文章编号:2095-6134(2024)02-0275-10

Actor-critic 框架下的二次指派问题求解方法*

李雪源,韩丛英†

(中国科学院大学数学科学学院,北京 100049)

(2022 年 1 月 30 日收稿; 2022 年 4 月 1 日收修改稿)

Li X Y, Han C Y. Solving quadratic assignment problem based on actor-critic framework[J]. Journal of University of Chinese Academy of Sciences, 2024, 41(2): 275-284. DOI: 10.7523/j.ucas.2022.031.

摘 要 二次指派问题(QAP)属于 NP-hard 组合优化问题,在现实生活中有着广泛应用。目前相对成熟的启发式算法通常以问题为导向来设计定制化算法,缺乏迁移泛化能力。为提供一个统一的 QAP 求解策略,将 QAP 问题的流量矩阵及距离矩阵抽象成两个无向完全图并构造相应的关联图,从而将设施和地点的指派任务转化为关联图上的节点选择任务,基于 actor-critic 框架,提出一种全新的求解算法 ACQAP。首先,利用多头注意力机制构造策略网络,处理来自图卷积神经网络的节点表征向量;然后,通过 actor-critic 算法预测每个节点被作为最优节点输出的概率;最后,依据该概率在可行时间内输出满足目标奖励函数的动作决策序列。该算法摆脱人工设计,且适用于不同规模的输入,更加灵活可靠。实验结果表明,在 QAPLIB 实例上,本算法在精度媲美传统启发式算法的前提下,迁移泛化能力更强;同时相对于 NGM 等基于学习的算法,求解的指派费用与最优解之间的偏差最小,且在大部分实例中,偏差均小于 20%。

关键词 二次指派问题;图卷积神经网络;深度强化学习;多头注意力机制;actor-critic 算法

中图分类号:TP391 **文献标志码**:A **DOI**:10.7523/j.ucas.2022.031

Solving quadratic assignment problem based on actor-critic framework

LI Xueyuan, HAN Congying

(*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract The quadratic assignment problem (QAP) is one of the NP-hard combinatorial optimization problems and is known for its diverse applications in real life. The current relatively mature heuristic algorithms are usually problem-oriented to design customized algorithms and lack the ability to transfer and generalize. In order to provide a unified QAP solution strategy, this paper abstracts the flow matrix and distance matrix of QAP problem into two undirected complete graphs and constructs corresponding correlation graphs, thus transforming the assignment task of facilities and locations into node selection task on the association graph. Based on actor-critic

* 国家重点研发计划专项(2021YFA1000403)、国家自然科学基金(11991022)和中国科学院战略性先导科技专项(XDA27000000)资助
† 通信作者,E-mail: hancy@ucas.ac.cn

framework, this paper proposes a new algorithm ACQAP (actor-critic for QAP). Firstly, the model uses a multi-headed attention mechanism to construct a policy network to process the node representation vectors from the graph convolutional neural network; Then, the actor-critic algorithm is used to predict the probability of each node being output as the optimal node. Finally, the model outputs an action decision sequence that satisfies the objective reward function within a feasible time. The algorithm is free from manual design and is more flexible and reliable as it is applicable to different sizes of inputs. The experimental results show that on QAPLIB instances, the algorithm has stronger transfer and generalization ability under the premise that the accuracy is comparable to the traditional heuristic algorithm, while the assignment cost for solving is less compared to the latest learning-based algorithms such as NGM, and the deviation is less than 20% in most instances.

Keywords quadratic assignment problem; graph convolutional neural network; deep reinforcement learning; multi-head-attention mechanism; actor-critic algorithm

二次指派问题(quadratic assignment problem, QAP)是一个典型的组合优化问题,同时也是一个NP-hard问题。自1957年Koopmans等[1]首次将QAP作为组合优化问题提出之后,便一直受到数学、计算机及诸多应用领域学者的关注。一方面,QAP在实际中应用广泛,许多现实问题都可以形式化为QAP,如集成电路布线[2-3]、工厂位置布局[4]、打字机键盘设计、作业调度[5-6]等。另一方面,一些经典的NP-hard组合优化问题,如旅行商问题、三角剖分问题以及最大团问题等也可以转化为QAP[7-9]。因此,寻找一种有效求解QAP的算法具有十分重要的理论研究意义和实际应用价值。

传统求解QAP的方法可以分为精确算法和近似算法两大类。精确算法能够找到全局最优解,但随着问题规模的扩大所需时间急剧增加,不适用于实际的应用。近似算法是以精度换时间,旨在合理的计算时间内找到尽可能接近最优解的一个可行解。启发式算法作为典型的近似算法,存在迁移性不强的问题,一旦问题设置发生变化,原来的方法便不再有优势,需要重新设计模型。而且,对于大规模复杂问题来说,传统方法会随着问题规模的增大出现“组合爆炸”现象,计算的时间和空间复杂度呈指数级增长。因此,如何针对QAP设计有效的求解算法仍然面临重重困难。

近年来,随着深度强化学习的迅速发展,涌现出一批利用深度强化学习解决组合优化问题的新方法,为QAP的求解提供了一种全新的思路。本文提出一种基于actor-critic框架的求解模型ACQAP(actor-critic for QAP),该模型通过结合深度强化学习[10]与图卷积神经网络(graph convolutional neural network, GCN)[11]求解QAP。为了更准确地处理二次指派图信息,引入GCN捕捉各个节点间的关系,并通过多头注意力机制指导节点选择序列的输出。整个模型只需要输入二次指派图信息,就可以预测出对应问题的最优决策节点解序列。不同于以往利用图像信息作为训练数据的机器学习模型,该模型针对QAP的图信息进行训练,其求解的指派总费用显著减少。并且相较于启发式算法而言,当输入图的节点或边信息与训练时有所变化时,模型依旧保持鲁棒性,为解决大规模图上的组合优化问题提供了新的方向。

本文主要贡献如下:

1)提出QAP的基于图结构的ACQAP算法:本算法基于图数据进行训练,而启发式算法需要以问题为导向,以流量及距离矩阵进行训练,不具备问题实例的可推广性;当前基于学习的QAP求解算法大多以图像数据进行训练而难以达到足够好的性能;本文以图数据作为训练数据,更加接近实际生活中的QAP,能够得到较优的性能。

