面向实时设置中数据库优化问题的量子-经典混合计算框架

Hanwen Liu[§]南加州大学
计算机科学系美国加利福尼亚州洛杉矶 hanwenliu@usc.edu

Ibrahim Sabek[§]南加州大学计算机科学系美国加利福尼亚州洛杉矶 sabek@usc.edu

arXiv:2602.14263v1 [cs.DB] 15 Feb 2026

摘要——**量子计算在求解数据库中的复杂优化问题（如连接顺序和索引选择）方面展现出良好前景。以往的工作往往将构造好的问题直接提交给黑盒量子求解器或量子启发式求解器，并期望直接获得良好的最终解。由于这些求解器具有黑盒特性，用户无法对求解过程进行细粒度控制，以平衡准确性和效率；这反过来限制了其在实时场景中的灵活性，而大多数数据库问题都出现在此类场景中。此外，这也限制了处理大规模数据库优化问题的潜力。本文提出构建首个实时量子增强数据库系统的愿景，以透明的方式求解数据库优化问题。我们设计了两种互补的可扩展性策略，以应对超出硬件限制的大规模挑战、过度复杂和规模过大问题。我们将该方法集成到数据库查询优化器中，构建了一个初步原型，并在真实工作负载上进行评估；与经典查询优化器相比，性能最高提升 14 倍。与黑盒量子求解器相比，我们在效率和解的质量方面也均有所提升。**

索引词—**量子计算、查询优化、现代硬件上的数据系统**

I. 引言

数据库系统涉及多种优化问题，例如连接顺序（如 [1]–[5]）、事务调度（如 [6], [7]）、多查询优化（如 [8], [9]）和索引调优 [10]–[12]。其解的质量对整体性能至关重要。随着数据量呈指数级增长、查询工作负载日益复杂，高效求解这些优化任务变得愈发困难 [13]。这些问题本质上具有组合性，数十年来，数据库系统一直依赖启发式方法和其他经典优化技术（如 [1], [2], [6], [8]–[11]），以平衡解的质量与计算效率。尽管这类方法在实践中行之有效，但当搜索空间庞大且高度复杂，或必须在实时或自适应条件下进行探索时，其可靠性会降低。在这些情形下，目标往往从寻找最优解转变为快速获得高质量的解；如果能够显著缩短优化时间，那么接受次优结果也是合理的。

一个典型例子是连接顺序优化问题，这是查询优化中的一项关键挑战，且属于NP难问题 [14] ，因为搜索空间会随着关系数量的增加而呈指数级增长。基于动态规划（DP）的方法（例如， [15]）可以通过穷举小型查询（即关系数量相对较少的查询）的所有可能连接序列来计算最优结果，但面对较大的查询时则变得不切实际。为了解决这一局限，人们提出了各种启发式方法（例如， [15]–[18]），以平衡解的质量与计算效率。

近年来，基于机器学习的技术已成为数据库优化的有前景的替代方案（例如， [3]–[5], [7], [12], [19]–[21]）。然而，在困难的组合问题中，这些技术仍面临重大挑战，包括冷启动问题（例如， [3]）、对大型训练数据集的依赖（例如， [22]），以及随着工作负载不断变化而需要频繁重新训练（例如， [19]）。

值得庆幸的是，量子硬件的最新进展（例如， [23]–[25]）使量子计算成为一种强大且前景广阔的计算范式。基于广为接受的复杂性理论假设 [26]–[28]，人们认为量子系统的计算能力将超越经典系统，有望 (i) 在特定问题上实现量子优势 [29]–[31] ，比已知最优的经典算法更快地求解这些问题，例如密码分析 [32], [33] 和模拟 [29], [30], ；以及 (ii) 高效探索指数级庞大的搜索空间，尤其是针对某些优化问题 [34], [35]。此外，通过 D-Wave [36] 和 IBM Quantum [24]等云平台，量子计算正变得越来越容易使用。前瞻性预测 [37], [38]预计量子计算能力将迅速提升，并且近期将出现本地部署硬件，从而实现更低的执行延迟和更紧密的系统集成。

尽管量子计算具有变革性潜力，但量子计算系统仍处于发展的早期阶段，并面临

[§]两位作者贡献相同，姓名按字母顺序排列。

重大的硬件相关挑战。当前设备运行于含噪中等规模量子（NISQ）阶段，并存在量子比特数量有限、连通性稀疏、噪声显著以及缺乏可靠纠错等局限 [39]。量子退火机 [23],（例如 D-Wave [36], 开发的设备）通过利用量子隧穿 [40] 和并行探索解空间，在现实应用中的组合优化问题（例如生产调度 [34], 、电力网络优化 [35]）方面展现出良好前景。然而，其有限的量子比特容量和稀疏的硬件拓扑结构（例如 Pegasus [41], 、Zephyr [42]）限制了能够高效嵌入的问题的规模和密度。因此，单纯依靠量子退火仍不足以解决复杂的大规模数据库优化任务（例如 [43]–[49]）。

为克服这些瓶颈，量子—经典混合计算应运而生，成为一种实用范式：量子处理单元（QPU）仅加速更广泛的经典优化流程中计算量最大的子问题 [50], [51]。这种混合方式充分发挥了两种计算模式各自的优势：经典处理器负责大规模编排和问题分解，而量子退火机则专注于采样并探索复杂的能量景观，这些任务若采用其他方式，计算成本将高得难以承受。然而，将现有的量子—经典混合方法应用于数据库优化，会带来若干独特挑战，尤其是在实时场景中：

C1. 求解器不透明性：依赖黑盒求解器。目前的大多数研究（例如 [52]–[54]）都高度依赖商业黑盒量子求解器（例如 D-Wave 的 CQM 求解器 [51]和 NL-Solver [55]），以及量子启发式求解器（例如数字退火器 [56]）。由于量子求解器缺乏必要的透明度，研究人员通常会将优化问题表述为二次无约束二元优化（QUBO）格式，并直接提交给这些求解器（例如， [57], [58]）。这种黑盒用法限制了用户对量子退火过程进行细粒度控制，也降低了调试和调优的灵活性。因此，这阻碍了面向实时场景的数据库优化高效算法设计。

C2. QUBO 过度复杂。 现有大多数方法都力求将原始问题忠实地映射为 QUBO 形式 [46], [54],[57], [59]–[61]。尽管这保留了语义，却会产生过于复杂的 QUBO，导致交互图稠密且辅助变量过多。这使变量到量子比特的映射更加困难，也减缓了在 D-Wave 的 Pegasus [41]和 Zephyr [42]等稀疏连接硬件上的嵌入过程。此外，增加的复杂性会降低求解器找到高质量解的性能。然而，许多数据库问题都存在冗余或稀疏性。例如，在连接顺序优化中，如果选择了连接操作 $A \bowtie B$ 和$B \bowtie C$ ，就无需再引入额外变量来编码数据表 A 和 C 之间存在连接这一事实 [54],[62]。盲目地保留所有相关性会增加复杂度，却无法带来相应的收益。

C3. 二次无约束二元优化规模过大：超出硬件限制与语义感知分解。随着数据库优化问题的规模不断扩大，将二次无约束二元优化映射到物理量子比特可能会超出硬件容量。要确保可行，必须将每个决策变量映射到一个物理量子比特。超出这一阈值的问题必须先分解，之后再重新组合。忽略数据库语义的通用分解会将每个二次无约束二元优化视为独立的问题。缺少结构性语义信息，会导致重新组合时无法利用潜在的简化机会，并限制系统在参数略有变化时的适应能力。因此，要实现实际应用中的扩展，分解过程必须保留任务语义，并尽量减少各子问题之间的共享变量，从而确保重新组合过程高效，且最终解保持有效。

为应对这些挑战，我们提出了一种可扩展的量子-经典混合框架，用于在**实时场景**中求解大规模数据库优化问题。具体而言，该框架在保持解正确性的同时，简化**（即迭代相关性松弛）**或**分解**（即问题感知分解）QUBO 结构。这一范式直接利用量子退火机的最低层级操作（即采样），从而实现透明、细粒度的控制，平衡解的质量与运行时效率。我们的初步原型和评估结果表明，与经典数据库组件（即 PostgreSQL 的查询优化器）以及商用黑盒量子求解器相比，性能最高可提升 14 倍。

