

# 基于异步 Dueling DQN 和计划时间预测网络的连接优化器

高瑞玮<sup>1</sup>, 乔少杰<sup>1</sup>, 韩楠<sup>2</sup>, 闵圣捷<sup>3</sup>, 李贺<sup>4</sup>, 覃晓<sup>5</sup>, 张桃<sup>6</sup>, 元昌安<sup>7</sup>

(1. 成都信息工程大学软件工程学院, 四川成都 610225; 2. 成都信息工程大学管理学院, 四川成都 610103;  
3. 中电智元数据科技有限公司, 北京 100081; 4. 西安电子科技大学计算机科学与技术学院, 陕西西安 710126;  
5. 南宁师范大学, 广西南宁 530001; 6. 宜宾学院, 四川宜宾 644000; 7. 广西科学院, 广西南宁 530007)

**摘要:** 连接顺序选择是查询优化领域中极具挑战性的研究方向, 对于数据库管理系统获得良好的查询性能至关重要. 然而, 传统优化方法和现有智能优化方法均存在着不足, 如规划时间过长、容易得到质量较差的连接计划、编码未考虑结构特征、依赖基数估计和代价估计使得连接计划无法反映真实的执行时间等. 针对上述问题, 提出了一种新型基于异步 Dueling DQN (Deep Q-network) 和计划时间预测网络的连接优化器: ADP-Join (Asynchronous Dueling DQN and Plan Latency Prediction Network for Join Order Selection). ADP-Join 集成了一种新的编码方法, 能够区分不同结构的连接计划. ADP-Join 设计了计划时间预测网络 PLN (Plan Latency Prediction Network) 来改善现有基于强化学习优化器的奖励机制. 再者, 提出异步更新机制改进 Dueling DQN 模型来提升训练性能和减少训练时间. 大量的实验结果表明, 在 TPC-H 和 JOB 真实数据集上 ADP-Join 的性能优于现有的智能优化器.

**关键词:** 连接顺序选择; 查询优化; 连接计划; 强化学习; 异步更新

**基金项目:** 国家自然科学基金 (No. 62272066, No. 61962006); 四川省科技计划资助 (No. 2021JDJQ0021, No. 2022YFG0186); 教育部人文社会科学研究规划基金 (No. 22YJAZH088); 成都市“揭榜挂帅”科技项目 (No. 2022-JB00-00002-GX, No. 2021-JB00-00025-GX); 成都市技术创新研发项目 (No. 2021-YF05-02413-GX, No. 2021-YF05-02414-GX); 中国电子科技集团公司第五十四研究所高校合作课题 (No. SKX212010057); 成都信息工程大学科技创新能力提升计划 (No. KYTD202222)

中图分类号: TP311

文献标识码: A

文章编号: 0372-2112(2023)07-1868-07

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20220490

## A Join Optimizer Based on Asynchronous Dueling DQN and Plan Latency Prediction Network

GAO Rui-wei<sup>1</sup>, QIAO Shao-jie<sup>1</sup>, HAN Nan<sup>2</sup>, MIN Sheng-jie<sup>3</sup>, LI He<sup>4</sup>,  
QIN Xiao<sup>5</sup>, ZHANG Tao<sup>6</sup>, YUAN Chang-an<sup>7</sup>

(1. School of Software Engineering, Chengdu University of Information Technology, Chengdu, Sichuan 610225, China;

2. School of Management, Chengdu University of Information Technology, Chengdu, Sichuan 610103, China;

3. CEC Zhiyuan Data Technology Co., Ltd., Beijing 610225, China;

4. School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi 710126, China;

5. Nanning Normal University, Nanning, Guangxi 530001, China;

6. Yibin University, Yibin, Sichuan 644000, China;

7. Guangxi Academy of Sciences, Nanning, Guangxi 530007, China)

**Abstract:** Join order selection is a challenging research topic in the field of query optimization, and it is very important for database management system to obtain good query performance. However, both traditional optimization methods and existing intelligent optimization methods have disadvantages such as long planning time, easily to obtain poor quality join plans, encoding without considering structural characteristics, making join plans unable to reflect the real execution time due to dependency on cardinality estimation and cost estimation. In order to solve the above problems, a new join opti-

mizer ADP-Join (Asynchronous Dueling DQN and Plan latency prediction network for Join order selection) is proposed. ADP-Join integrates a new encoding method that can distinguish join plans of different structures. ADP-Join designs a plan latency prediction network to improve the reward mechanism of existing reinforcement learning-based optimizers. Furthermore, the asynchronous update mechanism is proposed to improve the Dueling DQN model to improve the training performance and reduce the training time. Extensive experimental results show that ADP-Join outperforms existing intelligent optimizers on real TPC-H and JOB datasets.