2)采用无监督的方法进行训练:ACQAP算法利用强化学习进行无监督训练,克服了样本标签稀缺情况下无法充分训练的问题。

3)为求解QAP提供了一个统一的模型:针对不同维度的QAP,可以选择输入至同样的模型中进行预测,其性能仍然可观,具有鲁棒性。

1　相关工作

近年来随着人工智能技术的迅速发展,深度学习技术在很多领域打破了传统方法的壁垒,取得了令人瞩目的成就。而如何利用深度学习与强化学习解决图的组合优化问题也引起诸多学者的强烈兴趣。Vinyals 等[12]提出可以求解组合优化问题的指针网络,有效地解决了输入序列与输出序列长度不同的问题,但该模型基于监督学习,严重依赖于大量带标签的样本,很难实际应用。Bello 等[13]首次尝试用强化学习方法训练指针网络,解决了样本标签获取困难的问题,但模型缺乏对图结构信息的捕捉。Dai 等[14]首先研究如何利用图神经网络解决组合优化问题,采用 stucture2vec 图神经网络进行图嵌入,通过 deep-Q-learning[15]学习图嵌入网络的贪婪策略,但需要人为地设计辅助函数,泛化能力差。Kool 等[16]借鉴 Transformer 模型,将深度学习与注意力机制结合在一起,实现了数据的并行输入,有效地提高了模型的求解效率;同时加入一个 Critic 网络估计基准函数 $b(s)$,使模型的收敛能力大幅度增强。Nazari 等[17]将指针网络的编码层直接替换为一维卷积层,从而可以有效解决动态组合优化问题。在针对指派问题的学习算法研究中,Groß [18]构造了一个指派问题的深度强化学习求解模型,将指派问题重新规划为 MDP,但其只适用于一次指派问题。Tang 等[19]提出一种新的基于深度强化学习与半马尔科夫决策过程的智能匹配模型,在考虑时间与空间的长期优化目标基础上进行更有效的二部图最大匹配,但同样也只是在一次指派问题上得到了很好的应用。Wang 等[20]提出一个端对端的模型来求解 QAP,模型利用卷积神经网络提取图像特征,将每个特征视为图上的一个节点,得到图像的图结构信息后输入图神经网络,获得每个节点的表征输入后再进行节点的匹配任务,文章用图像数据作为数据集进行训练,在提取图像的过程中损失了大量的结构信息,从而在优化能力上表现平平,且基于监督学习,难以获得大量带标签的样本。

本文提出一种结合强化学习 actor-critic 框架和多头注意力机制[21]的求解策略,将 QAP 等价为图上的节点选择问题,训练出针对该问题的端对端的求解模型,为不同规模 QAP 提供了一个通用的求解模型,且相比于以往基于学习的算法,得到的结果更加接近最优解。

2　问题定义

2.1　QAP

QAP 一般可描述为:给定 n 个设施、n 个地点,以及对应的 2 个 n 阶矩阵 $\boldsymbol{A}=(a_{ij})\in\mathbb{R}^{n\times n}$,$\boldsymbol{B}=(b_{kl})\in\mathbb{R}^{n\times n}$,要求给每个设施指派一个对应的地点(同一地点有且仅有一个设施),并使得指派后的费用之和最小。其中元素 a_{ij} 表示设施 i 和设施 j 之间的流量,元素 b_{kl} 表示地点 k 到地点 l 的距离,那么设施 i 在地点 k 且设施 j 在地点 l 所导致的费用为 $a_{ij}\times b_{kl}$。若将 QAP 表示成二次目标函数的 0-1 整数规划问题,则其数学模型为

$$\begin{aligned}&\min\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}a_{ij}b_{kl}x_{ik}x_{jl}\\&\text{s.t.}\sum_{i=1}^{n}x_{ik}=1,k=1,2,\cdots,n,\\&\sum_{k=1}^{n}x_{ik}=1,i=1,2,\cdots,n,\\&x_{ik}\in\{0,1\},i,k=1,2\cdots,n.\end{aligned}\tag{1}$$

其中,当且仅当设施 i 被指派到地点 k 时,x_{ik} 值为 1。设 $\boldsymbol{X}=(x_{11},x_{12},\cdots,x_{1n},x_{21},\cdots,x_{nn})^{\mathrm{T}}$,$x_{ik}$ 满足上述约束条件,矩阵 $\boldsymbol{A}$ 和矩阵 $\boldsymbol{B}$ 的 Kronecker 内积 $\boldsymbol{A}\otimes\boldsymbol{B}=(a_{ik}b_{jl})$ 由流量矩阵 $\boldsymbol{A}$ 和距离矩阵 $\boldsymbol{B}$ 中元素的所有可能乘积组成。则 QAP 的数学模型也可以表示为

$$\min\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{A}\otimes\boldsymbol{B})\boldsymbol{X}.\tag{2}$$

2.2　QAP 的问题转化

为了适用于图神经网络,我们将 QAP 转化为图上的节点选择问题。为此,将 QAP 的 2 个矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 抽象为无向完全图 G_a 及 G_b,$G_a=(V_a,E_a)$ 表示设施之间的关系图,而 $G_b=(V_b,E_b)$ 表示位置之间的关系图。其中 V_a 由设施点 $\{a_0,a_1,\cdots,a_{n-1}\}$ 构成,$|V_a|=n$。$E_a=\{(a_i,a_j):a_i,a_j\in V_a,i<j\}$ 是连接 V_a 中各顶点边的集合,边 (a_i,a_j) 定义为设施 i 和设施 j 之间的流量。类似地,V_b 由位置点 $\{b_0,b_1,\cdots,b_{n-1}\}$ 构成,且 $|V_b|=n$,$E_b=\{(b_k,b_l):b_k,b_l\in V_b,k<l\}$,是连接 V_b 中各顶点边的集合,边 (b_k,b_l) 是位置 k 和位置 l 之间的距离。

进而,根据这两幅图节点与边的关系,构造相应的关联图 G_{ab}(如图 1 所示)。其中 G_a 和 G_b 之间的每个候选节点匹配对应关联图 G_{ab} 中的 1 个

顶点。另一方面,若 G_a 中 a_i,a_j 有边相连,并且 G_b 中 b_k,b_l 有边相连,则关联图 G_{ab} 中的顶点 a_ib_k 与顶点 a_jb_l 有边相连,反之则无边相连。关联图对应的邻接矩阵如图 2 所示。这样,QAP 就转换为了关联图的顶点选择问题。

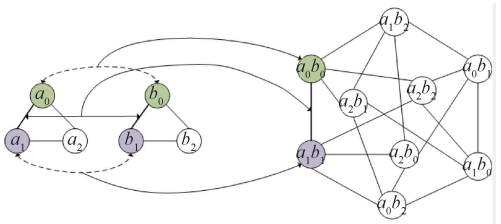

图 1 关联图转化过程

Fig. 1 Transformation process of association map

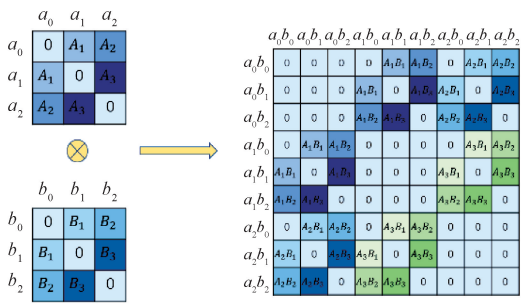

图 2 关联图对应的邻接矩阵

Fig. 2 Corresponding adjacency matrix of association map

3 模型设计

3.1 强化学习模型

强化学习,如图 3 所示,意在学习能够在环境中获得最大收益的行动。与生物学习过程类似,智能体在环境中通过交互试错,进而获得最优的策略。智能体通过和环境进行交互,得到关于环境状态变化的反馈信息,并根据这一反馈信息指导策略优化,从而使智能体的累计回报最大。强化学习的本质就是寻找最优决策的过程。

