具有访问模式的合取查询的时空权衡

Hangdong Zhao 威斯康星大学-麦迪逊 麦迪逊，威斯康星州，美国 hangdong@cs.wisc.edu

Shaleen Deep 微软 GraySystems 实验室 麦迪逊 威斯康星州 美国 shaleen.deep@microsoft.com

Paraschos Koutris 威斯康星大学-麦迪逊 麦迪逊，威斯康星州，美国 paris@cs.wisc.edu

arXiv:2304.06221v2 [cs.DB] 2 May 2023

摘要

本文研究如何为带访问模式的合取查询（CQAPs）回答问题时实现时空权衡。目标是在初始预处理阶段创建一种节省空间的数据结构，并在在线阶段利用它回答（多个）查询。以往的研究针对路径查询和三角形查询等具有实际意义的查询，开发了以空间使用量换取回答时间的数据结构。然而，这些方法缺乏全面的框架，也无法推广。我们的主要贡献是提出一个通用算法框架，用于为任意 CQAP 实现时空权衡。该框架建立在 PANDA 算法和树分解技术之上。我们证明，该框架涵盖了针对各种查询而独立提出的所有最先进的权衡方案。此外，我们还展示了相较于现有文献中已知的可达性查询最先进权衡方案所取得的出人意料的改进。

1 引言

我们研究一类将算法任务划分为两个阶段的问题：预处理阶段从输入计算出一种节省空间的数据结构；在线阶段则利用该数据结构，尽可能快速地回答输入中某种特定形式的请求。许多重要的算法任务都可以用这种方式表示，例如集合交集问题 [8, 15], 有向图中的可达性 [2, 3, 9], 直方图索引 [7, 25], 以及与文档检索相关的问题 [1, 26] 。与这些问题相关的基本算法问题是，如何确定空间之间的权衡 S （即存储数据结构所需的空间）与时间 T （即回答请求所需的时间）。

我们来看看这一设定中最简单的任务之一。考虑二集合不相交问题：给定一个由元素组成的全集 U和一组 m集合 $S_1, \ldots\ldots, S_m \subseteq U$，我们希望构建一种数据结构，使得对于任意一对整数 $1 \le i, \ j \le m$，都能高效判断 $S_i \cap S_j$ 是否为空。众所周知，二集合不相交问题的时空权衡由方程 $S \cdot T^2 = O(N^2)$刻画，其中 N 是所有集合的总大小。其他数据结构问题也有类似的权衡。

允许免费制作本作品全部或部分内容的数字版或纸质副本，供个人或课堂使用，但前提是不得为营利或商业利益制作或分发副本，且副本须在首页注明本声明及完整引文。作者以外的其他人拥有的本作品组成部分的版权必须得到尊重。注明出处后可以摘录。若要以其他方式复制、重新出版、发布到服务器或分发至列表，则须事先获得具体许可和／或支付费用。请向 permissions@acm.org 申请许可。*PODS '23，2023年6月18日至23日*，美国华盛顿州西雅图© 2023年版权归所有者／作者所有。出版权已授权给 ACM。ACM ISBN 979-8-4007-0127-6/23/06…$15.00
https://doi.org/10.1145/3584372.3588675

在 k-可达性问题 [8, 15] 中，输入为一个有向图 $G = (V, \ E)$和任意一对顶点 $u, \ v$，目标是判断 u和 v之间是否存在长度为 k 的路径。 [15], 猜想所得到的数据结构是最优的，并利用这一最优性猜想为其他问题建立了条件下界，例如近似距离预言机 [2, 3] ；在过去十年中，这类问题的上界一直未能得到改进。在边三角形检测问题 [15], 中，输入为一个图 $G = (V, \ E)$，目标是开发一种数据结构，能够判断给定的边 $e \in E$ 是否属于某个三角形。这些问题都是分别独立研究的，因此相关算法见解难以推广为一个全面的框架。

本文将上述许多问题表述为关系数据库上的带访问模式的合取查询（*CQAPs*）。例如，使用关系 $R(x, \ y)$ 对"元素 x 属于集合 y"进行编码，就可以用以下 CQAP 表示二集合不相交性：$\varphi(\mid y_1, \ y_2) \leftarrow R(x, \ y_1) \wedge R(x, \ y_2)$。表达式 $\varphi(\mid y_1, \ y_2)$ 可以解释如下：给定 $y_1, \ y_2$ 的值，计算该查询是否返回真。不同的访问模式对应着不同的 CQ 结果访问方式，并带来不同的权衡。

在静态和动态设置下枚举合取查询结果的权衡一直是以往研究的主题 [13, 17, 19–21, 30]。然而，以往工作要么关注预处理时间与回答时间之间的权衡 [19–21], ，要么关注枚举中的空间与延迟之间的权衡 [13, 30]。本文明确关注空间与回答时间之间的权衡，而不优化预处理时间。与我们的设置最密切相关的是布尔合取查询回答问题 [12]。在该工作中，作者对 [13] 提出的数据结构稍作改进，并将其用于布尔合取查询回答。此外， [12] 指出，关于 k-可达性问题的猜想权衡并非最优，并展示了对所有 $k \ge 3$ 略有改进的权衡。本文采用的技术截然不同，并且是 [12] 中所用技术的大幅推广。 [12] 对 k-可达性提出的改进已被本文涵盖，而且在许多情况下优于 [12] 中的改进。

我们的贡献。我们的主要贡献是提出一种通用的算法框架，用于为任意CQAP获得时空权衡。我们的框架建立在 PANDA算法 [24] 和树分解技术 [16, 28]之上。给定任意CQAP，它会计算一种权衡，从而在给定空间预算的情况下找到可能的最佳时间。为实现这一目标，我们需要两项关键的技术贡献。

首先，我们引入新颖的部分物化树分解（PMTD），它使我们能够捕捉给定树分解上不同的可能物化策略（第3节）。从高层次看，PMTD通过添加以下信息来扩展树分解：哪些袋应当被物化，以及哪些袋应当在线计算。为了使用PMTD，我们提出了Yannakakis算法的一个变体（第3.1节），使得在在线阶段我们只需承担访问非物化袋的成本。

第二个关键要素是对 PANDA 算法的扩展，该扩展 [24] 分两个阶段计算析取规则。计算析取规则允许将任意目标的答案放置到规则头中。PANDA算法中的一个关键技术组件是香农流不等式 的概念。对于任意香农流不等式，都可以构造一个与关系运算符直接对应的证明序列。因此，可以将证明序列转换为连接算法。我们考虑的析取规则分两个阶段进行计算：在第一阶段（预处理）中，我们只能将答案放置到将在预处理阶段进行物化的目标上。在第二阶段（在线）中，我们将答案放置到其余目标上。我们将这些规则称为两阶段析取规则 （第 4.1 小节）。为了实现这种两阶段计算，我们引入了一类称为联合香农流不等式的香农流不等式（第 5 节），使得每个不等式都能产生一种时空权衡。联合香农流不等式生成两个并行的证明序列：一个证明序列用于预处理阶段，另一个证明序列用于答案生成阶段。这种转换使我们能够将PANDA 算法作为黑盒分别应用于每个证明序列，并且对于实现时空权衡发挥了关键作用。

我们通过为布尔合取查询求值问题、二集合不相交性及其推广形式 k-集合不相交性，以及 k-可达性（第 6 节）重新推导出最先进的时空权衡，展示了我们框架的通用性。我们还将该框架应用于此前尚未研究的静态场景：针对一部分层次查询的访问模式的时空权衡。这类查询是无环合取查询的一个片段，备受关注 [5, 6, 10, 11, 18, 20]。有趣的是，我们能够重新推导出与该类合取查询中具有可证明保证的专用枚举算法所采用策略非常相似的策略 [11, 20]。更重要的是，我们改进了最先进的权衡。我们最有意思的发现是，对于 k-可达性，我们可以得到复杂的权衡，并且在 S 的不同范围内呈现不同的行为。对于三可达性问题，我们展示了如何改进其在相当大一部分范围内的权衡。对于四可达性问题，我们能够（通过相当复杂的分析）证明，与猜想中的最优权衡相比，时空权衡在所有范围内都可以改进！对于 k-可达性，这些结果推翻了 [12] 中提出的最优权衡 $S \cdot T^{2/(k-1)} = \mathrm{e}O(|E|^2)$，而且所涉及的范围甚至比该文所展示的更大。

组织结构。 我们在第2 节介绍基本术语和问题定义。在第3 节中，我们描述框架所需的增强型树分解。第 4 节介绍一般框架，而第 5 节则介绍框架中使用的算法。我们介绍第 6 节介绍框架的应用。第 7 节描述相关工作，最后在第 8 节列出尚待解决的问题。

2 背景

合取查询。 我们将合取查询（CQ） φ与超图 $\mathrm{H} = ([n], \mathrm{E})$ 相关联，其中 $[n] = \{1, \ldots, n\}$，且 $\mathrm{E} \subseteq [n]$。查询体包含原子 R_F，其中 $F \in \mathrm{E}$。对于每个节点 $i \in [n]$，我们关联一个变量 x_i。于是，CQ 为

$$\varphi(\mathbf{x}_H) \leftarrow \bigwedge_{F \in \mathcal{E}} R_F(\mathbf{x}_F),$$

其中 $\mathbf{x}_I$表示任意 $I \subseteq [n]$对应的元组 $(x_i)_{i \in I}$。$\mathbf{x}_H$中的变量称为 CQ 的头部变量。若 $H = [n]$，则该 CQ 是全的；若 $H = \emptyset$，则它是布尔的。我们用 φ表示 CQ φ的输出。

度约束。度约束是一个三元组 $(X, Y, N_{Y|X})$，其中 $X \subset Y \subseteq [n]$，且 $N_{Y|X}$是自然数。若 $X \subset Y \subseteq F$，并且对于每个元组 t_X（在 X上），$\max_{\mathrm{t}_X} \deg_F(Y|\mathrm{t}_X) \le N_{Y|X}$，其中 $\deg_F(Y|\mathrm{t}_X) = \left|\Pi_Y(\sigma_{X=\mathrm{t}_X} R_F)\right|$，则称关系 R_F守卫度约束 $(X, Y, N_{Y|X})$。我们用 DC 表示一组度约束；如果每个 $(X, Y, N_{Y|X}) \in$ DC 都由 D 中的某个关系守卫，则称数据库实例 D 守卫 DC。若 $X = \emptyset$，则度约束 $(X, Y, N_{Y|X})$ 是基数约束。在本文中，我们对由数据库实例 D 守卫的 DC 作如下假设：

- （最佳约束假设）</style}不失一般性，对于任意 ，至多存在一个 $(, Y, N_{Y|X}) \in$DC。若存在多个，只保留最小的 $N_{Y|X}$即可维持这一假设。

  对于每个关系 $R_F \in \mathrm{D}$，都存在一个基数约束$(\emptyset, F, |R_F| \mathrm{def}= N_{F|\emptyset}) \in$DC。数据库 D 的大小记作 $|\mathrm{D}| \mathrm{def}= \max_{R_F \in \mathcal{D}} |R_F|$。

在本文中，我们使用度约束来衡量数据复杂度。除非另有说明，所有对数均以 2 为底。

2.1 带访问模式的合取查询

我们按照 [21] 中的定义来定义带访问模式的合取查询：

定义 *2.1*（带访问模式的合取查询）。带访问模式的合取查询（CQAP）是一种如下形式的表达式

$$\varphi(\mathbf{x}_H \mid \mathbf{x}_A) \leftarrow \bigwedge_{F \in \mathcal{E}} R_F(\mathbf{x}_F),$$