**Key words:** join order selection; query optimization; join plan; reinforcement learning; asynchronous update

**Foundation Item(s):** National Natural Science Foundation of China (No.62272066, No.61962006); Sichuan Science and Technology Program (No.2021JDJQ0021, No.2022YFG0186); Planning Foundation for Humanities and Social Sciences of Ministry of Education of China (No.22YJAZH088); Chengdu 'Take the Lead' Science and Technology Project (No.2022-JB00-00002-GX, No.2021-JB00-00025-GX); Chengdu Technology Innovation and Research and Development Project(No.2021-YF05-02413-GX, No.2021-YF05-02414-GX); The 54th Research Institute of China Electronics Technology Group Corporation-University Cooperation Project (No.SKX212010057); Science and Technology Innovation Capability Improvement Project of Chengdu University of Information Technology (No.KYTD202222)

## 1 引言

查询优化是数据库领域中的重点和难题. 查询优化包括连接顺序选择、基数估计以及代价估计<sup>[1]</sup>. 近期,在数据库领域中研究人员将查询优化与人工智能技术结合,取得了一系列不错的成果<sup>[2-7]</sup>. 但是现有的优化器还存在着一些缺点:(1) 如 Rejoin<sup>[8]</sup>和 DQ<sup>[9]</sup>等现有优化器的编码方法不能捕捉连接计划的结构特征,使得不同结构的连接计划对应相同的编码,进而无法为后续强化学习算法提供准确的状态表示;(2) 如 Rejoin 和 DQ 等现有优化器的奖励机制基于基数估计和代价估计. 现实世界的数据库通常是非平衡的、相关的,这将导致错误的基数估计,进而生成次优的连接计划<sup>[10]</sup>;(3) 如 Neo<sup>[11]</sup>等现有优化器的搜索策略基于简单的启发式搜索算法,不能从全局的角度指导连接顺序选择,可能忽略没有尝试过的连接计划,进而陷入局部最优解;(4) 如 DQ 等现有优化器利用经验回放池<sup>[12]</sup>进行参数更新. 但是,经验回放池存在一些缺点如需要巨大的存储空间、数据样本由旧策略生成导致训练效果下降、训练时间长且收敛不稳定以及数据样本的相关性太强.

基于上述问题,本文提出一种新型基于异步 Dueling DQN 和计划时间预测网络的连接优化器:ADP-Join. ADP-Join 的创新性为:(1) ADP-Join 集成了一种新的编码方法,区分不同结构的连接计划,为 Dueling DQN 提供准确的状态表示;(2) ADP-Join 提出了计划时间预测网络来改进现有优化器的奖励机制;(3) ADP-Join 基于 Dueling DQN,相比简单的搜索策略,ADP-Join 能够从全局角度解决问题,尽可能避免陷入局部最优解;(4) ADP-Join 利用异步更新机制来改进 Dueling DQN,提出异步 Dueling DQN 算法,大大加快了训练速度,节省了存储空间,并提高了 ADP-Join 的性能.

## 2 ADP-Join 的架构和关键技术

### 2.1 ADP-Join 的整体架构和工作机制

如图 1 所示,ADP-Join 包含三个重要的组成部分:(1) 编码方法,包括对查询语句编码和对连接计划编码,为 Dueling DQN 提供准确的状态表示;(2) 计划时间预测网络,包括 CNN (Convolutional Neural Network) 模型和 GRU (Gate Recurrent Unit) 模型<sup>[13]</sup>,CNN 主要处理查询语句的编码,GRU 主要处理连接计划的编码,能够捕捉连接计划的结构特征,进而准确预测完整的执行计划的执行时间,并且使用预测执行时间作为奖励;(3) 异步 Dueling DQN,包括多个子网络和一个全局网络,子网络和全局网络均为 Dueling DQN 且具有相同的参数数量. 异步 Dueling DQN 根据当前部分连接计划的状态执行相应的动作(选择两个表进行连接),并且确定所有表的连接顺序.