强化学习的过程可描述为一个马尔科夫决策过程(Markov decision process, MDP)[22]。智能体在时刻 t 观测到所处环境和自身当前状态 $s_t \in S$,根据策略 π,采取一个动作 $a_t \in A(S)$。在下一个时刻 $t+1$,环境根据智能体所采取的动作 a_t 给予一个相应的回报 $r_{t+1} \in R$,并进入一个新的状态 s_{t+1}。进而,智能体根据奖励信息 r 评价动作 a_t,如果得到正向反馈,则增加当前动作被选择的概率;反之,该动作被选择的概率减小。然后进入下一个决策过程,MDP 过程中得到的序列为

$$s_0, a_0, r_1, s_1, a_1, r_2, \cdots, s_{n-1}, a_{n-1}, r_n.$$

通过这样的不断学习,找到能够带来最大长期累积回报的最优策略 π^*。需要注意的是,由于智能体所处环境的随机性,以及回报获取存在延迟,MDP 使用折扣因子 γ 来反映越是未来的回报对当前 t 时刻累积回报的贡献率越小。时刻 t 之后,带有折扣因子 $\gamma \epsilon [0,1]$ 的长期累积回报如下

$$R_t = r_{t+1} + \gamma r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}. \quad (3)$$

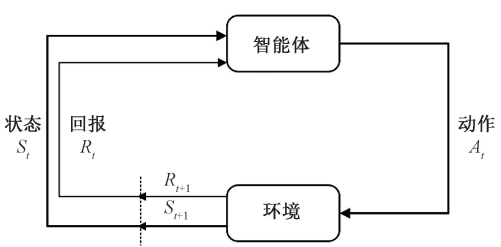

图 3 强化学习示意图

Fig. 3 The diagram of reinforcement learning

3.2 QAP 的强化学习模型构造

为便于表述,首先定义一些 QAP 转化为强化学习问题过程中的符号:S 表示状态空间,A 表示动作空间。每个状态 $s_t \in S$ 为之前已选择的顶点序列 $\{v_{\sigma(i)}\}_{i=1}^{t-1}$。动作 $a_t \in A(S)$ 被定义为下一个选择的顶点,即 $a_t = v_{\sigma(t)}$。基于 Bello 等[13]解决旅行商问题使用的 Actor-Critic 思想,引入 GCN 及多头注意力机制[21]构造了适合求解 QAP 的 ACQAP 模型,其包含一个策略网络和一个估值网络。其主体框架如图 4 所示。

3.2.1 策略网络构造

对于当前的状态 s_t,构造了一个策略网络 $\pi_{\theta_\pi}(a_t \mid s_t)$ 来获取相应的动作 a_t。本文借鉴 transformer 的多头注意力机制来构建策略网络。

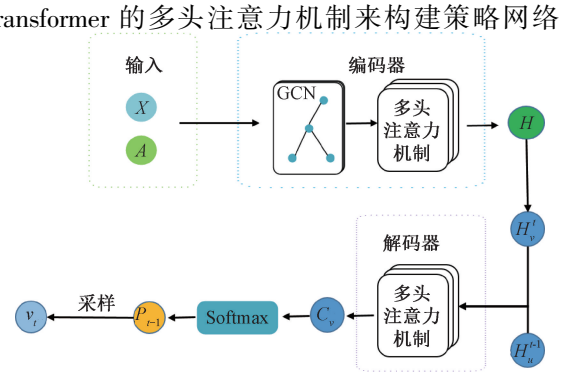

图 4 ACQAP 主体框架

Fig. 4 The main framework of ACQAP

通过多头注意力机制,能够从多个角度计算节点的表征序列对于预测模型输出节点序列的注意力权重,以此来指导模型节点输出,从而有效提高模型针对 QAP 解的预测准确度。

本文的策略网络在结合多头注意力机制的基础上引入 GCN,以此来更好地求解 QAP。如图 5 所示,策略网络中包含编码器和解码器。

在编码器上,采用有别于传统点编码器的图编码器,首先针对关联图利用 GCN 对所有顶点 $V=[v_{00},v_{01},\cdots,v_{nn}]$ 进行编码。通过 GCN,能够在节点的特征向量中嵌入图结构的信息,进而体现 QAP 中关联图顶点间的上下文信息。主要通过采样和聚合邻居节点的信息来实现,各个节点的嵌入通过如下方式进行更新

$$\boldsymbol{h}_{N_v}^{l}=\mathrm{AGGREGATE}_{l}(\{\boldsymbol{h}_{u}^{l-1},\forall u\in N_v\})$$

$$\boldsymbol{h}_{v}^{l}=\sigma(\boldsymbol{W}^{l}\cdot[\boldsymbol{h}_{v}^{l-1},\boldsymbol{h}_{N_v}^{l},\boldsymbol{x}_v,\boldsymbol{x}_{(v,u)}^{e}]),\quad(4)$$

其中:$\boldsymbol{W}^{l}$ 是第 l 层的参数,N_v 表示与顶点 v 相邻顶点的集合,$\boldsymbol{h}_{v}^{l}$ 是顶点 v 在第 l 层对应的嵌入,$\boldsymbol{x}_v$ 是顶点 v 的特征,$\boldsymbol{x}_{(v,u)}^{e}$ 是与 v 相邻的边特征。聚合(AGGREGATE)步骤可以有多种方式,在 ACQAP 中采用平均聚合的方法。

经过 GCN 的编码后,得到相应的表征向量 $\{\boldsymbol{h}_{v}^{l}\}$。进而将其拼接后的表示矩阵输入至多头注意力层中再次编码,获得编码矩阵 $\boldsymbol{H}$。

解码器在解码过程中存在两部分的输入,一部分是 t 时刻所有未输出节点的编码向量 $\boldsymbol{H}_{v}^{t}$,另一部分为当前时刻已输出节点序列的编码向量 $\boldsymbol{H}_{u}^{t-1}$。在多头注意力机制下,针对目标输出节点会产生 3 个向量:$\boldsymbol{k}_{v}^{t}$、$\boldsymbol{v}_{v}^{t}$ 和 $\boldsymbol{q}_{u}^{t}$,其中键向量 $\boldsymbol{k}_{v}^{t}$ 和值向量 $\boldsymbol{v}_{v}^{t}$ 对应未输出节点序列,而查询向量 $\boldsymbol{q}_{u}^{t}$ 对应已输出节点序列:

$$\boldsymbol{k}_{v}^{t}=\boldsymbol{W}_{k}\boldsymbol{H}_{v}^{t},\boldsymbol{v}_{v}^{t}=\boldsymbol{W}_{v}\boldsymbol{H}_{v}^{t},\boldsymbol{q}_{u}^{t}=\boldsymbol{W}_{q}\boldsymbol{H}_{u}^{t-1}.\quad(5)$$