其中， $A \subseteq [n]$ 称为该查询的访问模式。

访问模式说明用户如何获取合取查询的结果。具体来说，用户会提供一个关系 $Q_A(\mathbf{x}_A)$ 的实例，我们称之为访问请求。接下来，任务是返回以下合取查询的结果，记为 φ，其中

$$\varphi(\mathbf{x}_H) \leftarrow Q_A(\mathbf{x}_A) \wedge \bigwedge_{F \in \mathcal{E}} R_F(\mathbf{x}_F).$$

我们称之为 φ 访问合取查询。最自然的访问请求是满足 $|Q_A| = 1$的请求；换言之，用户为每个变量 x_i， $i \in A$ 只提供一个固定值。这可以看作是将合取查询的结果用作索引，搜索键为 $\mathbf{x}_A$。通过允许

访问请求 Q_A 包含更多元组，我们便能描述其他场景。例如，可以将一系列大小为 1 的访问请求批量处理，从而一次性获得所有请求的答案（可能更快）。先前工作 [13, 21] 只考虑了 $|Q_A| = 1$ 的情况。

2.2 问题陈述

设 $\varphi(\mathbf{x}_H \mid \mathbf{x}_A)$ 是一个受输入关系守卫的、满足度约束 DC 的 CQAP。此外，我们用 AC 表示另一组预先已知的度约束，这些约束由任意访问请求 Q_A 守卫。与 DC 类似，我们假设存在一个基数约束 $(\emptyset, A, |Q_A| = N_{A|\emptyset}) \in \mathsf{AC}$，由 Q_A 守卫。例如，$|Q_A| = 1$ 的情况可解释为一个基数约束 $(\emptyset, A, 1) \in \mathsf{AC}$。给定一个守卫 DC 的数据库实例 D，我们的目标是构建一种数据结构，以便尽可能快速地响应任意访问请求。更正式地说，我们将查询处理分为两个阶段：

预处理阶段：该阶段构建一个空间开销为 $\mathrm{e}O(S)$[1] 的数据结构。总空间开销的形式为 $\mathrm{e}O(S + |\mathsf{D}|)$，其中 S 称为该数据结构的内在空间开销，而 $|\mathsf{D}|$ 是存储数据库所需的（不可避免的）空间开销。

在线阶段：给定访问请求 Q_A（守卫 AC），它使用预处理阶段构建的数据结构，返回访问合取查询 φ 的结果。此时，（最坏情况下的）回答时间为 $O(T + |Q_A|) + O(|\varphi|)$，其中 T 称为内在时间开销，而 $|Q_A| + |\varphi|$ 是读取访问请求 Q_A 并枚举输出所需的（不可避免的）时间开销。对于布尔情形，且当 $|Q_A| = 1$时，回答时间就变为 $\mathrm{e}O(T)$。

在本文中，我们研究这两个内在量 S和 T之间的权衡，并将其称为内在权衡。在一个极端情况下，算法不存储任何内容，因此 $S = O(1)$，每个访问请求都从头开始回答。在另一个极端情况下，算法将合取查询 $\varphi_M(\mathbf{x}_{H\cup A}) \leftarrow \acute{\mathrm{O}}_{F\in\mathcal{E}} R_F(\mathbf{x}_F)$ 的结果存储为哈希表，索引键为$\mathbf{x}_A$。对于任意访问请求Q_A，我们只需在在线阶段通过查询哈希表，逐个探查 Q_A中的元组来计算查询 $\varphi(\mathbf{x}_H) \leftarrow Q_A \wedge \varphi_M$。若 $H \supseteq A$，则任意访问请求都能在（实例最优）时间 $O(|Q_A| + |\varphi|)$ 内得到回答，此时 $T = O(1)$。

示例 *2.2*（k-集合不相交性）。在这个问题中，给定集合 S_1、……、S_m，其元素取自同一个全集 U。每个访问请求都询问 k 个集合的交集是否为空。将这些集合构成的族编码为二元关系$R(y,\ x)$，使得元素 y 属于集合 x，我们就可以将该问题表示为以下 CQAP：

$$\varphi(\mid \mathbf{x}_{[k]}) \leftarrow \bigwedge_{i\in[k]} R(y, x_i). \tag{1}$$

如果我们还想枚举它们交集中的元素，则应改用非布尔版本：

$$\varphi(y \mid \mathbf{x}_{[k]}) \leftarrow \bigwedge_{i\in[k]} R(y, x_i). \tag{2}$$

[1]符号~O 隐藏了关于 $|\mathcal{D}|$ 的一个多对数因子。

示例 *2.3*（k-可达性）。给定一个有向图 G，k-可达性问题要求给定一对顶点（$u,\ v$），判断它们是否通过一条长度为 k 的路径相连。将该图表示为二元关系 $R(x,\ y)$，我们可以通过以下 CQAP（k-路径查询）来建模这个问题：

$$\phi_k(\mid x_1, x_{k+1}) \leftarrow \bigwedge_{i\in[k]} R(x_i, x_{i+1}).$$

我们还可以通过合并 k 个此类查询（每个 1、……、k对应一个查询）的结果，检查是否存在长度至多为 k的路径。

在本文中，我们关注满足 $H \supseteq A$的 CQAP。若给定的 CQAP 满足 $H \not\supseteq A$，我们就将其头部替换为 $\varphi(\mathbf{x}_{H\cup A} \mid \mathbf{x}_A)$，并只需在最后投影出所需结果。

3 部分物化的树分解

在本节中，我们介绍一种树分解类型，它通过补充有关我们想要物化哪些包的信息来增强分解。

定义 *3.1*（树分解）。超图 $\mathrm{H} = ([n], \mathrm{E})$ 的树分解是一个二元组 $(\mathrm{T},\ \chi)$，其中 (i) T 是一棵无向树，并且 (ii) $\chi\colon V(\mathrm{T}) \to 2^{[n]}$ 是一个映射，它为 $t \in V(\mathrm{T})$ 中的每个节点分配 $[n]$ 的一个子集，称为 t 的包，满足

(1) 对于每条超边 $F \in \mathrm{E}$，集合 F 包含于某个包中；以及
(2) 对于每个顶点 $x \in [n]$，节点集合 $\{t \mid x \in \chi(t)\}$ 在 T 中构成一个（非空）连通子树。

取一个树分解 $(\mathrm{T},\ \chi)$ 和一个节点 $r \in V(\mathrm{T})$。如果以 r 为根，我们将 $\mathrm{TOP}_r(x)$ 定义为 T 中包含 x 于其包中的最高节点。现在，如果对于任意 $x \in H$ 和 $y \in [n] \setminus H$，$\mathrm{TOP}_r(y)$ 都不是 $\mathrm{TOP}_r(x)$ 的祖先 [34]，则称 $(\mathrm{T},\ \chi)$ 是 关于 r 自由连通的。若 $(\mathrm{T},\ \chi)$ 关于某个 $r \in V(\mathrm{T})$ 自由连通，则称其为自由连通。

现在，我们可以引入一个针对带访问模式的合取查询的核心概念：部分物化树分解。设 $\varphi(\mathbf{x}_H \mid \mathbf{x}_A)$ 是一个带访问模式的合取查询，且 $A \subseteq H$。令 H 为与 $\varphi(\mathbf{x}_H)$（即访问合取查询）相关联的超图。

定义 *3.2*（*PMTD*）。合取查询 $\varphi(\mathbf{x}_H \mid \mathbf{x}_A)$ 在 $H \supseteq A$ 上的部分物化树分解$(\mathcal{T},\ \chi, M, r)$是满足以下性质的一个三元组：

(1) $(\mathrm{T},\ \chi)$ 是 H 关于节点r的自由连通树分解，该节点称为根；(2) $A \subseteq \chi(r)$；以及 (3) $M \subseteq V(\mathrm{T})$，且每当 $t \in M$ 时，其子树中的所有节点（相对于以 r 为起点、背离该点的树定向）都属于 M。

给定一个 PMTD $(\mathrm{T},\ \chi, M, r)$，我们称 M为物化集。我们还为每个节点 $t \in V(\mathrm{T})$ 关联一个变量为 $\mathbf{x}_v(t)$ 的视图，其中映射 v：$V(\mathrm{T}) \to 2^{[n]}$ 定义如下。若该节点为 $t \notin M$，则 $v(t)\ \mathrm{def}=\ \chi(t)$，且该视图形式为 $T_v(t)\,(\mathbf{x}_v(t))$，称为 T-视图。否则，$t \in M$。若 $t = r \in M$，则定义$v(r)\ \mathrm{def}=\ \chi(t) \cap H$。令 p为非根节点 $t \in M$ 的父节点

并定义

$$\nu(t) \stackrel{\text{def}}{=} \begin{cases} \chi(t) \cap (H \cup \chi(p)) & \text{if } p \notin M \\ \chi(t) \cap H & \text{if } p \in M \text{ and } \chi(t) \cap H \not\subseteq \chi(p) \cap H \\ \emptyset & \text{if } p \in M \text{ and } \chi(t) \cap H \subseteq \chi(p) \cap H. \end{cases}$$

对于每个 $t \in M$，视图的形式为 $S_{\nu(t)}(\mathbf{x}_{\nu(t)})$，称为 S-视图。这种 S-视图的定义，对应于在自由连通树分解 $(\mathrm{T}\ \chi)$ 的物化集 M 中运行 Yannakakis 算法的自底向上半连接归约过程。事实上，半连接约简之后，可以安全地投影掉 $\chi(t) \setminus \nu(t)$ 中的任何变量。

从高层来看，M指定与每个包关联的视图类型（S-视图或 T-视图），而 $\nu(\cdot)$则标明该视图的模式（可能为空）。PMTD指定在预处理阶段物化其 S-视图，并在在线阶段计算其 T-视图。在 $M = \emptyset$的情况下，分解中的每个视图都在在线阶段获得。当 $H = A$或$H = [n]$时，自由连通性质不会对 PMTD 的树分解施加任何额外限制。

例*3.3*。我们以用于 3-可达性的 CQAP 为例：

$$\phi_3(x_1, x_4 \mid x_1, x_4) \leftarrow R_1(x_1, x_2) \wedge R_2(x_2, x_3) \wedge R_3(x_3, x_4).$$

这里，$(x_1,\ x_4)$ 是访问模式。图1 展示了上述查询的三个 PMTD，以及每个PMTD中每个包的相关视图。最左侧的 PMTD 具有一个空的物化集。中间的 PMTD 将包 $\{x_1,\ x_2,\ x_3\}$ 物化，但相关视图 S_{13} 投影去除了 x_2。最右侧的 PMTD 将唯一的包$\{x_1,\ x_2,\ x_3,\ x_4\}$ 物化，但视图 S_{14} 仅保留变量 $x_1,\ x_4$。

冗余与支配。如果没有任何一个包是另一个包的子集，我们就称树分解为无冗余的。如果树分解 $(\mathrm{T1}, \chi_1)$ 中的每个包都是树分解 $(\mathrm{T2}, \chi_2)$ 中某个包的子集，我们就称 $(\mathrm{T1}, \chi_1)$ 被支配于另一个树分解 $(\mathrm{T2}, \chi_2)$。这里，我们将把这两个概念推广到 PMTD。

定义 *3.4*（*PMTD* 冗余）。如果满足以下条件，则称 PMTD $(\mathrm{T}\ \chi,\ M,\ r)$ 非冗余：(1) 对于 $t \in M$, $\nu(t) \neq \emptyset$，不存在 $\nu(t)$ 是另一个的子集；(2) 对于 $t \notin M$，不存在 $\nu(t)$ 是另一个的子集。