本文提出了三个关键要点：

1)新视角： 我们提出了一种构想：打造首个实时量子增强型数据库系统，将量子与经典优化器紧密集成，从而能够以透明的方式求解大规模数据库优化问题。

2) 新方法：我们开发了两种互补的可扩展性策略。首先，采用迭代式相关性松弛框架，在保留完整变量集的同时，根据语义及其对解的质量的影响，剪枝相关性并有选择地重新引入相关性。其次，采用问题感知的分解–嵌入–采样–组合流程，在保留结构的同时，将 QUBO 划分为具有语义意义的子问题，以求解大规模优化问题。

3) 新的研究方向与挑战：我们概述了一种前景广阔的量子-经典混合框架，用于大规模优化，并与数据库系统深度集成；同时指出，效率、简洁性和通用性仍是亟待应对的挑战。

本文结构安排如下。第二节介绍量子退火的基础知识，以及当前量子计算方法论的必要背景。第三节介绍系统架构并形式化问题设定。第四节详细介绍提出的两种方法。第五节介绍集成了数据库查询优化器的初步原型。第六节报告初步实验结果。最后，第七节总结本文的研究发现和

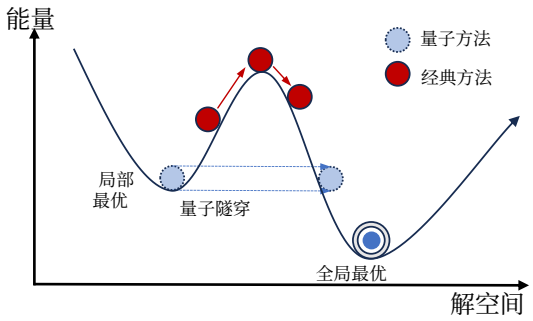


图 1：量子退火与经典方法的比较。

这篇愿景论文的贡献。

II. 背景与预备知识

量子退火（第 II-A 节）是本研究的理论基础和核心技术，支撑着混合量子经典优化工作流（第 II-B 节）。随后，我们将介绍用于二次无约束二元优化的相关性松弛（第 II-C 节）以及问题感知的混合优化流程（第 II-D 节）。

A. 量子退火

量子退火（QA）是一种利用量子硬件求解复杂优化问题的计算范式 [63]。它与经典模拟退火（SA） [64]有相似之处：两种方法都会通过对当前状态施加随机扰动来探索搜索空间。图 1 对比了 SA 和 QA。在 SA 中，如果提议的移动能够降低目标函数，就会接受该移动；否则，以取决于当前温度的概率接受。SA 中的探索由温度调度下的热涨落驱动。在 QA 中，系统在随时间变化的哈密顿量作用下演化；横向场会诱发量子涨落，从而促进探索。量子隧穿还能使系统穿越而非通过热力方式跨越能量势垒，从而逃离次优状态（即避免陷入局部极小值），进而提高收敛到全局最优的可能性。

QA 可求解可建模为二次无约束二元优化（QUBO） [65]的问题。在 QUBO 形式中，QA 寻找一个二元赋值 $s \in \{0,1\}^n$ ，使二次能量函数 [66]最小：

$$E(\vec{s}) = \sum_{ij} J_{ij}\, s_i\, s_j + \sum_i h_i\, s_i,$$

其中， s_i 和 s_j 是表示量子比特状态的二进制变量， h_i 表示量子比特 i上的局部偏置， J_{ij} 表示量子比特之间的相互作用强度。在量子退火过程中，初始叠加态会演化为使系统能量最小化的状态，从而编码原始组合优化问题的解。

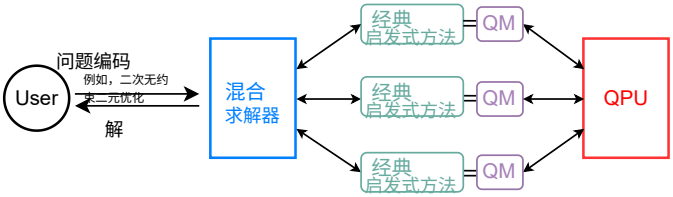


图 2：量子-经典混合优化的一般工作流程。

B. 量子-经典混合优化

量子-经典混合优化将经典启发式方法与量子退火（QA） [63] 相结合，以提高可扩展性和解的质量。如图 2 所示，用户对优化问题进行编码，并将其提交给混合求解器；该求解器会启动多个并行工作线程，每个线程都结合一个经典启发式模块和一个量子模块（QM）。启发式模块对搜索空间进行预处理、分解和细化，而 QM 则向量子处理单元（QPU）提交相应的量子查询，并返回带有能量值的样本；这些反馈会指导后续的启发式更新。该循环会持续进行，直到达到时间预算或满足停止条件，随后返回最佳解。大多数商业黑盒量子求解器，例如 D-Wave 的 CQM-Solver [51] 和 NL-Solver [55]，都采用这种混合工作流程。

C. *QUBO* 的相关性松弛

相关性松弛在保留全部决策变量的同时简化原始 QUBO。这种技术已在若干混合优化方法中得到探索 [36], [67], [68]。一种常见策略是优先处理能量贡献最大的相关性；在 QUBO 表示中，这相当于根据各成对项 J 对哈密顿量能量 [69]的影响进行排序。然而，纯粹基于能量的准则并未充分考虑数据库优化问题中的语义正确性，因为能量较低的相关性也可能保证关键语义。例如，在连接顺序优化 [48], [57], 中，低能量相关性可能编码了一项有效性约束，用以避免重复连接操作或不可行的连接结构；剪除这一相关性可能导致生成无效计划。同样，在索引调优 [70], 中，一些权重较低的约束可确保遵守存储预算，移除这些约束可能会产生不可行的解。因此，仅依据能量进行相关性剪枝而不考虑语义含义，并不足以应对数据库优化问题。

D. 问题感知型混合优化流水线

商业量子服务提供商（例如 D-Wave [36],IBM [71], 富士通 [72]）提供混合优化流水线。尽管这些流水线允许用户进行有限的自定义，但它们依赖次嵌入 [73]和无约束合并 [36], 等通用技术，而这些技术未考虑问题特定语义，因此限制了其在数据库优化场景中的有效性。要想在

实际应用中发挥作用，分解必须利用问题特征，而非通用的二次无约束二元优化图属性，并尽量减少子问题间的共享变量，以避免重组瓶颈。此前少有量子-经典数据库优化研究 [58], [60], [74]在二次无约束二元优化构造之前，利用逻辑层面的特征（例如，剪除不太可能的连接树 [58] 或低竞争事务 [60]）。虽然这能缩小规模，但却将二次无约束二元优化结构本身视为黑盒，未能考虑数据库语义如何映射到嵌入和求解器配置。因此，这些解无法复用，也无法适应工作负载的变化。例如，过于激进地剪除连接树，可能会排除一些在工作负载变化后可能成为最优方案的查询计划。

III. 系统概述

本节首先概述我们对量子增强数据库系统的构想及其关键组件（第 III-A 节）。随后，我们正式定义在给定时间约束下实现最佳解精度的问题设定，为系统的应用提供指导（第 III-B 节）。

A. 系统架构

图 3 概述了我们对量子增强数据库系统的构想。我们设计了一种量子-经典混合数据库优化器，使其与核心数据库管理系统组件紧密交互，包括事务管理器、查询优化器和执行引擎。这个前瞻性系统解决了求解大规模数据库优化问题时面临的三项挑战（参见第 I 节）。

在量子-经典混合数据库优化器的工作流程中，系统首先接收一个大规模数据库优化问题，例如事务调度、查询或算子调度，或连接顺序问题。随后，问题被编码为变量域模型，例如二次无约束二元优化（QUBO）。如果 QUBO 包含的变量数量超过量子退火机的容量（挑战 C3），系统便会采用问题感知分解（详见第四节 B）。之后，变量会映射到物理量子比特，作为在量子硬件上执行的初始步骤。如果 QUBO 过于稠密（挑战 C2），则会应用相关性松弛（详见第 IV-A 节）。随后，在量子硬件上执行量子采样这一最低层级且基础的操作。基于解的质量和用户-

