$ 进行存储. 其中，策略结果 o 即为目标，在具体实现中使用回合过程的状态信息作为策略结果. 之后在结果空间 O 中随机采样一个策略结果 o'，通过临近算法找到之前存储的元组中与 o' 相似的 o 与 θ. 随后，在找到的 θ 中加入高斯噪声作为随机扰动并使用其作为当前策略，同时在策略与环境交互的过

程中存储其产生的经验.在深度强化学习算法中,首先继承了之前探索算法的经验池,之后使用 DDPG 算法进行训练与学习.

GEP-PG 中的探索算法完全可以看作一种进化算法的变种,因为其具有进化算法的基本特征.它可以看作一种只具有一个个体的进化策略方法,对策略结果的扰动与选择可以看作变异和进化过程,且它在每次进化的时候都使用新个体而抛弃老个体,即使用经过选择与扰动后的策略结果.GEP-PG 充分利用了进化算法的探索能力对结果空间进行探索,虽然其仅在 2 个模拟环境下进行了实验,但其遵循了较为严格与严谨的实验方法[77],充分表明了这种方法的有效性.值得注意的是,这种对结果空间进行直接的探索的方法目前研究还较少,未来可以将其与 NS 与 QD 算法之间进行对比与结合或对其进行进一步改进,以便于与深度强化学习进行结合,从而进一步提升深度强化学习的探索能力.

5.2 有反馈的指导

无反馈的进化算法经验指导的深度强化学习将进化算法与深度强化学习过程进行了分离,两者进行了几乎完全的解耦,只通过经验池进行联系,且两个算法过程具有先后顺序.而有反馈的进化算法经验指导的深度强化学习使用了经验池与梯度将两个算法进行关联,其中梯度作为深度强化学习给进化算法提供的反馈信息.同时因为引入了梯度作用,两个算法过程需要交替或并行运行.有反馈的进化算法经验指导的强化学习算法框架如图 5 所示.

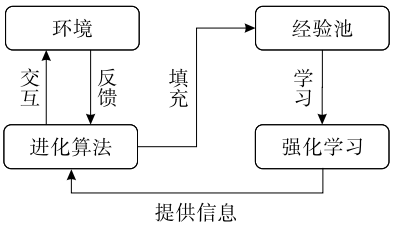


图 5 有反馈的进化算法经验指导的深度强化学习算法框架

无反馈的指导可以看作只利用到了进化算法高效的探索能力,但进化算法面对高维度和多参数问题时的样本效率问题依旧存在,而有反馈的指导将深度强化学习的梯度作为一种信息提供给进化算法从而达到提高进化算法样本效率的目的.

5.2.1 ERL 算法框架

Khadka 等人[78]提出了一种使用进化算法经验指导深度强化学习的框架 ERL,这是首次将进化算法的经验作为深度强化学习的指导且同时将深度强化学习的梯度信息引入进化算法的一种方法,同时这也是进化算法与深度强化学习进行结合的一个重要的阶段性成功,之后有较多的学者基于此框架提出了多种使用进化算法经验指导的深度强化学习的改进算法.

ERL 与环境交互的部分包括进化算法所产生的个体以及通过深度强化学习得到的个体,在进化算法种群中的个体与环境进行交互后会将得到的经验存储在经验池中,之后通过环境返回的累计奖励作为适应度对其个体进行以回合为周期的选择、交叉与变异从而生成新个体.同时,深度强化学习部分从经验池中获取并学习经验,并以一定周期在 DDPG 的 Actor 和 Critic 进行梯度更新后将 Actor 作为一个个体注入到进化算法部分的种群中,用于代替种群中适应度最差的个体.ERL 算法的基本框架如图 6 所示.

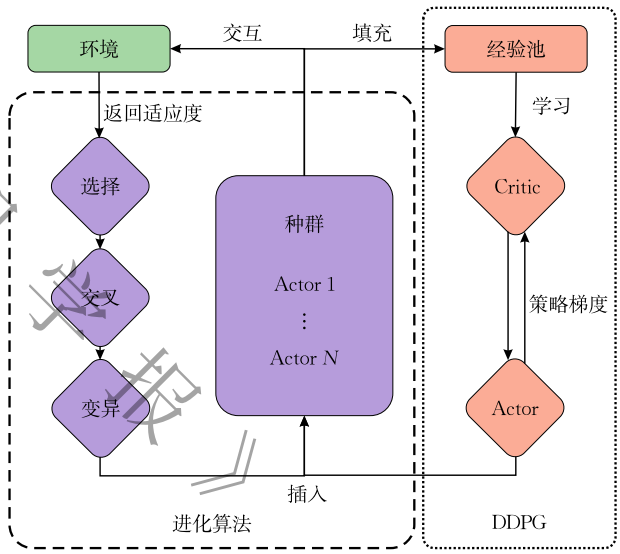


图 6 ERL 算法框架[78]

在 ERL 的进化算法部分,会根据种群中个体的适应度从高到低的顺序为个体排序.设种群总个体数为 k,那么其中适应度最高的 e 个个体被称为精英,可以直接作为下一代个体,之后通过轮盘赌的方式从种群中选择 $k-e$ 个个体形成集合 S 并进行交叉与变异操作.在交叉的过程中会随机选取一个精英个体与一个集合 S 中的个体,并将它们的网络参数进行与传统遗传算法相同的交叉操作,即对于每个网络参数子代个体有相同的概率获得任意一个父本个体的网络参数,最后将得到的个体加入集合 S.在变异过程中,ERL 会将个网络看作一个高维矩阵,并从中随机地选取行和列通过添加高斯噪声来进行变异,高斯噪声的大小会根据随机数选择不变

异、重置、低强度变异以及高强度变异 4 种操作中的任意一种. 通过将上述过程进行一个固定的次数，从而完成对个体的变异. ERL 算法框架完整过程见算法 2.

算法 2.　ERL 算法框架[78].

初始化：强化学习 Actor π_{rl} 和 Critic Q_{rl}，定义经验池 R，种群 P，其中包含 k 个 Actor

REPEAT

让种群 P 中的每个 Actor 与环境交互一回合，将得到的累积回报作为其适应度

将上述交互过程中获得的经验存储到经验池 R 中

根据适应度从 P 中以一定方式选择出 e 个精英个体

根据精英个体对 P 中的个体进行交叉和变异产生新个体从而形成新种群 P'

$P \leftarrow P'$

让强化学习 Actor 与环境进行交互一回合

将上述交互过程中获得的经验存储到经验池 R 中

计算强化学习 Critic 损失，更新 Q_{rl}

计算强化学习 Actor 损失，更新 π_{rl}

将种群中的某个 Actor 用 π_{rl} 进行替换

UNTIL 终止条件

ERL 的关键在于插入到种群中的通过深度强化学习得到的 Actor，如果深度强化学习个体适应度较高则可以直接作为新种群中的个体并在之后影响种群子代的发展，而即使其适应度较低那么它也会在变异的过程中为进化算法的种群提供一些信息. Khadka等人[78]的实验表明ERL兼具了深度强化学习的样本效率和进化算法的探索能力，在 MuJoCo 的多个环境中达到了远超部分深度强化学习算法的效果. ERL 这种进化算法与深度强化学习相互指导的方法作为一种框架，对其中的进化算法部分和深度强化学习部分分别进行改进再以这种方式组合则可以达到更好的效果，目前已有很多学者提出了基于 ERL 框架改进的方法. 此外，进一步修改框架中深度强化学习部分与进化算法部分的耦合方式也是未来工作的一个方向.

5.2.2　基于 ERL 的算法

由于 ERL 框架的成功，出现了很多基于 ERL 的改进算法，其中大多数均是在 ERL 算法框架上对其使用的算法进行替换或增加新的组成部分，并没有修改 ERL 的基本框架，本节将按照这些方法的提出顺序对这些方法进行详细的介绍，图 7 列出了部分基于 ERL 的方法的算法框架.

