强化学习求解组合优化问题的研究综述*

张宏立，朱家政，董颖超

(新疆大学 电气工程学院，新疆 乌鲁木齐 830017)

摘　要： 组合优化问题广泛的存在于生产实践的各个领域，解决组合优化问题的主要手段通常包括使用由领域专家人工设计的启发式算法以及设计成熟的求解器，按照一定顺序构建一个解决方案. 而随着实际问题复杂度逐渐的增加，这类方法无法于在线求解方面取得很好的效果，得到的结果可能往往是次优的. 而强化学习给出了一个很好的替代方案，通过对智能体模型的良好训练，迅速地对此类问题进行求解. 故回顾了近年来将强化学习框架应用于组合优化问题的研究，对其基本原理、相关方法、应用研究进行总结和综述，并指出未来该方向亟待解决的若干问题.

关键词： 强化学习；组合优化问题；深度神经网络；指针网络

DOI： 10.13568/j.cnki.651094.651316. 2023.02.02.0001

中图分类号： O224；TP18　**文献标识码：** A　**文章编号：** 2096-7675(2023)02-0129-013

引文格式： 张宏立, 朱家政, 董颖超. 强化学习求解组合优化问题的研究综述[J]. 新疆大学学报(自然科学版)(中英文), 2023, 40(2): 129-141.

英文引文格式： ZHANG Hongli, ZHU Jiazheng, DONG Yingchao. A review of research on reinforcement learning for solving combinatorial optimization problems[J]. Journal of Xinjiang University(Natural Science Edition in Chinese and English), 2023, 40(2): 129-141.

A Review of Research on Reinforcement Learning for Solving Combinatorial Optimization Problems

ZHANG Hongli, ZHU Jiazheng, DONG Yingchao

(*School of Electrical Engineering, Xinjiang University, Urumqi Xinjiang 830017, China*)

Abstract： Combinatorial optimization problems are widespread in all areas of production practice, and the main means of solving combinatorial optimization problems usually include the use of heuristic algorithms manually designed by domain experts and the design of sophisticated solvers to construct a solution in a certain order. As the complexity of the actual problem gradually increases, such methods can not achieve good results in online solving, and the results obtained may often be suboptimal. And reinforcement learning gives a good alternative to solve such problems rapidly by well-training an intelligent body model. Therefore, this paper reviews the recent research on applying the reinforcement learning framework to combinatorial optimization problems, summarizes and reviews its basic principles, related methods and application studies, and points out several problems that need to be addressed in this direction in the future.

Key words： reinforcement learning; combinatorial optimization problems; deep neural networks; pointer network

0　引 言

优化问题是指在不同的可能性中寻找出最佳方案，根据求解问题的不同，对应的求解问题可分为三大类：连续、离散、混合整数规划问题. 例如，寻找一个凸优化问题的解是一个连续优化问题，而从一个图的所有路径中找到最短路径则是一个离散优化问题，将此类离散空间的优化问题称为组合优化问题，其数学模型如下所

* **收稿日期：** 2023-02-02

基金项目： 国家自然科学基金“新能源电力系统的随机动力学分析”（52267010），“高寒高海拔环境下的无人机集群电力巡检关键控制问题研究”（62263030）；新疆维吾尔自治区自然科学基金“复杂环境下高维电力系统混合动力学行为分析与控制研究”（2022D01C367），“风电系统混沌振荡识别定位、耦合交互传播机理及抑制策略研究”（2022D01E33）.

作者简介： 张宏立（1972-），男，博士，教授，主要从事复杂生产过程优化与调度的研究，E-mail: zhlxju@163.com.

示：

$$\begin{aligned} &\min\ f(x) \\ &s.t. g(x) \geq 0 \\ &x \in D \end{aligned} \tag{1}$$

式中：$f(x)$ 为目标函数，$g(x)$ 为问题的约束条件，D 表示有限的离散决策空间. 其中的一个常见问题是旅行商问题，该问题目标是寻找到一条能够访问所有城市并返回初始城市的最短路线，其本质是在一个全连接的加权图中找到一个长度最小的哈密顿回路. 哈密顿回路中所有的节点组成了 D，回路中每条边权值的累加构成了 $f(x)$.

大多数组合优化问题都是 NP 难问题，通常无法在多项式时间内找到解决方案. 因此，许多近似方法或启发式算法被设计用以解决此类问题，但此类方法往往计算耗时、计算成本过高. 而近些年随着人工智能技术的不断发展，一个新兴趋势是使用训练机器学习算法解决此类问题. 强化学习算法作为机器学习算法的一个特殊分支已经在旅行商问题中得到应用，对于一个组合优化问题，它通过定义环境、训练环境中的智能体的方式，最终产生一个解决方案. 将强化学习应用到组合优化问题中，首先将问题建模为序列决策过程，其中智能体通过执行一系列的动作同环境进行交互，最终找到一个可行的解决方案. 马尔可夫决策过程为此类问题的建模提供了一个广泛使用的数学框架[1]. 该过程可以被描述为一个五元组：

$$M = (S, A, R, T, \gamma) \tag{2}$$

式中：S 为状态空间，以旅行商问题为例，可以通过两种方式进行状态空间的设计，一种方法是渐进式地构建解决方案，将问题中已去过的城市作为状态空间，逐步地得到完整的解决方案；另一种方法是从问题的次优解出发，在该解的基础上不断进行改进. A 为动作空间，不同时刻的动作会对当前的状态产生改变. R 为奖励函数，是对当前状态下所产生动作的一种评价，反映了当前选择的动作是否改善了需要解决的问题. T 是转移函数，其制约着不同时刻状态的转移. γ 为折扣因子，$0<\gamma<1$，折扣因子的大小展示了智能体对未来奖励的期望程度.

在马尔可夫决策过程中智能体的目标是找到一个策略函数，使得预期的累计奖励最大化：

$$\pi^* = \arg\max_{\pi} \mathrm{E}\left(\sum_{t=0}^{H} \gamma^t R(s_t, a_t)\right) \tag{3}$$

如果一个组合优化问题能够定义为马尔可夫决策过程，就需要训练智能体如何寻找到最优策略，一般而言，有两种类型的强化学习算法：

第一种为基于价值的方法，智能体不需要制定显式的策略，它维护一个价值表格或价值函数，并通过这个价值表格或价值函数来选取价值最大的动作.

第二种为基于策略的方法，智能体会制定一套动作策略（确定在给定状态下需要采取何种动作），并根据这个策略进行操作. 强化学习算法直接对策略进行优化，使制定的策略能够获得最大的奖励.

可以看出，强化学习算法的关键在于将马尔可夫决策过程的状态转化为输入并输出行动值或者行动的策略. 状态包含了关于问题的一些特征信息，如旅行商问题中给定的图或者当前的行程. 而动作值或者策略则为数字. 因此强化学习算法中应该包含编码器，能够将状态转化为数字形式. 针对不同的组合优化问题，已经有许多的编码器被提出，包括递归神经网络[2]、图神经网络[3]、基于注意力的网络[4]和多层感知器[5].

本文的组织结构如下：第 1 节对常见的组合优化问题、编码器和强化学习算法进行了研究背景概述. 第 2 节对当前主流的将强化学习应用于各种组合优化问题的研究进行了综述. 第 3 节进行了总结与展望.