在既定的时间约束下，我们的系统会确定是否调用迭代相关性松弛机制（详情见第 IV-A 节）以进一步提升解的质量。如果调用，则利用获得的样本及其能量更新 QUBO 参数，并继续执行该过程。否则，将执行分解的逆过程——问题感知组合（详情见第四节 B），以生成最终的 QUBO 解。最后，对其进行解码，以获得原始问题的解。这种端到端工作流不依赖任何黑盒量子求解器（挑战 C1），并提供

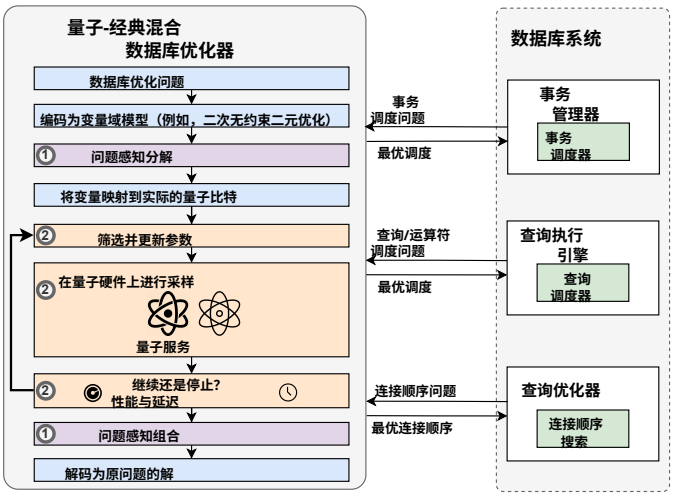


图 3：量子增强数据库系统概览。

对量子硬件上的大规模优化问题求解过程进行透明的量子采样控制。

B. 问题陈述

我们考虑在数据库 $\mathcal{D}$上执行查询 $\mathcal{Q}$的优化任务。候选解记为 $s(z)$（例如，查询计划），通过量子解空间中的编码 $z \in \{0,1\}^n$表示，其中 n 是该编码中的二元决策变量数量。目标是在遵守量子处理时间严格预算的同时，最大化解的质量。

性能。我们通过在数据库 $\mathcal{D}$上执行查询 $\mathcal{Q}$ 的性能来衡量解的质量。具体而言，

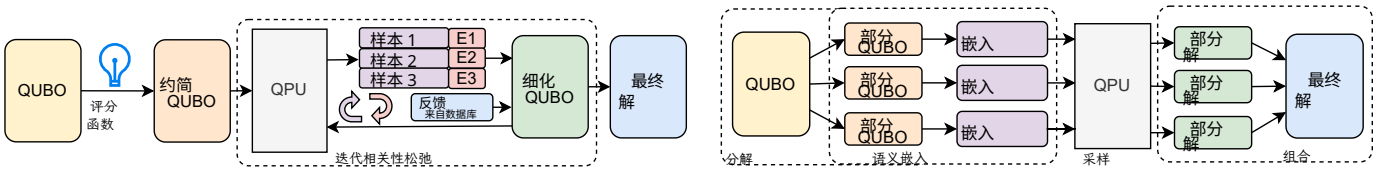
$$\max \text{Perf}(s;\mathcal{Q};\mathcal{D}) \iff \min \text{Latency}(s;\mathcal{Q};\mathcal{D}),$$

其中，Latency$(s;\mathcal{Q};\mathcal{D})$ 是在 $\mathcal{D}$上采用解 s执行 $\mathcal{Q}$所需的时间，因此延迟越低，质量越高。**量子生命周期**。该指标表示量子计算流水线为获得候选解 z所耗费的时间。为了获得高质量的解，算法可能需要与量子硬件交互多次（参见第 IV-A 节）。设 N 为迭代次数。在第 i次迭代中，令 T_{Q_i} 表示量子求解器的计算时间，T_{C_i} 表示数据库管理系统与量子服务之间的通信时间，T_{R_i} 表示客户端的二次无约束二元优化细化时间。量子生命周期总时间为

$$\text{Time}_{\text{Quantum}}(z) = \sum_{i=1}^{N} \left(T_{Q_i} + T_{C_i} + T_{R_i}\right),$$

其中，Time$_{\text{Quantum}}(z)$ 是在量子生命周期内生成 z所花费的端到端时间。

时间预算下的性能最大化。给定时间预算 $\tau > 0$，我们的目标是在此预算内找到



（a）迭代相关性松弛。　　（b）问题感知分解与组合　　on.

图 4：可扩展混合量子-经典方法。

能够使查询执行延迟最低的解。该问题表述为

$$
\begin{aligned}
&\textbf{Given: } \mathcal{Q},\ \mathcal{D},\ \tau \in \mathbb{R}_{>0} \\
&\textbf{Decisions: } z \in \{0,1\}^n,\ s = s(z) \\
&\textbf{Minimize: } \mathrm{Latency}(s; \mathcal{Q}; \mathcal{D}) \\
&\quad \textbf{s.t. } \mathrm{Time}_{\mathrm{Quantum}}(z) \ \leq\ \tau \qquad (\textit{Time budget}) \\
&\qquad\ \ g_j(s) \ \leq\ 0,\quad j = 1, \ldots, m \quad (\textit{Original constraints})
\end{aligned}
$$

IV. 面向大规模数据库优化的可扩展混合量子-经典方法

本节介绍两种方法。二者都通过避免依赖黑盒求解器，并提供对量子硬件的透明控制，来应对第一个挑战（C1）。第一种方法针对 QUBO 过度复杂这一第二个挑战（C2），在效率与准确性之间实现可调节的权衡。该方法引入迭代相关性松弛框架，在保留完整变量集的同时，通过剪枝并逐步重新引入相关性来简化 QUBO（第 IV-A 节）。第二种方法针对第三个挑战（C3），即 QUBO 超出当前硬件容量的问题。该方法采用问题感知分解与组合策略，同时保留变量和相关性，将 QUBO 划分为规模更小且具有语义意义的子问题，独立求解这些子问题，然后协调其结果（第四节 B）。

A. 迭代相关性松弛

有选择地松弛变量间的相关性，可以形成更稀疏的交互结构，从而简化将相关性映射到量子退火机的过程，使过于复杂的 QUBO 实例也能更自然地实现高效执行。通过恰当的相关性松弛，即使约简后的 QUBO 忽略了若干相关性，仍能生成质量相近的解。为解决这些局限，我们提出一种迭代相关性松弛框架，通过问题特定启发式和反馈驱动细化，在 QUBO 稀疏性（即过度松弛会导致 QUBO 稀疏）与解精度（即有限松弛可最大限度地保留原始 QUBO 特征）之间取得平衡。该迭代机制还能够灵活控制解精度与量子计算所带来的延迟之间的权衡。

图 4a 展示了一个完整框架。我们首先得到一个约简 QUBO，仅保留与初始 QUBO 中估计影响最大的相关性。这些相关性根据针对各类数据库问题设计的评分函数进行筛选。例如，在连接顺序优化中，我们优先考虑会产生较大中间结果基数的连接操作之间的相关性 [75]。对于事务调度，我们保留涉及频繁发生争用的事务的边 [76]。对于索引调优，我们着重关注查询中经常共同访问的索引之间的访问模式相关性 [77]。在这些情况下，评分函数会为这些相关性赋予较高分数，因为它们在原始数据库优化问题中起着关键作用。

量子处理单元（QPU），例如 D-Wave 量子退火机 [36]，，用作参数化采样器 [78]，在当前系数 h 和 J（见 II-A 节）下，针对约简后的相关性集合生成候选解（即样本及其对应的能量）。这些样本的经验分布提供了第一种反馈来源。此外，通过与实际数据库系统紧密集成，还可获得更多反馈，例如解质量指标（如连接顺序问题中当前查询计划的成本）以及约束违反统计（如无效连接子计划的数量）。我们会分析这些信号，以识别对实际查询执行性能有显著影响的缺失相关性。在后续迭代中，有选择地重新引入已识别出的相关性，即 J 中的成对项，从而逐步细化 QUBO 结构。