ADP-Join 的工作机制分为训练阶段和运行阶段. 在训练阶段,包括异步 Dueling DQN 的参数更新过程和计划时间预测网络的参数更新过程. 异步 Dueling DQN 的参数更新过程:ADP-Join 开启多个线程,在每个线程中不断与数据库管理系统交互,收集基于当前策略产生的数据样本(包括状态、动作、奖励等信息),用于计算每一个线程中的子网络的参数梯度,并将每个线程中的子网络的参数梯度传递给全局网络并更新全局网络的参数. 由于很难获得中间过程的奖励,ADP-Join 将中间过程的奖励设置为 0,类似现有基于强化学习的优化器<sup>[4, 5, 8, 9]</sup>. 计划时间预测网络的参数更新过程:计划时间预测网络和数据库的执行器分别对完整的连接计划给出预测执行时间和真实执行时间,用于更新计划时间预测网络的参数.

在运行阶段,首先,ADP-Join 对查询语句进行编码和连接计划进行编码. 需要注意的是,在初始阶段,所

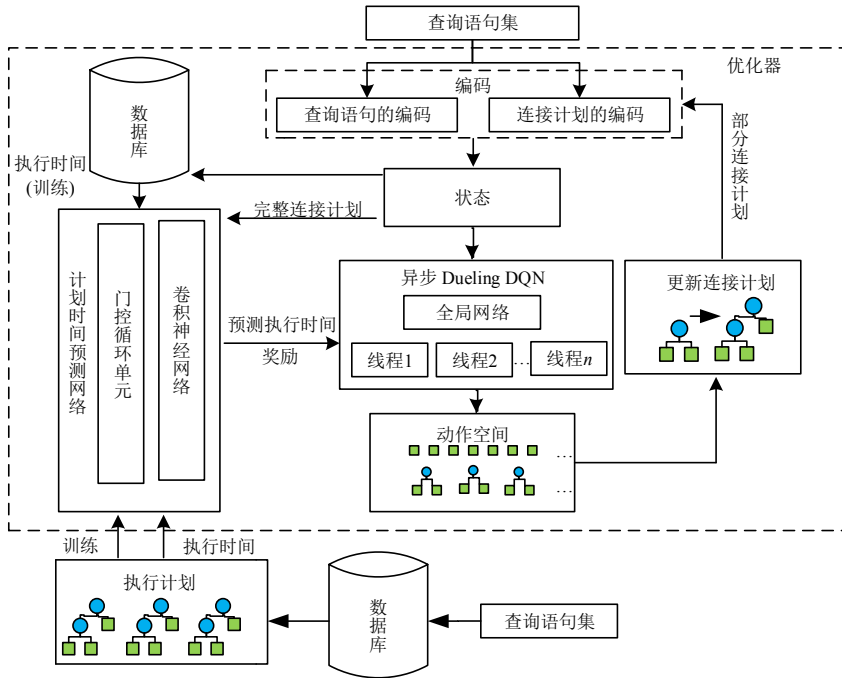


图1 ADP-Join的整体架构

有的表并未连接。其次,基于初始状态  $s_0$ ,ADP-Join 选择两个表进行连接并更新连接计划的编码,初始状态  $s_0$  被转换为下一阶段状态  $s_1$ 。接着,ADP-Join 根据状态  $s_{t-1}$  执行相应的动作,直到达到终止状态  $s_t$ (所有表均已经连接)。

### 2.2 ADP-Join 的编码方法

编码方法包括查询语句的编码 SQL-Encoding 和连接计划的编码 Join-Plan-Encoding。

#### 2.2.1 SQL-Encoding

如图2所示,给定查询语句的连接条件包括  $A. a=B. a$ ,  $A. b=C. b$  和  $A. c=D. c$ 。SQL-Encoding 将给定查询的连接条件编码为对称矩阵  $M$ ,其中,  $n$  表示数据库中所有表的数量。 $M$  中的元素取值为0或1,1表示两个表具有连接条件,反之则为0。