其中 $\boldsymbol{W}_k$、$\boldsymbol{W}_v$、$\boldsymbol{W}_q$ 为线性映射的参数矩阵。在多头注意力机制下,不同层的 $\boldsymbol{k}_{v}^{t}$、$\boldsymbol{v}_{v}^{t}$ 和 $\boldsymbol{q}_{u}^{t}$ 可以在多个层面上获取 $\boldsymbol{H}_{v}^{t}$ 与 $\boldsymbol{H}_{u}^{t-1}$ 之间的关系,从而得到不同角度下序列之间的关联关系。

通过 $\boldsymbol{k}_{v}^{t}$、$\boldsymbol{v}_{v}^{t}$ 和 $\boldsymbol{q}_{u}^{t}$ 计算已输出节点与目标输出节点 v 之间的相容性,用 $\boldsymbol{C}_v$ 表示,其定义如下

$$\boldsymbol{C}_v=\begin{cases}\dfrac{(\boldsymbol{q}_{u}^{t})^{\mathrm{T}}\boldsymbol{k}_{v}^{t}}{\sqrt{d}},\text{若 } v \text{ 与所有已输出节点相连},\\ -\infty,\text{其他}.\end{cases}\quad(6)$$

其中 d 为缩放因子,用于控制后续反向传播中梯度过小的问题。$\boldsymbol{C}_v$ 的每一项分别表示目标输出节点 v 与已输出节点序列 $\{u\}$ 之间的注意力系数,通过求和的方式,可以求得 v 与所有已输出节点之间的相容性 c_v。最后策略网络利用 softmax 函数使所有未输出节点的相容性 $\{c_v\}$ 归一化至 0~1 之间的实数,并将其作为选择节点的概率值。

因此所有候选顶点的指派政策如下所示

$$\pi_{\theta_\pi}(a_t\mid s_t)=p_{t-1}=\mathrm{softmax}(c_v).\quad(7)$$

根据策略 $\pi_{\theta_\pi}(a_t\mid s_t)$ 进行采样或贪婪选择,预测下一个选择的顶点 $a_t=x_{\sigma(t)}$。

3.2.2　估值网络构造

估值网络根据策略网络的动作来评价策略的价值,并反馈给策略网络。网络输出的是对目标函数的预测,将输入序列映射成一个基线预测 $b_{\theta_v}(s)$。估值网络主要包括 3 个模块:1)编码器:与策略网络相同的编码器结构;2) glimpse 计算模块:该计算模块对编码器的输出使用了 glimpse 函数[23],并且将该函数的输出作为解码器的输入;3)解码器:带有 ReLU 函数的两层全连接神经网络,输出基线预测 $b_{\theta_v}(s)$。其中,glimpse 计算模块通过 glimpse 指向机制进行整合,假设估值网络的参数为 θ_v,则 $p_{\theta_v}(s)$ 为所有顶点的概率分布,$\{\boldsymbol{a}_i\}$ 为节点的编码向量。相应地,glimpse 向量的计算公式为

$$\mathrm{glimpse}=\sum_{i=1}^{n^2}p_{\theta_v}(s_i)\boldsymbol{a}_i.\quad(8)$$

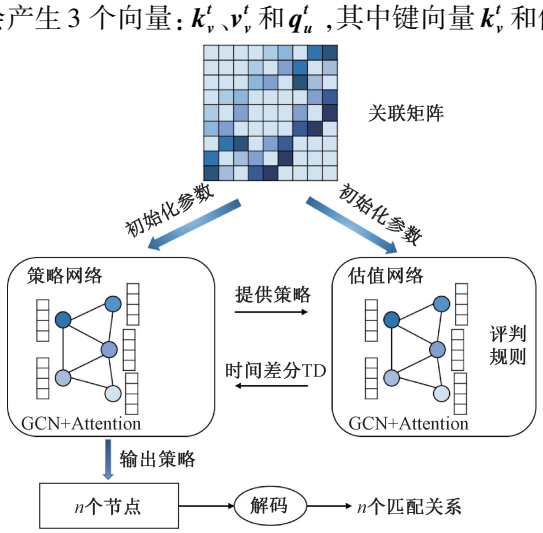

图 5　策略网络

Fig. 5　Actor network

3.2.3　Actor-critic 训练

采用策略梯度算法[24]最小化奖励函数,在训练中使用带基准线 REINFORCE 方法来训练

actor-critic 框架,同时使用随机采样和贪心的方式进行顶点的选择,之后使用 Adam 优化器对网络参数进行更新。

策略网络输出所选择的 n 个顶点,其目标是使这选择的 n 个顶点对应的 QAP 解所需费用最少。用 θ_π 表示策略网络的参数,g 表示输入的待指派的两幅图,则优化目标为

$$\begin{aligned} J(\theta_\pi \mid g) &= \boldsymbol{E}_{\pi \sim p_{\theta_\pi}(\cdot \mid g)} C(\pi \mid g) \\ &= \boldsymbol{E}_{\pi \sim p_{\theta_\pi}(\cdot \mid g)} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{A} \otimes \boldsymbol{B} \boldsymbol{X}, \end{aligned} \quad (9)$$

其中策略网络的优化目标与式(2)的优化目标是等价的。

接着引入估值网络构造基线函数 $b_{\theta_v}(g)$,同时为便于计算,根据蒙特卡洛方法采样 Batch 个实例估计 θ_π 的梯度如下:

$$\begin{aligned} &\nabla_{\theta_\pi} J(\theta_\pi \mid g) \\ &= \sum_{i=1}^{\text{Batch}} \frac{C(\pi_i \mid g_i) - b_{\theta_v}(g_i) \ \nabla_{\theta_\pi} \log p_\theta(\pi_i \mid g_i)}{\text{Batch}}. \end{aligned} \quad (10)$$

在估值网络的训练中,采用随机梯度下降的方法训练网络参数,其优化目标为

$$L(\theta_v) = \frac{\sum_{i=1}^{\text{Batch}} [\ \| b_{\theta_v}(g_i) - C(\pi_i \mid g_i) \|_2^2]}{\text{Batch}}. \quad (11)$$

重复交错训练上述的策略网络及估值网络,直至训练完成。其算法流程如算法 1 所示。这一与 GCN 结合的 actor-critic 深度强化学习方法是一种完全端到端的求解方法,可将问题的已知条件输入训练好的神经网络快速输出相应的 n 个节点,从而获得匹配关系,也即 QAP 的近似解。

算法 1 求解 QAP 的 actor-critic 算法流程

Require:策略网络 $\pi_{\theta_\pi}(\pi \mid g)$ 和估值网络 $V_{\theta_v}(g)$,训练步数 T,批量大小 B