迭代过程在满足以下任一条件时终止：(i) 解的质量趋于稳定，例如连续若干次迭代后成本不再进一步降低；(ii) 能量景观达到预定义的稳定性阈值；或 (iii) 达到最大迭代次数。

收敛策略。在迭代式 QUBO 精化中，一个核心挑战是设计有效的策略以确保收敛。仅以贪心方式重新引入相关性，可能导致振荡或过早收敛，尤其是在数据库优化问题具有稠密成本结构时。为应对这一挑战，我们最小化理想的玻尔兹曼分布 [78], [80]与量子处理单元生成的经验分布之间的 Kullback-Leibler（KL）散度 [79]。使这两个分布保持一致，可近似完整约束问题的能量景观。由此产生的迭代更新类似于基于机器学习的梯度下降，逐步缩小两个分布之间的差异。作为补充方案，当最小化 KL 散度出现

在特定工作负载下出现不稳定。

B. 问题感知分解与组合

为应对第三个挑战，即 QUBO 的规模超出量子硬件的容量，我们通过保留所有变量和成对交互来维持完整的问题结构，同时将问题分解为较小的子问题，分别求解后再重新组合（例如， [36], [60], [71], [72], [82]–[84]）。我们提出了一种面向问题的混合优化流程，充分考虑底层数据库优化问题的关键语义。我们重点关注连接顺序优化 [54], [57], [66]和事务调度 [6], [59] 这两种代表性应用场景。该流程可推广到其他问题。它包含四个阶段，具体如下。

1) 分解阶段： 在此阶段，我们将大型二次无约束二元优化问题划分为较小的子问题，以减少分区间依赖，同时保留与数据库操作语义相关的所有相关性。对于连接顺序优化，我们利用 [62],中介绍的连接图表示法，将候选二元连接操作编码为节点（即 QUBO 变量），并用边（即 QUBO 相关性）表示是否存在共享基关系。该图中的每棵生成树都对应一个有效的连接计划，而边和顶点上的权重则表示连接代价和选择率因子。基于此模型，我们设计了分解启发式方法，以尽量减少分区间共享连接键的数量，从而使部分计划能够独立求解，同时保持代价估算的准确性。

对于事务调度，我们根据争用特征进行分区 [85]，：将经常发生争用的事务分组，并将相互独立的事务隔离开来，以便并行求解，同时保证正确性。两种场景面临的一个核心挑战是选择合适的分解粒度。分区过粗可能超出硬件容量，而分区过细则会增加后续组合的复杂度。为此，我们开发了由工作负载统计信息指导的自适应分解策略，例如连接选择率和事务争用程度，以平衡子问题规模与后续组合代价。

2) 嵌入阶段*:* 在此阶段，我们将每个分解后的子问题的变量映射到量子处理单元上，同时保留底层数据库操作的语义局部性。这与通用的次嵌入方法 [73],不同，后者往往会产生又长又脆弱的链。链将多个物理量子比特组合成一个超量子比特，以提供比单量子比特之间的耦合更强的有效连通性。然而，链在实际量子执行中很脆弱，因此不适合且不可靠地表示问题中固有的逻辑关系。对于连接顺序优化，我们嵌入涉及给定基表的所有二元连接操作，以反映它们共享的数据路径。对于事务调度，我们将关键冲突对映射到设备的低噪声区域，以减少链断裂。我们维护一个可复用的嵌入模板库（例如，用于循环连接模式 [48], [57]），这些模板可以适配到不同的硬件架构，例如 D-Wave Pegasus [41]和 Zephyr [42]。

3) 采样阶段： 在此阶段，我们会根据每个分解后子问题的特性，有策略地进行量子处理单元采样，以获得高质量的解。例如，对于连接顺序中涉及大量关系或大型中间结果的子问题，我们会分配更长的退火时间或采用自适应退火调度，使量子处理单元能够更充分地探索解空间。对于复杂度较低或不太关键的子问题，我们则采用较短的运行时间以提高效率。目前，量子硬件通常对所有量子处理单元任务使用静态参数；当子问题复杂度各不相同时，这会导致效率低下 [36], [72]。该策略会根据子问题成本估算和求解器反馈（例如能量方差），调整退火时间、读取次数和偏置调度，从而采用自适应采样策略。通过根据子问题的难度分配资源，并依据反馈更新参数，该采样策略能更充分地利用现有量子硬件，提高性能，同时自适应地处理大型 QUBO。

4) 组合阶段： 在此阶段，我们采用两阶段协调框架，将各个部分解整合为全局一致的结果。第一阶段，我们采用受结构化伊辛模型和域分解相关先前研究启发的、具备领域感知能力的置信传播方法 [69], [78], [80]。这些技术在相互重叠的变量之间传递消息，以协调局部最优解，并推动算法收敛到一致的全局解。例如，在连接顺序优化中，共享连接键的分区会交换共识更新，以统一成本估算。第二阶段，我们采用超大邻域搜索 [86]和禁忌搜索 [84]等后处理启发式方法，迭代改进合并后的解；对于事务调度工作负载尤其如此，因为协调过程必须遵守全局约束，包括锁定顺序和延迟预算。当置信传播效果不佳或计算成本过高时，这些方法可作为回退过程，并且无需完全重新优化即可逐步改进结果。

V. 初步原型：量子增强查询优化器

在我们的初步原型中，我们将迭代相关性松弛方法（第 IV-A 节）与 Post- greSQL 查询优化器 [87] 集成。图 5 展示了迭代式量子增强查询优化器的框架，称为 *Iter-Q²O*，它是将我们提出的方法与数据库系统紧密耦合的初始原型。

在收到用户查询后，*Iter-Q²O* 首先从数据库管理系统引擎中获取基数和选择率统计信息。这些统计信息用于构建 QUBO 公式，其中对目标函数（例如成本函数）和约束（例如连接树结构）进行参数化。预期输出连接顺序序列被编码在 QUBO 中的决策变量内。构建完成后，该 QUBO 被称为初始 QUBO，并提交至量子处理单元（即我们初始原型中的量子退火机）。量子处理单元执行量子

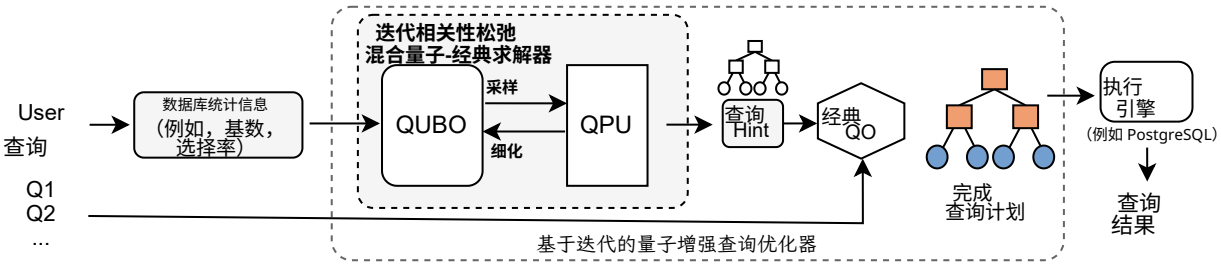


图 5：基于迭代的量子增强查询优化器框架。

采样并返回 n候选解及其对应的能量。随后，根据第 IV-A 节提出的方法，利用这些 n 样本的分布及其能量，将初始 QUBO 精化为精化 QUBO。采样与精化步骤合并视为一次迭代。可用时间预算决定迭代次数，该次数可由用户直接指定，也可由系统自动确定。在完成规定的迭代后，基于迭代的混合量子-经典求解器会返回在时间约束下找到的最佳连接顺序序列。

该序列随后被转换为查询计划提示，用于引导经典查询优化器生成完整的查询计划[1]。在我们的初步原型中，我们使用*pg hint plan* [91] 通过在 SQL 注释中嵌入提示来调整 PostgreSQL 执行计划。连接顺序由以关键字 Leading 开头的子句指定；内层括号表示连接顺序序列中的分组优先级。例如，Bushy 连接提示 Leading((a(b c))(d e)) 表示首先执行 $b \bowtie c$，随后执行 $a \bowtie (b \bowtie c)$，然后执行 $d \bowtie e$，最后将两个中间结果进行连接。PostgreSQL 随后为每个连接选择连接算子（例如，哈希连接、嵌套循环连接）。一旦优化器接受该提示，它就会被转换为完整的查询计划，由执行引擎运行以生成并返回查询结果。