Pourchot 等人将 ERL 中传统的遗传算法替换为了 CEM 并将深度强化学习部分替换为了 TD3 算法[20]提出了CEM-RL 算法[79]，其算法框架如图 6(a)所示，从而进一步提高了 ERL 算法的性能. CEM 属于分布估计算法，在 CEM-RL 中会使用适应度最高的前 1/2 个体用于计算新的均值 μ_{new} 与协方差矩阵 Σ，其计算方式如式(15)与式(16).

$$\mu_{\text{new}} = \sum_{i=1}^{K_e} \lambda_i z_i \tag{15}$$

$$\Sigma_{\text{new}} = \sum_{i=1}^{K_e} \lambda_i (z_i - \mu_{\text{old}})^2 + \varepsilon \mathcal{I} \tag{16}$$

其中：K_e 为种群中适应度最高的前 1/2 个体数，z_i 为种群个体参数，$\varepsilon\mathcal{I}$ 为噪声项，λ_i 为种群个体在更新时所占的权重，使用 $\lambda_i = \dfrac{\log(1+K_e)/i}{\sum_{i=1}^{K_e} \log(1+K_e)/i}$ 进行计算，即适应度越高的个体所占权重越大. 之后，每回合在个体均值上根据协方差矩阵加入一个高斯噪声来生成新个体. 另外，CEM-RL 没有使用与 ERL 相同的将深度强化学习的 Actor 个体注入种群的方式，而是直接使用梯度更新种群适应度最高的前1/2个体. 此外，Pourchot 等人[80]还引入了此前提出的重要性融合(Importance Mixing)机制，但其对算法效果没有明显改进.

Khadka 等人[78]将 ERL 的深度强化学习部分的单个学习者改为了多个具有不同 γ 的学习者共同学习，提出了 CERL 算法[81]，希望以此增加算法对解空间探索的多样性，其算法框架如图 6(b)所示. CERL 通过一个资源管理器对不同学习者的计算资源进行动态的分配，其分配方式使用了一种基于 UCB(Upper Confidence Bound)的方法对每一个学习者 i 计算其得分 $\mathcal{U}$，并通过 $\mathcal{U}$ 构建分布并采样决定计算资源的分配.

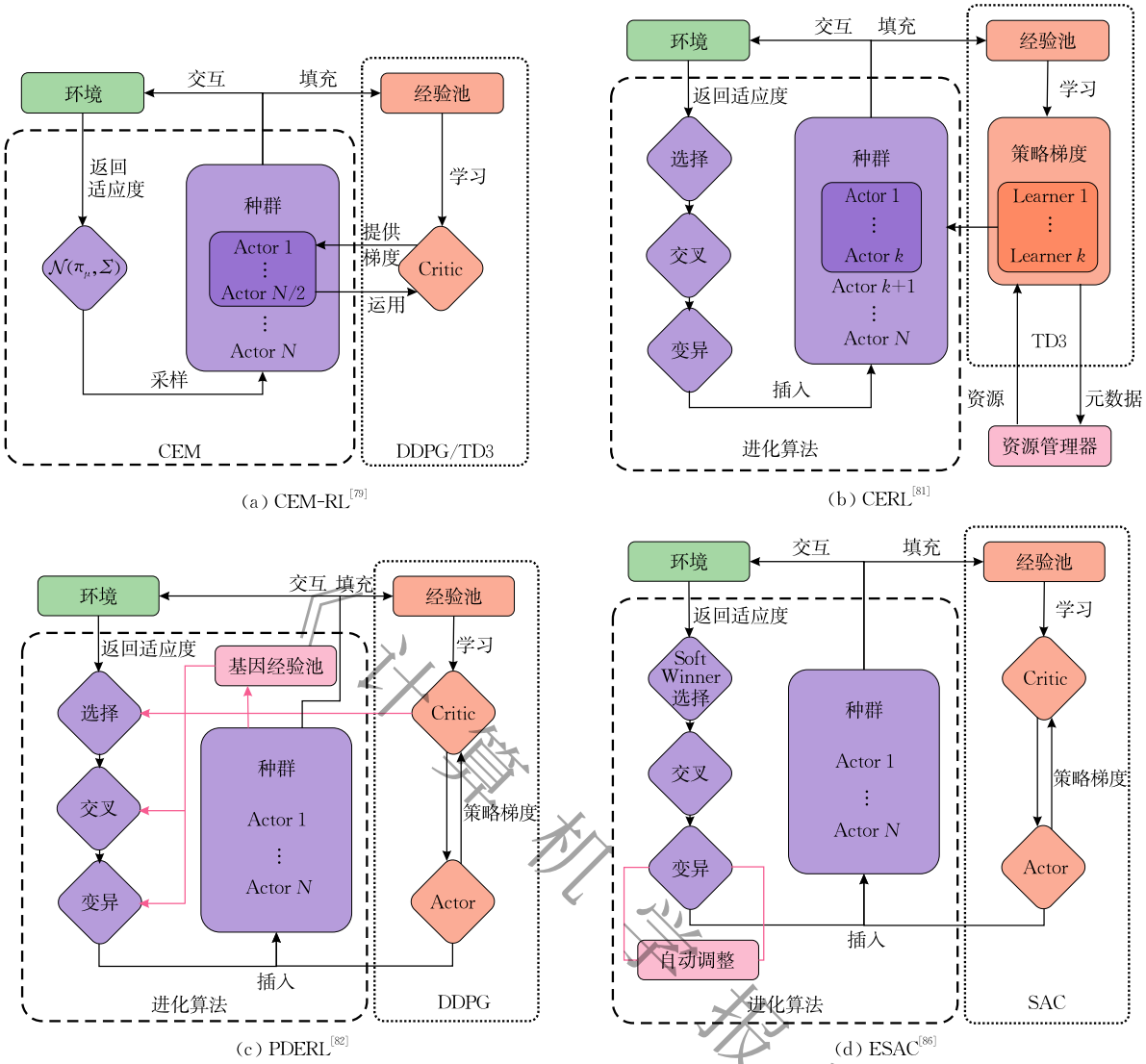
在计算得分 $\mathcal{U}$ 的过程中，CERL 首先会根据使用式(17)这种软更新(Soft Update)的方式计算其适应度评价 v_i，之后将 v_i 经过标准化得到 v_i^n，并使用 v_i^n 计算每个学习者的得分 $\mathcal{U}_n$，如式(18).

$$v' \leftarrow \alpha \times R + (1-\alpha) \times v \tag{17}$$

$$\mathcal{U}_n = v_i^n + c \times \sqrt{\frac{\log\left(\sum_{i=1}^{b} y_i\right)}{y_i}} \tag{18}$$

其中：R 为回合总回报，y_i 为每一个学习者所经历的回合总数，α 和 c 为调节因子，b 为学习者总数.

虽然 CERL 在进化算法部分与 ERL 完全一致，但由于引入了多个学习者，因此在将深度强化学习的 Actor 插入到进化算法的种群中时会在一定的



(a) CEM-RL[79] (b) CERL[81]

(c) PDERL[82] (d) ESAC[86]

图 7 进化算法经验指导的深度强化学习算法框架

周期将适应度最低的 n 个 Actor 进行替换.