定义 *3.5*（*PMTD* 支配）。如果满足以下条件，则称 PMTD $(\mathrm{T1},\ \chi_1,\ M_1,\ r_1)$ 被另一个 PMTD $(\mathrm{T2},\ \chi_2,\ M_2,\ r_2)$ 支配：(1) 对每个节点$t_1 \in M_1$，都存在某个节点 $t_2 \in M_2$，使得 $\nu(t_1) \subseteq \nu(t_2)$；(2) 对每个节点 (T) \，都存在某个节点 (T) \，使得 ( ) ( )。

对于 PMTD，冗余和支配都使用物化集和视图来定义，而不是使用包。对于包含 $M = \emptyset$ 的 PMTD，PMTD 冗余和支配都等同于标准定义。

例*3.6*。继续例3.3，假设我们考虑一个 PMTD，它采用与左侧 PMTD 相同的树分解，但物化集包含两个袋。与根袋关联的 S 视图是 S_{14}，与子袋关联的视图是 $S_\emptyset$；因此，这个 PMTD 是冗余的。此外，假设我们考虑一个只有一个袋 $\{x_1,\ x_2,\ x_3,\ x_4\}$ 的 PMTD，该袋是根，但不属于 M。与该袋关联的T视图是 T_{1234}；因此，这个 PMTD 支配图 1 中的左侧 PMTD。另一方面，图 1 中的所有 PMTD 彼此之间都既不冗余，也不互相支配。

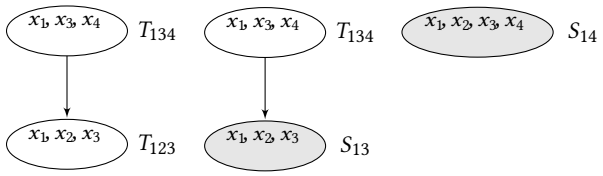


图 1：用于 3-可达性 CQAP 的三个 PMTD。物化节点以阴影标出，并标记为 S-视图。

正如我们稍后在一般框架中所述，我们主要关注由非冗余且非支配的 PMTD 构成的集合。注意，非冗余 PMTD (T

$\chi\ M\ r$ $\nu\ t \neq \emptyset$ $t \in$, ,) 对任意 V (T) 都满足 ( )，因此我们可以安全地假设所有视图均非空。

3.1 面向 PMTD 的 Online Yannakakis

我们介绍一种针对非冗余 PMTD（即不含空视图）的 Yannakakis 算法变体 [37]，称为 *Online Yannakakis*。回顾一下，对于非冗余 PMTD，预处理阶段会存储 S-视图（每个 $t \in M$对应一个），而T-视图（每个 $t \in \mathrm{T} \setminus M$对应一个）以及访问请求 Q_A 仅在在线阶段可用。

定理 3.7。考虑一个 *PMTD* $(\mathcal{T}\ \chi, M, r)$ 及其视图 $\nu(\cdot)$。给定 S-视图，我们可以在线性于其大小的空间内对其进行预处理，从而计算自由连通无环合取查询

$$\psi(\mathbf{x}_H) \leftarrow Q_A \wedge \bigwedge_{t \in M} S_{\nu(t)} \wedge \bigwedge_{t \in V(\mathcal{T}) \setminus M} T_{\nu(t)} \tag{3}$$

对于任意 T-视图和 Q_A，时间为 $O(\max_{t \in V(\mathcal{T}) \setminus M} |T_{\nu(t)}| + |Q_A| + |\psi|)$，其中 $|\psi|$ 是式 (3) 的输出大小。

请注意，时间开销不受 S-视图大小的影响，因为在整个 Online Yannakakis 过程中，S-视图仅用于半连接操作中的哈希探测。算法的详细信息及其正确性证明将在附录 A 中给出。

4 一般框架

考虑一个 CQAP $\varphi(\mathbf{x}_H \mid \mathbf{x}_A) \leftarrow \acute{\mathrm{O}}_{F \in \mathcal{E}} R_F(\mathbf{x}_F)$，其具有 $H \supseteq A$。回顾一下，我们的目标是在度约束 DC（由输入关系守卫）和 AC（由任意访问请求 Q_A守卫）下，找到最佳时空权衡，具体如第 2.2 小节所述。我们的主算法由以下参数决定：

- P $=\{P_i\}_{i \in I}$，这是一个（有限的）非冗余且非支配 PMTD 索引集，使得对于每个 $i \in I$，都有 $P_i = (\mathrm{T}_i,\ \chi_i,\ M_i,\ r_i)$。将所有此类 PMTD（其数量有限）纳入 P，将得到可能的最佳权衡。不过，正如我们稍后将看到的，考虑规模较小的 PMTD 集合也有意义，因为这样得到的时空权衡更易于理解。
- S，即空间预算。

4.1 两阶段析取规则

在本节中，我们定义一种特定类型的析取规则，这是为 PMTD获取 S-视图和 T-视图所必需的。我们首先回顾析取规则的概念。析取规则具有与合取查询完全相同的体，而其头部是一个析取

输出关系 $T_B(\mathbf{x}_B)$，我们称之为目标。令 BT $\subseteq 2^{[n]}$ 为一个非空集合，则一个析取规则 ρ 的形式为：

$$\rho: \bigvee_{B\in \mathsf{BT}} T_B(\mathbf{x}_B) \leftarrow \bigwedge_{F\in\mathcal{E}} R_F(\mathbf{x}_F). \tag{4}$$

给定一个数据库实例 D，ρ 的模型是一个元组 $(T_B)_{B\in\mathsf{BT}}$，由一组关系构成，每个目标对应一个关系，并且满足式 (4) 所表示的逻辑蕴含。更准确地说，对于任何满足主体的元组 $\mathbf{a}$，都存在一个目标 $T_B \in (T_B)_{B\in\mathsf{B}}$，使得 $\Pi_B(\mathbf{a}) \in T_B$。模型的大小定义为其输出关系的最大大小，而析取规则 ρ 的输出大小 $|\rho|$ 定义为所有模型大小中的最小值。

就我们的目的而言，我们定义了一类析取规则，称为 两阶段析取规则。

定义 *4.1*（两阶段析取规则）。由 CQAP $\varphi(\mathbf{x}_H \mid \mathbf{x}_A)$ 定义的两阶段析取规则 ρ 是一条单一析取规则，其主体取自访问合取查询 φ，而头部包含两组输出关系。换言之，ρ 的形式为

$$\rho: \bigvee_{B\in\mathsf{BS}} S_B(\mathbf{x}_B) \vee \bigvee_{B\in\mathsf{BT}} T_B(\mathbf{x}_B) \leftarrow Q_A(\mathbf{x}_A) \wedge \bigwedge_{F\in\mathcal{E}} R_F(\mathbf{x}_F), \tag{5}$$

其中 BS、BT $\subseteq 2^{[n]}$ 且至多一个可以为空。ρ 的模型 因此由两组输出关系组成，即 S-目标 $(S_B)_{B\in\mathsf{BS}}$ 和 T-目标 $(T_B)_{B\in\mathsf{BT}}$。

顾名思义，两阶段析取规则 ρ 的模型分两个阶段计算：预处理阶段和在线阶段：

预处理阶段： 我们使用预处理析取规则获得 S-目标 $(S_B)_{B\in\mathsf{BS}}$

$$\rho_S: \bigvee_{B\in\mathsf{BS}} S_B(\mathbf{x}_B) \leftarrow \bigwedge_{F\in\mathcal{E}} R_F(\mathbf{x}_F), \tag{6}$$

e 存储 S-目标的空间开销为 $O(S_\rho)$，总空间开销为 e$O(S_\rho + |\mathsf{D}|)$。预处理阶段除了度约束 AC 外，不掌握 Q_A 的任何信息，因此会显式地强制要求 S-目标对于任何访问请求实例都是通用的。

在线阶段： 给定访问请求 Q_A（在 AC 下），我们使用在线析取规则获得 T-目标 $(T_B)_{B\in\mathsf{BT}}$

$$\rho_T: \bigvee_{B\in\mathsf{BT}} T_B(\mathbf{x}_B) \leftarrow Q_A(\mathbf{x}_A) \wedge \bigwedge_{F\in\mathcal{E}} R_F(\mathbf{x}_F) \tag{7}$$

在时间和空间e$O(T_\rho)$ 中。总体时间为 e$O(T_\rho + |Q_A|)$。

如果 BS $= \emptyset$，则模型会在在线阶段从头开始计算（反之亦然）。与第 2.2 小节一样，我们关注的是分析两个内在量 S_ρ 和 T_ρ 之间的时空权衡。

接下来，假设我们有一个两阶段算法（称为2PP），给定空间预算 S，该算法有一个使用空间 $S_\rho \le S$ 的预处理过- 程2PP-预处理，以及一个使用时间（以及空间） T_ρ的在线过程2PP-Online。我们将在下一节讨论此算- 法。

4.2 预处理阶段

首先，我们从 P 构造一组两阶段析取规则，具体如下。设 v_i 为 P_i 的关联视图的映射。定义笛卡尔积 $\mathbf{A} = \times_{i\in I}\{V(\mathsf{T}_i)\}$，并令 $M = |\mathbf{A}|$。直观地说，集合中的每个元素 $\mathbf{a} \in \mathbf{A}$ 都从索引集合中的每个 PMTD 中选取一个视图。对于每个 $\mathbf{a} \in \mathbf{A}$，我们构造如下两阶段析取规则（回忆一下，$M_i$ 是 PMTD $(\mathsf{T}_i, \chi_i, M_i, r_i)$ 的物化集合）：

$$\bigvee_{\mathbf{a}_i\in M_i} S_{v_i(\mathbf{a}_i)}(\mathbf{x}_{v_i(\mathbf{a}_i)}) \vee \bigvee_{\mathbf{a}_i\notin M_i} T_{v_i(\mathbf{a}_i)}(\mathbf{x}_{v_i(\mathbf{a}_i)}) \leftarrow Q_A(\mathbf{x}_A) \wedge \bigwedge_{F\in\mathcal{E}} R_F(\mathbf{x}_F)$$

无论 $\mathbf{a} \in \mathbf{A}$ 为何值，规则的主体都相同。当对应的包位于 PMTD 的物化集内时，规则头部会引入一个 S-目标（使用对应的视图）；否则会引入一个 T-目标。根据给定的 PMTD 集构造的析取规则恰好有 M 2 个阶段，这是一个查询复杂度量。

例 *4.2.* 继续使用贯穿全文的示例，考虑图 1 中的三个 PMTD。它们会产生四条两阶段析取规则（移除冗余的 T-目标和 S-目标后）：

$$\begin{aligned}
T_{134}(x_1,x_3,x_4) \vee S_{14}(x_1,x_4) &\leftarrow \text{body}\\
T_{134}(x_1,x_3,x_4) \vee S_{13}(x_1,x_3) \vee S_{14}(x_1,x_4) &\leftarrow \text{body}\\
T_{123}(x_1,x_2,x_3) \vee T_{134}(x_1,x_3,x_4) \vee S_{14}(x_1,x_4) &\leftarrow \text{body}\\
T_{123}(x_1,x_2,x_3) \vee S_{13}(x_1,x_3) \vee S_{14}(x_1,x_4) &\leftarrow \text{body}
\end{aligned}$$

其中

$$\text{body} = Q_{14}(x_1,x_4) \wedge R_1(x_1,x_2) \wedge R_2(x_2,x_3) \wedge R_3(x_3,x_4)$$