VI. 初步实验

本节评估我们的初步原型（见第五节），该原型将迭代相关性松弛方法（见第 IV-A 节）应用于实时数据库系统中的查询优化问题。我们首先在第 VI-A 节介绍实验设置。随后，我们通过回答以下问题来评估我们的方法：(1) 与经典查询优化器相比，*Iter-Q²O* 在查询优化方面的效果如何（以查询执行延迟衡量）？(2) 在效率和解的质量方面，*Iter-Q²O* 与黑盒量子求解器相比表现如何？(3) 该方法的端到端延迟是多少？在与外部量子服务通信时，它是否仍能保持优势？(4) 量子硬件在核心量子计算过程中的详细计时分解情况如何？

[1]这种方法常用于许多知名的查询优化器（例如 PostgreSQL [88], MySQL [89], Oracle [90]）和学习型查询优化器（LQOs）（例如, [20], [21]）。

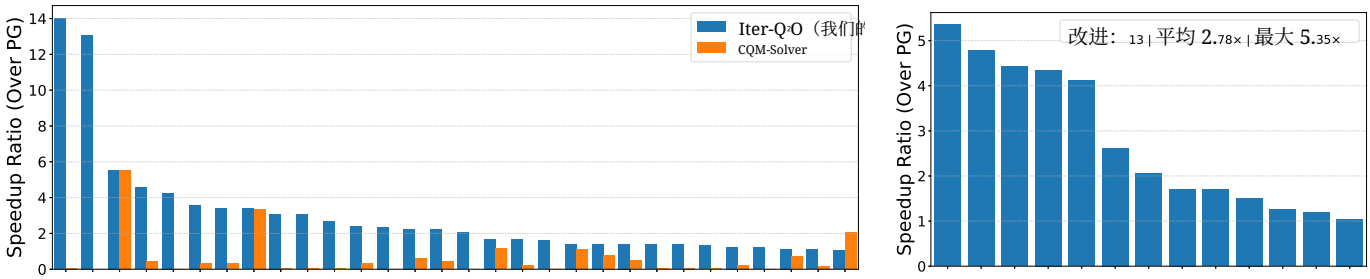
A. 实验设置

我们在一台配备 Intel Core i9-9980XE CPU 和 128 GB 随机存取存储器的服务器上运行所有实验。我们使用 Advantage2 系统1.7 作为D-Wave Leap 量子服务平台上的量子采样器。我们配置了 PostgreSQL 16。4 及相应版本的 *pg hintplan*。我们将 D-Wave 的 CQM-Solver [51]，这一具有代表性的黑盒量子求解器用作黑盒方法的基线。我们使用 连接顺序基准测试（*JOB*） [92] 和 基数估计基准测试（*CEB*） [93]评估我们的方法。JOB 和 CEB 均基于 IMDB 电影数据库模式及其数据构建；与 JOB 相比，CEB 包含更复杂的查询模式，并涵盖范围更广的基数估计挑战。这种复杂性使 JOB 和 CEB 适用于测试连接顺序优化问题，并成为广泛采用的基准测试。

在以下实验中，为了测试第五节介绍的初步基于迭代的量子增强查询优化器，我们将 QUBO 采样与精化的迭代次数设为两次，以启动迭代相关性松弛。在这种情况下，完整的生命周期（在我们的两次迭代设置中）包括：(i) 根据查询相关统计信息生成问题；(ii) 连接量子求解器；(iii) 提交问题并检索结果；(iv) 执行局部 QUBO 精化；以及 (v) 重新提交问题并检索最终结果。

B. 实验结果

查询计划质量。我们使用 JOB 和 CEB 查询在 PostgreSQL 上的实际执行时间来评估查询计划质量。图 6 对比了我们的方法生成的计划与 PostgreSQL 生成的计划的延迟。在 JOB 工作负载（113 个查询）上，我们的方法在 36 个查询上实现了加速，最高加速比为 14.02 倍，平均加速比为 2.70 倍。我们使用 CEB 查询来考察*Iter-Q²O*处理复杂工作负载的能力。在 CEB 工作负载（40 个查询）上，我们的方法在 13 个查询上实现了加速，最高加速比为 5.35 倍，平均加速比为 2.78 倍。对于其余查询，我们的方法仅表现出有限的性能下降。



（a）*Iter-Q²O* 和 CQM-Solver（在 JOB 上）　　(b)迭代-*Q²O*（在 CEB 上）

图 6：我们的方法在迭代 = 2中生成的部分查询计划优于 PostgreSQL 的默认计划。

表 I：我们的方法与 PostgreSQL 之间的端到端延迟分解及比较。

	工作负载	JOB		CEB
	时间（毫秒）	q102	q62	8a9
PG	规划	3172.32	17.33	82.33
	执行	8039.26	5177.80	5992.39
CQM	带提示的规划	2467.99	4.54	43.89
	执行	44968.41	311489.41	512241.75
	量子生命周期	9092.41	7591.69	9002.50
Iter-Q²O （我们的方法）	带提示的规划	2548.51	7.31	65.91
	执行	**573.27**	**395.74**	**1119.09**
	量子生命周期	3168.31	1553.35	1772.95
我方增益（相较于 PostgreSQL）	**绝对时间节省**	**4921.49**	**3238.73**	**2957.95**
	执行时间 ↑	**14.02x**	**13.08x**	**5.35x**
	端到端延迟 ↑	**1.78x**	**2.65x**	**2.05x**

与黑盒量子求解器的比较。为进行公平比较，我们对原型和 D-Wave 的 CQM-Solver 使用相同的初始 QUBO 公式。图 6a 报告了两种方法相对于 PostgreSQL 的加速比。对于大多数查询，我们的方法优于黑盒量子求解器。性能更优的原因在于，我们的迭代求解器专注于重要的相关性，并通过重新引入关键相关性来迭代修正松弛。相比之下，由于 CQM-Solver 以黑盒方式运行，我们无法对其进行问题感知的细粒度控制。

此外，表 I 显示，CQM-Solver 在量子生命周期中的开销要大得多。这是因为 CQM-Solver 在返回结果前会强制设置 5 秒的最短时间限制。相比之下，我们的方法仅依赖底层操作——量子采样，从而绕过了黑盒量子求解器施加的约束。

端到端（E2E）延迟。我们评估端到端延迟，以比较完整的查询执行流水线。对于 PostgreSQL，端到端延迟由规划时间和执行时间组成。对于我们的方法，端到端延迟包括 PostgreSQL 规划时间（使用我们的方法生成的查询提示）、PostgreSQL 执行时间，以及完整量子生命周期所耗费的时间。表 I 详细列出了两个 JOB 查询和一个 CEB 查询的这些组成部分。尽管我们的方法会产生额外的量子开销（主要来自云服务通信），但它生成的连接顺序序列优于 PostgreSQL 的默认设置，因此产生的查询计划能够缩短执行时间并抵消这部分开销。所有报告的结果均基于本地 IMDB 数据集。随着数据规模增大，执行时间的绝对改善预计也会相应扩大，而量子开销不会按同等比例增长。总体而言，我们的方法可将端到端延迟缩短数秒（即 2957.95ms、3238.73ms 和 4921.49ms），实现 1.78 至 2.65 倍的端到端延迟加速。

表 II：量子服务上的时间开销细分（毫秒）。

指标	**迭代 1**	**迭代2**	**总计**	**Avg**
入口（已创建→已接收）	87.896	77.505	165.401	82.701
求解（已接收→已求解）	361.214	246.652	607.866	303.933
出口（已求解→已解决）	122.164	122.176	244.340	122.170
端到端（已创建→已解决）	**571.274**	**446.333**	**1017.607**	**508.803**
QPU 编程	15.760	15.760	31.520	15.760
QPU 采样	26.356	26.356	52.712	26.356
QPU 访问时间（报告值）	42.116	42.116	84.232	42.116
QPU 访问开销	0.822	0.798	1.620	0.810
后处理开销	0.037	0.036	0.073	0.036