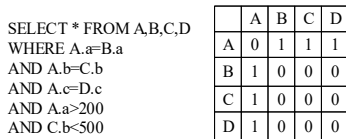


图2 查询示例与SQL-Encoding

#### 2.2.2 Join-Plan-Encoding

连接计划由不同类型的节点(叶子节点和内部节点)组成。叶子节点是指单表。内部节点是指多个表连接后的中间结果集。Join-Plan-Encoding 对连接计划中的每个节点进行编码,包括:节点的顺序、节点的类型、

节点中存在的列和列相关的选择谓词。具体为:(1) 使用两位整数编码节点的顺序,例如(00)表示未选择叶子节点,(01)表示第一次选择叶子节点或第一次生成内部节点,(02)表示第二次选择叶子节点或第二次生成内部节点,依此类推;(2) 使用独热向量编码节点的类型,例如,(01)表示左子叶节点,(10)表示右子叶节点,(11)表示内部节点;(3) 使用独热向量编码节点中存在的列,其中独热向量的维度为数据库中所有列的数量。节点存在某个列则独热向量中对应位置的值为1;(4) 列相关的选择谓词包括选择运算符和给定数值(或字符串)。使用独热向量编码选择运算符,例如:(001)表示等于,(010)表示小于,(100)表示大于。使用数据库现有的直方图统计值编码给定数值,取值范围为0到1。

### 2.3 ADP-Join 的计划时间预测网络

计划时间预测网络由CNN模型和GRU组成。如图3所示,将查询语句的编码和完整连接计划的编码输入到CNN和GRU模型进行处理。将CNN模型的输出与GRU模型的输出进行级联,经过Sigmoid激活函数层输出预测执行时间 pred-latency。最后,使用均方差损失函数最小化 pred-latency 和真实执行时间 latency 之间的差距。

需要注意的是,GRU不适宜处理树形结构的编码。通过遍历算法将树形结构的连接计划的编码转换为节点编码序列集。GRU处理节点编码序列集的计算公式如下:

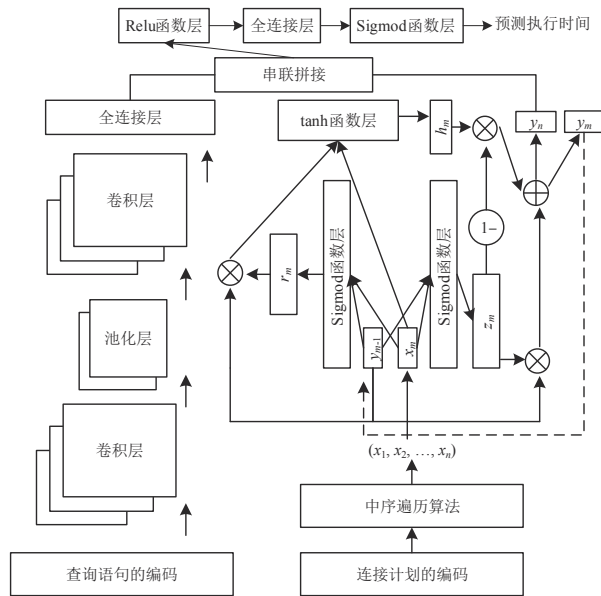


图3 计划时间预测网络的结构

$$r_m = \text{Sigmoid}(x_m W_r + y_{m-1} U_r) \quad (2)$$

$$z_m = \text{Sigmoid}(x_m W_z + y_{m-1} U_z) \quad (3)$$

$$h_m = \tanh(x_m W_h + (r_m \odot y_{m-1}) U_h) \quad (4)$$

$$y_m = y_{m-1} \odot z_m + h_m \odot (1 - z_m) \quad (5)$$

其中,  $x_m$  为节点编码序列集  $(x_1, x_2, \dots, x_n)$  中的特征向量,  $m$  为下标,  $0 < m < n$ ,  $n$  为节点编码序列集中的特征向量的数量.  $y_{m-1}$  为  $x_{m-1}$  作为输入向量时输出的特征向量,  $\text{Sigmoid}(\cdot)$  和  $\tanh(\cdot)$  是激活函数,  $r_m$ 、 $z_m$  和  $h_m$  是中间计算结果,  $\odot$  表示哈达玛积,  $W_r$  和  $U_r$  为重置门的参数矩阵,  $W_z$  和  $U_z$  为更新门的参数矩阵,  $W_h$  和  $U_h$  为用于计算  $h_m$  的参数矩阵,  $y_m$  表示  $x_m$  作为输入向量时输出的特征向量.