对于每条两阶段析取规则 ρ_k，其中 $k \in [M]$，我们使用空间预算 S 运行 2PP-预处理。2PP-预处理会为 ρ_k 生成 S-目标。接下来，我们通过合并所有与 S-视图具有相同模式的 S-目标，计算 PMTDP_i 的每个 S-视图（这些目标可能来自不同析取规则的输出）。然后，我们使用全连接 $\bowtie_{F\in\mathcal{E}} R_F$ 对每个 S-视图进行半连接约简。进行此半连接约简时，可以暂时将 $\bowtie_{F\in\mathcal{E}} R_F$存储为中间真值表，并在对所有 S-视图完成半连接约简后将其移除。此步骤确保 S-视图中的任何元组都参与 $\bowtie_{F\in\mathcal{E}} R_F$。最后，我们按照定理 3.7 所述对 S-视图进行预处理。

4.3 在线阶段

请回想，收到访问请求实例 Q_A 后，我们需要返回合取查询的结果，即 $\varphi(\mathbf{x}_H)$。我们按如下方式获得 $\varphi(\mathbf{x}_H)$。首先，我们对每个 ρ_k应用 2PP-Online，以在时间 e$O(T_{\rho_k} + |Q_A|)$ 内获取其 T-目标（数量为 e$O(T_{\rho_k})$）。令 $T_{\max} = \max_{k\in[M]} T_{\rho_k}$。

接下来，我们通过合并所有与 T-视图具有相同模式的 T-目标（可能来自不同析取规则的输出），计算 PMTD P_i 的每个 T-视图。我们将每个 T-视图（大小为 e$O(T_\rho)$）与每个输入关系以及 Q_A 进行半连接约简。然后，对于 $\{P_i\}_{i\in I}$ 中的每个 PMTD，我们计算自由连通无环合取查询。

$$\psi_i(\mathbf{x}_H) \leftarrow Q_A \wedge \bigwedge_{t\in M_i} S_{v_i(t)} \wedge \bigwedge_{t\in V_i(\mathcal{T}_i)\setminus M_i} T_{v_i(t)} \tag{8}$$

通过按照第3.1节所述应用 Online Yannakakis，在 e$O(T_{\text{最大值}}) + O(|Q_A| + |\phi_i \cap \varphi|)$ 时间内完成。我们通过合并集合 $\varphi = \biguplus_{i\in I} \psi_i$ 中所有 PMTD 的输出，得到最终结果。总而言之，我们在 e$O(T_{\text{最大值}} + |Q_A|) + O(|\varphi|)$ 时间内处理访问请求 Q_A。

5 构造权衡关系

设 ρ 为采用形式(5) 的两阶段析取规则，在度约束 DC（由输入关系守卫）和度约束 AC（由 Q_A守卫）下成立。本节将讨论如何使用PANDA 分两个阶段获得 ρ 的模型，以及由此产生的时空权衡。由于空间有限，我们将以非正式的方式进行介绍，并通过一个示例阐述关键思想。完整的细节和证明将推迟至附录C和附录D。我们将使用以下规则作为贯穿全文的示例，其中 $|R_1| = |R_2| = |\mathbb{D}|$:

$$T_{123} \vee S_{13} \leftarrow Q_{13}(x_1, x_3), R_1(x_1, x_2), R_2(x_2, x_3).$$

对于 2-可达性查询，考虑两个 PMTD 得到的唯一规则就是这条规则。要计算析取规则，PANDA从一个香农流不等式开始，它是关于集合函数的不等式 $h:\ 2^{[n]} \to \mathbb{R}_+$ 它必须对任何作为多拟阵的集合函数成立[2]。对于我们的目的，我们需要一个联合香农流不等式，它适用于两个集合函数h_S，h_T且这两个函数都必须是多拟阵。直观地说，h_S支配预处理阶段，而h_T支配在线阶段。我们示例中的联合香农流不等式为:

$$\underbrace{h_S(1) + h_T(2|1)}_{R_1} + \underbrace{h_S(3) + h_T(2|3)}_{R_2} + 2\underbrace{h_T(13)}_{Q_{13}} \geq \underbrace{h_S(13)}_{S_{13}} + 2\underbrace{h_T(123)}_{T_{123}}$$

其中 $h(Y|X) = h(Y) - h(X)$。右侧包含h_S 中对应于 S-目标的项，以及 h_T 中对应于 T-目标的项。左侧包含 h_T 中对应于访问请求 Q_A 的一项，还可能包含 h_S (h_T) 中编码度约束 DC（DC $\cup$ AC）的项。更重要的是，它包含通过将具有属性 Y 的输入关系拆分为两部分来关联两个多拟阵的项，即 $(i)\ h_S(X) + h_T(Y|X)$，或 $(ii)\ h_T(X) + h_S(Y|X)$，其中 $X \subseteq Y$。直观地说，第一次拆分将重 X-值实体化，并将其他所有内容发送到在线阶段；而第二次拆分则对轻 X-值进行预处理，并将重 X-值发送到在线阶段。在我们的示例中，关系 R_1 被拆分为 $h_S(1) + h_T(2|1)$，并且每个部分都被发送到不同的多拟阵。利用上述联合香农流不等式的系数，我们得到如下固有时空权衡:

$$S \cdot T^2 \cong |Q_{13}|^2 \cdot |\mathcal{D}|^2$$

我们用 $\cong$记号表示 $S \cdot T^2 = eO(|Q_{13}|^2 \cdot |\mathbb{D}|^2)$。一般而言，我们证明（形式化定义见定理 D.6）:

定理 5.1（非正式表述）。对于两阶段析取规则，任何联合香农流不等式都蕴含一种时空权衡，可通过读取该不等式的系数计算得出。

上述定理要求我们先获得一个联合香农流不等式，才能得出时空权衡。我们还证明，给定空间预算 S 后，也可以通过线性规划计算出能带来最佳回答时间的最优不等式。

2PP算法。现在介绍主算法的工作方式（详细说明请参见附录D）。对于运行示例，我们取 $|Q_{13}| = 1$，而 S 是固定的空间预算。

[2]多拟阵是一个集合函数 $h:\ 2^{[n]} \to \mathbb{R}_+$，它非负、单调且次模，并满足 $h(\emptyset) = 0$。

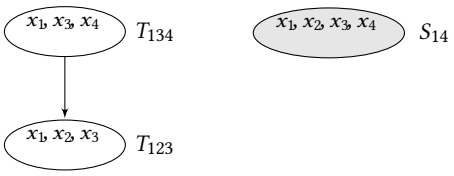


图 2：正方形 CQAP 的两个 PMTD。物化节点以阴影标出，并标记为 S-视图。

首先，2PP 扫描联合香农流不等式，并将 $R_1(x_1,\ x_2)$（在 x_1 上）划分为 R_1^H和 R_1^L，其中 R_1^H包含所有满足 $|\sigma_{x_1=t}(R_{12})| \geq |\mathbb{D}|/\sqrt{S}$ 的 $(x_1,\ x_2)$ 元组，而 R_1^L包含满足 $\deg 12\ (x_2|x_1) \leq |\mathbb{D}|/\sqrt{S}$ 的元组。R_2则对称地（在 x_3 上）划分为 R_2^H和 R_2^L。由此产生四个子问题：$\{R_1^H,\ R_2^H\}$、$\{R_1^H,\ R_2^L\}$、$\{R_1^L,\ R_2^H\}$ 和 $\{R_1^L,\ R_2^L\}$。一般来说，这些划分将依据联合流中的相关项进行。

预处理阶段（2PP-预处理）受以下对象的香农流不等式支配：h_S，即 $h_S(1) + h_S(3) \geq h_S(13)$。现在我们沿用PANDA 的方法，为该不等式构造证明序列。证明序列通过一系列较小的步骤证明该不等式，每一步都可以解释为一个关系运算符。本例的证明序列如下:

$$\begin{aligned} h_S(1) + h_S(3) &\geq h_S(13|3) + h_S(3) && (submodularity) \\ &= h_S(13) && (composition) \end{aligned}$$

在这种情况下，PANDA 会尝试连接每个子问题中的两个关系。不过，只有当结果所需的空间不超过 S时，我们才允许这样做。由于 R_1^H和 R_2^H的大小分别不超过$|\mathbb{D}|/(|\mathbb{D}|/\sqrt{S}) = \sqrt{S}$个值，因此对于 x_1，x_3，子问题$\{R_1^H,\ R_2^H\}$可以存储在 S_{13} 中，所需空间至多为$\sqrt{S} \cdot \sqrt{S} = S$。

在线阶段（2PP-Online）接收一个包含单个元组的访问请求$Q_{13}(x_1, x_3)$。现在，2PP-Online 按照多拟阵 h_T 的第二个证明序列进行:

$$\begin{aligned} h_T(2|1) + h_T(2|3) + 2h_T(13) &\geq 2h_T(2|13) + 2h_T(13) && (submod.) \\ &= 2h_T(123) && (comp.) \end{aligned}$$

对于其他 3 个子问题，2PP-Online 计算以下 3 个连接：$Q_{13}(x_1, x_3) \bowtie R_2^L(x_2, x_3)$、$Q_{13}(x_1, x_3) \bowtie R_1^L(x_1, x_2)$ 和 $Q_{13}(x_1, x_3) \bowtie R_1^L(x_1, x_2)$。在次模性步骤中，2PP-Online 确定，对于第一个连接，$\deg 23\ (x_2|x_3) \leq |\mathbb{D}|/\sqrt{S}$，因此该连接需要 $|Q_{13}| \cdot |\mathbb{D}|/\sqrt{S} \leq |\mathbb{D}|/\sqrt{S}$ 的时间；并且由于 $\deg 12(x_2|x_1) \leq |\mathbb{D}|/\sqrt{S}$，最后两个相同的连接需要 $|Q_{13}| \cdot \deg 12(x_2|x_1) \leq |\mathbb{D}|/\sqrt{S}$ 的时间。因此，总体在线计算时间为 $|\mathbb{D}|/\sqrt{S}$。

例*5.2*（正方形查询）。现在，我们通过以下 CQAP 给出如何构建权衡的完整示例:

$$\varphi(x_1, x_3 \mid x_1, x_3) \leftarrow R_1(x_1, x_2) \wedge R_2(x_2, x_3) \wedge R_3(x_3, x_4) \wedge R_4(x_4, x_1).$$

这描述了以下任务：给定图中的两个顶点，判断它们是否分别位于正方形的两个对角顶点。我们考虑两个 PMTD。第一个 PMTD 有一个根袋 {1，3、4}，它关联到一个 T-视图 T_{134}；另有一个袋 {1、3、2}，它关联到一个 T-视图 T_{132}。第二个 PMTD 有一个袋 {1、2、3、4}，它关联到

一个带有 an S-视图 S_{13}。这两个 PMTD 如图 2 所示。由此生成两条析取规则：

$$T_{134} \vee S_{13} \leftarrow \text{body}, \qquad T_{132} \vee S_{13} \leftarrow \text{body}$$

其中主体 $= Q_{13}\ x_1\ x_3\ \wedge R_1\ x_1\ x_2\ \wedge R_2\ x_2\ x_3\ \wedge R_3\ x_3\ x_4\ \wedge(,\ )(,\ )(,\ )(,\ )(,\ )$ 我们可以为第一条规则构造如下联合香农流不等式（及其证明序列）：

$$\begin{aligned}
&\underbrace{h_S(1) + h_T(4|1)}_{R_4} + \underbrace{h_S(3) + h_T(4|3)}_{R_3} + 2 \cdot \underbrace{h_T(13)}_{Q_{13}} \\
&\quad \geq h_S(13) + h_T(4|1) + h_T(4|3) + 2 \cdot h_T(13) \\
&\quad \geq h_S(13) + h_T(4|13) + h_T(13) + h_T(4|13) + h_T(13) \\
&\quad = \underbrace{h_S(13)}_{S_{13}} + 2 \cdot \underbrace{h_T(134)}_{T_{134}}.
\end{aligned}$$