量子生命周期的详细计时分解。在表 I 中，JOB 查询 q102 在量子生命周期中耗时 3168.31 毫秒。我们进一步量化了与量子硬件本身直接相关的部分。

表 II 对与量子退火机进行两次交互的耗时进行了详细分解。这两次迭代的端到端时间分别为 571.274 ms 和 446.333ms（平均为 508.803 ms）。端到端时间包括在商业量子云平台上，从创建问题（即 created）到量子硬件将解返回至量子云平台（即 resolved）所消耗的全部时间。约 500 ms 的端到端量子求解时间令人鼓舞。我们还观察到，这两次迭代在量子云平台上消耗的总时间为 1017.607 ms，仅占客户端观测到的 3168.31 ms 的一小部分（32.12%）。其余时间主要耗费在我们的数据库平台与

商业云平台。

在量子硬件本身，QPU 的平均编程时间和采样时间分别为 15.760ms 和 26.356 ms。报告的 QPU 访问时间为 42.116 ms，代表核心量子计算过程所耗费的时间。QPU 访问开销和后处理开销相对较小。总体而言，这些测量结果反映了当前在量子硬件上求解所需的时间及开销。尽管大部分时间耗费在云端通信上，但量子硬件的端到端延迟以及完整量子生命周期的延迟，在解决实际问题乃至实时数据库问题时都切实可行。**挑战与未来改进**。在所提出的迭代框架的每次迭代中，我们根据获得的样本和能量在本地精化二次无约束二元优化，然后提交精化后的二次无约束二元优化。如第六节-B 所示，每次获取和提交都会产生不可忽视的云端通信开销，量子生命周期时间中约有 60% 花在通信上。为降低这一开销，我们提议将细化循环移至服务器端，例如，编译所提出的框架并将其部署在尽可能靠近量子硬件的位置，从而显著提高运行效率，使该方法更适用于解决实际问题和实时问题。

VII. 结论

本文提出了首个实时量子增强型数据库系统，该系统紧密集成了量子-经典混合优化器。该系统能够提供透明的解决方案，而非仅依赖黑盒量子求解器。随后，我们提出了两种应对大规模挑战的可扩展性策略。为解决 QUBO 过度复杂这一挑战，我们引入了迭代相关性松弛框架，在平衡准确性与效率的同时简化 QUBO 结构。为解决二次无约束二元优化规模过大这一挑战，我们提出了问题感知的分解与组合流程，利用数据库语义将 QUBO 划分为子问题，从而能够求解大规模优化问题。初步原型及评估结果表明，与经典数据库组件和黑盒量子求解器相比，该系统的性能均有所提升。未来，我们将致力于设计更高效、更准确的量子增强型系统，用于大规模数据库优化，并使其能够应用于实时场景。

参考文献

[1] Y. E. Ioannidis，《查询优化》，*ACM Comput. Surv.*，第 28 卷，第 1 期，第 121–123 页，1996 年 3 月。 [在线]。可访问：https://doi.org/10.1145/234313.234367[2] J. Zhao、K. Su、Y. Yang、X. Yu、P. Koutris 和 H. Zhang，《揭穿连接顺序的迷思：迈向稳健的 SQL 分析》，*Proc. ACM Manag. Data*，第 3 卷，第 3 期，2025 年 6 月。 [在线]。可访问：https://doi.org/10.1145/3725283。/ R. Marcus、P. Negi、H. Mao、N. Tatbul[3]、M. Alizadeh 和 T. Kraska，《Bao：让学习式查询优化走向实用》，载于 *SIGMOD*，2021 年。[4] X. Yu、G. Li、C. Chai 和 N. Tang，《利用树形长短期记忆网络进行连接顺序选择的强化学习》，载于 *2020 IEEE 第36届国际数据工程会议 (ICDE)*。IEEE，2020 年，第 1297–1308 页。

[5] R. Marcus与O. Papaemmanouil，“用于连接顺序枚举的深度强化学习”，载于《第一届利用人工智能技术进行数据管理国际研讨会论文集》，丛书编号 aiDM’18。美国纽约州纽约：计算机协会，2018年。 [在线]。可获取：https://doi.org/10.1145/3211954.3211957[6] A. Cheng、A. Kabcenell、J. Chan、X. Shi、P. Bailis、N. Crooks与I. Stoica，“迈向最优事务调度”，VLDB 基金会会议录，第17卷，第11期，第2694–2707页，2024年7月。 [在线]。可获取：https://doi.org/10.14778/3681954.3681956。/ S. Chen、C. Shen与C. Wu，“智能事务调度以增强高竞争工作负载中的并发性”，《应用科学》，第15卷，第11期，2025年。 [在线]。可获取：https://www.mdpi.com/2076-3417/15/11/6341[8] T. K. Sellis，“多查询优化”，《ACM数据库系统汇刊》，第13卷，第1期，第23–52页，1988年3月。 [在线]。可获取：https://doi.org/10.1145/42201.42203。/ T. Kathuria与S. Sudarshan，“高效且可证明的多查询优化”，载于《第36届ACM SIGMOD-SIGACT-SIGAI数据库系统原理研讨会论文集》，丛书编号PODS ’17。美国纽约州纽约：计算机协会，2017年，第53–67[9]页。在线。可获取：https://doi.org/10.1145/3034786.3034792。/ S. Agrawal、S. Chaudhuri与V. R. Narasayya，“SQL数据库中物化视图和索引的自动选择”，载于《第26届超大型数据库国际会议论文集》，丛书编号VLDB ’00。美国加利福尼亚州旧金山：摩根·考夫曼出版社，2000年，第496–505页。[11] S. Chaudhuri与V. R. Narasayya，“面向Microsoft SQL Server的高效成本驱动的索引选择工具”，载于《第23届超大型数据库国际会议论文集》，丛书编号VLDB ’97。美国加利福尼亚州旧金山：摩根·考夫曼出版社，1997年，第146–155页。[12] W. Wu、C. Wang、T. Siddiqui、J. Wang、V. Narasayya、S. Chaudhuri与P. A. Bernstein，“基于强化学习的预算感知索引调优”，载于《2022年数据管理国际会议论文集》，丛书编号SIGMOD ’22。美国纽约州纽约：计算机协会，2022年，第1528–1541页。 [在线]。可获取：https://doi.org/10.1145/3514221.3526128。/ D. Shasha与P. Bonnet[13]，《数据库调优：原理、实验与故障排查技术》。美国加利福尼亚州旧金山：摩根·考夫曼出版社，2002年。[14] S. Cluet与G. Moerkotte，“论生成含笛卡尔积的最优左深处理树的复杂性”，载于《数据库理论国际会议》。施普林格，1995年，第54–67页。[15] P. G. Selinger、M. M. Astrahan等，“关系数据库管理系统中的访问路径选择”，载于《ACM SIGMOD数据管理国际会议论文集》（SIGMOD ’79），1979年，第23–34页。 [在线]。可获取：https://doi.org/10.1145/582095.582099。/ T. Ibaraki与T. Kameda[16]，“论计算n元关系连接的最优嵌套顺序”，《ACM数据库系统汇刊》，第9卷，第3期，第482–502页，1984年9月。在线。可获取：https://doi.org/10.1145/1270.1498。/ G. Moerkotte与T. Neumann，“动态规划的回归”，载于《2008年ACM SIGMOD数据管理国际会议论文集》，2008年，第539–552页。[18] T. Neumann与B. Radke，“超大型连接查询的自适应优化”，载于《2018年数据管理国际会议论文集》，2018年，第677–692页。[19] I. Sabek、T. S. Ukyab与T. Kraska，“LSched：面向分析型数据库系统的工作负载感知型学习查询调度器”，载于SIGMOD，2022年。[20] H. Liu、Q. Zhang、R. Marcus与I. Sabek，“Serag：用于查询优化的自演进RAG系统”，2025年。 [在线]。可获取：https://viterbi-web.usc.edu/~sabek/pdf/25 workshop serag.pdf[21] ——，“Sefrqo：自演进的微调型RAG查询优化器”，2025年[22]。在线。可获取：https://arxiv.org/abs/2508.17556。/ R. Marcus、P. Negi、H. Mao、C. Zhang、M. Alizadeh、T. Kraska、O. Papaemmanouil与N. Tatbul，“Neo：学习型查询优化器”，VLDB基金会会议录，第12卷，第11期，第1705–1718页，2019年7月。 [在线]。可获取：https://doi.org/10.14778/3342263.3342644。/ “TechRepublic：D-Wave发布量子比特第五代量子退火机[23]，5,000 https://www.dwavequantum.com/company/newsroom/media-coverage/techrepublic-d-wave-announces-5-000-qubit-fifth-generation-quantum-annealer/。