1 基础知识概述

1.1 组合优化问题概述

首先介绍混合整数线性规划模型，一种受限的优化问题，许多实际的应用都可以归结为此类问题. 已经有许多使用分支定界技术的工业优化器用以求解此类问题的实例[6−10]. 该问题能够表现为如下形式[11]：

$$\arg\min\{c^T x | Ax \leq b, 0 \leq x, x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}\} \tag{4}$$

式中：$c\in\mathbb{R}^n$ 为目标系数向量，$A\in\mathbb{R}^{m\times n}$ 为约束系数矩阵，$b\in\mathbb{R}^m$ 为约束向量，$p\leq n$ 为整数变量的个数.

解决混合整数线性规划模型所使用的工业求解器，往往也可用于求解离散的组合优化问题，常见的离散的组合优化问题如下所示.

首先，旅行商问题为给定一个完整的加权图 $G=(V,E)$，找出其中权重最小的路线，这是一个典型的组合优化问题，已经在路径规划、数据聚类、基因测序等领域得到了应用[12]. 旅行商问题属于 NP-hard 问题[13]，为了解决该问题，已经有许多精确算法、启发式算法、近似算法被提出. 其中最著名的精确算法为 1962 年被提出的霍普克洛夫特-卡普算法[14]，该算法求解的时间复杂度为 $O(n^2 2^n)$，是目前精确算法中理论速度最好的算法，但并不实用，且在此之后一直未得到改进提升. 旅行商问题也可以被表述为混合整数线性规划模型[15]，可以通过混合整数规划模型的求解器得到其精确或近似的解决方案.

最大割问题为给定一个完整的加权图 $G=(V,E)$，从包含顶点 $S\subset V$ 的集合中，找出一个子集，使切口 $C(S,G)=\sum_{i\in S,j\in V\setminus S}\omega_{ij}$，其中 $\omega_{ij}\in W$ 为连接顶点 i 和 j 的边的权重. 最大割问题的解决方法已经在许多实际问题中得到了应用，包括蛋白质折叠[16]、金融投资管理[17]、寻找物理学中的基态[18]等. 最大割是一个完全的 NP 难问题[19]，因此没有一个明确的多项式时间算法. 存在近似算法对其求解，包括确定性的近似算法[20]和随机的近似算法[21].

装箱问题为给定一个物品集 I，每个 $i\in I$ 的尺寸大小为 $s(i)\in\mathbb{Z}^+$，以及一个正整数容量的箱子 B. 要求将这些物品按一定的数量放入箱子中，使得每个箱子中的物品大小之和不超过箱子容量并使所用的箱子数目最少. 装箱问题有多种变体，如二维装箱、三维装箱、不同表面积的装箱等[22]，该问题在许多领域都有应用，如资源优化、物流和电路设计等[23]. 其同样也是一个完全的 NP 难问题，许多近似算法被提出以解决该问题. 首选递减和最佳匹配递减是两种简单的近似算法，首先按照成本的递减顺序对项目进行排序，然后将每个项目分配到它所适合的第一个或最完整的箱子. 这两种方法的时间复杂度都为 $O(n\log n)$[24]. 精确算法中，最早使用的是 Martello-Toth 算法，该算法原理为分支定界算法[25].

最大独立集问题为给定图 $G=(V,E)$，寻找到一个集合，该集合任意两点所构成的边都不是图 G 中的边，且 $|S|$ 最小. 该问题是一个受关注的组合优化问题，应用于分类理论、分子对接、推荐系统等[26−28]. 很容易发现，该问题中图 G 中独立集的补集是图 G 中的顶点的覆盖以及补图 $\bar{G}$ 的团. 因此，求解该问题的思路在于找出图 G 中的最小顶点覆盖或者图 $\bar{G}$ 中的最大团. 使用遍历算法求解该问题的时间复杂度为 $O(n^2 2^n)$，该算法时间复杂度被 Tarjan 等[29]提升至 $O(2^{n/3})$. 而 Xiao 等找到其最佳边界为 $O(1.199\ 6^n)$[30].

最小顶点覆盖问题为给定图 $G=(V,E)$，需找到使得每条边都被覆盖的节点最少的节点集 $S\subseteq V$，即 $(u,v)\in E\Leftrightarrow u\in S$，且 $|S|$ 最小. 该问题是一个应用于计算生物化学[31]和计算机网络安全[32]的基本问题. 求解该问题的一种方法为近似算法[33]，通过将任意一条边的两个端点添加至解中，并将这两点从图中删除.

为了使用强化学习来处理上述问题，必须将问题中涉及到的图表示为向量，并进一步提供给机器学习算法作为输入，即进行编码操作.

1.2　编码器概述

强化学习中，编码器主要是为了处理组合优化问题中的输入状态 S，将目前得到的输入状态 S 映射为 d 维的实数. 编码器的结构类型会因输入状态的变化而变化，常用的一些架构如下所示.

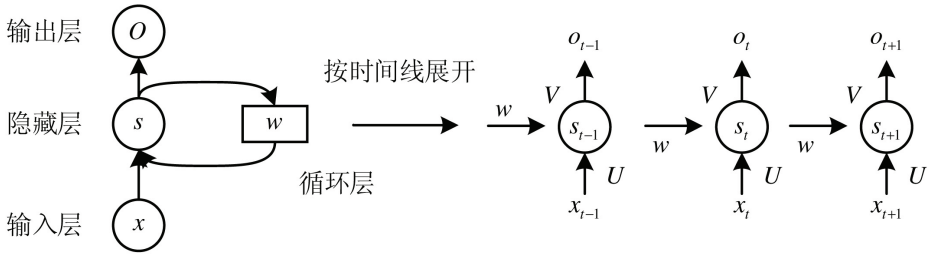

图 1　循环神经网络结构

第一个经常使用的架构为循环神经网络，循环神经网络可以对序列数据进行操作，将序列中的每个元素编码为一个向量，如图 1 所示. 该网络由成块的形式组成，它把序列的当前元素和先前的输出作为一个输入. 序

列的当前元素和前一个输出构成循环神经网络块，并输出一个向量，传递给序列的下一个元素. 例如，在旅行商问题中，可以通过对当前节点使用循环神经网络进行编码，得到其后续的一个行程. 常用的循环神经网络有长短时记忆网络[34]和门控循环单元[35].

循环神经网络的一个基本局限性与其长期依赖关系的建模有关：当模型获取最后一个时间步的输出时，它可能会丢失序列中先前保存的元素信息. 而注意力模型通过与所有输入元素形成连接来解决这个问题. 因此，注意力模型的输出取决于序列的当前元素和所有先前的元素. 特别是，输入元素和之前的每个元素之间的相似性分数（例如点积）被计算出来，这些分数被用来确定之前的每个元素对当前元素的重要性权重. 注意力模型在语言处理上能够取得不错的表现[36]，并在组合优化问题中得到了应用，例如为旅行商问题逐步建立解决方案[37].