对于第二条规则，我们对称地构造一个证明序列，其长度为 $2\log|\mathcal{D}| + 2\log|Q_{13}| \geq h_S(13) + 2 \cdot h_T(132)$。因此，读取上述联合香农流不等式的系数，我们得到给定正方形 CQAP 的以下固有时空权衡 $S \cdot T^2 \cong |\mathcal{D}|^2 \cdot |Q_{13}|^2$。

6 应用

在本节中，我们应用我们的框架，为几个具体问题获得最先进的时空权衡结果，并获得新的权衡结果。我们将关于层次化 CQAP 的讨论推迟到论文的完整版本 [38]。

6.1 k-集合交集的权衡

我们首先研究与非布尔型 k-集合不相交问题（set $y = x_{k+1}$）相对应的 CQAP (2)

$$\varphi(\mathbf{x}_{[k+1]} \mid \mathbf{x}_{[k]}) \leftarrow \bigwedge_{i \in [k]} R(x_{k+1}, x_i)$$

从包含单个节点 t、带有 $\chi(t) = [k+1]$ 的分解出发，我们构造两个 PMTD，一个带有 $M_1 = \emptyset$，另一个带有 $M_2 = \{t\}$。因此，$v_1(t) = v_2(t) = [k+1]$。由此得到以下（唯一的）两阶段析取规则：

$$T_{[k+1]} \vee S_{[k+1]} \leftarrow Q_{[k]}(\mathbf{x}_{[k]}) \wedge \bigwedge_{i \in [k]} R(x_{k+1}, x_i)$$

对于这条规则，我们有以下联合香农流不等式：

$$\begin{aligned}
&h_S(k, k+1) + \sum_{i \in [k-1]} \{h_S(i|k+1) + h_T(k+1)\} + (k-1) \cdot h_T([k]) \\
&\quad \geq h_S([k+1]) + (k-1) \cdot h_T([k+1]).
\end{aligned}$$

根据定理 5.1，我们得到权衡 $S \cdot T^{k-1} \cong |\mathbb{D}|^k \cdot |Q_A|^{k-1}$。

6.2 通过分数边覆盖实现的权衡

设 $\varphi(\mathbf{x}_A \mid \mathbf{x}_A)$ 是一个超图为 $([n], \mathrm{E})$ 的 φ 合取查询访问模式。一种 $S \subseteq [n]$ 的分数边覆盖是一个赋值 $\mathbf{u} = (u_F)_{F \in \mathcal{E}}$，满足 $(i)\ u_F \geq 0$，且对于每个 $i \in S$，$(ii)\ \sum_{F: i \in F} u_F \geq 1$。对于 $[n]$ 的任意分数边覆盖 $\mathbf{u}$，我们定义 $\mathbf{u}$ 相对于 $A \subseteq [n]$ 的 松弛量：

$$\alpha(\mathbf{u}, A) \stackrel{\text{def}}{=} \min_{i \notin A} \sum_{F \in \mathcal{E}: i \in F} u_F.$$

换言之，松弛量是我们可以将分数覆盖 $\mathbf{u}$ 缩小的最大倍数，使其仍然是一个有效的边不在 A 中的变量的覆盖。因此，$(u_F/\alpha(\mathbf{u}\ A))_{F \in \mathcal{E}}$ 是 $[n] \setminus A$ 的分数边覆盖。我们总有 $\alpha(\mathbf{u}\ A) \geq 1$。

定理 6.1。 设 $\varphi(\mathbf{x}_A \mid \mathbf{x}_A)$ 为一个 *CQAP*。设 $\mathbf{u}$ 为 φ 的超图的任意分数边覆盖。那么，对于任意输入数据库 $\mathcal{D}$ 和任意访问请求，以下内在权衡成立：

$$S \cdot T^{\alpha(\mathbf{u},A)} \cong |Q_A|^{\alpha(\mathbf{u},A)} \cdot \prod_{F \in \mathcal{E}} |R_F|^{u_F}$$

上述定理也可以作为 [13] 中定理 1 的推论来证明。不过，[13] 中使用的数据结构复杂得多，因为其目标还包括限制枚举期间的延迟（而我们关注的是总时间）。 [12] 给出了一种更简单、具有相同权衡的构造。我们的框架通过采用一种包含两个 PMTD 的简单物化策略，得到了相同的结果。

例 *6.2.* 考虑 $\varphi(\mathbf{x}_{[k]} \mid \mathbf{x}[k]) \leftarrow \acute{O}_{i \in [k]} R(y,\ x_i)$（对应于 k-集合不相交问题），其分数边覆盖为 $\mathbf{u}$，其中 $u_j = 1$ for $j \in \{1, \ldots\ldots, k\}$。相对于 $[k]$ 的松弛量为 k，因为分数边覆盖 $\hat{}\ \mathbf{u}$，其中 $\hat{u}_i = u_i/k = 1/k$ 覆盖 x。应用定理 6.1，我们得到时空权衡 $S \cdot T^k \cong |Q_A|^k \cdot |\mathbb{D}|^k$。当 $|Q_A| = 1$时，这与已知的 k-集合不相交问题的最佳时空权衡相匹配。

6.3 通过树分解实现的权衡

设 $\varphi(\mathbf{x}_A \mid \mathbf{x}_A)$ 是一个 CQAP。在上一节中，我们利用两个平凡的 PMTD 得到了时空权衡。这里，我们将展示如何通过考虑与一个分解相对应的更大 PMTD 集合，使我们的框架得到更优的时空权衡。

任取一个非冗余的自由连通分解 $(\mathrm{T}\ \chi, r)$。首先，将分解中任意一组互不为彼此祖先的节点作为物化集。然后，对于物化集中的每个节点 t，将子树 t 中的所有包含并到 t 的包中（并截断该子树）。遍历所有此类物化集，即可构造出一个固定的（有限）PMTD 集合。我们称这个 PMTD 集合是由 $(\mathrm{T}\ \chi, r)$诱导而来的。现在，将诱导得到的 PMTD 集合输入我们的一般框架。为讨论由此得到的时空权衡，任取一组分配给每个节点 $t \in V(\mathrm{T})$ 的分数边覆盖 $\mathbf{u}_t$，并令 u_t^* 为其总权重。令 A_t 表示节点 t 与其父节点的公共变量（对于根节点，$A_r = A$），并将 $\alpha_t = \alpha(\mathbf{u}_t, A_t)$ 定义为节点 t 相对于 A_t 的松弛量。现在，取 T 中任意一条从根到叶的路径上的节点 P。我们可以证明，任意此类路径 P 都会产生以下内在权衡：

$$S^{\sum_{t \in P} 1/\alpha_t} \cdot T \cong |Q_A| \cdot |\mathcal{D}|^{\sum_{t \in P} u_t^*/\alpha_t}$$

为了获得最终的时空权衡，我们取所有根到叶路径中此类权衡的最差值。我们在附录E.2 中展示了如何获得这一权衡，并说明了为什么它能够恢复先前的结果 [12]。我们的框架保证，将PMTD 添加到我们在此处考虑的诱导集合中，只会使这一权衡变得更好。

示例 *6.3.* 考虑 4-可达性 CQAP。这里，$H = A = \{x_1,\ x_5\}$。我们将考虑具有袋 $t_1 = \{x_1,\ x_2,\ x_4,\ x_5\} \rightarrow t_2 = \{x_2,\ x_3,\ x_4\}$ 的树分解。

为包 t_1 取边覆盖 $u_1 = 1, u_4 = 1$，为包 t_2 取边覆盖 $u_2 = 1$, $u_3 = 1$。第一个包的松弛量为 $\alpha_1 = 1$

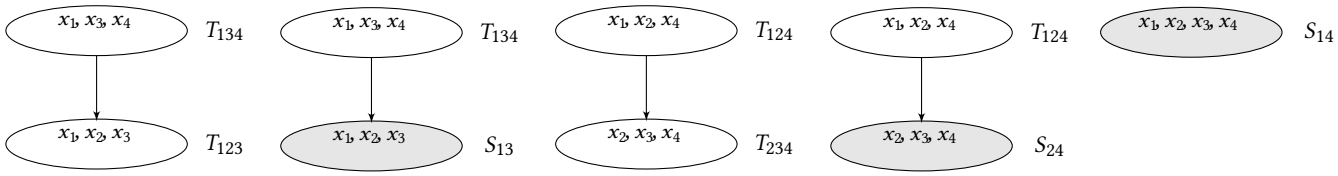


图 3：3-可达性 CQAP 的 PMTD。

表 1：用于3-可达性的两阶段析取规则

rule	head	权衡
ρ_1	$T_{134} \vee T_{124} \vee S_{14}$	$S \cdot T^2 \cong \lvert D\rvert^2 \cdot \lvert Q_A\rvert^2$
ρ_2	$T_{123} \vee S_{13} \vee T_{124} \vee S_{14}$	$S^2 \cdot T^3 \cong \lvert D\rvert^4 \cdot \lvert Q_A\rvert^3$ $T \cong \lvert D\rvert \cdot \lvert Q_A\rvert$
ρ_3	$T_{134} \vee T_{234} \vee S_{24} \vee S_{14}$	$S^2 \cdot T^3 \cong \lvert D\rvert^4 \cdot \lvert Q_A\rvert^3$ $T \cong \lvert D\rvert \cdot \lvert Q_A\rvert$
ρ_4	$T_{123} \vee S_{13} \vee T_{234} \vee S_{24} \vee S_{14}$	$S \cdot T \cong \lvert D\rvert^2 \cdot \lvert Q_A\rvert$ $S^4 \cdot T \cong \lvert D\rvert^6 \cdot \lvert Q_A\rvert$ $T \cong \lvert D\rvert \cdot \lvert Q_A\rvert$

（相对于 x_1, x_5），而第二个具有松弛量 $\alpha_2 = 2$（相对于 x_2, x_4）。这里有一条根到叶路径，因此我们得到权衡 $S^{1+1/2} \cdot T \cong \lvert Q_A\rvert \lvert \quad \rvert^{2\ 1+2\ 2}$, $S^{3\ 2}\ T \cong \lvert Q_A\rvert \lvert \quad \rvert^3 \cdot D//$，或者等价地 $/ \cdot \cdot D$。

6.4 k-可达性的权衡

在本部分中，我们将考虑与示例 2.3 中描述的 k-可达性问题相对应的 CQAP：

$$\phi_k(x_1, x_{k+1} \mid x_1, x_{k+1}) \leftarrow \bigwedge_{i \in [k]} R(x_i, x_{i+1}).$$

先前工作 [15] 给出了针对输入 D 的以下权衡，并猜想其对于 $\lvert Q_A\rvert = 1$ 是渐近最优的：

$$S \cdot T^{2/(k-1)} \cong \lvert \mathcal{D}\rvert^2 \cdot \lvert Q_A\rvert^{2/(k-1)}.$$