1. 初始化策略网络和估值网络参数 θ_π 和 θ_v
2. for t=1,…,T do
3. $g_i \sim$ SampleInput(G) for $i=1,2,\cdots,B$
4. $\pi_i \sim$ SampleSolution($P_{\theta_\pi}(\cdot \mid g)$) for $i=1,2,\cdots,B$
5. $b_i \sim V_{\theta_v}(g)$ for $i=1,2,\cdots,B$
6. 计算估值网络目标函数

$$L(\theta_v) \leftarrow \sum_{i=1}^{B} [\ \| b_{\theta_v}(g_i) - C(\pi_i \mid g_i) \|_2^2]/B$$

7. 更新策略网络参数 θ_π

$$\nabla_{\theta_\pi} J(\theta_\pi \mid g) \leftarrow \sum_{i=1}^{B} [(C(\pi_i \mid g_i) - b_{\theta_v}(g_i) \nabla_{\theta_\pi} \log p_\theta(\pi_i \mid g_i)]/B$$

$$\theta_\pi \leftarrow \text{Adam}(\theta_\pi, \nabla_{\theta_\pi} J)$$

8. 更新估值网络参数 θ_v

$$\theta_v \leftarrow \text{Adam}(\theta_v, \nabla_{\theta_v} C)$$

9. end for

输出:最优策略 π_{θ_π}

4 数值实验及分析

4.1 实验环境及评估指标

实验环境如表 1 所示。在实验中,基于 U(0,1)随机生成训练数据。使用 Adam 优化器对网络参数进行更新,学习率为 $1e^{-3}$。

表 1 实验环境

Table 1 Experimental environment

CPU	Intel Core i7-9700K CPU@3.6 GHz
操作系统	Ubuntu16.4
GPU	NVIDIA Tesla V100
深度学习平台	Pytorch1.6.0

为评估所提出算法的有效性,将实验结果与目前的最优解进行比较,利用与目前最优解之间的偏差大小及执行时间来反映模型的好坏。假设 O_{cost}、B_{cost} 分别代表算法所求的最优解、QAPLIB 公开库[25] 里对应的目前最优解。则偏差(Deviation,简记为 D)定义如下

$$D = \frac{O_{\text{cost}} - B_{\text{cost}}}{B_{\text{cost}}} \times 100\%. \quad (12)$$

4.2 对比实验

定义 QAP-X 为指派节点数为 x 的 QAP,ACQAP-Y 为训练数据节点数为 y 的 ACQAP 模型。实验过程中随机生成指派节点数目为 20、50、100 的 1 000 个图,将这些图 $G(V,E)$ 输入至本文的模型 ACQAP 中。基于此,训练了 3 个不同的 ACQAP 模型(ACQAP-20、ACQAP-50 以及 ACQAP-100)。

4.2.1 与启发式算法比较

启发式算法是解决复杂 QAP 问题的常用算法,它们可以在可行时间内提供一个近似的解决方案。为了评估提出的 ACQAP 模型在大规模和复杂 QAP 实例上的性能,将其与一些经典的启发式算法[26] 进行了比较,如迭代局部搜索(iterated local search, ILS)、模拟退火(simulated annealing, SA)、遗传算法(genetic algorithm, GA)、粒子群优化(particle swarm optimization, PSO)、乌鸦搜索算法(crow search algorithm, CSA)等。为此,从公开库 QAPLIB 中的 134 个实例挑出具有代表性的例

子进行仿真实验,其中包含了规模在 50 以上的实例。QAPLIB 是一个由 QAP 实例组成的公开库,其实例可以根据其属性划分为不同的类型,实例类型的不同会影响 QAP 算法的性能[27]。一般情况下,QAPLIB 实例可分为 4 类[28]:

Ⅰ 非结构化:按均匀分布随机生成流量矩阵和距离矩阵,这些例子是最难求解的。例如:tai20a,tai30a,tai40a,tai80a,tai100a 等。

Ⅱ 类真实实例:随机生成的实例,分布与真实实例类似,但其规模可能比真实实例的更大。例如:tai20b,tai25b,tai30b,tai60b,tai80b,tai100b 等。

Ⅲ 基于网格的距离矩阵:距离矩阵来源于 $n_1 \times n_2$ 网格,其距离定义为网格点之间的曼哈顿距离。例如:nug20,nug30,sko42,sko56,sko100a 等。

Ⅳ 真实实例:来自 QAP 实际应用的实例。在现实生活中,流量矩阵有许多零项,其余的项显然不是均匀分布的。例如:ste36x, kra30x, bur26x 等。

其中,Ⅰ和Ⅲ为规则实例,Ⅱ和Ⅳ为不规则实例。

与启发式算法的比较结果如表 2 和图 6 所示。

表 2　不同算法求解 QAP 产生的偏差及执行时间比较

Table 2　Comparison of the deviation and execution time generated by different algorithms for solving QAP

实例		ILS		SA		GA		PSO		CSA		ACQAP-100	
		D/%	*T*/s	*D*/%	*T*/s	*D*/%	*T*/s	*D*/%	*T*/s	*D*/%	*T*/s	*D*/%	*T*/s
Ⅰ 非结构化	tai30a	**4.87**	0.281	7.33	1.125	13.39	0.664	12.37	0.278	14.79	0.214	9.80	**0.086**
	tai35a	**5.78**	0.332	7.14	1.232	15.49	0.729	12.30	0.364	16.73	0.257	10.39	**0.128**
	tai40a	**7.13**	0.362	8.18	1.275	14.70	0.761	14.06	0.482	15.80	0.302	9.45	**0.182**
	tai50a	8.04	0.428	8.98	1.457	15.22	0.889	13.15	0.750	16.25	0.366	**6.25**	**0.313**
	tai60a	**8.47**	0.493	9.56	1.656	14.72	1.050	13.15	1.093	15.49	0.453	9.00	**0.447**
	tai80a	**8.43**	0.641	9.30	2.175	12.80	1.296	12.54	1.959	13.19	**0.601**	9.57	0.815
	tai100a	**8.87**	0.918	9.22	2.884	12.32	1.596	11.67	3.754	12.53	**0.764**	9.23	1.846
Ⅱ 类真实实例	tai60b	14.55	0.461	**14.06**	1.575	45.18	1.021	35.96	0.803	41.36	**0.450**	17.03	0.451
	tai80b	**17.29**	**0.614**	18.33	2.054	38.61	1.274	35.59	1.410	41.00	0.643	17.54	0.783
	tai100b	17.47	0.894	19.12	2.674	38.50	1.601	32.20	2.326	37.97	**0.759**	**16.31**	1.825
	tai150b	17.64	2.262	17.58	6.130	25.99	2.532	23.80	7.474	26.08	**1.533**	**16.25**	6.542
	tai64c	**0**	0.496	**0**	1.665	13.77	1.120	8.63	0.876	14.08	**0.471**	3.21	0.493
	tai256c	4.91	10.773	**4.66**	18.964	12.33	4.564	10.20	45.963	12.99	**2.318**	5.12	84.678
Ⅲ 基于网格的距离矩阵	sko72	**8.83**	0.544	16.47	1.645	16.01	1.245	14.60	0.913	15.68	**0.541**	10.76	0.643
	sko81	**9.09**	0.632	15.63	1.962	14.71	1.382	14.46	1.325	16.11	**0.625**	10.54	0.804
	sko90	**9.03**	0.756	15.06	2.280	15.29	1.426	14.45	1.730	15.16	**0.745**	11.23	1.231
	sko100a	**9.45**	1.016	14.58	2.583	14.14	1.575	13.76	2.172	14.40	**0.861**	11.04	1.852
Ⅳ 真实实例	els19	**0.61**	0.224	0	0.494	64.19	0.573	53.20	0.143	47.73	0.173	4.90	**0.061**
	kra30a	**9.20**	0.275	33.41	1.113	31.70	0.662	29.38	0.262	33.95	0.224	17.69	**0.085**
	kra32	**9.41**	0.315	35.33	1.128	34.90	0.683	29.36	0.273	36.18	0.242	19.21	**0.089**
	ste36a	**21.58**	0.324	88.37	1.254	86.77	0.720	59.16	0.357	88.35	0.278	36.01	**0.139**