Bodnar 等人[82]认为 ERL 中使用的常规的遗传算子可能对训练具有破坏性并且会导致灾难性遗忘,为了解决这一问题他们提出了 PDERL 算法,其算法框架如图 6(c)所示. PDERL 引入了两个基于反向传播的遗传算子,并且为每一个个体加入了一个基因经验池用于存储个体各自的经验. 其中,第一个算子为 Q 过滤蒸馏交叉算子(Q-filtered Distillation Crossovers),它首先通过适应度或策略距离选择两个父代个体并选择其中一个父代个体的参数初始化子代的网络参数,之后使用两个父代个体的基因经验池通过模仿学习训练出子代,其损失函数如式(19).

$$L(C)=\sum_{i}^{N_C}\|\mu_z(s_i)-\mu_x(s_i)\|^2\,\mathbb{I}_{Q(s_i,\mu_x(s_i))>Q(s_i,\mu_y(s_i))}+\sum_{j}^{N_C}\|\mu_z(s_j)-\mu_y(s_j)\|^2\,\mathbb{I}_{Q(s_j,\mu_y(s_j))>Q(s_j,\mu_x(s_j))}+\frac{1}{N_C}\sum_{k}^{N_C}\|\mu_z(s_k)\|^2 \tag{19}$$

其中:N_C 为种群个体数,μ 为策略,x 与 y 为父代个体,z 为交叉算子产生的子代个体. 从式(19)中可以看出子代 z 的策略会对父代个体 x 与 y 中具有更大 Q 值的动作进行模仿,从而使得子代学习到更好的策略. 第 2 个算子为近端变异算子(Proximal Mutations),采用了 SM-G-SUM 算法[83]在传统的高斯噪声项的分母加入了一个敏感度 s,用于控制更新子代策略时变异的程度,其更新子代策略更新公式为:$\theta\leftarrow\theta+\frac{x}{s}$,$x\sim\mathcal{N}(0,\sigma^2 I)$.

Lü 等人[84]认为 ERL 框架的深度强化学习部分只能从经验池中进行学习而不能直接利用精英个体,为了解决这一问题他们提出了 RIM 算法. RIM 可以将适应度较高的种群个体招募成为深度强化学

习智能体，而种群中适应度较低的个体则可以对深度强化学习智能体进行模仿学习以提高其适应度. 此外，RIM 还将深度强化学习的智能体网络修改为一个梯度策略网络和一个招募策略网络共同接受输入并由 Critic 给出输出的形式，以此统一了深度强化学习智能体和种群个体的结构.

Suri 等人[85]基于 ERL 框架提出了一种结合了 SAC 算法[86]与进化策略的 ESAC 算法，其算法框架如图 6(d)所示，在 ESAC 中提出了一种自动变异调整机制(Automatic Mutation Tuning). 其具体做法是使用种群中最大适应度与平均适应度作为变异强度 σ 调整的依据，如式(20)和式(21).

$$\sigma \leftarrow \sigma + clip\left(\frac{\alpha_{es}}{n\sigma} SmoothL1(F_{\max}, F_{\text{avg}}), 0, \zeta\right) \tag{20}$$

$$SmoothL1(x_i, y_i) = \begin{cases} 0.5(x_i - y_i)^2, & |x_i - y_i| < 1 \\ |x_i - y_i| - 0.5, & \text{其他} \end{cases} \tag{21}$$

其中：$F_{\max}$ 为种群个体最大的适应度，F_{avg} 为种群个体平均的适应度，α_{es} 为进化策略的学习率，ζ 为调整因子. 从上述公式可以看出，当种群个体最大适应度与平均适应度相差不大时 ESAC 使用 $L2$ 距离计算变异程度，而在两者相差较大时使用线性函数计算，但变异程度被限制在了0到ζ之间，使得变异程度在自动调整的同时控制在一定的范围内. 在进化算法部分，ESAC 使用了一种软胜利者选择(Soft Winner Selection)种群个体选择策略，这种方法会从种群的个体中选择出适应度最高的部分个体作为胜利者并组成集合，之后通过此集合进行交叉操作，而变异算子则使用进化策略的形式.

Cideron 等人[87]在 ERL 的基础上提出了一种可以在结果空间进一步平衡探索和利用的算法 QD-RL. 在进化算法部分，种群中所有被训练过的个体会被保存到一个集合中，对此集合的选择过程类似于进化算法中优势个体的选择过程，但其并不会只将累计回报作为其唯一标准，而是基于帕累托前沿(Pareto front)的方式根据累计回报和新颖性两种标准选择优势个体. 其中，对于个体新颖性的评价是根据个体的结果空间与在集合中 k 个临近个体结果空间的距离计算得到的. 在强化学习部分，QD-RL 基于TD3算法，但不同的是，其具有两个独立的 Critic. 一个 Critic 用于评价种群的质量，而另一个 Critic 用于评价种群的多样性. 另外，与 ERL 不同的是，在种群中的所有个体都需要进行梯度更新，其中一半的个体按照最大化质量的方式使用质量 Critic 进行更新，而另一半个体按照最大化多样性的方式使用多样性 Critic 进行更新.

Marchesini 等人[88]指出此前的基于 ERL 的算法你存在无法与基于价值的深度强化学习和评估时间开销大的问题，并提出了 SUPE-RL，其算法框架如图 8. SUPE-RL 改变了 ERL 的算法流程，它采用了周期性评估的方式，在强化学习的算法流程中每间隔一定的回合对种群进行一次并行的评估，这样可以通过降低评估次数降低时间开销. 之后选择出最优个体并通基因软更新的方式更新强化学习的策略网络，其具体公式如式(22).

$$\theta_a \leftarrow \tau\theta_a + (1-\tau)\theta_{\text{best}} \tag{22}$$

其中：θ_a 为强化学习策略网络的参数，θ_{best} 为种群中最优个体的网络参数，τ 为调节因子. 正是因为上述的更新方式，SUPE-RL 不仅可以应用于 DDPG、TD3、SAC 等连续动作域的方法，还可以应用于 PPO 和 Rainbow[89]等方法从而解决离散动作域任务，这使得有反馈的指导方法能够应用在更广的领域中.

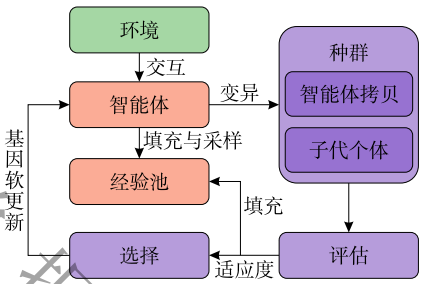


图 8　SUPE-RL 算法框架[88]

除了上述完全基于 ERL 的算法外，还出现了一些与 ERL 算法在算法结构与流程类似的基于种群的深度强化学习算法[70,90]. 其也使用经验进行指导并使用累积奖励选择优势个体，主要不同在于，他们不会将 Actor 网络作为种群的个体，也不会对 Actor 网络的参数进行变异，而是将一个 Actor 网络和一个 Critic 网络的整体视为一个个体，通过梯度进行更新. 这类算法可以看作是受 ERL 框架思想影响而产生的一类基于种群的深度强化学习算法. 总的来说，ERL 算法提供了一个进化算法与深度强化学习结合的优秀范例，因此产生了上述众多 ERL 算法的衍生算法，使得结合进化算法的深度强化学习方法得到了极大的扩充.

5.3　算法对比与分析

上文分别介绍了部分进化算法对深度强化学习进行无反馈的指导和有反馈的指导的算法，本节将对这些方法进行综合性的对比与分析.

目前无反馈的指导算法还较少，大多数为有反馈的指导算法.这可能是由于无反馈的指导算法应用的范围较小，目前仅适合于具有大量欺骗性奖励的问题，在其它问题上逐渐被有反馈的指导算法所替代.因此需要对无反馈的指导算法进行进一步的探究使其专注于解决这种需要更强探索度的问题，同时为有反馈的指导算法提供一些的思路.此外，无反馈指导的算法由于使用了两个算法进行线性组合，因此需要使两个算法在切换的过程中更加平滑，但是目前还没有类似的改进.