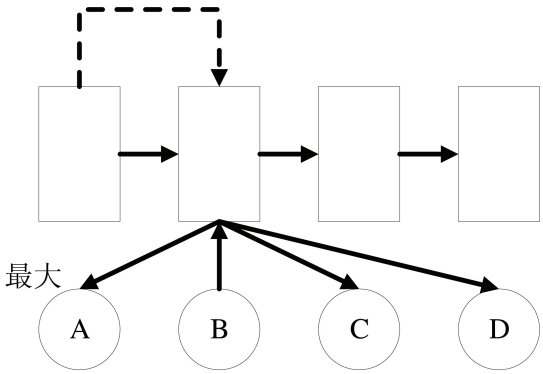

图 2 指针网络结构

值得注意的一点是注意力模型依赖于对输入结构中每一对元素之间的依赖关系进行建模，如果只有少数相关的依赖关系，则是低效的. 另一种对注意力模型简单的扩展是指针网络[38]. 指针网络不使用所有配对中的权重来计算每个输入元素的影响，而是使用权重来选择一个将被用于编码的单一输入元素. 如图 2 所示示例中，元素“A”与元素“B”的相似度最高，因此，它被用于计算元素“B”的表示（与注意力模型不同，在这种情况下，元素“C”和“D”也会被使用）.

尽管上述的模型具有足够的通用性，可以应用在各种输入状态下，但许多的组合优化问题都可以通过图的形式进行表示，使用图的形式能够对组合优化问题产生更直观的理解，而传统的神经网络只能通过节点的特征来理解图中的依赖关系，并不全面，而图神经网络则可以直接学习两个节点间的依赖关系. 首先，将节点关系通过向量进行表示；然后，每个节点的表示通过该节点的邻域结构进行更新. 在最常见的消息传递范式中，相邻的节点交换它们当前的表示，以便在下一次迭代中更新它们. 这个框架看作是注意力模型的一般化，其中的元素并不需要关注其它所有的元素，仅需关注与当前图相关的元素. 常用的图神经网络包括图卷积网络[39]、图注意力网络[40]、图同构网络[41]和图嵌入网络[42].

1.3 强化学习算法

第 1 节对马尔可夫决策过程进行了定义，其中包括了状态、动作、奖励等. 接下来，将深入介绍搜索马尔可夫决策过程最优策略的强化学习算法. 强化学习算法大体上分为基于模型和无模型两大类. 如图 3 所示.

无模型的方法更专注于环境，其中的转移函数是已知的或者可以学习的，能够在算法决策时使用的. 包括蒙特卡洛树搜索算法，如 AlphaZero[43] 和 MuZero[44].

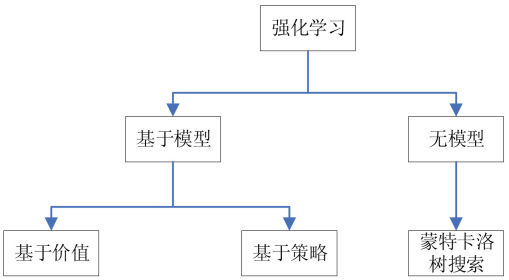

图 3 强化学习算法分类

基于模型的方法则不依赖环境的转移函数的可用性，只使用智能体所收集到的经验进行学习. 基于模型的方法按照得到解决方案的方法又可分为基于策略和基于价值两大类方法. 在基于策略的方法中，得到的为解决该问题所选择的策略，而基于价值的方法则侧重于得到一个价值函数，这个价值函数是对给定环境中状态-动作对的评价. 此外，还有对上述两种方法进行结合的方法，比较典型的是演员-评论家方法[45]. 这种方法的基本原理是演员起到产生策略的作用，而评论家近似于价值函数的作用. 通常为了做到这一点，行为者和评论家都会使用上述的基于策略和基于价值的强化学习. 这样一来，批判者提供了衡量行为者所采取的行动有多好的标准，从而可以适当地调整下一个训练步骤的可学习参数.

1.3.1　基于价值的方法

正如前面提到的，所有强化学习方法的主要目标是找到一个策略，该策略将持续允许智能体获得大量奖励. 基于价值的强化学习方法主要是通过对价值函数 $V(s)$ 和行动价值函数 $Q(s,a)$ 的近似来寻找这种策略. 在本节中，将定义这两个函数，哪些价值和动作-价值函数可以被称为最优，以及在知道最优价值函数的情况下，如何能得出最优策略.

状态 s 的价值函数是对未来折现回报的期望，当从状态 s 开始并遵循一些策略 π 时，价值函数如下所示：

$$V^{\pi}(s)=\mathrm{E}[\sum_{t=0}^{\infty}\gamma^{t}r(s_t)|\pi,s_0=s] \tag{5}$$

式中：V^{π} 为策略时的价值函数 V，在有限马尔可夫决策过程中，其最终状态的值为 0. 同时，把价值函数看作不仅取决于状态而且取决于行动的函数可能更容易理解.

动作价值函数 $Q(s,a)$ 是对未来折现回报的期望，当从状态 s 开始并遵循一些策略 π 时，价值函数如下所示：

$$Q^{\pi}(s,a)=\mathrm{E}[\sum_{t=0}^{\infty}\gamma^{t}r(s_t,a_t)|\pi,s_0=s,a_0=a] \tag{6}$$

式 (5) 很明显与式 (6) 有下列的关系：

$$V^{\pi}(s)=\max_{a}Q^{\pi}(s,a) \tag{7}$$

从价值函数的定义中产生了一个非常重要的递归属性，代表了状态 s 的价值 $V^{\pi}(s)$ 和后面可能的状态 s' 的价值 $V^{\pi}(s')$ 之间的关系，这也是许多基于价值的强化学习方法的基础. 这一特性可以用一个方程来表示，称为贝尔曼方程[46]：

$$V^{\pi}(s)=r(s)+\gamma\sum_{s'}T(s,\pi(s),s')V^{\pi}(s') \tag{8}$$

贝尔曼方程使用动作价值函数的表达形式为：

$$Q^{\pi}(s,a)=r(s)+\gamma\sum_{s'}T(s,\pi(s),s')\max_{a'}Q^{\pi}(s',a') \tag{9}$$

在本节开始时已经指出，所有强化学习任务的目标是找到一个策略，它可以积累大量的奖励. 这意味着，如果一个政策的预期收益大于另一个政策所取得的收益，那么这个政策就可以比（或等于）另一个政策更好. 除此以外，根据价值函数的定义，我们可以得到策略 $\pi'\geq\pi$ 当且仅当对于所有的状态来讲都有 $V^{\pi'}(s)\geq V^{\pi}(s)$.

得到了策略之间的这种关系，则可以确定一定有一种策略优于或等于所有其它可能的策略. 这个策略被称为最优策略 π^{*}. 显而易见，行动-价值函数和价值函数的最优性与它们所遵循的策略的最优性是密切相关的. 如果一个价值函数是所有策略中价值函数的最大值，则被称为最优价值函数. 如下所示：

$$V^{*}(s)=\max_{\pi}V^{\pi}(s),\forall s\in S \tag{10}$$

同样的，可以得到最佳行动-价值函数的定义：

$$Q^{*}(s,a)=\max_{\pi}Q^{\pi}(s),\forall s\in S,\forall a\in A \tag{11}$$

根据贝尔曼方程 (8) 和 (9)，如果行动-价值或价值函数是已知的，可以推导出最佳策略. 根据已有的最佳的价值函数 $V^{*}(s)$，可以通过做贪婪搜索找到最佳动作：选择对应于贝尔曼方程 (8) 所计算的状态 s 中的最大值 $V^{*}(s)$ 的动作. 在行动-价值函数的情况下，不需要进行单步式的搜索. 对于每个状态 s，可以很容易地找到这样的动作 a，使动作函数最大化，只需要计算 $Q^{*}(s,a)$，而不需要知道下一状态 s' 中的奖励和价值.