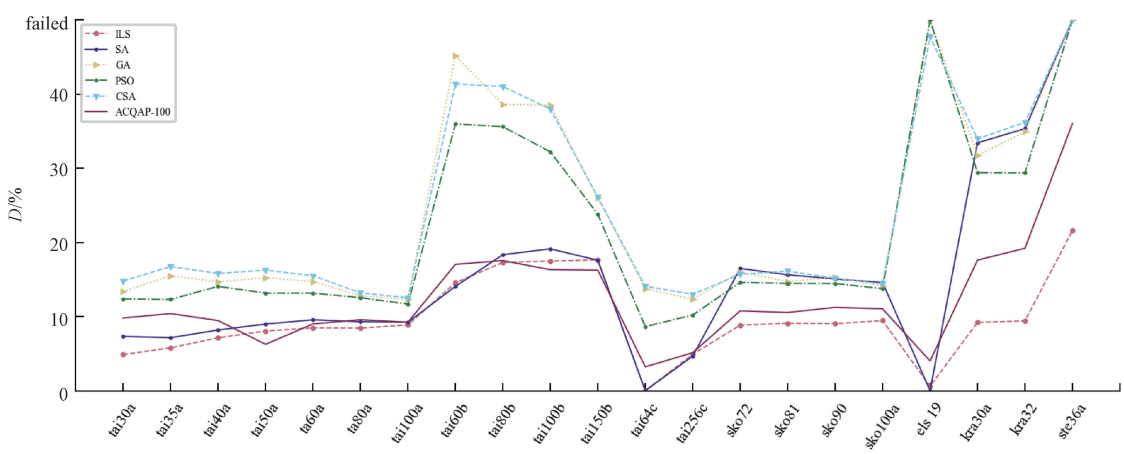

图 6　ACQAP-100 与元启发式算法在求解 QAPLIB 实例上产生的偏差比较

Fig. 6　Comparison of bias produced by ACQAP-100 and meta-heuristic algorithms on solving QAPLIB instances

基于多个 QAPLIB 实例比较不同算法在 GPU 上的平均偏差 D 和平均执行时间 T。从表 2 可以看出,在精度上,ACQAP-100 相比当前最好的启发式算法 ILS 相差不大,要优于其他的启发式算法,尤其在 Tai50a、Tai100b 及 Tai150b 实例上达到了最优。为更直观地感受,用折线图对其进行了表示,如图 6 所示。从执行时间上看,当节点数目在 50 以内时,ACQAP-100 的性能要优于现有的启发式算法。虽然当规模过大时,ACQAP-100 需要较长的执行时间,但在实际应用中,相较于启发式算法需要针对具体问题的性质和规模设置不同的参数,不具有泛化性,ACQAP 是一种统一的模型,无需针对问题进行手工设计。

4.2.2 与基于学习的算法比较

神经图匹配网络(neural graph matching network,NGM)是由 Wang 等[20]提出的基于学习的方法求解 QAP 的网络。将 ACQAP 模型与多个 NGM 模型(标准 NGM 模型(NGM)以及利用 Gumbel 采样的 NGM 模型(NGM-Gx,x=样本数))进行比较,以此对所提出的 ACQAP 模型做进一步的评估。从 QAPLIB 中的 134 个实例中随机选取 24 个实例进行模拟实验,其中包含规模在 50 以上的实例,并将实验结果与当前的最优解以及基于学习的方法进行比较,实验结果如表 3 和表 4 所示。从表 4 可以看出基于 gumbell 采样的 NGM-G5k 始终优于确定性的 NGM,而 ACQAP 模型则要优于 NGM-G5k 模型。可视化结果如图 7 所示。对不同模型在这 24 个实例上的实验结果进行可视化,定义偏差大于 50% 的实例是失败的实例(失败的实例被绘制在 y 轴的顶部)。从图 7 中可以看出,我们的模型在应对 QAP 上要优于现有的基于学习的方法。

4.3 鲁棒性分析

在 QAP 问题中,鲁棒性是一个很重要的研究内容。为研究模型的鲁棒性,通过在 1 000 个随机生成的图(模拟 QAP 算例)上运行 3 种 ACQAP 模型获取相应的指派费用。其结果如表 5 所示,指派费用是指 1 000 个算例的平均指派费用。

通过比较 3 种 ACQAP 模型在不同规模问题下的运行结果,可以看出 ACQAP 模型具有较好的鲁棒性,能够更好地适应实际应用需求,即利用单一规模训练的模型可以直接用于其他规模的问题。如通过 20 个节点数训练的 ACQAP-20 模型求解 100 节点规模的 QAP-100 时,其指派费用与利用 100 个节点数训练的 ACQAP-100 求解得到的差别不大。

表 3 ACQAP 与 NGM 模型在求解 QAPLIB 实例上产生的费用(O_{cost})比较

Table 3 Comparison of the cost(O_{cost}) produced by ACQAP and NGM models in solving QAPLIB instances