对于以 ERL 为代表的有反馈指导的方法来说，从表 2 中可以看出大多数算法使用了不同的深度强化学习算法和进化算法，但是这些方法大多数没有进行消融实验，从而无法得知具体是因为算法的替换还是因为其它组件的引入所带来的性能上的提升.在 ERL 中由于采用了将深度强化学习个体直接插入到种群中的更新方式从而减少了时间开销，而以 CEM-RL 为例，其使用的将梯度直接作用于多个个体的更新方式虽然提高了算法的样本效率，但大大增加了算法的时间开销.因此，在未来的研究中还需要注意引入新组件所带来的算法时间开销问题.在面对更大规模的问题时，有反馈指导的方法可能由于其进化算法的本质而产生样本效率问题，从而难以以较高的效率解决此类问题[24,91].因此，对有反馈指导的方法的效率问题进行改进是使这类方法能继续发展和应用的一个难点.此外，上文提到大多数有反馈指导的算法只是在 ERL 算法框架上进行简单的改进，而 ERL 框架的耦合方式未必为最优形式，因此探究其它的耦合方式也是未来的一个研究方向.

表 2 进化算法经验指导的深度强化学习算法对比

指导方法	算法	文献	特点	进化模块算法	进化包含的操作	深度强化学习模块算法
无反馈的指导	神经进化自助 Q 学习算法	[72]	进化算法在前，深度强化学习随后的算法结构	多目标遗传算法、神经进化	选择、交叉、变异	Deep Q-learning
	GEP-PG	[75]	目标探索过程	目标探索过程	变异	DDPG
有反馈的指导	ERL	[78]	第一次提出进化算法经验指导的深度强化学习框架	遗传算法	选择、交叉、变异	DDPG
	CEM-RL	[79]	梯度直接作用于个体、重要性融合	分布估计算法	变异	DDPG/TD3
	CERL	[81]	多个学习者、资源管理器分配资源	遗传算法	选择、交叉、变异	TD3
	PDERL	[82]	基因经验池、两个新算子	遗传算法	选择、Q 过滤蒸馏交叉、近端变异	DDPG
	RIM	[84]	招募-模仿机制、独特的网络结构	遗传算法	选择、交叉、变异	DDPG 变种
	ESAC	[85]	自动变异调整机制、新选择策略	进化策略	Soft Winner 选择、交叉、变异	SAC
	QD-RL	[87]	在结果空间平衡探索和利用	帕累托最优	选择、变异	TD3
	SUPE-RL	[88]	周期性评估、基因软更新、可用于基于价值函数的强化学习方法	遗传算法	选择、变异	PPO/Rainbow

有反馈的指导的方法作为一类基于常规深度强化学习算法的改进方法，其固然具有和常规深度强化学习一样的可以解决一系列决策问题的能力.在强化学习领域中，通常使用强化学习算法在 MuJoCo 环境上的得分来评估其在连续动作域上的性能.在强化学习算法与环境交互相同的步数下，通过对比不同的算法在同一环境上的得分的高低可以方便地对比不同算法的性能，也可以直观地体现出算法的数据利用率.表 3 对比了分布估计算法、部分具有代表性的常规深度强化学习方法与部分进化指导的深度强化学习方法在 MuJoCo 中不同测试环境上的性能表现.其中，表格中的所有算法所使用的参数均与原始论文相同，所有算法都是选取 10 个不同的随机种子训练得到的结果，前 6 列实验结果在每个环境中训练 100 万步后得到的结果，后 3 列实验结果在 Half-Cheetah 环境中训练 200 万步，Hopper、Walker2d、Swimmer 和 Ant 环境中训练 600 万步后得到的结果.从表 3 中可以看出：有进化指导的深度强化学习比一般深度强化学习方法相比在大部分环境上都能获得不同程度的提高，这说明使用进化算法的经验指导深度强化学习这一方向的正确性和合理性.

从表 3 中的一些环境中还可以看出，一般的深度强化学习在一定的训练步数后算法效果的提升相较于结合进化算法的深度强化学习来说较小，这也说明结合进化算法的深度强化学习在训练的前期采样效率相较于一般的深度强化学习算法更低.但随着训练步数的增加，进化算法对环境探索度更大的优势逐渐就能够逐渐显现出来，而一般的深度强化学习算法在此时已经接近收敛.因此，总的来说，在训练步数较大的情况下，结合进化算法的深度强化

表 3　进化算法经验指导的深度强化学习算法性能对比①②

算法	HalfCheetah			Hopper			Walker2d			Swimmer			Ant		
	Mean	Std.	Median	Mean	Std.	Median	Mean	Std.	Median	Mean	Std.	Median	Mean	Std.	Median
CEM[40,79]	2940	353	3045	1055	14	1040	928	50	934	**351**	**9**	**361**	487	33	506
TD3[20,79]	9630	202	9606	3355	171	3626	3808	339	3882	63	9	47	4027	402	4587
ERL[78,79]	8684	130	8675	2288	240	2267	2188	328	2338	350	8	360	3716	673	4240
CEM-DDPG[79]	**11035**	**298**	**11276**	3444	55	3499	2865	218	2985	268	32	279	2170	1128	3574
CEM-TD3[79]	10725	397	11539	**3613**	**105**	**3722**	**4711**	**155**	**4637**	75	11	62	**4251**	**251**	**4310**
PDERL[82]	7891	445	7983	3178	101	3190	1484	493	1155	331	30	352	3282	1056	3201
TD3*[20,82]	11534	713	11334	3231	213	3282	4925	476	5190	53	26	51	6212	216	6121
ERL*[78,82]	10963	225	11025	2049	841	1807	1666	737	1384	334	20	246	4330	1806	5164
PDERL*[82]	**13522**	**287**	**13553**	**3397**	**202**	**3400**	**5184**	**477**	**5333**	**337**	**12**	**348**	**6845**	**407**	**6948**

学习能取得更好的效果. 此外，在表 3 中结合进化算法的深度强化学习算法通常具有比一般的深度强化学习更低的标准差，这是由于对种群进行评估时直接使用了回合奖励，这使得进化算法中鲁棒性好的特点被很好的结合到强化学习之中，从而可以解决强化学习中出现的欺骗性梯度问题.

值得注意的是，CEM 作为一种最简单的分布估计算法在 Swimmer 环境上能取得比深度强化学习和进化算法经验指导的深度强化学习更高的分数，这是因为 Swimmer 环境对算法的环境探索度具有较高的要求，同时算法在此环境更容易陷入局部最优解. 由此可以看出，将进化算法与深度强化学习以这种相互指导的方式进行结合虽然会提升算法的探索能力和性能，但算法的探索能力相较于进化算法会有一定的减弱，因此在结合的过程中需要更多的考虑算法探索能力与利用能力之间的平衡. CEM-TD3 由于直接将深度强化学习的梯度信息作用于种群个体，因此失去了较大一部分探索能力，使得其在 Swimmer 中的表现与 TD3 接近，而 PDERL 在此处处理得更好，从而在 Swimmer 中的表现与 CEM 接近，但依旧不能达到 CEM 的效果，因此在此类方法上探究其探索度的自适应机制也是未来研究的一个重要方向.

综上所述，进化算法经验指导的深度强化学习在结构、样本效率和探索度自适应机制上还有较大的改进空间，此外，目前相关研究相较于其它领域较少，将传统的方法和技巧引入到进化算法经验指导的深度强化学习中也是未来研究的方向之一.

6　进化算法模块嵌入的深度强化学习

进化算法经验指导的深度强化学习方法会通过共享经验池和梯度信息的方式双向地指导与结合，除此之外近期的研究中还有一类将进化算法模块嵌入或应用于深度强化学习的过程中的方法，从而使得两者更为紧密的耦合，本节将选择部分近期进化算法模块嵌入的深度强化学习方法进行简单的介绍.