因此，在基于价值的方法的情况下，为了找到最佳策略，需要找到最佳价值函数. 值得注意的是，当转移函数已知的情况下，有可能明确地解出贝尔曼方程，即找到最佳价值函数. 然而在实践中，这种情况很少发生，所以需要一些方法来近似贝尔曼方程的解决方案.

基于近似值的方法的代表是 Q-learning[47] 以及其的一个变体 Deep Q-learning（DQN）[48]. 在 Q-learning 中，通过学习当前策略所产生的经验，对动作-价值函数 $Q(s,a)$ 进行迭代更新，由这种规则更新的函数收敛于最优价值函数[49].

随着深度学习的发展，神经网络已被证明在各种数据集上取得了最先进的结果，它通过高维输入学习有用的函数近似. 强化学习发展出了同深度学习相结合的 DQN，可以直接使用端到端强化学习来学习策略[50]. 该网络根据当前的输入状态为每个动作近似地计算出 Q 值. 其损失函数如下所示：

$$L(\theta_i)=\mathrm{E}_{(s,a,r,s')\sim D}[(r+\gamma\max_{a'}Q_{\theta_i}-(s',a')-Q_{\theta_i}(s,a))^2] \tag{12}$$

式中：D 为记忆回放机制容量，用来存储训练中产生的轨迹 (s,a,r,s'). 其结果是当前 Q 函数的近似值与某个最大化目标值之间的均方误差. DQN 的训练已被证明是比较稳定的，因此，DQN 对许多组合优化问题都是有效的.

1.3.2 基于策略的方法

基于价值的方法旨在找到最佳状态-动作价值函数 $Q^*(s,a)$，并对其进行贪婪的行动以获得最佳策略 π，与之不同的是，基于策略的方法试图通过对式（3）中的策略参数进行优化，直接找到由一些参数 θ 表示的最佳策略 π_θ^*：该方法收集环境中使用当前策略的经验，并利用这些收集的经验对其进行优化. 已经有许多方法来优化策略函数，接下来讨论解决组合优化问题的常用方法.

第一种为策略梯度算法，为了对式（3）中的策略参数 θ 进行优化，可以应用策略梯度定理[51]来估计策略函数的梯度，其形式如下：

$$\nabla_\theta J(\pi_\theta)=\mathrm{E}_{\pi_\theta}[\sum_{t=0}^{H}\nabla_\theta\log\pi_\theta(a_t|s_t)\hat{A}(s_t,a_t)] \tag{13}$$

式中：$\hat{A}(s_t,a_t)=\sum_{t=t'}^{H}\gamma^{t'-t}r(s_t',a_t')-b(s_t)$ 为回报估计，H 为智能体的回合次数，$b(s)$ 为基线函数，该策略使用梯度下降算法的梯度来优化参数 θ.

第二种为 REINFORCE 算法，基线 $b(s)$ 的作用为减少回报估计的方差 $\hat{A}(s_t,a_t)$，由于它是通过运行当前的策略 π_θ 计算出来的，初始参数可能导致训练开始时的性能不佳，基线 $b(s)$ 试图通过减少方差来缓解这种情况. 当 $b(s)$ 被排除在回报估计计算之外时，得到了一个 REINFORCE 算法[52]. 另外，可以通过计算采样轨迹的平均奖励来计算基线值 $b(s_t)$，或者使用参数化的价值函数估计器 $V_\phi(s_t)$.

第三种为 Actor-Critic 算法，该算法通过使用自举法从后续状态-价值估计的值中更新状态值，进一步扩展了 REINFORCE 与基线. 一个常见的方法是使用参数值函数来计算每一步的收益估计值：

$$\hat{A}(s_t,a_t)=r(s_t,a_t)+V_\phi(s_t')-V_\phi(s_t) \tag{14}$$

虽然这种方法会给梯度估计带来偏差，但它往往能进一步减少方差. 此外，这种方法可以应用于在线和持续学习，因为它们不再依赖蒙特卡洛滚动，即把轨迹展开到一个终端状态.

第四种为近端策略优化算法，第三种强化学习算法的进一步发展推出了一些更先进的方法，如近端策略优化（PPO）[53]，该算法在策略空间中以约束条件进行策略更新，试图学习一个参数化的状态-行动价值函数 $Q_\phi(s,a)$，对应于当前策略，并使用它来计算引导的回报估计. PPO 算法公式如下：

$$J_{\mathrm{PPO}}^{\theta'}(\theta)=J^{\theta'}(\theta)-\beta KL(\theta,\theta') \tag{15}$$

$$J^{\theta'}(\theta)=E_{(s_t,a_t)\sim\pi_{\theta'}}[\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)}A^{\theta'}(s_t,a_t)] \tag{16}$$

1.3.3 蒙特卡洛树搜索

基于价值的方法和基于策略的方法都不使用环境的模型（无模型的方法），即模型的转移概率，因此，这种方法不通过展开环境到下一个步骤. 然而，为组合优化问题定义一个马尔可夫决策过程是可行的，可以利用

环境的知识，通过提前几步计划来提高预测能力. 蒙特卡洛树搜索算法的一般程序包括选择、扩展、模拟和回溯[54]，如图 4 所示. 其中，不是通过进行遍历来评估树中的叶子节点，而是使用神经网络 f_θ 来为树中的新节点提供策略 $P(s,*)$ 和状态-价值估计 $V(s)$. 树中的节点表示状态 s，每条边指代动作 a. 在选择阶段，从根状态 s_0 开始，不断选择下一个状态，使置信度上限最大化，公式如下所示：

$$UCB = Q(s,a) + c \cdot P(s,a) \cdot \frac{\sqrt{\sum_{a'} N(s,a')}}{1+N(s,a)} \tag{17}$$

当遇到搜索树中以前未见过的节点时，对该节点的策略 $P(s,*)$ 、状态价值函数以及状态-价值估计 $V(s)$ 进行估计. 之后，$V(s)$ 的估计值沿着搜索树向后传播，更新 $Q(s,a)$ 和 $N(s,a)$ 的值. 经过若干次搜索迭代后，根据改进的策略从根状态中选择下一个动作，公式如下所示：

$$\pi(s_0) = \frac{N(s_0,a)}{\sum_{a'} N(s_0,a')} \tag{18}$$

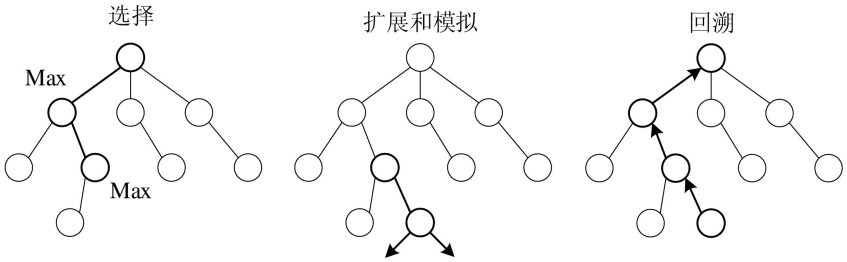

图 4　蒙特卡洛树搜索操作

2　强化学习求解组合优化问题

本节对近年来现有的解决组合优化问题的强化学习方法进行了介绍，这些问题包括旅行商问题、最大剪问题、装箱问题、最小顶点覆盖问题和最大独立集问题. 对解决这些问题的算法进行了罗列对比.