#### 2.4 ADP-Join 的异步 Dueling DQN 算法

异步 Dueling DQN 算法主要是利用异步更新机制来改进 Dueling DQN. 异步 Dueling DQN 包括一个全局网络和多个子网络. 全局网络和子网络拥有相同的网络结构和参数数量. 全局网络为 Dueling DQN, 包括: 输入层、隐藏层、价值函数层、优势函数层以及输出层.

Dueling DQN 以状态  $s$  作为输入, 其中状态  $s$  由查询语句的编码和连接计划的编码组成, 价值函数层输出状态值  $V$ , 优势函数层输出动作优势值  $A$ , 输出层输出  $Q$  值, 并由  $Q$  值确定需要执行的动作  $a$ . 计算  $Q$  值的公式如下所示.

$$Q(s_t, a_t, \alpha, \beta) = V(s_t, \alpha) + A(s_t, a_t, \beta) \quad (6)$$

其中,  $s_t$  表示  $t$  时刻的状态,  $a_t$  表示  $t$  时刻的动作,  $\alpha$  为价值函数层的参数,  $\beta$  为优势函数层的参数,  $V(s_t, \alpha)$  表示状态  $s_t$  的状态价值函数,  $A(s_t, a_t, \beta)$  为状态  $s_t$  时执行动

作  $a_t$  的动作优势函数. 由于连接顺序选择的目标是寻找最小执行时间的执行计划, 而 Dueling DQN 需要最大化奖励. 因此, 将预测执行时间作为奖励并取负值. Dueling DQN 的损失函数如下所示.

$$\text{loss}_1 = \sum_{t=0}^T (\text{pred} - \text{latency}) - Q(s_t, a_t, \alpha, \beta) \quad (7)$$

$$\text{loss}_2 = \sum_{t=0}^T \gamma \cdot Q(s_{t+1}, a_{t+1}, \alpha, \beta) - Q(s_t, a_t, \alpha, \beta) \quad (8)$$

$$\text{loss}_3 = W_1 \text{loss}_1 + W_2 \text{loss}_2 \quad (9)$$

其中  $\text{loss}_1$  表示状态  $s_t$  为终止状态时的损失函数,  $\text{loss}_2$  表示状态  $s_t$  为非终止状态时的损失函数.  $\text{pred} - \text{latency}$  为预测执行时间, 由计划时间预测网络给出.  $\gamma$  为折扣因子, 表示未来奖励的影响程度.  $\gamma \in [0, 1]$ ,  $\gamma = 0$  表示未来的奖励没有影响,  $\gamma = 1$  表示只关注未来的奖励.  $\text{loss}_3$  表示联合损失函数. 当  $s_t$  为终止状态时,  $W_1 = 1$  且  $W_2 = 0$ ; 当  $s_t$  为非终止状态时,  $W_1 = 0$  且  $W_2 = 1$ .

### 3 ADP-Join 的实验评估

#### 3.1 数据集与实验环境

实验数据集为 TPC-H<sup>[14]</sup> 和 JOB (Join Order Benchmark)<sup>[15]</sup>. TPC-H 包含 8 个表和 22 个查询语句模板. JOB 包含 21 个表和 33 个查询模板, 113 个基于查询模板的查询语句. JOB 一共包含 3.6 GB 的数据, 最大的表有 3 000 万行, 查询语句包含的表的数量最多为 17 个, 平均为 8 个.

实验服务器配置为 16 核 32 线程的 CPU、128 GB 内存和 2 TB 硬盘. 数据库为 Postgresql, 软件环境为 Python 3.7、TensorFlow 2.0 和 RLLib3.

实验评估中的对比模型分别为 Rejoin、DQ、Neo、Alphajoin、QP 和 RD. 其中对比模型的主要参数设置: RLLib 提供的 PPO 常规参数配置<sup>①</sup>和 DQN 常规参数配置<sup>②</sup>. 实验评估指标为平均相对代价和平均相对延迟.