在结合进化算法的传统强化学习方法中，除了使用进化算法对网络参数进行搜索的方法外也有进化算法作为一个附加组件对强化学习进行辅助的方法，但其进化算法与强化学习相对来说比较独立且耦合度较低，且强化学习一般用于网络的更新. 在进化算法经验指导的深度强化学习中，对于无反馈的指导来说，进化算法与深度强化学习是完全分离的，对于有反馈的指导来说，ERL 框架通过经验池与梯度信息将两者进行联系，这种方式可以看作进化算法与深度强化学习两者以对等的地位进行协助. 而本节的算法一般是通过进化算法解决深度强化学习中某一子过程的问题，使得进化算法作为一个关键的组件嵌入并参与到深度强化学习的过程中从而提升深度强化学习方法的表现，其主要特点是一般不能再以回合为单位来进行种群的评估，同时也不能直接使用累计回报作为进化算法的适应度函数，其适应度函数需要进行特定的设计.

Hämäläinen 等人[92]提出了一种使用 CMA 方法自适应的改善 PPO 算法的探索度的 PPO-CMA 算法，并认为 PPO 的探索方差会过早收缩并导致算法过早收敛. PPO-CMA 会记录最近 n 次迭代的历史信息，并使用一种类似于 CMA-ES 进化路径启发式的方式训练一个方差网络，并通过在策略的损失函数加入此方差网络输出的均值和方差来控制采样探索与利用的平衡. 在 CMA 中还使用了一种 rank-μ 的更新方式，先估计协方差再更新均值，这样可以使方差在搜索方向上被拉长，从而提高下回合的探

① 由于部分最新的算法未提供源代码或详细实验结果，此表实验数据选自文献[79]与文献[82].

② 其中标 * 的算法在 HalfCheetah 中训练 200 万步，在 Hopper、Walker2d、Swimmer、Ant 中训练 600 万步.

索效率.此外,PPO-CMA 还会对 GAE 为负值的项根据策略均值进行镜像翻转.PPO-CMA 可以看作将进化算法作为深度强化学习中一部分网络的替代,从而使得进化算法模块嵌入到深度强化学习的过程中.

Houthooft 等人[93]提出了一种使用进化算法的元强化学习方法 EPG,它对损失函数使用了神经网络进行建模,其中外层循环使用进化策略来优化和调整损失函数的网络参数 ϕ,而内层循环则使用外层所提供的损失函数来进行梯度上升.在学习的过程中,先使用内层循环使智能体学习最小化外层提供的损失函数 L_ϕ:

$$\theta^* = \arg\min_\theta \mathbb{E}_{\tau\sim\mathcal{M}}\left[L_\phi(\pi_\theta,\tau)\right] \tag{23}$$

再使用外层循环调整损失函数的参数使得智能体的回报最大:

$$\phi^* = \arg\max_\phi \mathbb{E}_{\mathcal{M}\sim p(\mathcal{M})}\mathbb{E}_{\tau\sim\mathcal{M}}\left[r(\tau)\right] \tag{24}$$

其中:$\mathcal{M}$ 为 MDP 过程,而 $p(\mathcal{M})$ 为 MDP 的分布.这种方法可以看作使用进化算法对深度强化学习的目标函数进行调整从而直接干预深度强化学习的过程,因此也可以看为一种进化算法模块嵌入深度强化学习的方法.

Simmons-Edler 等人[94]提出了 CGP 算法,它将迭代采样策略的稳定性和高性能与策略网络的低计算开销进行了结合.在 CGP 中,使用 CEM 策略对动作进行迭代采样从而使得 Q-learning 算法可以应用于连续动作域环境,并使用 $L2$ 范数对 CEM 策略 π_{CEM} 进行模仿从而训练一个策略网络 π_ϕ:

$$J(\phi) = \mathbb{E}_{s_t\sim\rho^{\pi_{\mathrm{CEM}}}}\left(\nabla_{\pi_\phi}\pi_\phi(s_t) - \pi_{\mathrm{CEM}}(s_t)^2\right) \tag{25}$$

具体来说,其 CEM 会对动作空间按照高斯分布进行采样得到 n 个动作作为种群的个体,之后通过 Q 函数对其计算适应度并排序,最后根据适应度最高的前 k 个动作更新高斯分布的均值与方差,并经过多次迭代最终得到动作选择.在这一过程中,相当于使用进化算法模块嵌入到深度强化学习之中从而使得离散动作域的方法可以应用在连续动作域的环境之中.CGP 方法虽然只能达到与当前大多数策略梯度方法近乎相同的性能,但其在训练的过程中表现得更为稳定.

Shao 等人[95]认为 Q-learning 中的目标网络的延迟更新会减慢算法的学习过程,因此提出了一种不需要目标网络的 GRAC 算法.此方法会使用 CEM 算法寻找临近当前策略的动作中具有更大 Q 值的动作,之后会将这一动作用于更新目标 Q 网络,并将这种 CEM 策略与策略梯度进行结合从而加速策略学习过程.这种方法将 CEM 算法嵌入到了一般的深度强化学习过程之中,并通过 Q 值作为其适应度函数,不仅提升了 Q 值估计对噪声的鲁棒性,同时在性能提升上也有较为可观的表现.

进化算法需要在一个回合之后才能进行种群的评估和进化,而深度强化学习则需要在每一步做出决策.这两者各自的特点决定了进化算法不能直接插入在深度强化学习的任意一个过程之中,且一般只适用于不需要在学习的每一步都给出结果的步骤之中.因此,如何选择一个合适的函数或者基准作为此类方法进化算法部分的评估标准成为此类方法面临的一个重要的问题.如果评估标准选取的不当,可能会对算法的稳定性造成较大的影响,甚至使得网络向错误的方向进行更新,从而导致算法崩溃或无法收敛.PPO-CMA 和 EPG 算法均使用损失函数作为适应度函数,使用这种方式需要算法中存在与深度强化学习过程相关的辅助网络,而 CGP 与 GRAC 均选择了 Q 值作为进化算法的适应度函数,因为 Q 值能够表现当前时间步状态动作对的价值.虽然损失函数和 Q 值不像累积回报一样是一个准确值,但却够在每一步的训练过程中得到,并对当前的时间步有一个比较好的估计,这对于进化算法模块嵌入的深度强化学习算法来说是至关重要的.目前,类似的适应度函数还比较难寻找,此外如何解决引入这种估计值所带来的鲁棒性问题也需要进行考虑.因此,目前将进化算法模块嵌入到深度强化学习的这类方法还较少,这也是将来进化算法与深度强化学习结合的另一个重要的方向.

7 总结与展望

在策略搜索的角度上来看,强化学习可以看作沿着梯度进行的策略搜索过程,而进化算法则是沿着种群个体适应度高的方向进行搜索的过程,这两个方向在解决不同问题上具有不同的优势.此外,由于强化学习中环境通常会直接给出智能体与其交互过程中每一步的立即奖励,这大大简化了进化算法适应度函数的设计过程.综上所述,近年来出现了众多将进化算法与深度强化学习进行结合的方法,这些方法表现出了比传统强化学习更好的性能,同时也使得深度强化学习算法一直所欠缺的环境探索能力得到了有效的提升,因此也越来越受到研究者们的关注.

本文将进化算法与强化学习结合的方法分为两

大类，并将上文所提到的相关算法、强化学习模块所使用的方法、所适用的问题以及其特点总结为表 4. 从表 4 中可以直观地看出每类方法的当前的发展状况以及所存在的问题.