2.1　旅行商问题

Bello 等[55]最早将策略梯度算法应用于组合优化问题. 在解决旅行商问题时，其马尔可夫决策过程的表现形式为：状态是一个 d 维的图嵌入向量，代表节点在时间 t 的当前行程，而动作是挑选另一个节点，该节点在当前状态下没有被使用过. 初始状态 s_0 即为初始节点的嵌入表示. 在这种情况下，转移函数 $T(s,a,s')$ 会返回所构建的旅行的下一个节点，直到所有的节点都被访问过. 最后，其奖励函数与总的行程进行负相关. 指针网络被用来对输入序列进行编码，而解决方案则是利用解码器的指针机制从输入的分布中依次构建的，并进行并行异步训练.

Dai 等[56]的工作又对其进行了提升，在状态、动作等改变不大的情况下，将奖励进行了改进. 奖励被定义为从状态 s 过渡到状态 s' 后，采取某种动作 a 时，成本函数的差异，即 $r(s,a) = c(h(s'),G) - c(h(s),G)$，其中：$h$ 为图嵌入函数，G 为该问题的图表示，c 为代价函数. 除此以外，在神经网络中使用了图嵌入网络进行了编码工作，DQN 作为网络参数更新的强化学习算法.

受到 Bello 等[55]的启发，Nazari 等[57]使用强化学习求解了旅行商问题的一种延续问题——车辆路径问题. Bello 等[55]提出的方法不能直接应用于解决车辆路径问题，因为它具有动态性质，即一旦节点被访问，该节点的需求就会变成零，因为它嵌入了输入的顺序和静态性质. Nazari 等[57]扩展了以前用于解决 TSP 的方法来规避这个问题，并找到了车辆路径问题及其随机变体的解决方案. 其状态设置为一个包含两元素的向量，其中一个值为当前时刻的位置坐标，另一个值为当前时刻的任务需求. 动作设置为选定下一个将要去的节点. 奖励则与旅行商问题中使用的奖励类似，它是负的总路线长度，只有在所有客户的需求得到满足后才会给智能体. 除此以外，编码器被简化，用 1-d 卷积嵌入层取代长短时记忆单元，使模型对输入序列的顺序不变，从而能够处理动态状态变化. 然后，通过使用 REINFORCE 算法进行策略学习，而对于随机的车辆路径问题，则使用 A3C 算法进行训练.

与 Nazari 等[57]的工作类似，Deudon 等[58]采用了相同的方法，但改变了编码器和解码器的网络结构. 其图神经网络的编码器架构不包括长短时记忆单元，而是完全基于注意力机制，因此输入被编码为一个集合而不是一个序列. 解码网络仍然使用指针网络. 此外，作者还研究了将强化学习智能体提供的解决方案与 2-Opt 启发式方法相结合，以进一步改善得到的结果. 带有批评基线的 REINFORCE 算法被用来更新所描述的编码解码器的网络参数.

Kool 等[59]提出了一种构造启发式学习方法，以解决旅行商问题和车辆路径问题的两种变体. 其编码器使用了一种类似于 Transformer 的基于注意力的编码器，而解码器则与指针网络相类似.

Liu 等[60]提出了一种改进的强化突变遗传算法，命名为 RMGA，用于解决 TSP. RMGA 的核心在于使用异构配对选择 EAX 中的随机配对选择和强化突变算子（RLM）的构造，方法是修改 Q 学习算法并将其应用于修改后生成的个体 EAX. TSP 中对小型和大型 TSP 实例的实验结果表明，RMGA 几乎每次都能在合理的时间内获得最佳游览，在解决方案质量和运行时间方面优于已知的 EAX-GA 和 LKH.

Cappart 等[61]的工作结合了两种解决带时间窗口的旅行商问题的方法，即强化学习方法和约束编程方法，以便学习分支策略. 为了对组合优化问题进行编码，作者提出了一个动态编程公式，作为两种技术之间的桥梁. 其状态 s 是由三种元素组成的向量，包括仍需访问的剩余城市集、已访问的最后一个城市以及当前时间. 其动作为选择一个相应的城市. 奖励为两个城市花费时间的负相关. 然后，这个马尔可夫决策过程可以转化为一个动态编程模型. 通过 DQN 和 PPO 算法进行训练.

Drori 等[62]的工作与之前的工作不同，他们的方法针对特殊的问题给出解决方案. 相比之下，这项工作为无模型强化学习提供了一个通用框架，使用图神经网络表示，通过改变奖励来适应不同的问题类别. 这个框架通过使用边缘到顶点的线图来模拟问题，并将其制定为一个单人游戏框架. 将策略表示为具有基于注意力的解码器的图嵌入网络编码器，在树形搜索过程中训练学习.

Xu 等[63]考虑到现有的强化学习模型都是简单地聚合节点嵌入来生成前后关联的嵌入，而没有考虑到动态的网络结构，使得它们不能对动态的状态转换和行动选择进行建模. 其开发了一个新的基于注意力的强化学习模型，通过批量归一化重排和门聚合提供增强的节点嵌入，以及通过对多个关系结构的注意力聚合模块提供动态感知的嵌入. 通过实验表明，其设计的模型不仅优于基于学习的基线，而且解决问题的速度也比传统基线快很多. 此外，在评估大规模问题以及不同数据分布的问题时显示出更好的通用性.

Zhang 等[64]通过结合熵正则化最优传输技术与强化学习算法以解决旅行商问题，其将熵正则化技术整合为编码器网络的一个层，并构建为一个能够在没有监督和推论的情况下学习的模型，其速度明显快于目前的自回归方法. 通过实验评估了在深度学习模型中包含最优传输算法的优势. 旅行商问题总结见表 1.

表 1　强化学习求解旅行商问题总结

作者	编码器	强化学习算法
Bello 等[55]	指针网络	REINFORCE with baseline
Dai 等[56]	图嵌入网络	DQN
Nazari 等[57]	带有卷积的指针网络	REINFORCE
Deudon 等[58]	带有注意力机制的指针网络	REINFORCE with baseline
Kool 等[59]	带有注意力机制的指针网络	REINFORCE with baseline
Liu 等[60]	前馈神经网络	DQN
Cappart 等[61]	图注意力网络	DQN
Drori 等[62]	带有注意力机制的图嵌入网络	MCTS
Xu 等[63]	多重注意力机制的神经网络	Policy Gradient
Zhang 等[64]	多头注意力机制的神经网络	REINFORCE

2.2　最大剪问题

Dai 等[56]首次将强化学习应用于求解最大剪问题，提出了通过结合图嵌入和 Q-learning 来构造启发式的原则性方法，他们将该问题表述为马尔可夫决策过程，其中状态空间 s 被定义为问题的部分解决方案，即添加到图中的所有节点的子集，该子集能使最大切口最大化. 动作空间为一组不在当前状态下的节点. 转移函数

$T(s_{t+1}|s_t,a_t)$ 是确定性的，对应于用特征 $x_v=1$ 标记最后选定的节点. 奖励的计算方式是立即改变减重，当减重无法通过进一步的行动得到改善时，该回合就会终止. 图嵌入网络被用作状态的编码器. Q-learning 算法的一个变种被用来学习构建解决方案，该算法在随机生成的图形实例上进行了训练. 与常用的启发式解决方案相比，这种方法实现了更好的近似率以及泛化能力，这一点已经通过在由 50~100 个节点组成的图上的训练和在多达 1 000~1 200 个节点的图上的测试得到了证明，实现了与精确解决方案非常好的近似率.