我们将展示，通过应用我们的框架，可以显著改进上述权衡 $k \geq 3$。

***3*-可达性。** 首先，我们考虑所有非冗余且非支配的 PMTD 集合（共五个），如图 3 所示。这五个 PMTD 将导出 $2^4 = 16$ 析取规则，但我们可以舍弃目标严格多于其他规则的规则，从而减少需要分析的规则数量。例如，可以忽略头部为 $T_{134} \vee S_{13} \vee T_{124} \vee S_{14}$ 的析取规则，因为另一个析取规则的目标是其真子集，即 $T_{134} \vee T_{124} \vee S_{14}$。我们在表 1 中列出需要考虑的两阶段析取规则的头部（为简洁起见，省略变量），以及每条规则的内在权衡。

请注意，规则可能存在两种（或更多种）互不支配的权衡；因此，我们需要根据所考虑的情形选择最佳权衡（关于如何证明每种权衡，见附录 E）。为了理解我们通过分析得到的综合权衡，我们在图 4a 中绘制每条权衡曲线，方法是取 $\log_{\lvert \mathcal{D}\rvert}$，然后将 x 轴设为 $\log_{\lvert \mathcal{D}\rvert}(T)$，将 y 轴设为 $\log_{\lvert \mathcal{D}\rvert}(S)$（固定 $\lvert Q_A\rvert = 1$）。图中的虚线表示得到的权衡，它是一个分段线性函数。请注意，每条线段代表在相应空间范围内最优的不同策略。还要注意，图 4a 不一定针对 $\lvert Q_A\rvert > 1$ 进行了优化。假设 $S = \lvert D\rvert$，并且我们在在线阶段收到 $\lvert D\rvert$ 个单元组访问请求。使用上述权衡逐个回答这些请求，所需时间为 $eO(\lvert D\rvert^2)$。不过，一种更好的策略是将这 $\lvert D\rvert$ 个元组批量处理（组成一个 4-循环查询），并使用 PANDA 从头开始回答，所需时间为 $eO(\lvert D\rvert^{3/2})$。

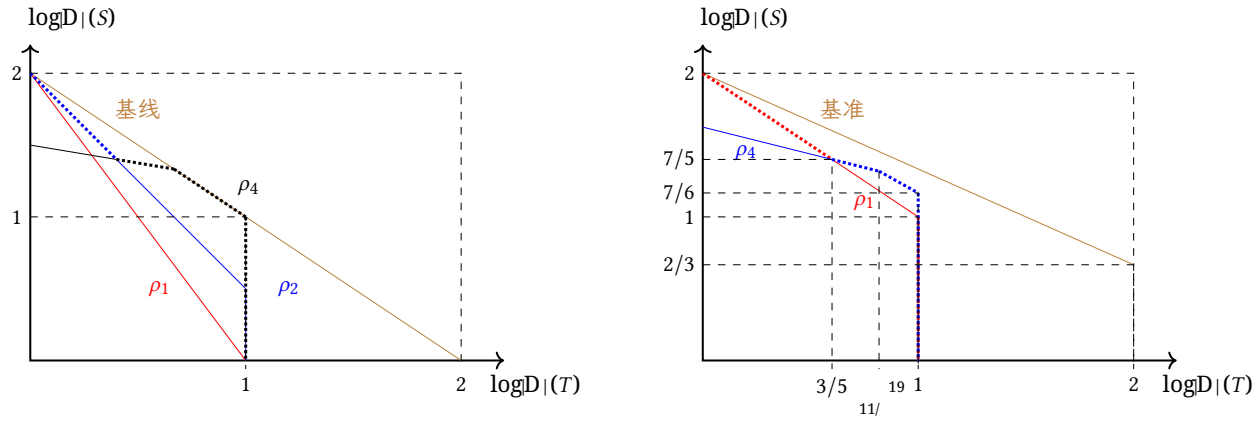
***4*-可达性。** 我们还研究了四可达性问题的 CQAP。相关计算（相当复杂）留待附录 E 介绍，但这里给出一张与图 4a 类似的图（图 4b）。一个令人意外的观察是，在所有空间范围内，新的时空权衡都优于此前最先进的结果。还应指出，通过纳入更多 PMTD，权衡可能进一步改善（我们的分析涉及 11 个 PMTD！），但相关计算超出了本文的范围。

一般可达性。 分析 $k \geq 5$ 的最优权衡是一项非常复杂的工作。不过，根据第 6.3 小节以及对 [12], 的分析，我们的框架至少可以获得 $S \cdot T^{2/(k-1)} \cong \lvert D\rvert^2$ 权衡，而且很可能还能对其进行严格改进。

7 相关工作

集合交集与距离预言机。 过去十年间，精确和近似查询回答的时空权衡一直是多个研究领域的活跃课题 [1, 8, 25, 26] [8] 。

Cohen 和 Porat 提出了快速交集问题，并给出了一种数据结构，可枚举两个集合的交集，并保证总回答时间。Gold- stein 等人 [15] 提出了图上的 k-可达性问题，并展示了一种简单的递归数据结构，实现了 $S \cdot T^{2/(k-1)} = O(\lvert D\rvert^2)$ 的权衡。他们还推测该权衡是最优的，并以此论证 [2]提出的近似距离预言机具有最优性。Patrascu 和 Roditty [32], 开创了图上（近似）距离预言机的研究，并基于关于集合交集问题最优数据结构的猜想，证明了稀疏图距离预言机大小的下界。Cohen 和 Porat [9] 也将集合交集与距离预言机联系起来。Agar- wal 等人 [2, 3] 引入了预言机的伸展概念，用以控制答案所允许的误差。此外，对于伸展度为 2 和伸展度为 3 的预言机，我们可以实现 $S \cdot T = O(\lvert D\rvert^2)$ 的权衡，并且



(a) 3-可达性 CQAP。 **(b) 4-可达性 CQAP。**

图 4：3-可达性和 4-可达性 CQAP 的空间-时间权衡。通过我们的框架获得的新权衡以点线段表示。棕色线条（基准）展示了此前最先进的权衡。

$S \cdot T^2 = O(|\mathbb{D}|^2)$，而对于任意整数 $k > 0$，伸展度为 $(1 + 1/k)$ 的预言机呈现出 $S \cdot T^{1/k} = O(|\mathbb{D}|^2)$ 的权衡。遗憾的是，对于非恒定查询时间，目前尚无已知下界。

空间/延迟权衡。另一类研究关注枚举非布尔查询的查询结果，目标是尽量缩短输出中相邻元组之间的延迟 。在常数延迟枚举中， [4, 33], 目标是对合取查询进行线性时间（和空间）预处理后，实现常数延迟；然而，只有一部分合取查询能够实现这种权衡。因子化数据库 [31] 则能在更昂贵的超线性预处理步骤后，对任意合取查询实现常数延迟枚举。若要进一步缩短预处理时间，就必须增加延迟。Kara 等人 [20] 针对静态（和动态）设置，提出了在预处理时间与枚举任意层次化合取查询结果的延迟之间的权衡。Deng 等人 [14] 开始研究范围子图计数和范围子图列举问题中的空间-查询权衡。带访问模式的合取查询问题最初由 Deep 和 Koutris [13] 提出（在限制 $|Q_A| = 1$ 下），但作者只考虑了完整 CQAP。先前的工作 [36] 研究了构建空间高效的图视图以执行图分析的问题，但没有提供任何理论保证。更近期，Kara 等人 [21] 研究了 CQAP 的预处理时间、延迟和更新时间之间的权衡。他们刻画了这样一类 CQAP：经过线性预处理时间后，能够实现常数延迟枚举和常数更新时间。上述所有结果都关注空间（或预处理时间）与延迟之间的权衡，而我们的工作聚焦于回答查询所需的总时间。与我们的工作最为接近的是 [12] 中未经同行评审的工作。该文也研究了为布尔合取查询建立权衡的问题。作者提出了两项结果，对 [13] 作出了小幅改进。他们还率先认识到， k-可达性权衡并非最优，并针对 $k \geq 3$ 提出了一项小幅改进。我们的工作借助更全面的框架实现了对这些结果的大幅推广。对于动态设置， [5] 开启了在更新过程中回答合取查询的研究。近期， [19] 提出了一种在更新过程中统计三角形数量的算法。 [21] 提出了用于 CQAP 的动态算法，并给出了查询的句法刻画：这类查询能够以常数时间处理每次单元组更新，且其输出元组可通过常数延迟枚举。

合取查询评估。我们提出的框架基于高效合取查询评估领域的近期进展，尤其是 PANDA 算法 [24]。这一强大的算法成果承袭了长期以来关于查询分解 [16, 27, 28], 、最坏情形最优算法 [29], 以及合取查询评估与信息论之间联系 [22, 23]的研究脉络。

8 结论

在本文中，我们提出了一个用于计算回答带访问模式的合取查询时的一般空间-时间权衡的框架。我们通过展示该框架如何捕捉针对以往分别研究的问题的最先进的权衡，说明了其多功能性。应用我们的框架还发现了此前未知的权衡。仍有许多开放问题，其中包括获得与我们的上界相匹配的（条件）下界，以及研究如何使我们的方法具有实用性。

仍有许多值得进一步研究的开放问题。尤其是，目前尚无已知下界能够证明这些时空权衡的最优性。近似距离预言机现有时空权衡的最优性如今也再次成为一个开放问题；我们相信，所提出的框架应当能够改进上界。若能在实践中检验这一框架的适用性，也将十分有趣。具体而言，我们的框架还可以扩展为纳入预先计算好的视图，这是一种常见的应用场景。对此，仍需设法优化时间复杂度中的常数，以确保实现具有可行性。

参考文献

[1] Peyman Afshani 和 Jesper Asbjørn Sindahl Nielsen。文档索引问题的数据结构下界。载于 *ICALP*，2016年。[2] Rachit Agarwal。距离预言机中的空间-伸展-时间权衡。载于 *ESA*，第 49–60 页。施普林格，2014年。[3] Rachit Agarwal、P Brighten Godfrey 和 Sariel Har-Peled。稀疏图中的近似距离查询与紧凑路由。载于 *INFOCOM*，第 1754–1762 页。IEEE，2011年。[4] Guillaume Bagan、Arnaud Durand 和 Etienne Grandjean。关于无环合取查询与常数延迟枚举。载于 *CSL*，第 4646 卷《计算机科学讲义》，第 208–222 页。施普林格，2007年。[5] Christoph Berkholz、Jens Keppeler 和 Nicole Schweikardt。更新操作下的合取查询回答。载于 *PODS*，第 303–318 页。ACM，2017年。[6] Angela Bonifati、Wim Martens 和 Thomas Timm。对大型 SPARQL 查询日志的分析研究。《*VLDB*期刊》，29(2):655–679，2020年。[7] Timothy M Chan 和 Moshe Lewenstein。通过加法组合学求解聚类整数 3SUM。载于 *STOC*，第 31–40 页，2015年。[8] Hagai Cohen 和 Ely Porat。快速集合交集与双模式匹配。《理论计算机科学》，411(40-42):3795–3800，2010年。[9] Hagai Cohen 和 Ely Porat。稀疏图距离预言机的难度。*arXiv* 预印本 *arXiv:1006.1117*，2010年。[10] Nilesh Dalvi、Christopher Ré 和 Dan Suciu。概率数据库：尘土中的钻石。《*ACM*通讯》，52(7):86–94，2009年。[11] Shaleen Deep、Xiao Hu 和 Paraschos Koutris。带投影的合取查询枚举算法。载于 第*24*届国际数据库理论会议，第 1 页，2021年。[12] Shaleen Deep、Xiao Hu 和 Paraschos Koutris。回答布尔合取查询的时空权衡。*arXiv* 预印本 *arXiv:2109.10889*，2021年。[13] Shaleen Deep 和 Paraschos Koutris。合取查询结果的压缩表示。载于 *PODS*，第 307–322 页。ACM，2018年。[14] Shiyuan Deng、Shangqi Lu 和 Yufei Tao。范围子图计数与枚举中的空间-查询权衡。载于 第*26*届国际数据库理论会议，*ICDT 2023*，*2023*年*3*月*28*日至*31*日，希腊约阿尼纳，第 6:1–6:25 页，2023年。[15] Isaac Goldstein、Tsvi Kopelowitz、Moshe Lewenstein 和 Ely Porat。空间/时间权衡的条件下界。载于 *WADS*，第 421–436 页。施普林格，2017年。[16] Georg Gottlob、Gianluigi Greco 和 Francesco Scarcello。树宽与超树宽。《可处理性：困难问题的实用方法》，第 1 卷，2014年。[17] Gianluigi Greco 和 Francesco Scarcello。枚举 CSP 解的结构可处理性。《*Constraints*》，18(1):38–74，2013年。[18] Muhammad Idris、Martín Ugarte 和 Stijn Vansummeren。动态 Yannakakis 算法：更新操作下紧凑而高效的查询处理。载于*2017*年 *ACM* 数据管理国际会议论文集，第 1259–1274 页，2017年。