表 4　进化算法与强化学习结合的方法总结

结合方式	方法类别		相关算法	强化学习模块	适用的问题	特点
进化算法引导策略搜索的强化学习	参数分布搜索方法		PEPG[54]、NES[60]等	传统策略梯度	高维连续动作域	直接在参数空间上进行探索，更为充分地利用了进化算法的探索优势
	策略梯度近似方法		OpenAI-ES[61]、GA[63]、NS-ES[66]、NSR-ES[66]、NSRA-ES[66]等	策略梯度思想	高维离散/连续动作域、稀疏奖励问题	无梯度，具有与深度强化学习算法性能、解决稀疏奖励能力较强
	策略种群搜索方法		PBT[27]、PB2[71]、SEARL[28]、DERL[29]等	任意强化学习算法	高维离散/连续动作域	直接在策略空间上进行搜索，可以高效地进行大规模策略和超参数搜索
进化算法与深度强化学习结合	进化算法经验指导的深度强化学习	无反馈的指导	ERQL[72]、GEP-PG[75]等	深度Q学习、基于Actor-Critic的策略梯度	高维离散/连续动作域、欺骗性奖励问题、稀疏奖励问题	先使用进化算法进行探索，再在此基础上使用深度强化学习
		有反馈的指导	ERL[78]、CEM-RL[79]、CERL[81]、PDERL[82]、RIM[84]等	基于Actor-Critic的策略梯度	高维离散/连续动作域	进化算法与深度强化学习相辅相成，相互指导，相互作用，形成了较为成熟的框架
	进化算法模块嵌入的深度强化学习		PPO-CMA[92]、EPG[93]、CGP[94]、GRAC[95]等	深度Q学习、基于Actor-Critic的策略梯度	高维离散/连续动作域、元强化学习问题、强化学习鲁棒性问题	使用进化算法处理深度强化学习中某一部分的问题，适用于多种问题

(1) 进化算法引导策略搜索的强化学习中的参数搜索方法由于没有引入深度神经网络，因此解决问题的能力较为低下. 尽管如此，这类方法充分地利用了进化算法的探索能力，将进化算法直接用于参数空间的探索这一思想与之后将进化算法用于深度强化学习中的神经网络的参数探索的思想是一致的.

(2) 进化算法引导策略搜索的强化学习中的梯度近似方法最为重要的特点就是其无梯度的特性，与其它基于梯度的方法相比其主要优势在于更为容易进行并行计算，从而大幅度减少时间开销. 未来可以致力于让此类方法解决更高维度的问题，同时可以将强化学习中的搜索与探索方式引入其中，探究例如 NS-ES、NSR-ES、NSRA-ES 这类的探索机制，使得此类方法可以解决奖励更为稀疏的问题，此外还可以考虑将此类方法可并行的特性引入进化算法与深度强化学习结合的方法之中.

(3) 进化算法与深度强化学习进行结合的方法目前是解决问题能力最强且适用范围最广泛的一类方法，从表 4 中可以看出与深度强化学习相结合的方法大多数采用了基于 Actor-Critic 的策略梯度，这是目前深度强化学习领域中最为成熟的算法框架之一，这也使得其能解决大部分高维离散或者连续动作域的问题. 对于这类方法的更深入的研究，可以从以下几个方向着手进行：

① 进化算法普遍存在着样本效率低下的问题，虽然其探索度较高，但其本质为全局随机性搜索，需要在整个回合结束后才能更新其种群，而深度强化学习在每个回合步中都会得到大量的信息并使用这些信息进行梯度更新，因此进化算法相较于深度强化学习来说样本效率较低. 针对进化算法样本效率的问题，可以使用深度强化学习中的梯度和回合步中的其它信息对其进行指导，指引进化算法种群在解空间中的位置与下一代进化的方向.

② 进化算法与深度强化学习的兼容性较差，从表 4 中可以看出目前进化算法与深度强化学习的组合与耦合方式较为单一，本文认为可以从探索和利用的角度进一步的分析和探究两者的其它结合方式. 平衡强化学习中的探索和利用一直是强化学习领域中的一个重要问题，在与进化算法结合的深度强化学习中也需要对两者进行更为合理的平衡，如引入新颖度与探索度等一些度量方式在进化算法的探索和深度强化学习的利用中自动调节也是未来的方向之一.

③ 目前结合进化算法与深度强化学习的方法中均与最新的同类方法进行了比较，但很少有进行消融实验并进行进一步分析其算法获得提升的工作. 进化算法本身为一种启发式算法，其理论基础较为薄弱，且与深度强化学习一样均不能保证其收敛性，因此本文认为需要加强对算法性能提升的分析与实验，从而为进一步的研究打下基础.

总的来说，尽管结合进化算法的强化学习方法在以上几个方向上取得了一定的成果且出现了较为成熟的进化算法经验指导的深度强化学习框架，但其研究和发展目前还处于较为初级的阶段. 目前使

用进化算法与深度强化学习结合的方法还较少，将进化计算领域与深度强化学习领域中最新的研究成果应用于结合进化算法的强化学习方法之中还有较大的空间可以发掘.

参 考 文 献

[1] Watkins C J C H. Learning from delayed rewards. UK: King's College, Cambridge, 1989

[2] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning//Proceedings of the Workshops at the 26th Neural Information Processing Systems (NIPS). Lake Tahoe, USA, 2013: 201-220

[3] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529-533

[4] Liu Q, Zhai J, Zhang Z, et al. A survey on deep reinforcement learning. Chinese Journal of Computers, 2018, 41(1): 1-27

[5] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search. Nature, 2016, 529(7587): 484-489

[6] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of go without human knowledge. Nature, 2017, 550(7676): 354-359

[7] Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature, 2019, 575(7782): 350-354

[8] Polydoros A S, Nalpantidis L. Survey of model-based reinforcement learning: Applications on robotics. Journal of Intelligent & Robotic Systems, 2017, 86(2): 153-173

[9] Yu L, Zhang W, Wang J, et al. Seqgan: Sequence generative adversarial nets with policy gradient//Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI). San Francisco, USA, 2017: 2852-2858

[10] Ng A Y, Coates A, Diel M, et al. Autonomous Inverted Helicopter Flight Via Reinforcement Learning. Experimental Robotics IX. Berlin, Germany: Springer, 2006: 363-372

[11] Yu K, Dong C, Lin L, et al. Crafting a toolchain for image restoration by deep reinforcement learning//Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, USA, 2018: 2443-2452

[12] Arulkumaran K, Deisenroth M P, Brundage M, et al. A brief survey of deep reinforcement learning. IEEE Signal Processing Magazine, 2017, 34(6): 26-38

[13] Bellemare M, Srinivasan S, Ostrovski G, et al. Unifying count-based exploration and intrinsic motivation//Proceedings of the 30th in Neural Information Processing Systems (NIPS). Barcelona, Spain, 2016: 1471-1479

[14] Pathak D, Agrawal P, Efros A A, et al. Curiosity-driven exploration by self-supervised prediction//Proceedings of the 34th International Conference on Machine Learning (ICML). Sydney, Australia, 2017: 2778-2787

[15] Fortunato M, Azar M G, Piot B, et al. Noisy networks for exploration//Proceedings of the 6th International Conference on Learning Representations (ICLR). Vancouver, Canada, 2018: 1-12

[16] Gong X, Yu J, Lü S, et al. Actor-critic with familiarity-based trajectory experience replay. Information Sciences, 2022, 582: 633-647

[17] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. arXiv preprint arXiv: 1509.02971, 2015

[18] Schulman J, Levine S, Moritz P, et al. Trust region policy optimization//Proceedings of the 32nd International Conference on Machine Learning (ICML). Lille, France, 2015, 3: 1889-1897

[19] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. arXiv preprint arXiv: 1707.06347, 2017

[20] Fujimoto S, Van Hoof H, Meger D. Addressing function approximation error in actor-critic methods//Proceedings of the 35th International Conference on Machine Learning (ICML). Stockholm, Sweden, 2018, 4: 2587-2601

[21] Liu Jian-Wei, Gao Feng, Luo Xiong-Lin. Survey of deep reinforcement learning based on value function and policy gradient. Chinese Journal of Computers, 2019, 42(6): 1406-1438(in Chinese)
(刘建伟，高峰，罗雄麟. 基于值函数和策略梯度的深度强化学习综述. 计算机学报，2019，42(6)：1406-1438)