Barrett 等[65]在 Dai 等[56]的工作基础上，提出 ECO-DQN 算法，在近似率以及泛化方面进行了改进. 该算法保持了原算法的一般框架，但是引入了一些修改. 智能体被允许从部分构建的解决方案中删除顶点，以更好地探索解决方案空间. 奖励函数也作出改进，以提供一个正常化的增量奖励当找到一个比迄今为止的回合中看到的更好的解决方案，以及给予较小的奖励当找到一个在回合中尚未看到的局部最优解决方案. 此外，对降低切割值没有添加惩罚措施. 状态编码器的输入被修改以适应奖励函数的变化. 因为在这种情况下，智能体已经能够无限期地探索，所以情节长度被设定为 $2|V|$. 作者允许算法从一个任意的状态开始，这对于将这种方法与其它方法（例如启发式方法）结合起来是很有效的. 该方法显示了比原本算法更好的近似率，以及更好的泛化能力.

Cappart 等[66]通过将强化学习纳入决策图框架来学习启发式，设计了解决最大切割问题的联合方法. 强化学习的整合允许通过学习变量排序的启发式方法以向决策图提供更严格的目标函数界限的解决方案. 状态空间 s 被表示为一组选定变量的有序序列以及部分构建的决策图. 动作空间 a 由未被选定的变量组成. 转移函数则将已选用的变量加入已用集合与决策图中. 奖励函数旨在缩小决策图的界限，并被编码为变量加入集合以改进相对上限和下限. 实验表明，他们的方法优于几个排序启发式方法，对较大的图实例有很好的泛化作用，但没有与其它基于强化学习的方法进行比较.

Tang 等[67]将强化学习框架同切割面方法进行结合，其状态空间包括了原始的线性约束和迄今为止所添加的切割，动作空间则使用了割平面法，转移函数将所选择的切割添加到问题中，从而形成一个新的状态. 奖励函数被定义为两个连续线性问题解决方案的目标函数之差. 策略梯度算法被用来选择新的割平面，并用基于注意力机制的长短时记忆网络对状态进行编码. 该算法使用进化策略对生成的图实例进行了训练，与通常用于选择割平面的启发式方法相比，其已被证明能够提高切割的效率、完整性差距和概括性.

Gu 等[68]使用了带有指针网络的 Actor-Critic 算法迭代构建出了解决方案. 状态空间定义为一个对称矩阵，其值是节点之间的边缘权重. 该矩阵以每列的形式作为输入进入到指针网络，由指针网络产生当前时刻的动作，以指向输入向量的指针形式，加上一个特殊的序列结束符号“EOS”. 由此产生的“EOS”符号分隔的节点序列代表了问题的解决方案，并从中计算出奖励. 作者用多达 300 个节点的模拟图进行了实验，并产生了相当好的近似率，但并没有与以前的工作或已知的启发式方法进行比较. 最大剪问题总结见表 2.

表 2　强化学习求解最大剪问题总结

作者	编码器	强化学习算法
Dai 等[56]	图嵌入网络	DQN
Barrett 等[65]	图嵌入网络	DQN
Cappart 等[66]	图嵌入网络	DQN
Tang 等[67]	带有注意力机制的长短时记忆网络	Policy Gradient + ES
Gu 等[68]	指针网络	A3C

2.3　装箱问题

Zhao 等[69]解决的是一个具有挑战性但实际有用的三维垃圾箱包装问题（3D-BPP）的变体. 智能体对要装入一个仓的物品的信息有限，而且一个物品必须在到达后立即被装入，不需要缓冲或重新调整. 物品的摆放也受制于顺序依赖和物理稳定性. 其将此在线 3D-BPP 表述为一个受限的马尔可夫决策过程（CMDP）. 为了解决这个问题，提出了一个有效的、易于实施的、在行为人批判框架下的约束性深度强化学习方法. 特别是，引入了一个预测和投射方案. 代理人首先预测安置行动的可行性掩码，作为一项辅助任务，然后使用掩码来调节训练期间演员输出的行动概率. 这样的监督和投射有利于智能体有效地学习可行的政策. 方法可以很容易地扩展到处理超前物品、多仓包装和物品重新定位.

Duan 等[70]试图解决上述中方法的局限性，具体来说，作者提出构建一个端到端的通道，通过使用注意力机制来选择一个物品、方向和位置坐标. 其状态空间包括物品是否被包装的标志位、物品的尺寸以及相对箱子的坐标. 动则空间则包括物品的选择、旋转和物品在垃圾箱中的位置. 奖励函数是递增的，通过箱中的空闲体积进行计算，即当前箱子的体积减去包装物品的体积. 训练算法选用了 Actor-Critic 算法. 与遗传算法和以前的强化学习方法相比，对于大小不超过 30 个物品的问题，所提出的方法实现了较小的箱体间隙比率.

Jiang 等[71]通过设计端到端的多模态的强化学习智能体解决了三维装箱问题，通过多模态的处理，将整体的任务分为三个子任务，包括顺序、方向和位置，智能体将会按照顺序解决三个子任务. 其状态空间由箱体状态和视图状态构成，箱体状态主要包含了箱体的相关信息，视图状态则确定了箱体的整体位置，将原本的三维进行了降维处理. 动作空间的输出则是通过解码器直接进行位置选择. 在算法选择中，选用了广义优势估计同 A2C 算法进行结合. 作者通过设计的多模态强化学习算法，解决了原本只能处理 50 个箱体问题的局限性，使智能体能够解决 100 个箱体及更大的实例. 装箱问题总结见表 3.

表 3 强化学习求解装箱问题总结

作者	编码器	强化学习算法
Zhao 等[69]	前馈神经网络	REINFORCE
Duan 等[70]	注意力机制	Actor-Critic
Jiang 等[71]	卷积神经网络	A2C

2.4 最小顶点覆盖问题

Song 等[72]提出了在分类领域得到普及的联合协同训练方法，以构建顺序性策略. 其使用了两种学习策略：第一种参考了 Dai 等[56]描述的方法，第二种是通过分支定界法解决的整数线性编程方法. 所提出的算法在直觉上类似模仿学习，算法中的两个策略产生出两个政策，并对其进行估计以找出哪个政策更好，在它们之间交换信息，最后，进行算法更新. 所进行的实验中，列出了与 S2V-DQN、模仿学习和 Gurobi 求解器的比较，并显示在 500 个节点以下的问题上有较小的优化差距. 最小顶点覆盖问题总结见表 4.

表 4 强化学习求解最小顶点覆盖问题总结

作者	编码器	强化学习算法
Song 等[72]	图嵌入网络	DQN + Imitation Learning

2.5 最大独立集问题

Cappart 等[66]首先使用强化学习解决了最大独立集问题. 通过强化学习算法寻找决策图中变量的最佳排序，以缩小该问题中的松弛边界. 具体的实施方法与 2.2 节中相同. Alipour 等[73]提出了一种多智能体强化学习方法以解决最大独立集问题，该方法中，每个智能体试图自主解决最大独立集问题，并使用强化学习改善其局部行为，通过分享成功的结果直接与其它智能体进行交流. 同时，2-1 局部搜索被用于进一步改进每一步的解决方案. 实验在多组实例上进行测试，结果表明所提方法能够竞争甚至超过一些最先进的算法. 最大独立集问题总结见表 5.