实例	B_{cost}	ACQAP-20	ACQAP-50	ACQAP-100	NGM	NGM-G5	NGM-G50	NGM-G500	NGM-G5k
lipa20a	3 638	3 851	**3 836**	3 845	3 929	3 904	3 891	3 864	3 853
lipa30a	13 178	13 525	**13 499**	13 502	13 841	13 756	13 742	13 660	13 631
lipa40a	31 538	32 301	**32 107**	32 114	32 666	32 658	32 521	32 504	32 454
lipa50a	62 093	63 107	**62 937**	62 950	64 100	63 886	63 816	63 705	63 671
rou12	235 528	257 769	257 761	257 681	321 082	301 728	275 876	**252 156**	264 898
rou15	354 210	**410 076**	410 153	410 203	469 592	453 260	430 000	413 294	403 872
rou20	725 522	817 793	821 069	821 077	897 348	876 282	867 682	845 494	**817 776**
tai40a	3 139 370	3 447 918	3 436 119	**3 436 002**	3 610 604	3 755 024	3 672 066	3 660 256	3 610 604
tai50a	4 938 796	5 563 996	**5 213 447**	5 247 496	5 891 066	5 788 660	5 704 692	5 714 682	5 677 282
nug27	5 234	5 828	**5 798**	5 812	6 332	6 860	6 546	6 276	6 208
nug28	5 166	6 057	**6 003**	6 024	6 128	6 764	6 332	6 424	6 128
nug30	6 124	7 143	7 221	**7 119**	7 608	7 666	7 780	7 530	7 294
scr12	31 410	**35 292**	35 312	35 306	44 400	38 228	42 014	40 908	36 292
scr15	51 140	**64 407**	64 597	64 611	81344	77 376	75 746	75 224	68 768
scr20	110 030	152 634	152 647	**152 622**	182 882	212 432	175 534	171 666	154 636
sko42	15 812	18 617	**18 233**	18 242	20 192	19 274	19 340	19 218	18 716
sko49	23 386	27 160	26 934	**26 922**	28 712	28 584	27 718	27 238	27 554
tho30	149 936	184 632	184 589	**184 573**	187 062	207 424	198 456	196 072	185 622
tho40	240 516	294 878	**293 465**	293 877	313 026	323 808	311 780	318 188	304 878
wil50	48 816	53 418	53 260	**53 167**	55 390	54 962	54 552	54 086	53 418
bur26a	5 426 670	5 621 443	**5 620 996**	5 621 332	5 684 628	5 828 287	5 650 343	5 650 343	5 621 774
chr25a	3 796	**5 152**	5 246	5 207	18 950	16 704	13 758	13 162	11 648
els19	17 212 548	**18 041 490**	18 065 473	18 056 732	34 880 280	53 830 864	31 247 564	28 600 336	27 029 748
esc16a	68	78	**76**	78	88	84	86	82	78

表 4　ACQAP 与 NGM 模型在求解 QAPLIB 实例上产生的偏差比较

Table 4　Comparison of the deviation produced by ACQAP and NGM models in solving QAPLIB instances

实例	B_{cost}	ACQAP-20	ACQAP-50	ACQAP-100	NGM	NGM-G5	NGM-G50	NGM-G500	NGM-G5k
lipa20a	3 638	5. 85	**5. 44**	5. 69	8. 00	7. 31	6. 95	6. 21	5. 91
lipa30a	13 178	2. 63	**2. 44**	2. 46	5. 03	4. 39	4. 28	3. 66	3. 44
lipa40a	31 538	2. 42	**1. 80**	1. 83	3. 58	3. 55	3. 12	3. 06	2. 90
lipa50a	62 093	1. 63	**1. 36**	1. 38	3. 23	2. 89	2. 77	2. 60	2. 54
rou12	235 528	9. 44	9. 44	**9. 41**	36. 32	28. 11	17. 13	7. 06	12. 47
rou15	354 210	**15. 77**	15. 79	15. 81	32. 57	27. 96	21. 40	16. 68	14. 02
rou20	725 522	**12. 72**	13. 17	13. 17	23. 68	20. 78	19. 59	16. 54	12. 72
tai40a	3 139 370	9. 83	**9. 45**	**9. 45**	15. 01	19. 61	16. 97	16. 59	15. 01
tai50a	4 938 796	12. 66	**5. 56**	6. 25	19. 28	17. 21	15. 51	15. 71	14. 95
nug27	5 234	11. 35	**10. 78**	11. 04	20. 98	31. 07	25. 07	19. 91	18. 61
nug28	5 166	17. 25	**16. 20**	16. 61	18. 62	30. 93	22. 57	24. 35	18. 62
nug30	6 124	16. 64	17. 91	**16. 25**	24. 23	25. 18	27. 04	22. 96	19. 11
scr12	31 410	**12. 36**	12. 42	12. 40	41. 36	21. 71	33. 76	30. 24	15. 54
scr15	51 140	**25. 94**	26. 31	26. 34	59. 06	51. 30	48. 11	47. 09	34. 47
scr20	110 030	38. 72	38. 73	**38. 71**	66. 21	93. 07	59. 53	56. 02	40. 54
sko42	15 812	17. 74	**15. 31**	15. 37	27. 70	21. 89	22. 31	21. 54	18. 37
sko49	23 386	16. 14	15. 17	**15. 12**	22. 77	22. 23	18. 52	16. 47	17. 82
tho30	149 936	23. 14	23. 11	**23. 10**	24. 76	38. 34	32. 36	30. 77	23. 80
tho40	240 516	22. 60	**22. 01**	22. 19	30. 15	34. 63	29. 63	32. 29	26. 76
wil50	48 816	9. 43	9. 10	**8. 91**	13. 47	12. 59	11. 75	10. 80	9. 43
bur26a	5 426 670	3. 59	**3. 58**	3. 59	4. 75	7. 40	4. 12	4. 12	3. 60
chr25a	3 796	**35. 72**	38. 20	37. 17	399. 21	340. 04	262. 43	246. 73	206. 85
els19	17 212 548	**4. 82**	4. 96	4. 90	102. 64	212. 74	81. 54	66. 16	57. 04
esc16a	68	14. 71	**11. 76**	14. 71	29. 41	23. 53	26. 47	20. 59	14. 71

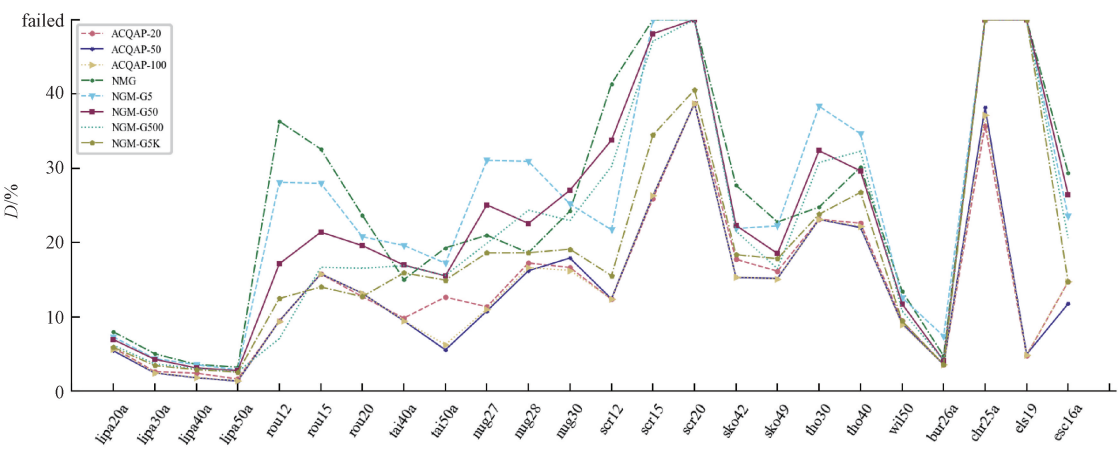

图 7　ACQAP 与 NGM 模型在求解 QAPLIB 实例上产生的偏差比较

Fig. 7　Comparison of deviation produced by ACQAP and NGM models on solving QAPLIB instances