[19] Ahmet Kara、Hung Q Ngo、Milos Nikolic、Dan Olteanu 和 Haozhe Zhang。更新情况下以最坏情形最优时间计数三角形。发表于 *ICDT*，2019年。[20] Ahmet Kara、Milos Nikolic、Dan Olteanu 和 Haozhe Zhang。层次查询静态与动态求值中的权衡。发表于 *PODS*，第 375–392 页，2020年。[21] Ahmet Kara、Milos Nikolic、Dan Olteanu 和 Haozhe Zhang。更新情况下具有自由访问模式的合取查询。发表于 第 *26* 届国际数据库理论会议（*ICDT 2023*），2022年。第 26 届国际数据库理论会议，2023年，ICDT 2023；会议日期：2023年 3 月 28 日至 2023 年 3 月 31 日。[22] Mahmoud Abo Khamis 、 Phokion G. Kolaitis 、 Hung Q. Ngo 和 Dan Suciu。包查询包含与信息论。发表于 *PODS*，第 95–112 页。ACM，2020年。[23] Mahmoud Abo Khamis、Hung Q. Ngo 和 Atri Rudra。FAQ：常见问题。发表于 *PODS*，第 13–28 页。ACM，2016年。[24] Mahmoud Abo Khamis、Hung Q. Ngo 和 Dan Suciu。香农型不等式、次模宽度和析取 Datalog 之间有何关联？发表于 *PODS*，第 429–444 页。ACM，2017年。[25] Tomasz Kociumaka、Jakub Radoszewski 和 Wojciech Rytter。适用于常数大小字母表的高效乱序模式匹配索引。发表于 *ESA*，第 625–636 页。施普林格，2013年。[26] Kasper Green Larsen、J Ian Munro、Jesper Sindahl Nielsen 和 Sharma V Thankachan。若干字符串索引问题的困难性。理论计算机科学，582：74–82，2015年。[27] Dániel Marx。你能超越树宽吗？*Theory Comput.*，6(1)：85–112，2010年。[28] Dániel Marx。适用于约束满足和合取查询的易处理超图性质。*J. ACM*，60(6)：42:1–42:51，2013年。[29] Hung Q. Ngo、Christopher Ré 和 Atri Rudra。偏斜再度来袭：连接算法理论的新进展。*SIGMOD Rec.*，42(4)：5–16，2013年。[30] Dan Olteanu 和 Maximilian Schleich。因子化数据库。*ACM SIGMOD Record*，45(2)：5–16，2016年。[31] Dan Olteanu 和 Jakub Závodný。查询结果因子化表示的大小界。*ACM Trans. Database Syst.*，40(1)：2:1–2:44，2015年。[32] Mihai Patrascu 和 Liam Roditty。超越 Thorup-Zwick 界的距离预言机。发表于 *FOCS*，第 815–823 页。IEEE，2010年。[33] Luc Segoufin。以常数延迟枚举查询的答案。发表于 *ICDT*，第 10–20 页。ACM，2013年。[34] Yilei Wang 和 Ke Yi。安全的 Yannakakis：针对私有数据的连接聚合查询。发表于 *SIGMOD Conference*，第 1969–1981 页。ACM，2021年。[35] Yilei Wang 和 Ke Yi。通过电路进行查询求值。发表于 *PODS '22*，第 67–78 页。ACM，2022年。[36] Konstantinos Xirogiannopoulos 和 Amol Deshpande。从关系数据库中提取和分析隐藏图。*CoRR*，abs/1701.07388，2017年。[37] Mihalis Yannakakis。无环数据库模式的算法。发表于 *VLDB*，第 82–94 页。IEEE 计算机学会，1981年。[38] Hangdong Zhao、Shaleen Deep 和 Paraschos Koutris。具有访问模式的合取查询的时空权衡。*arXiv* 预印本 *arXiv:2304.06221*，2023年。

A 第 3 节缺失的细节

我们将把 Yannakakis 算法改造为适用于自由连通树分解的两个轮次：自底向上半连接归约过程和自顶向下连接轮次。算法首先根据边连接的是两个 S 视图、两个 T 视图，还是一个 S 视图和一个 T 视图，对所有边进行分组。我们分别将它们称为 SS 边、TT 边和 ST 边。PMTD 的一个关键观察是，每个分支上的边总是按以下自底向上顺序排列：先是若干条 SS 边，然后至多一条 ST 边，接着是若干条 TT 边。

为了预处理这些 S-视图，我们首先在 SS-边上执行一次自底向上的半连接遍历。然后，对于每个 S-视图，我们创建一个哈希索引，其搜索键为它与其（唯一）父视图共有的变量。现在，我们来说明OnlineYannakakis的两遍处理过程。这使得父视图与作为 S-视图的子视图之间的半连接可以在与父视图大小呈线性关系的时间内完成。

自底向上半连接归约过程。 首先，我们自底向上执行半连接归约过程，通过半连接和投影移除树中的所有非头部变量。向上遍历时，根据所访问边的类型，会出现三种情形。设 $(t, p) \in E(\mathrm{T})$ 为当前访问的边，其中 t 是子节点，p 是 t 的父节点。我们区分以下情况：

(1)(t, p) 是一条 SS-边：我们跳过这条边（请回想我们在 M 中的自底向上半连接归约过程中已经处理过这条边）。（2）(t, p) 是一条 ST-边：我们更新视图 $T_v(p) \leftarrow T_v(p) \ltimes S_v(t)$。如果 $v(t)$中的每个头部变量也都在 $v(p)$ 中，我们就从树中移除 $S_v(t)$。（3）(t, p)是一条 TT-边：我们更新 $T_v(p) \leftarrow T_v(p) \ltimes T_v(t)$。如果 $v(t)$中的每个头部变量也都在 $v(p)$ 中，我们就从树中移除 $T_v(t)$；否则，我们更新 $v(t) \leftarrow v(t) \cap H$以及 $T_v(t) \leftarrow \Pi_{v(t) \cap H}(T_v(t))$.

在自底向上轮次结束时，对于根节点 r，如果 $r \in M$，我们更新 $Q_A \leftarrow Q_A \ltimes S_{v(r)}$；或者如果 $r \notin M$，我们更新 $v(r) \leftarrow v(r) \cap H$，$T_{v\ r} \leftarrow \Pi_{v\ t \cap H}\ T_{v\ r}$ $Q_A \leftarrow Q_A \ltimes T_{v\ r}$ $()\,()(())$ 和 $()$。

现在，在约简树上完成了自底向上的半连接约简。我们在引理 A.2 中证明，该约简树仅包含头部变量。

自顶向下连接轮次。 如果 $r \in M$，我们计算 $Q_A \bowtie S_v\ (r)$；或者如果 $r \notin M$，我们计算 $Q_A \bowtie T_v\ (r)$。接下来，我们在约简树上应用 Yannakakis 的精确自顶向下全连接轮次（从 r 开始，使用父视图探测子视图），以获得输出 ψ。

示例 *A.1.* 我们使用图 5 中所示的 CQAP $\varphi(x_1, x_2, x_3, x_4, x_7, x_8 \mid x_1, x_2)$ 的非冗余 PMTD，其中 (x_1, x_2) 是访问模式。我们使用以下自由连通无环合取查询来演示 Online Yannakakis：

$$\psi(\mathbf{x}_H) \leftarrow Q_{12}(x_1, x_2) \wedge T_{12}(x_1, x_2) \wedge T_{13}(x_1, x_3) \wedge T_{345}(x_3, x_4, x_5) \wedge S_{45}(x_4, x_5) \wedge S_{37}(x_3, x_7) \wedge S_{78}(x_7, x_8),$$

其中 $\mathbf{x}_H = (x_1, x_2, x_3, x_4, x_7, x_8)$。以下是自底向上半连接归约过程中的半连接约简序列（跳过 SS-边（S_{37}, S_{78}））

$$\begin{aligned}
TS\text{-edge } (T_{345}, S_{45}): &\quad T_{345}^{(1)} \leftarrow T_{345} \ltimes S_{45}, \quad \text{remove } S_{45} \\
TS\text{-edge } (T_{13}, S_{37}): &\quad T_{13}^{(1)} \leftarrow T_{13} \ltimes S_{37} \\
TT\text{-edge } (T_{13}, T_{345}): &\quad T_{13}^{(2)} \leftarrow T_{13} \ltimes T_{345}^{(1)}, \quad T_{34}^{(1)} \leftarrow \Pi_{34}(T_{345}^{(1)}) \\
TT\text{-edge } (T_{12}, T_{13}): &\quad T_{12}^{(1)} \leftarrow T_{12} \ltimes T_{13}^{(2)} \\
\text{root}: &\quad Q_A^{(1)} \leftarrow Q_A \ltimes T_{12}^{(1)}
\end{aligned}$$

在自顶向下轮次中，为了获得 ψ 的结果，Online Yannakakis 从根到叶节点依次计算以下连接，从 $Q(1)\ A$ 开始。

$$Q_A^{(1)} \bowtie T_{12}^{(1)} \bowtie T_{13}^{(2)} \bowtie T_{34}^{(1)} \bowtie S_{37} \bowtie S_{78}.$$

引理 A.2。*Online Yannakakis* 的自底向上半连接归约过程之后的约简树仅包含头部变量，即 $\mathbf{x}_H$。

[{"id":0,"output":"证明。显然，约简树中的 T-视图只包含头部变量。因此，我们只需证明每个 S-视图都具有这一性质。根据 $v(\cdot)$ 的定义，对于根 S-视图，或具有父 S-视图的非根 S-视图，这一点显然成立。剩下需要证明的是，对于一条 ST-边（t, p），存在一个头部顶点 $y \in v(t) \setminus\setminus v(p)$，这表明 t 是包含 y 的最顶层包。假设 S-视图 $S_{v(t)}$ 包含某个 $z \notin H$，那么根据 $v(\cdot)$ 的定义，z 必须位于 $v(p) = \chi(p)$ 中。这与自由连通性质相矛盾，因为 $\mathrm{TOP}_r(z)$ 是 $\mathrm{TOP}_r(y)$ 的祖先，即非头部顶点 $z \in v(p)$ 位于头部顶点 $y \in v(t)$ 之上。□"}]