表 5 强化学习求解最大独立集问题总结

作者	编码器	强化学习算法
Cappart 等[66]	图嵌入网络	DQN
Alipour 等[73]	前馈神经网络	MARL

3 结论与展望

前面几节已经介绍了利用强化学习算法解决典型组合优化问题的几种方法. 由于这个领域已经证明了能够与最先进的启发式方法和求解器相媲美的性能，我们期望在以下方向继续深入研究.

1）扩大组合优化问题的求解范围. 当前的主要问题之一在于求解问题所作的实验对比的数量有限. 实际的组合优化问题所涉及到的数学问题非常庞大，而目前的方法往往需要针对一组具体的问题来实施. 尽管强化学习算法能够将学到的策略应用到未训练过的问题上，但是这些问题往往是同一问题的较小实例，或者具有不同分布的问题实例，甚至是来自另一组组合优化问题的实例. 如何能够从当前已经训练好的算法出发，解决更多类型的组合优化问题是一个很好的方向.

2）提高算法求解质量. 本文所介绍的许多研究工作，与商业求解器相比，表现出了卓越的性能，其中一些还实现了与最佳解决方案或启发式算法实现的解决方案的求解质量相等. 然而，这些结果在更加复杂的组合优化问题中往往表现欠佳，因此，将经典的组合优化算法与强化学习方法进行结合是很好的发展方向.

3）从算法模型上进行改进. 目前的研究中，许多优化模型是以端到端的方法进行求解，其解的质量较差，

大部分文献都需要进一步通过波束搜索、局部搜索、采样策略等方式进一步提升解的质量，这说明当前的模型仍然有很大的提升空间，未来需要进一步对求解组合优化问题的深度神经网络模型进行研究.

参考文献：

[1] BELLMAN R. A Markovian decision process[J]. Journal of Mathematics and Mechanics, 1957, 6: 679-684.

[2] 章炯民, 陶增乐, 吴文娟. 组合优化问题的神经网络解——装箱问题和背包问题的求解[J]. 华东师范大学学报(自然科学版), 1998, 4: 102-105.

[3] ZHANG C, SONG W, CAO Z, et al. Learning to dispatch for job shop scheduling via deep reinforcement learning[J]. Advances in Neural Information Processing Systems, 2020, 33: 1621-1632.

[4] ZHUANG W W, CHEN C, QIU G X. A new deep reinforcement learning model for dynamic portfolio optimization[J]. Journal of University of Science and Technology of China, 2022, 52(11): 15-28.

[5] RUSKONE-FOURMESTRAUX A, DELMER A, LAVERGNE A, et al. Multiple lymphomatous polyposis of the gastrointestinal tract: prospective clinicopathologic study of 31 cases[J]. Gastroenterology, 1997, 112(1): 7-16.

[6] MANUAL C U. IBM ILOG CPLEX optimization studio[J]. Version, 1987, 12: 1-586.

[7] MAHER S, MILTENBERGER M, PEDROSO J P, et al. PySCIPOpt: mathematical programming in python with the SCIP optimization suite[C]//Mathematical Software-ICMS 2016: 5th International Conference. Berlin: Springer, 2016.

[8] BELLM E C, KULKARNI S R, BARLOW T, et al. The zwicky transient facility: surveys and scheduler[J]. Publications of the Astronomical Society of the Pacific, 2019, 131(1000): 068003.

[9] GRAY M A. Sage: a new mathematics software system[J]. Computing in Science and Engineering, 2008, 10(6): 72-75.

[10] DANTZIG G B. Linear programming[J]. Operations Research, 2002, 50(1): 42-47.

[11] BALAS E. Projection, lifting and extended formulation in integer and combinatorial optimization[J]. Annals of Operations Research, 2005, 140(1): 125.

[12] FLOOD M M. The traveling-salesman problem[J]. Operations Research, 1956, 4(1): 61-75.

[13] WONG R T. Combinatorial optimization: algorithms and complexity (christos h. papadimitriou and kenneth steiglitz)[J]. SIAM Review, 1983, 25(3): 424-425.

[14] HELD M, KARP R M. A dynamic programming approach to sequencing problems[J]. Journal of the Society for Industrial and Applied Mathematics, 1962, 10(1): 196-210.

[15] DANTZIG G, FULKERSON R, JOHNSON S. Solution of a large-scale traveling-salesman problem[J]. Journal of the Operations Research Society of America, 1954, 2(4): 393-410.

[16] PERDOMO-ORTIZ A, DICKSON N, DREW-BROOK M, et al. Finding low-energy conformations of lattice protein models by quantum annealing[J]. Scientific Reports, 2012, 2(1): 1-7.

[17] AOUNI B, COLAPINTO C, LA TORRE D. Financial portfolio management through the goal programming model: current state-of-the-art[J]. European Journal of Operational Research, 2014, 234(2): 536-545.

[18] BARAHONA F. On the computational complexity of Ising spin glass models[J]. Journal of Physics A: Mathematical and General, 1982, 15(10): 3241-3253.

[19] JONES N D. Space-bounded reducibility among combinatorial problems[J]. Journal of Computer and System Sciences, 1975, 11(1): 68-85.

[20] PAPA D A, MARKOV I L. Hypergraph partitioning and clustering[J]. Handbook of Approximation Algorithms and Metaheuristics, 2007, 20073547: 1-15.

[21] GOEMANS M X, WILLIAMSON D P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming[J]. Journal of the ACM, 1995, 42(6): 1115-1145.

[22] WU Y, LI W K, GOH M, et al. Three-dimensional bin packing problem with variable bin height[J]. European Journal of Operational Research, 2010, 202(2): 347-355.

[23] VARNAMKHASTI M J. Overview of the algorithms for solving the multidimensional knapsack problems[J]. Advanced Studies in Biology, 2012, 4(1): 37-47.

[24] BLUM C, ROLI A. Metaheuristics in combinatorial optimization: overview and conceptual comparison[J]. ACM Computing Surveys, 2003, 35(3): 268-308.

[25] MARTELLO S, TOTH P. Lower bounds and reduction procedures for the bin packing problem[J]. Discrete Applied Mathematics, 1990, 28(1): 59-70.

[26] FEO T A, RESENDE M G C, SMITH S H. A greedy randomized adaptive search procedure for maximum independent set[J]. Operations Research, 1994, 42(5): 860-878.

[27] GARDINER E J, WILLETT P, ARTYMIUK P J. Graph-theoretic techniques for macromolecular docking[J]. Journal of Chemical Information and Computer Sciences, 2000, 40(2): 273-279.

[28] AGRAWAL R, MANNILA H, SRIKANT R, et al. Fast discovery of association rules[J]. Advances in Knowledge Discovery and Data Mining, 1996, 12(1): 307-328.

[29] TARJAN R E, TROJANOWSKI A E. Finding a maximum independent set[J]. SIAM Journal on Computing, 1977, 6(3): 537-546.

[30] XIAO M, NAGAMOCHI H. Exact algorithms for maximum independent set[J]. Information and Computation, 2017, 255: 126-146.