[24] IBM Quantum，“由IBM提供的量子计算机云访问”，https://quantum-computing.ibm.com，2022年。[25] Rigetti Computing，“Rigetti量子处理器”，https://qcs.rigetti.com/ qpus，2022年。[26] S. Arora与B. Barak，《计算复杂性：现代方法》，第1版。美国：剑桥大学出版社，2009年。[27] E. Bernstein与U. Vazirani，“量子复杂性理论”，《SIAM计算期刊》，第26卷，第5期，第1411–1473页，1997年。[28] L. K. Grover，“一种用于数据库搜索的快速量子力学算法”，载于《第28届ACM计算理论年度研讨会论文集》，丛书编号STOC ’96。美国纽约州纽约：计算机协会，1996年，第212–219页。 [在线]。可获取：https://doi.org/10.1145/237814.237866[29] Google，“谷歌称取得突破：量子计算机超越超级计算机的能力”，https://www.theguardian.com/technology/2025/oct/22/google-hails-breakthrough-as-quantum-computer-surpasses-ability-of-supercomputers，2025年。[30] A. D. King、A. Nocera、M. M. Rams等，“量子模拟中的超越经典计算”，《科学》，第388卷，第6743期，第199–204页，2025年。 [在线]。可获取：https://www.science.org/doi/abs/ 10.1126/science.ado6285[31] Z. Liu，“用于提升引用的实用量子优势”，2025年3月。[32] D. Boneh与R. J. Lipton，“隐线性函数的量子密码分析”，载于《密码学进展——CRYPT0’ 95》，D. Coppersmith编。柏林、海德堡：施普林格柏林海德堡，1995年，第424–437页。[33] Y. Zhao，“密码分析中量子计算的演示与实现”，第38卷，第431–436页，2023年3月。 [在线]。可获取：https://drpress.org/ojs/index.php/HSET/article/view/ 3855
[34] A. Awasthi、N. Kraus等，“量子-经典优化在生产调度中的实际应用”，载于《2024年IEEE量子计算与工程国际会议》（QCE），第02卷，2024年，第239–244页。[35] G. Colucci等，“电力网络优化：量子方法”，IEEE Access，第11卷，第98 926–98 938页，2023年。[36] D-Wave Leap的混合求解器，https://docs.dwavesys.com/docs/latest/ doc leap hybrid.html，2024年。[37] Google，“谷歌：认识Willow，我们最先进的量子芯片”，https://blog.google/technology/research/google-willow-quantum-chip/#:~:text=General%20summary,medicine%2C%20energy%2C%20and% 20AI，2024年。[38] Microsoft，“微软的Majorana 1芯片为量子计算开辟新路径”，https://news.microsoft.com/source/features/innovation/microsofts-majorana-1-chip-carves-new-path-for-quantum-computing/，2025年。[39] J. Preskill，“含噪声中等规模量子时代及以后的量子计算”，《量子》，第2卷，第79页，2018年8月。 [在线]。可获取：https: //doi.org/ 10 22331 2018 08 06 79 . /q- - - - M. Razavy，《量子隧穿理论》。[40] 世界科技出版社，2013年。[41] D.-W.文档，https://docs.dwavesys.com/docs/latest/c gs 4. html，2024年。[42] K. Boothby、A. D. King与J. Raymond，“D-Wave量子处理器的Zephyr拓扑”，载于《D-Wave 技术报告》，2021年，第1–18页。[43] R. Hai、S.-H. Hung、T. Coopmans、T. Littau与F. Geerts，“含噪声中等规模量子时代的量子数据管理”，载于《超大型数据库国际会议论文集》，VLDB（即将发表），2026年。[44] T. Bittner与S. Groppe，“利用量子退火调度事务以避免阻塞”，载于《第24届国际数据库工程与应用研讨会论文集》，丛书编号IDEAS ’20。美国纽约州纽约：计算机协会，2020年。
[45] [ ] 10 1145 3410566 3410593 在线。可获取：https://doi.org/ . / . S. Roy、L. Kot与C. Koch，“量子数据库”，载于《第六届创新数据系统研究双年会》，CIDR 2013，美国加利福尼亚州阿西洛马，2013年1月6日至9日，在线论文集。www.cidrdb.org，2013年。 [在线]。可获取：http://cidrdb.org/cidr2013/ Papers/CIDR13\ Paper86.pdf [46] I. Trummer与C. Koch，“D-Wave 2X绝热量子计算机上的多查询优化”，载于VLDB，2016年。[47] M. Sch¨onberger、I. Trummer与W. Mauerer，“一般连接树的量子优化”，载于VLDB研讨会，2023年。[48] M. Sch¨onberger、S. Scherzinger等，“准备好飞跃（通过协同设计）？ 量子硬件上的连接顺序优化”，Proc. ACM Manag. Data，第1卷，第1期，2023年5月。 [在线]。可获取：https://doi.org/10.1145/3588946
[49] H. Liu 和 I. Sabek，“量子计算是否已准备好用于实时数据库优化？”2026年。 [在线]。获取地址：https://arxiv.org/abs/ 2601.12123[50] S. Endo、Z. Cai、S. C. Benjamin 和 X. Yuan，“混合量子-经典算法与量子误差缓解”，《日本物理学会誌》，第90卷，第3期，032001页，2021年。[51] 约束二次模型混合求解器 [白皮书], https://www.dwavesys.com/media/rldh2ghw/14-1055a-a hybrid solver for constrained quadratic models.pdf，2021年。[52] H. Liu、F. Spedalieri 和 I. Sabek，“q2o演示：量子增强型查询优化器”，载于超大型数据库国际会议论文集，VLDB，2025年。[53] H. Liu、P. Saxena、F. Spedalieri 和 I. Sabek，“通过约束二次模型型优化连接顺序（摘要）”，载于第1届量子数据与机器学习研讨会论文集，丛书 QDML ’25，2025年。[54] H. Liu、A. Kumar、F. Spedalieri 和 I. Sabek，“灌木状连接树的混合量子-经典优化”，载于第2届面向数据密集型系统与应用的量子计算及量子启发技术研讨会论文集，丛书 Q-Data ’25。美国纽约州纽约市：计算机协会，2025年，第
20 24 [ ] 10 1145 3736393 3736695 –页。在线。获取地址：https://doi.org/ . / . E. Osaba 和 P. Miranda-Rodriguez，“D-Wave非线性规划混合求解器：描述与性能分析”，IEEE Access，第13卷，第4724–4736页，2025年。[56] M. Aramon、G. Rosenberg 等，“使用数字退火器对无约束二次问题进行物理启发式优化”，《物理学前沿》，第7卷，48页，2019年。[57] M. Sch¨onberger、I. Trummer 和 W. Mauerer，“用于连接顺序的量子启发式数字退火”，载于 VLDB，2023年。[58] N. Nayak、J. Rehfeld 等，“在量子硬件和模拟器上求解 QUBO 问题以构造最优灌木状连接树”，载于新兴分布式环境中的大数据国际研讨会论文集，丛书 BiDEDE ’23，2023年。 [在线]。获取地址：https://doi.org/ 10 1145 3579142 3594298 。 / . S. Groppe 和 J. Groppe，“通过为 Grover搜索算法生成代码，在通用量子计算机上优化事务调度”，载于第25届国际数据库工程与应用研讨会论文集，丛书 IDEAS ’21。美国纽约州纽约市：计算机协会，2021年，第149–156页。
[59] [ ] 10 1145 3472163 3472164 在线。获取地址：https://doi.org/ . / . N. Nayak、A. Prisacaru、U. C¸alıkyılmaz、J. Groppe 和 S. Groppe，“通过使用锁机制迭代求解 QUBO，以降低复杂度实现量子增强型事务调度”，载于第2届面向数据密集型系统与应用的量子计算及量子启发技术研讨会论文集，丛书 Q-Data ’25。美国纽约州纽约市：计算机协会，2025年，第
26 35 [ ] 10 1145 3736393 3736701 –页。在线。获取地址：https://doi.org/ . / . T. Fankhauser、M. E. Soler、R. M. Fuchslin 和 K. Stockinger，“采用经典与量子计算混合方法的多查询优化”，arXiv预印本 arXiv:2107.10508，2021年。[62] C. Lee、C.-S. Shih 和 Y.-H. Chen，“采用基于图的方法优化大型连接查询”，《IEEE 知识与数据工程汇刊》，第13卷，第2期，第298–315页，2001年。[63] T. Kadowaki 和 H. Nishimori，“横场伊辛模型中的量子退火”，Phys. Rev. E，第58卷，第5355–5363页，1998年11月。
[64] [ ] 10 1103 58 5355 在线。获取地址：https://link.aps.org/doi/ . /PhysRevE. . P. J. Van Laarhoven 和 E. H. Aarts，“模拟退火”，载于《模拟退火：理论与应用》。施普林格，1987年，第7–15页。[65] A. Lucas，“许多 NP问题的伊辛形式化”，《物理学前沿》，第2卷，2014年。 [在线]。获取地址：https://www.frontiersin.org/ articles/ 10 3389 2014 00005 。 /fphy. . P. Saxena、I. Sabek 和 F. Spedalieri，“用于优化连接顺序的约束二次模型”，载于第1届量子计算及量子启发式数据密集型系统与应用技术研讨会论文集，丛书 Q-Data ’
[66] 24, 2024, 38 44 [ ] 2 第–页。在线。获取地址：https://doi-org.libproxy .usc.edu/ 10.1145/3665225.3665447[67] H. Karimi 和 G. Rosenberg，“通过样本持久化提升量子退火机性能”，《量子信息处理》，第16卷，第7期，1–24页，2017年7月。 [在线]。获取地址：https://doi.org/10.1007/s11128- 017-1615-x[68]“dimod文档”，https://docs.ocean.dwavesys.com/en/stable/ docsdimod/.[69] Z. Bian、F. Chudak、R. Israel、B. Lackey、W. G. Macready 和 A. Roy，“使用稀疏伊辛模型进行离散优化的量子退火