表 5　不同规模训练下模型的指派费用比较

Table 5　Comparison of the cost of models under different scale training

	指派费用		
	QAP-20	QAP-50	QAP-100
ACQAP-20	**48. 143**	266. 648	1 109. 538
ACQAP-50	48. 691	**261. 920**	1 102. 769
ACQAP-100	48. 794	263. 412	**1 001. 453**

5　结束语

本文提出一个端到端机器学习模型 ACQAP,将强化学习与多头注意力机制有机结合起来,并引入 GCN,用于解决复杂组合优化问题中的 QAP。通过将图作为一个环境,简称图环境,智能体与图环境进行交互,在每一次交互中,图环境 G 给予智能体一定的奖励 r,并输出当前环境的随

机状态 s,而智能体则根据当前的奖励 r 和随机状态 s 做出动作 a。最后智能体输出的动作是图上的一个顶点子集,即为 QAP 的近似解。最终实验结果表明 DRL+multi-head-attention 的方法可以很好地从小规模问题扩展到大规模问题,且优于传统启发式算法。

参考文献

[1] Koopmans T C, Beckmann M. Assignment problems and the location of economic activities[J]. Econometrica, 1957, 25(1):53. DOI:10.2307/1907742.

[2] Pardalos P M, Xue J. The maximum clique problem[J]. Journal of Global Optimization, 1994, 4(3): 301-328. DOI: 10.1007/BF01098364.

[3] Steinberg L. The backboard wiring problem: a placement algorithm[J]. SIAM Review, 1961, 3(1): 37-50. DOI: 10.1137/1003003.

[4] Kusiak A, Heragu S S. The facility layout problem[J]. European Journal of Operational Research, 1987, 29(3): 229-251. DOI:10.1016/0377-2217(87)90238-4.

[5] İşeri A, Ekşioğlu M. Estimation of digraph costs for keyboard layout optimization[J]. International Journal of Industrial Ergonomics, 2015, 48: 127-138. DOI: 10.1016/j.ergon.2015.04.006.

[6] Manne A S. On the job-shop scheduling problem[J]. Operations Research, 1960, 8(2): 219-223. DOI:10.1287/opre.8.2.219.

[7] Flood M M. The traveling-salesman problem[J]. Operations Research, 1956, 4(1): 61-75. DOI:10.1287/opre.4.1.61.

[8] Kirby R C, Siebenmann L C. On the triangulation of manifolds and the hauptvermutung[J]. Bulletin of the American Mathematical Society, 1969, 75(4): 742-750. DOI:10.1090/s0002-9904-1969-12271-8.

[9] Pardalos P M, Xue J. The maximum clique problem[J]. Journal of Global Optimization, 1994, 4(3): 301-328. DOI: 10.1007/BF01098364.

[10] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[EB/OL]. arXiv:1312.5602. (2013-12-19)[2013-12-19]. https://arxiv.org/abs/1312.5602.

[11] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[EB/OL]. arXiv:1609.02907. (2016-09-09)[2017-02-22]. https://arxiv.org/abs/1609.02907.

[12] Vinyals O, Fortunato M, Jaitly N. Pointer networks[C]// Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2 (NIPS'15). MIT Press, Cambridge, MA, USA, 2692-2700. DOI: 10.5555/2969442.2969540.

[13] Bello I, Pham H, Le Q V, et al. Neural combinatorial optimization with reinforcement learning[EB/OL]. arXiv: 1611.09940. (2016-11-29)[2017-01-12]. https://arxiv.org/abs/1611.09940.

[14] 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 (NIPS' 17). Curran Associates Inc., Red Hook, NY, USA, 6351-6361, 2017. DOI: 10.5555/3295222.3295382.

[15] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518, 529-533. DOI:10.1038/nature14236.

[16] Kool W, Hoof H V, Welling M. Attention, learn to solve routing problems![EB/OL]. arXiv:1803.08475. (2018-03-22)[2019-02-07]. https://arxiv.org/abs/1803.08475v3.

[17] Nazari M, Oroojlooy A, Takáč M, et al. Reinforcement learning for solving the vehicle routing problem[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS' 18). Curran Associates Inc., Red Hook, NY, USA, 9861-9871. DOI: 10.5555/3327546.3327651.

[18] Groß J. Using deep reinforcement learning to optimize assignment problems [D/OL]. Saarbrücken: Saarland University, 2021[2021-01-06]. https://mosi.uni-saarland.de/assets/theses/ma_joschka.pdf.

[19] Tang X C, Qin Z T, Zhang F, et al. A deep value-network based approach for multi-driver order dispatching[C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage AK USA. New York, NY, USA: ACM, 2019: 1780-1790. DOI: 10.1145/3292500.3330724.

[20] Wang R Z, Yan J C, Yang X K. Neural graph matching network: learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8053, PP(99): 1. DOI: 10.1109/TPAMI.2021.3078053.

[21] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000-6010. DOI: 10.5555/3295222.3295349.

[22] Puterman M L. Markov decision processes[M]//Handbooks in Operations Research and Management Science, 1990, 2: 331-434. DOI:10.1016/S0927-0507(05)80172-0.

[23] Ba J, Mnih V, Kavukcuoglu K. Multiple object recognition with visual attention[EB/OL]. arXiv:1412.7755. (2014-12-24)[2015-04-23]. https://arxiv.org/abs/1412.7755v2.

[24] Sutton R S, McAllester D, Singh S, et al. Policy gradient methods for reinforcement learning with function approximation[C]//Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS'99). MIT Press, Cambridge, MA, USA, 1057-1063. DOI: 10.5555/3009657.3009806.

[25] Burkard R E, Karisch S, Rendl F. QAPLIB-A quadratic assignment problem library[J]. European Journal of Operational Research, 1991, 55(1): 115-119. DOI: 10.1016/0377-2217(91)90197-4.

[26] Commander C W. A survey of the quadratic assignment problem, with applications[J]. Morehead Electronic Journal of Applicable Mathematics, 2005, (4):1-15.

[27] Tseng L Y, Liang S C. A hybrid metaheuristic for the quadratic assignment problem[J]. Computational Optimization and Applications, 2006, 34(1): 85-113. DOI: 10.1007/s10589-005-3069-9.

[28] Taillard É D. Comparison of iterative searches for the quadratic assignment problem[J]. Location Science, 1995, 3(2): 87-105. DOI:10.1016/0966-8349(95)00008-6.