[31] LANCIA G, BAFNA V, ISTRAIL S, et al. SNPs problems, complexity, and algorithms[C]//European Symposium on Algorithms. Berlin: Springer, 2001.

[32] FILIOL E, FRANC E, GUBBIOLI A, et al. Combinatorial optimisation of worm propagation on an unknown network[J]. International Journal of Computer Science, 2007, 2(2): 124-130.

[33] 闫兴篡, 殷建平, 蔡志平, 等. 求图的最小顶点覆盖集的一个近似算法[J]. 哈尔滨工业大学学报, 2008, 40(7): 1131-1135.

[34] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[35] DEY R, SALEM F M. Gate-variants of gated recurrent unit(GRU) neural networks[C]//2017 IEEE 60th International Midwest Symposium on Circuits and Systems(MWSCAS). Boston: IEEE, 2017.

[36] SUBAKAN C, RAVANELLI M, CORNELL S, et al. Attention is all you need in speech separation[C]//2021 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). Online: IEEE, 2021.

[37] OUYANG W, WANG Y, HAN S, et al. Improving generalization of deep reinforcement learning-based TSP solvers[C]//2021 IEEE Symposium Series on Computational Intelligence(SSCI). Online: IEEE, 2021.

[38] FEI H, JI D, LI B, et al. Rethinking boundaries: end-to-end recognition of discontinuous mentions with pointer networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Online: AAAI, 2021.

[39] WU F, SOUZA A, ZHANG T, et al. Simplifying graph convolutional networks[C]//International Conference on Machine Learning. Long Beach: PMLR, 2019.

[40] XIE Y, ZHANG Y, GONG M, et al. Mgat: multi-view graph attention networks[J]. Neural Networks, 2020, 132: 180-189.

[41] PENG Y, LIN Y, JING X Y, et al. Enhanced graph isomorphism network for molecular admet properties prediction[J]. IEEE Access, 2020, 8: 168344-168360.

[42] DAI H, DAI B, SONG L. Discriminative embeddings of latent variable models for structured data[C]//International Conference on Machine Learning. New York: PMLR, 2016.

[43] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

[44] SCHRITTWIESER J, ANTONOGLOU I, HUBERT T, et al. Mastering Atari, Go, chess and shogi by planning with a learned model[J]. Nature, 2020, 588(7839): 604-609.

[45] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C]//International Conference on Machine Learning. New York: PMLR, 2016.

[46] BELLMAN R. On the theory of dynamic programming[J]. Proceedings of the National Academy of Sciences, 1952, 38(8): 716-719.

[47] WATKINS C J C H, DAYAN P. Q-learning[J]. Machine Learning, 1992, 8(3): 279-292.

[48] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[49] SUTTON R S. Learning to predict by the methods of temporal differences[J]. Machine Learning, 1988, 3(1): 9-44.

[50] GAO G, JIN R. An end-to-end flow control method based on DQN[C]//2022 International Conference on Big Data, Information and Computer Network(BDICN). Sanya: IEEE, 2022.

[51] PETERS J, VIJAYAKUMAR S, SCHAAL S. Reinforcement learning for humanoid robotics[C]//Proceedings of the Third IEEE-RAS International Conference on Humanoid Robots. Karlsruhe: IEEE, 2003.

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

[53] WANG Y, HE H, TAN X. Truly proximal policy optimization[C]//Uncertainty in Artificial Intelligence. Online: PMLR, 2020.

[54] BROWNE C B, POWLEY E, WHITEHOUSE D, et al. A survey of Monte Carlo tree search methods[J]. IEEE Transactions on

Computational Intelligence and AI in Games, 2012, 4(1): 1-43.

[55] BELLO I, PHAM H, LE Q V, et al. Neural combinatorial optimization with reinforcement learning[C]//Proceedings of the 5th International Conference on Learning Representations(ICLR 2017). Toulon: ICLR, 2017.

[56] DAI H J, KHALIL E B, ZHANG Y Y, et al. Learning combinatorial optimization algorithms over graphs[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017.

[57] NAZARI M, OROOJLOOY A, TAKAC M, et al. Reinforcement learning for solving the vehicle routing problem[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montreal: Curran Associates Inc., 2018.

[58] DEUDON M, COURNUT P, LACOSTE A, et al. Learning heuristics for the TSP by policy gradient[C]//International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Cham: Springer, 2018.

[59] KOOL W, VAN HOOF H, WELLING M. Attention, learn to solve routing problems![C]//Proceedings of the 7th International Conference on Learning Representations(ICLR 2019). New Orleans: ICLR, 2019.

[60] LIU F, ZENG G. Study of genetic algorithm with reinforcement learning to solve the TSP[J]. Expert Systems with Applications, 2009, 36(3): 6995-7001.

[61] CAPPART Q, MOISAN T, ROUSSEAU L M, et al. Combining reinforcement learning and constraint programming for combinatorial optimization[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Online: AAAI, 2021.

[62] DRORI I, KHARKAR A, SICKINGER W R, et al. Learning to solve combinatorial optimization problems on real-world graphs in linear time[C]//19th IEEE International Conference on Machine Learning and Applications(ICMLA). Online: IEEE, 2020.

[63] XU Y, FANG M, CHEN L, et al. Reinforcement learning with multiple relational attention for solving vehicle routing problems[J]. IEEE Transactions on Cybernetics, 2021, 52(10): 11107-11120.

[64] ZHANG R, PROKHORCHUK A, DAUWELS J. Deep reinforcement learning for traveling salesman problem with time windows and rejections[C]//International Joint Conference on Neural Networks(IJCNN). Online: IEEE, 2020.

[65] BARRETT T, CLEMENTS W, FOERSTER J, et al. Exploratory combinatorial optimization with reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Online: AAAI, 2020.

[66] CAPPART Q, GOUTIERRE E, BERGMAN D, et al. Improving optimization bounds using machine learning: decision diagrams meet deep reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019.

[67] TANG Y, AGRAWAL S, FAENZA Y. Reinforcement learning for integer programming: learning to cut[C]//International Conference on Machine Learning. Online: PMLR, 2020.

[68] GU S, YANG Y. A deep learning algorithm for the max-cut problem based on pointer network structure with supervised learning and reinforcement learning strategies[J]. Mathematics, 2020, 8(2): 298.

[69] ZHAO H, SHE Q, ZHU C, et al. Online 3D bin packing with constrained deep reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Online: AAAI, 2021.

[70] DUAN L, HU H, QIAN Y, et al. A multi-task selected learning approach for solving 3D flexible bin packing problem[C]//Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems. Montreal: IFAAMAS, 2019.

[71] JIANG Y, CAO Z, ZHANG J. Solving 3D bin packing problem via multimodal deep reinforcement learning[C]//Proceedings of the 20th International Conference on Autonomous Agents and Multi-Agent Systems. Auckland: IFAAMAS, 2021.

[72] SONG J, LANKA R, YUE Y, et al. Co-training for policy learning[C]//Uncertainty in Artificial Intelligence. Toronto: PMLR, 2020.

[73] ALIPOUR M M, ABDOLHOSSEINZADEH M. A multiagent reinforcement learning algorithm to solve the maximum independent set problem[J]. Multiagent and Grid Systems, 2020, 16(1): 101-115.

责任编辑：张自强