#### 3.2 ADP-Join 的代价评估

如图 4(a) 所示, 在 JOB 数据集上, ADP-Join 的平均相对代价低于其他方法. 并且基于强化学习的方法如 ADP-Join、Rejoin、DQ、Neo 以及 Alphajoin<sup>[16]</sup> 等均优于传统方法如 QP (QuickPick)<sup>[17]</sup>、RD (Right-Deep)<sup>[15]</sup>. 如图 4(b) 所示, 在 TPC-H 数据集上, ADP-Join 的性能依然优于其他方法. 但是, 相比于 JOB 数据集, 在 TPC-H 数据集上各个方法的平均相对代价的差距很小. 并且基于强化学习的方法和传统方法的平均相对代价均接近 1. 因为 TPC-H 中查询语句包含表的数量较少 (最多不超过 8 个), 各种方法都容易为给定查询找到一组最佳连接顺序. 与 TPC-H 数据集不同, JOB 数据集的查询语句包含多个表 (最多为 17 个), 传

① <https://docs.ray.io/en/latest/rllib/rllib-algorithms.html#ppo>

② <https://docs.ray.io/en/latest/rllib/rllib-algorithms.html#dqn>

统方法很难枚举所有候选连接计划,而基于强化学习方

法能够找到较好连接计划.

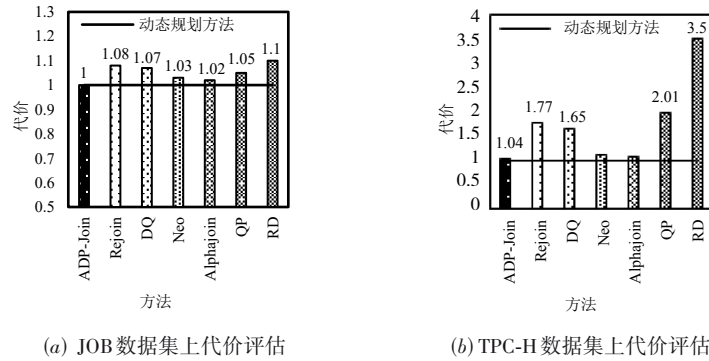


图4 基于不同数据集的ADP-Join的代价评估

### 3.3 ADP-Join 的执行时间评估

如图5(a)所示,在JOB数据集上,ADP-Join的平均相对延迟为0.68,低于其他方法. ADP-Join的平均相对延迟比动态规划方法低32.0%. 基于强化学习的方法如ADP-Join、Rejoin、DQ、Neo以及Alphajoin均优于传统方法. 并且Neo和Alphajoin的平均相对延迟低于动态规划方法. 然而,传统方法如QP和RD的平均相对延迟均

大于1. 如图5(b)所示,在TPC-H数据集上,ADP-Join的性能优于其他方法. 并且基于强化学习的方法的平均相对延迟均远小于1,传统方法的平均相对延迟均远大于1. 换言之,基于强化学习的方法与传统方法的平均相对延迟的差距很大,与图4(b)显示的结论相反. 主要原因是错误的基数估计导致不准确的代价估计,而执行时间能够较好地衡量连接计划的质量.

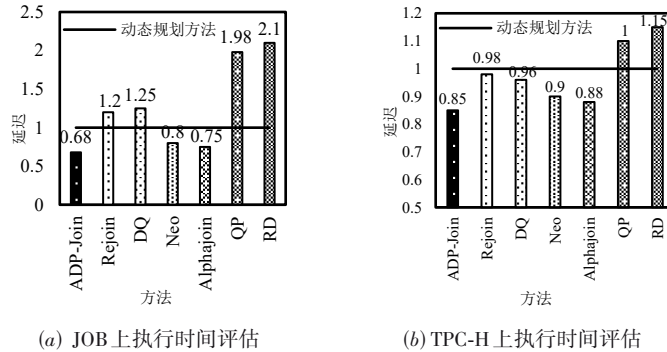


图5 基于不同数据集的ADP-Join的延迟评估

### 3.4 编码方法有效性评估

为了证明ADP-Join编码方法的有效性,引入对照方法ADP-Join-Encoding. ADP-Join-Encoding是指将ADP-Join的编码方法替换为DQ的编码方法,其余组件与ADP-Join保持一致. 如图6(a)和(b)所示,在JOB数据集上,随着查询语句中表数量的增加,ADP-Join与ADP-Join-Encoding的平均相对代价以及平均相对延迟的差距均增大,因为对于越复杂的多表连接查询,需要区分不同结构的连接计划,提供更加准确的状态表示,才能使得优化器生成更好的执行计划. 并且在平均相对代价和平均相对延迟方面,ADP-Join始终小于ADP-Join-Encoding.