模型”，《物理学前沿》，第2卷，2014年9月。 [在线]。获取地址：https://doi.org/ 10 3389 2014 00056 。 /fphy. . I. [70] Trummer 和 D. Venturelli，“利用量子计算进行数据库索引选择”，载于第1届面向数据密集型系统与应用的量子计算及量子启发技术研讨会论文集，丛书 Q-Data '24。美国纽约州纽约市：计算机协会，2024年，第14–26页。 [在线]。获取地址：https://doi.org/ 10 1145 3665225 3665445 。 / . “ [71] qiskit-optimization文档”，https://qiskit.org/ecosystem/optimization/index.html。[72] F. Limited，“富士通数字退火器——即时求解大规模组合优化问题”，2023年，https://www.fujitsu.com/global/services/business-services/digital-annealer/what-is-digital-annealer/.[73] D-Wave Advantage 系统 [白皮书], https://www.dwavesys. _ com/media/3xvdipcn/14-1058a-a advantage processor overview.pdf, 2022年。[74] M. Sch¨onberger、I. Trummer 和 W. Mauerer，“采用增量式量子（启发式）退火的大规模多查询优化”，载于 ACM 数据管理国际会议论文集，SIGMOD（即将发表），2026年。[75] I. Haffner 和 J. Dittrich，“通过启发式搜索高效计算连接顺序”，Proc. ACM Manag. Data，第1卷，第1期，2023年5月。 [ ] 10 1145 3588927 在线。获取地址：https://doi.org/ . / T. Zhang、A. Tomasic 和 A. Pavlo，“通过冲突预测实现 OLTP 数据库管理系统中的智能事务调度”，2024年。 [在线]。获取地址：https://arxiv.org/abs/ 2409 01675 。V. Sharma 和 C. [77] Dyreson，“Indexer++：结合 Transformer 与强化学习的工作负载感知在线索引调优”，载于第37届 ACM/SIGAPP 应用计算研讨会论文集，丛书 SAC '22。美国纽约州纽约市：计算机协会，2022年，第 372 380 [ ] 10 1145 3477314 3507691 –页。在线。获取地址：https://doi.org/ . / . B. Lackey，“一种基于域分解的置信传播算法”，2018年。 [在线]。获取地址：https://arxiv.org/abs/1810. 10005[79] S. Kullback 和 R. A. Leibler，“论信息与充分性”，《数学统计年刊》，第22卷，第1期，第79–86页，1951年。[80] Z. Bian、F. A. Chudak、R. B. Israel、B. Lackey、W. G. Macready 和 A. Roy，“将约束优化问题映射到量子退火并应用于故障诊断”，Frontiers ICT，第3卷，14页，2016年。 [在线]。获取地址：https://api.semanticscholar.org/ CorpusID:70[81] J. Lin，“基于香农熵的散度度量”，《IEEE 信息论汇刊》，第37卷，第1期，第145–151页，1991年。[82] T. Noguchi、K. Fukada、S. Bao 和 N. Togawa，“针对具有独热约束的组合优化问题，基于高效二进制变量选择的子 QUBO 退火”，载于2025年 IEEE 消费电子国际会议（ICCE），2025年，第1–2页。[83] W. Zhao 和 G. Tang，“面向混合 QUBO 求解器的基于聚类的子 QUBO 提取”，2025年。 [在线]。获取地址：https://arxiv.org/abs/ 2502.16212[84] M. Booth、S. P. Reinhardt 和 A. Roy，“面向经典-量子混合执行的优化问题划分”，载于 D-Wave 技术报告，2017年，第1–9页。[85] E. Zamanian、J. Shun、C. Binnig 和 T. Kraska，“Chiller：面向争用的事务执行与现代网络的数据分区”，载于2020 年 ACM SIGMOD 数据管理国际会议论文集，丛书 SIGMOD '20。美国纽约州纽约市：计算机协会，2020年，第511–526页。 [在线]。获取地址：https://doi.org/ 10 1145 3318464 3389724 。 / . D. Pisinger 和 S. [86] Ropke，《大邻域搜索》。瑞士尚贝：Springer 国际出版公司，2019年，第99–127页。 [在线]。获取地址：https://doi.org/ 10 1007 978 3 319 91086 4 4 。 / - - - - PostgreSQL, [87] https://www.postgresql.org/.[88] PostgreSQL 全球开发组, https://www.postgresql.org/.[89] MySQL 文档团队，《MySQL 8.4 参考手册》，第8版，2025年。 [在线]。获取地址：https://dev.mysql.com/doc/en/ [90] 甲骨文公司，《Oracle 数据库参考手册》，19c，第19版，2025年4月。 [在线]。获取地址：https://docs.oracle.com/en/database/oracle/oracle-database/19/refrn/index.html[91] OSSC-DB，“pg hint plan”, https://github.com/ossc-db/pg hint plan. [92] V. Leis、A. Gubichev、A. Mirchev、P. Boncz、A. Kemper 和 T. Neumann，“查询优化器的实际表现究竟如何？”，VLDB基金会会议录，第9卷，第3期，第204–215页，–2015年11月。 [在线]。获取地址：https://doi.org/10.14778/2850583.2850594 [93] P. Negi、R. Marcus、A. Kipf、H. Mao、N. Tatbul、T. Kraska 和 M. Alizadeh，“Flow-loss：学习具有实际意义的基数估计”，*VLDB*基金会会议录，第14卷，第11期，第2019–2032页，2021年7月。 [在线]。可访问：https://doi.org/10.14778/3476249.3476259