[22] Liu Q, Yan Y, Zhu F, et al. A deep recurrent q network with exploratory noise. Chinese Journal of Computers, 2019, 42(7): 1588-1604(in Chinese)
(刘全，闫岩，朱斐等. 一种带探索噪音的深度循环 Q 网络. 计算机学报，2019，42(7)：1588-1604)

[23] Jiang Yu-Bin, Liu Quan, Hu Zhi-Hui. Actor-critic algorithm with maximum-entropy correction. Chinese Journal of Computers, 2020, 43(10): 1897-1908(in Chinese)
(姜玉斌，刘全，胡智慧. 带最大熵修正的行动者评论家算法. 计算机学报，2020，43(10)：1897-1908)

[24] Ge Ji-Ke, Qiu Yu-Hui, Wu Chu-Ming, Pu Guo-Lin. Summary of genetic algorithms research. Application Research of Computers, 2008(10): 2911-2916(in Chinese)
(葛继科，邱玉辉，吴春明，蒲国林. 遗传算法研究综述. 计算机应用研究，2008(10)：2911-2916)

[25] Yu Yang, Qian Chao. Perface of special issue on evolutionary learning. Journal of Software, 2018, 29(9): 2545-2546(in Chinese)
(俞扬，钱超. 演化学习专题前言. 软件学报，2018，29(9)：2545-2546)

[26] Real E, Aggarwal A, Huang Y, et al. Regularized evolution for image classifier architecture search//Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI). Honolulu, USA, 2019, 33: 4780-4789

[27] Jaderberg M, Dalibard V, Osindero S, et al. Population based training of neural networks. arXiv preprint arXiv: 1711.09846, 2017

[28] Franke J K H, Koehler G, Biedenkapp A, et al. Sample-efficient automated deep reinforcement learning//Proceedings of the 8th International Conference on Learning Representations (ICLR). Addis Ababa, Ethiopia, 2020: 1-12

[29] Gupta A, Savarese S, Ganguli S, et al. Embodied intelligence via learning and evolution. arXiv preprint arXiv:2102.02202, 2021

[30] Holland J H. Genetic algorithms and adaptation. Adaptive Control of Ill-Defined Systems. Boston, USA: Springer, 1984: 317-333

[31] Beyer H G, Schwefel H P. Evolution strategies-a comprehensive introduction. Natural Computing, 2002, 1(1): 3-52

[32] Fogel D B, Fogel L J, Atmar J W. Meta-evolutionary programming//Proceedings of the 25th Asilomar Conference on Signals, Systems & Computers. IEEE computer Society (ACSSC). Pacific Grove, USA, 1991: 540-545

[33] Banzhaf W, Nordin P, Keller R E, et al. Genetic Programming: An Introduction on the Automatic Evolution of computer programs and its Applications. Burlington, USA: Morgan Kaufmann Publishers Inc, 1998

[34] Storn R, Price K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 1997, 11(4): 341-359

[35] Larrañaga P, Lozano J A, Ref L, et al. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Berlin, Germany: Springer, 2001

[36] Hansen N, Müller S D, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation. Evolutionary Computation, 2003, 11(1): 1-18

[37] Baluja S. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Pittsburgh, Carnegie Mellon University, 1994

[38] Harik G R, Lobo F G, Goldberg D E. The compact genetic algorithm. IEEE Trans on Evolutionary Computation, 1999, 3(4): 287-297

[39] Pelikan M, Goldberg D E, Cantú-Paz E. BOA: The Bayesian optimization algorithm//Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). Orlando, USA, 1999, 1: 525-532

[40] Rubinstein R. The cross-entropy method for combinatorial and continuous optimization. Methodology and Computing in Applied Probability, 1999, 1(2): 127-190

[41] Rubinstein R Y, Kroese D P. The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning. Berlin, Germany: Springer, 2013

[42] Stulp F, Sigaud O. Path integral policy improvement with covariance matrix adaptation//Proceedings of the 29th International Conference on Machine Learning (ICML). Edinburgh, UK, 2012: 1-8

[43] Sigaud O, Stulp F. Policy search in continuous action domains: An overview. Neural Networks, 2019, 113: 28-40

[44] Gong Mao-Guo, Jiao Li-Cheng, Yang Dong-Dong, Ma Wen-Ping. Research on evolutionary multi-objective optimization algorithm. Journal of Software, 2009, 20(2): 271-289 (in Chinese)
(公茂果，焦李成，杨咚咚，马文萍. 进化多目标优化算法研究. 软件学报，2009，20(2)：271-289)

[45] Szita I, Lörincz A. Learning Tetris using the noisy cross-entropy method. Neural Computation, 2006, 18(12): 2936-2941

[46] Thiery C, Scherrer B. Improvements on learning Tetris with cross entropy. International Computer Games Association Journal, 2009, 32(1): 23-33

[47] Mahmud M, Kaiser M S, Hussain A, et al. Applications of deep learning and reinforcement learning to biological data. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(6): 2063-2079

[48] Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge, USA: MIT Press, 1998

[49] Singh S, Jaakkola T, Littman M L, et al. Convergence results for single-step on-policy reinforcement-learning algorithms. Machine Learning, 2000, 38(3): 287-308

[50] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning//Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI). Phoenix, USA, 2016: 2094-2100

[51] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning//Proceedings of the 33rd International Conference on Machine Learning (ICML). New York City, USA, 2016, 4: 1995-2003

[52] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 1992, 8(3&4): 229-256

[53] Di Mario E, Talebpour Z, Martinoli A. A comparison of PSO and reinforcement learning for multi-robot obstacle avoidance//Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC). Cancun, Mexico, 2013: 149-156

[54] Atiya A F, Parlos A G, Ingber L. A reinforcement learning method based on adaptive simulated annealing//Proceedings of the 46th IEEE Midwest Symposium on Circuits and Systems (MWSCAS). Cairo, Egypt, 2003, 1: 121-124

[55] Gabillon V, Ghavamzadeh M, Scherrer B. Approximate dynamic programming finally performs well in the game of Tetris//Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS). Lake Tahoe, USA, 2013, 26: 1754-1762

[56] Jaśkowski W, Szubert M. Coevolutionary CMA-ES for knowledge-free learning of game position evaluation. IEEE Transactions on Computational Intelligence and AI in Games, 2015, 8(4): 389-401

[57] Scherrer B, Ghavamzadeh M, Gabillon V, et al. Approximate modified policy iteration and its application to the game of Tetris. Journal of Machine Learning Research, 2015, 16: 1629-1676

[58] Krawiec K, Szubert M G. Learning n-tuple networks for Othello by coevolutionary gradient search//Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). Dublin, Ireland, 2011: 355-362

[59] Sehnke F, Osendorfer C, Rückstieß T, et al. Parameter-exploring policy gradients. Neural Networks, 2010, 23(4): 551-559

[60] Wierstra D, Schaul T, Peters J, et al. Natural evolution strategies//Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC). Hong Kong, China, 2008: 3381-3387

[61] Salimans T, Ho J, Chen X, et al. Evolution strategies as a scalable alternative to reinforcement learning. arXiv preprint arXiv:1703.03864, 2017

[62] Todorov E, Erez T, Tassa Y. Mujoco: A physics engine for model-based control//Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vilamoura, Portugal, 2012: 5026-5033

[63] Such F P, Madhavan V, Conti E, et al. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. arXiv preprint arXiv:1712.06567, 2017

[64] Lehman J, Stanley K O. Abandoning objectives: Evolution through the search for novelty alone. Evolutionary Computation, 2011, 19(2): 189-223

[65] Conti E, Madhavan V, Such F P, et al. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents//Proceedings of the 32nd in Neural Information Processing Systems (NeurIPS). Montréal, Canada, 2018: 5027-5038