### 3.5 计划时间预测网络有效性评估

为了证明ADP-Join的计划时间预测网络的有效性,引入对照方法ADP-Join-Cost. ADP-Join-Cost是指ADP-Join不使用计划时间预测网络提供的预测执行时间作为奖励,而是使用数据库优化器提供的估计代价. 如图7(a)和7(b)所示,在JOB数据集上,ADP-Join的平均相对代价和平均相对延迟均低于ADP-Join-Cost. 相比Rejoin和DQ,ADP-Join-Cost的平均相对代价和平均相对延迟更低. 换言之,即使ADP-Join, Rejoin和DQ均使用类似的奖励机制,ADP-Join的性能依然优于Rejoin和DQ. 主要原因是ADP-Join的其余组件依然能够提升性能.

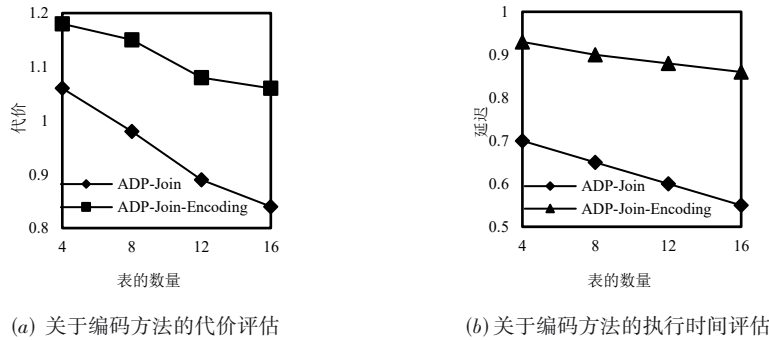
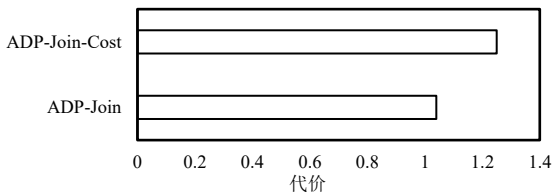
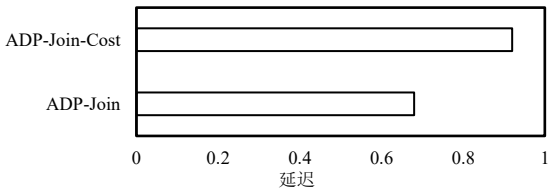


图 6 JOB数据集上ADP-Join的编码方法有效性评估



(a) 关于计划时间预测网络的代价评估

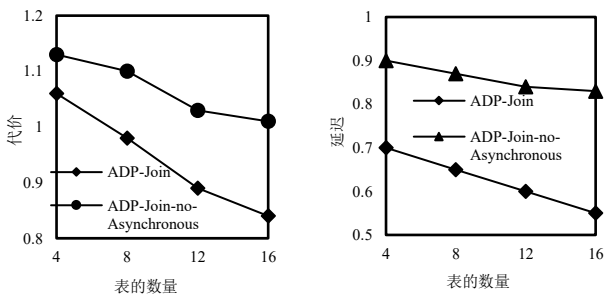


(b) 关于计划时间预测网络的执行时间评估

图 7 JOB数据集上ADP-Join的计划时间预测网络有效性评估

### 3.6 ADP-Join的异步更新机制有效性评估

为了证明ADP-Join的异步更新机制的有效性,引入对照方法ADP-Join-no-Asynchronous. ADP-Join-no-Asynchronous是指ADP-Join并不设计异步更新机制改进Dueling DQN. 如图8(a)和(b)所示,在各个包含不同数量的表的查询语句集上,ADP-Join的平均相对代价和平均相对延迟均低于ADP-Join-no-Asynchronous. 这



(a) 异步更新机制的代价评估 (b) 异步更新机制的执行时间评估

图 8 JOB数据集上ADP-Join的异步更新机制有效性评估

表明异步更新机制能够提升Dueling DQN的性能.