[66] Pugh J K, Soros L B, Szerlip P A, et al. Confronting the challenge of quality diversity//Proceedings of the 2015 IEEE Conference on Genetic and Evolutionary Computation (CEC). Sendai, Japan, 2015: 967-974

[67] Liu G, Zhao L, Yang F, et al. Trust region evolution strategies//Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI). Honolulu, USA, 2019, 33: 4352-4359

[68] Maheswaranathan N, Metz L, Tucker G, et al. Guided evolutionary strategies: Augmenting random search with surrogate gradients//Proceedings of the 36th International Conference on Machine Learning (ICML). Long Beach, USA, 2019: 4264-4273

[69] Jaderberg M, Czarnecki W M, Dunning I, et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning. Science, 2019, 364(6443): 859-865

[70] Jung W, Park G, Sung Y. Population-guided parallel policy search for reinforcement learning//Proceedings of the 7th International Conference on Learning Representations (ICLR). New Orleans, USA, 2019

[71] Parker-Holder J, Nguyen V, Roberts S J. Provably efficient online hyperparameter optimization with population-based bandits//Proceedings of the 33th Neural Information Processing Systems (NeurIPS). Vancouver, Canada, 2020

[72] Zimmer M, Doncieux S. Bootstrapping Q-learning for robotics from neuro-evolution results. IEEE Transactions on Cognitive and Developmental Systems, 2017, 10(1): 102-119

[73] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197

[74] Floreano D, Dürr P, Mattiussi C. Neuroevolution: from architectures to learning. Evolutionary intelligence, 2008, 1(1): 47-62

[75] Colas C, Sigaud O, Oudeyer P Y. GEP-PG: Decoupling exploration and exploitation in deep reinforcement learning algorithms//Proceedings of the 35th International Conference on Machine Learning (ICML). Stockholm, Sweden, 2018: 1039-1048

[76] Forestier S, Mollard Y, Oudeyer P Y. Intrinsically motivated goal exploration processes with automatic curriculum learning. arXiv preprint arXiv:1708.02190, 2017

[77] Henderson P, Islam R, Bachman P, et al. Deep reinforcement learning that matters//Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI). New Orleans, USA, 2018: 3207-3214

[78] Khadka S, Tumer K. Evolution-guided policy gradient in reinforcement learning//Proceedings of the 32nd in Neural Information Processing Systems (NeurIPS). Montréal, Canada, 2018: 1188-1200

[79] Pourchot A, Sigaud O. CEM-RL: Combining evolutionary and gradient-based methods for policy search//Proceedings of the 7th International Conference on Learning Representations (ICLR). New Orleans, USA, 2019

[80] Pourchot A, Perrin N, Sigaud O. Importance mixing: Improving sample reuse in evolutionary policy search methods. arXiv preprint arXiv:1808.05832, 2018

[81] Khadka S, Majumdar S, Nassar T, et al. Collaborative evolutionary reinforcement learning//Proceedings of the 36th International Conference on Machine Learning (ICML). Long Beach, USA, 2019: 3341-3350

[82] Bodnar C, Day B, Lió P. Proximal distilled evolutionary reinforcement learning//Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI). New York, USA, 2020, 34(4): 3283-3290

[83] Lehman J, Chen J, Clune J, et al. Safe mutations for deep and recurrent neural networks through output gradients//Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). Kyoto, Japan, 2018: 117-124

[84] Lü S, Han S, Zhou W, et al. Recruitment-imitation mechanism for evolutionary reinforcement learning. Information Sciences, 2021, 553: 172-188

[85] Suri K, Shi X Q, Plataniotis K N, et al. Maximum mutation reinforcement learning for scalable control. arXiv preprint arXiv:2007.13690, 2020

[86] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor//Proceedings of the 35th International Conference on Machine Learning (ICML). Stockholm, Sweden, 2018, 5: 2976-2989

[87] Cideron G, Pierrot T, Perrin N, et al. QD-RL: Efficient mixing of quality and diversity in reinforcement learning. arXiv preprint arXiv:2006.08505, 2020

[88] Marchesini E, Corsi D, Farinelli A. Genetic soft updates for policy evolution in deep reinforcement learning//Proceedings of the 8th International Conference on Learning Representations (ICLR). Addis Ababa, Ethiopia, 2020

[89] Hessel M, Modayil J, Van Hasselt H, et al. Rainbow: Combining improvements in deep reinforcement learning// Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI). New Orleans, USA, 2018: 3215-3222

[90] Doan T, Mazoure B, Abdar M, et al. Attraction-repulsion actor-critic for continuous control reinforcement learning. arXiv preprint arXiv:1909.07543, 2019

[91] Müller N, Glasmachers T. Challenges in high-dimensional reinforcement learning with evolution strategies//Proceedings of the 15th International Conference on Parallel Problem Solving from Nature (PPSN). Coimbra, Portugal, 2018: 411-423

[92] Hämäläinen P, Babadi A, Ma X, et al. PPO-CMA: Proximal policy optimization with covariance matrix adaptation// Proceedings of the 30th IEEE International Workshop on Machine Learning for Signal Processing (MLSP). Espoo, Finland, 2020: 1-6

[93] Houthooft R, Chen Y, Isola P, et al. Evolved policy gradients// Proceedings of the 32nd in Neural Information Processing Systems (NeurIPS). Montréal, Canada, 2018: 5400-5409

[94] Simmons-Edler R, Eisner B, Mitchell E, et al. Q-learning for continuous actions with cross-entropy guided policies// Proceedings of the Workshops at the 36th International Conference on Machine Learning (ICML). Long Beach, USA, 2019

[95] Shao L, You Y, Yan M, et al. GRAC: Self-guided and self-regularized actor-critic//Proceedings of the 5th Conference on Robot Learning (CoRL 2021). London, UK, 2021

LÜ Shuai, Ph. D., associate professor, Ph. D. supervisor. His main research interests include artificial intelligence, machine learning and automated reasoning.

GONG Xiao-Yu, M. S. candidate. His main research interests include artificial intelligence and machine learning.

ZHANG Zheng-Hao, M. S. candidate. His main research interests include artificial intelligence and machine learning.

HAN Shuai, Ph. D. candidate. His main research interests include artificial intelligence and machine learning.

ZHANG Jun-Wei, M. S. candidate. His main research interests include artificial intelligence and machine learning.

Background

Deep reinforcement learning is one of the most important branches in the field of machine learning. Both theoretical and applied researches of this field are current hot topics. Deep reinforcement learning is an end-to-end learning method that does not require labeled data as input, but rather learns policies by interacting with the environment, and forms an intelligent agent with strong learning capabilities through continuous trial and error. It is capable of solving high-dimensional and large-scale problems. Although deep reinforcement learning has achieved remarkable results, it still faces problems such as insufficient exploration of the environment, poor robustness, and susceptibility to deceptive gradients caused by deceptive rewards. Evolutionary algorithms are a class of generic population-based optimization algorithms inspired by biological evolution. These methods usually have good global search ability, robustness, parallelism and other advantages, so they have been widely used in various fields as a classical universal optimization algorithm. Evolutionary reinforcement learning is a method combines evolutionary algorithms with deep reinforcement learning methods. It can compensate for the inadequacy of deep reinforcement learning methods mentioned above, and it usually can achieve better performance than classical deep reinforcement learning methods especially in environments with local optima, so it has become a research hotspot recently.

In order to help researchers who are interested in deep reinforcement learning follow the recent hotspot about evolutionary reinforcement learning, our paper introduces the main idea of evolutionary algorithms and reinforcement learning, focuses to investigate the approaches of combining these two methods, analyze the advantages and disadvantages, applications and performance of these methods, and discuss the future research trends and directions in this field.

This paper is supported by the National Key R&D Program of China (2017YFB1003103), National Natural Science Foundation of China (61763003), and Natural Science Research Foundation of Jilin Province of China (20180101053JC).