## 4 结论与未来工作

本文提出了新型连接优化器ADP-Join. ADP-Join能够区分不同结构的连接计划. ADP-Join设计了计划时间预测网络. 将计划时间预测网络提供的预测执行时间作为奖励能够避免错误的基数估计导致生成次优的连接计划. 此外,提出异步Dueling DQN算法,能够避免经验回放池的缺点和提高ADP-Join的性能. 在未来的工作中,基于ADP-Join进一步地研究端到端的查询优化器.

### 参考文献

- [1] QIAO S J, YANG G P, HAN N, et al. Cardinality estimator: Processing SQL with a vertical scanning convolutional neural network[J]. Journal of Computer Science and Technology, 2021, 36(4): 762-777.
- [2] LI G L, ZHOU H X, GAO L. AI meets database: AI4DB and DB4AI[C]//Proceedings of SIGMOD Conference. New York: ACM, 2021: 2859-2866.
- [3] LI G L, ZHOU X H, SUN J. openGauss: An autonomous database system[J]. Proceedings of VLDB Endowment, 2021, 14(12): 3028-3041.
- [4] MARCUS R, PAPAEMMANOUIL O. Towards a hands-free query optimizer through deep learning[EB/OL]. (2018)[2022]. <https://api.semanticscholar.org/CorpusID:52880119>.
- [5] YU X, LI G L, CHAI C L. Reinforcement learning with tree-LSTM for join order selection[C]//Proceedings of 36th International Conference on Data Engineering. Washington: IEEE, 2020: 1297-1308.
- [6] ZHOU H X, LI G L, CHAI C L. A learned query rewrite system using Monte Carlo tree search[J]. Proceedings of VLDB Endowment, 2021, 15(1): 46-58.
- [7] KAOUDI Z. ML-based cross-platform query optimization

- [C]//Proceedings of the 36th International Conference on Data Engineering. Washington: IEEE, 2020: 1489-1500.
- [8] MARCUS R. Deep reinforcement learning for join order enumeration[C]//Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. New York: ACM, 2018: 31-34.
- [9] KRISHNAN S, YANG Z H, GOLDBERG K, et al. Learning to optimize join queries with deep reinforcement learning[EB/OL].(2018)[2022]. <https://api.semanticscholar.org/CorpusID:51952349>.
- [10] WANG J Y, CHAI C L, LIU J B. FACE: A normalizing flow based cardinality estimator[J]. Proceedings of VLDB Endowment, 2021, 15(1): 72-84.
- [11] MARCUS R, NEGI P, MAO H Z. Neo: A learned query optimizer[J]. Proceedings of VLDB Endowment, 2019, 12(11): 1705-1718.
- [12] MNIH V, KAVUKCUOGLU K, SILVER D. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.
- [13] DEY R, SALEM F M. Gated recurrent unit neural networks gate-variants[C]//Proceedings of 60th International Midwest Symposium on Circuits and Systems. Washington: IEEE, 2017: 1597-1600.
- [14] POSS M, FLOYD C. New TPC benchmarks for decision support and web commerce[J]. SIGMOD Record, 2000, 29(4): 64-71.
- [15] LEIS V, GUBICHEV A, MIRCHEV A. How good are query optimizers, really? [J]. Proceedings of VLDB Endowment, 2015, 9(3): 204-215.
- [16] ZHANG J, ABEDJAN Z, HOSE K. AlphaJoin: Join order selection à la AlphaGo[C]//Proceedings of the VLDB 2020 PhD Workshop Co-located with the 46th International Conference on Very Large Databases. Trondheim: VLDB Endowment, 2020: 5-8.
- [17] WASS F. Join order selection[C]//Proceedings of British National Conference on Databases. Berlin: Springer, 2000: 256-265.



乔少杰(通讯作者) 男,1981年10月出生  
于山东招远,博士,现为成都信息工程大学软件  
工程学院教授. 主要研究方向为人工智能  
数据库、时空数据库、机器学习.

E-mail: sjqiao@cuit.edu.cn

#### 作者简介



高瑞玮 男,1996年12月出生于四川射  
洪,现为成都信息工程大学软件工程学院硕士  
研究生. 主要研究方向为人工智能数据库.

E-mail: 1013384678@qq.com