定理 3.7 的证明。定义 $T = \max_{t \in V(\mathcal{T}) \setminus M} |T_{v(t)}|$。自底向上轮次的时间开销为 $O(T + |Q_A|)$，因为只有 T-视图和 Q_A 经过了半连接约简（根据索引构建，S-视图也经过了自底向上的半连接约简）。此外，根据引理 A.2，自底向上轮次之后的约简树仅包含 H 中的变量。因此，从根开始自顶向下连接可避免任何中间悬空元组，时间开销为 $O(|\psi|)$。总时间开销为 $O(T + |Q_A| + |\psi|)$。□

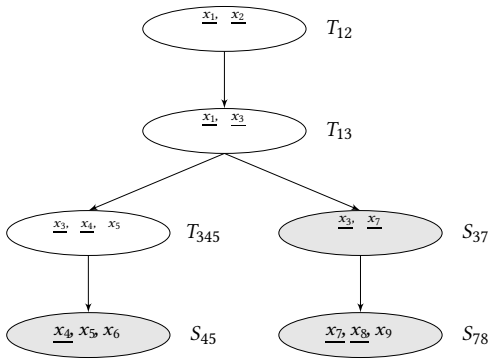


图 5：一个非冗余PMTD，作为OnlineYannakakis的示例。物化集以阴影标示（并标记为 S-视图），头部变量带下划线。

B 第4节中缺失的细节

我们证明，对于任意访问请求 Q_A，一般框架都会返回正确的结果 φ，即 $\varphi = \text{Đ}_{i\in I}\,\psi_i$。由于 PMTD 中的每个视图都经过了半连接约简，显然有 $\text{Đ}_{i\in I}\,\psi_i \subseteq \varphi$。对于反向包含关系，我们参照 [24] 中推论 7.13 的证明，证明以下两个论断。

断言 ***1.***从每条规则 ρ_k 的头部中选出一个目标（S-目标或 T-目标），记该目标为 U_k，其中 $k \in [M]$。我们将由每条规则各选出的一个目标组成的元组 $(U_k)_{k\in[M]}$ 称为完整选择。那么，对于任意完整选择 $(U_k)_{k\in[M]}$，存在一个 PMTD $P \in \{P_i\}_{i\in I}$，使得对于每个树节点 $t \in M$，S-目标 $S_{v(t)}$ 都在该完整选择中，并且对于每个树节点 $t \notin M$，T-目标 $T_{v(t)}$ 都在该完整选择中。任意打破平局后，我们将此 PMTD 称为与完整选择 $(U_k)_{k\in[M]}$ 关联的 PMTD。

命题 1 的证明。固定一个完整选择 $(U_k)_{k\in[M]}$。反设，对于每个 PMTD $P_i \in \mathrm{P}$，都存在一个树节点，其对应的目标（T-目标或 S-目标），记为 U_i^*，不属于该完整选择。接下来考察以 $\hat{\text{O}}_{i\in I}\,U_i^*$ 为头部的两阶段析取规则 ρ^*。根据完整选择的定义，必须从 ρ^* 中选取一个目标。然而，这就产生了矛盾，因为 ρ^* 中的每个头部 U_i^* 都没有出现在固定的完整选择 $(U_k)_{k\in[M]}$ 中。□

命题 ***2***。对于任意完整选择 $\boldsymbol{U}\text{def}=(U_k)_{k\in[M]}$ 及其关联的 PMTD $(\mathrm{T}_i,\ \chi_i,\ M_i,\ r_i)$，我们定义一个合取查询

$$\varphi_{\boldsymbol{U}}(\mathbf{x}_H) \leftarrow Q_A \wedge \bigwedge_{k\in[M]} U_k$$

令 U 表示所有完整选择构成的集合。那么，

$$\varphi \subseteq \bigcup_{\boldsymbol{U}\in\mathcal{U}} \varphi_{\boldsymbol{U}} \subseteq \bigcup_{i\in I} \psi_i \tag{9}$$

命题 2 的证明。取任意输出元组 $\mathbf{a}_H \in \varphi$。那么，存在某个元组 $\mathbf{a}$ 满足 φ 的主体，使得 $\mathbf{a}_H = \Pi_H(\mathbf{a})$。对于每条规则 ρ_k，其中 $k \in [M]$，令 U_k^* 为与视图 v_k^* 相关联的目标（S-目标或 T-目标），使得 $\Pi_{v_k^*}(\mathbf{a}) \in U_k^*$。因此，$\boldsymbol{U}^* = (U_k^*)_{k\in M}$ 是完整选择，并且 $\mathbf{a}$ 满足 $\varphi_{\boldsymbol{U}^*}$ 的主体。于是，$\mathbf{a}_H \in \varphi_{\boldsymbol{U}^*}$，我们证明了 (9) 中的第一个包含关系。第二个包含关系可通过删除 $\varphi_{\boldsymbol{U}^*}$ 主体中不属于 $\boldsymbol{U}^*$ 关联 PMTD 的任何视图的原子而得。□

C 两阶段析取规则的算法

ρ 是一个采用形式（5）的两阶段析取规则，在度约束 DC（由输入关系守卫）以及度约束 AC（由访问请求 Q_A 守卫）下成立。本节将介绍一种使用 PANDA 算法分两个阶段获取 ρ 模型的朴素算法。首先，我们介绍一些必要的术语和结果。

C.1 背景

熵函数。 给定一个析取规则 (4)，若存在定义在 $[n]$ 上的联合概率分布，使得对于任意 $F \subseteq [n]$，$h(F)$ 都是 F 的边缘熵，则称集合函数 $h\colon 2^{[n]} \to \mathbb{R}_+$ 为熵函数。令 Γ_n^* 为所有熵函数构成的集合，并令 $h(Y|X) \stackrel{\text{def}}{=} h(Y) - h(X)$。在给定一组度约束 DC 的情况下，$[n]$ 上的任意联合分布都满足约束 $h(Y|X) \le n_{Y|X}$，其中 $n_{Y|X} \stackrel{\text{def}}{=} \log N_{Y|X}$，对每个 $(X, Y, N_{Y|X}) \in \text{DC}$。

多拟阵。 多拟阵是一个非负、单调且次模的集合函数 $h\colon 2^{[n]} \to \mathbb{R}_+$，其中 $h(\emptyset) = 0$。具体而言，单调性意味着对于任意 $X \subseteq Y \subseteq [n]$，$h(Y) \ge h(X)$；令 $h(Y|X) \stackrel{\text{def}}{=} h(Y) - h(X)$，则次模性意味着对于任意 $I, J \subseteq [n]$，$h(I|I \cap J) \ge h(I \cup J|J)$。令 Γ_n 为定义在 $[n]$ 上的所有多拟阵构成的集合。由于每个熵函数都是多拟阵，因此有 $\Gamma_n^* \subseteq \Gamma_n$。

析取规则的大小界 则。令 ρ 为形如 (4) 的析取规则。令 D 为在给定度约束集合下的数据库实例 ree DC。集合

$$\text{HDC} \stackrel{\text{def}}{=} \left\{ h : 2^{[n]} \to \mathbb{R}_+ \,\middle|\, \bigwedge_{(X,Y,N_{Y|X}) \in \text{DC}} h(Y|X) \le \log N_{Y|X} \right\}$$

包含 $[n]$ 上满足度约束 DC 的所有熵函数 h。固定一个闭子集 $\text{F} \subseteq \mathbb{R}_+^{2^n}$。我们将析取规则 ρ 关于 F 的对数大小界定义为以下量：

$$\text{LogSizeBound}_{\mathcal{F}}(\rho) \stackrel{\text{def}}{=} \max_{h \in \mathcal{F}} \min_{B \in \text{BT}} h(B)$$

那么，对于 ρ 的输出大小，我们有以下定理（见 [24] 中的定理 1 5）：.

定理 C.1（[24]）。令 ρ 为任意 析取规则 (4) 在度约束 DC下 .那么，对于任意满足 DC的数据库实例 $\mathcal{D}$，该以下结论成立：

$$\log |\rho| \le \underbrace{\text{对数大小界}_{\Gamma_n^* \cap \text{HDC}}(\rho)}_{\text{熵界}} \le \underbrace{\text{对数大小界}_{\Gamma_n \cap \text{HDC}}(\rho)}_{\text{多拟阵界}},$$

在最坏情况下，熵界在度约束下是紧的。然而，通常很难计算熵界。若 ρ 是合取查询（即只有一个目标），且只有基数约束，则多拟阵界是紧的；在这种情况下，多拟阵界退化为 AGM 界。然而，在一般的度约束下，多拟阵界并不紧。

该PANDA算法。给定一个形式如 (4) 的析取规则 ρ，PANDA 算法以数据库实例 D 和一组度约束 DC（由 D 守卫）为输入，并计算一个模型，其时间和空间由其多拟阵界预测：

$$\widetilde{O}(2^{\text{LogSizeBound}_{\Gamma_n \cap \text{HDC}}(\rho)}).$$

读者可以参阅 [24] 中的定理 1.7 了解详情。目前，我们将 PANDA 算法视为黑盒；稍后会介绍它的工作原理。

C.2 两阶段框架

本节将介绍一个两阶段算法框架，我们将遵循该框架，为形如 (5) 的两阶段析取规则设计两阶段算法。

设 D 为一个数据库实例，Q_A 为任意访问请求。我们假设，根据框架的需要，可以在预处理阶段开始时预先构建关于必要索引键（输入关系的索引键以及输入关系中元组的度）的哈希表。也就是说，我们假设在两个阶段中，访问输入关系中的元组及其度均可在常数时间内完成。对于每个输入关系 R_F，至多需要预先构建 $O(2^{2n})$ 个哈希表（即，对于每个满足 $X \subset Y \subseteq F \in \text{E}$ 的 (Y, X) 对），因此，在数据复杂度下，存储所有必要哈希表的空间开销为 $O(|\text{D}|)$。回顾一下，我们还不失一般性地假设满足以下最佳约束假设：对于每个 $X \subset Y \subseteq [n]$，我们最多只保留一个 $(X, Y, N_{Y|X}) \in \text{DC}$（如果有多个，则保留 $N_{Y|X}$ 最小的那个）。

拆分步骤。 设 $R \in \text{D}$ 为基数约束 $(\emptyset, Z, N_{Z|\emptyset}) \in \text{DC}$ 的守卫。对于一个 (Y, X)-对（其中 $\emptyset \ne X \subset Y \subseteq Z$），在其上执行的拆分步骤 应用 [24] and 的引理 6.1，并将 $R_Y \stackrel{\text{def}}{=} \Pi_Y(R)$ 划分为 $k = 2 \log N_{Z|\emptyset}$ 个子表，即 $R(1)_Y, \ldots, R(k)_Y$，使得对于所有 $j \in [k]$，$N_{(j)X|\emptyset} \cdot N_{(j)Y|X} \le N_{Z|\emptyset}$，其中

$$N_{X|\emptyset}^{(j)} \stackrel{\text{def}}{=} \left| \Pi_X(R_Y^{(j)}) \right|$$

$$N_{Y|X}^{(j)} \stackrel{\text{def}}{=} \deg_{R_Y^{(j)}}(Y|X).$$

对于这些子表 $R_{(j)Y}$ 中的每一个，我们都创建一个输入为 $(\mathbf{D}_{(j)}, \mathbf{DC}_{(j)})$ 的子问题，其中 $\mathbf{D}_{(j)} \text{def=} \mathbf{D} \cup \{R_{(j)Y}\}$ 表示输入表，而

$$\mathbf{DC}^{(j)} \stackrel{\text{def}}{=} \mathbf{DC} \cup \left\{(\emptyset, X, N_{X|\emptyset}^{(j)}), (X, Y, N_{Y|X}^{(j)})\right\}$$

表示由 $\mathbf{D}_{(j)}$ 守卫的度约束。拆分步骤序列作用于 $(Y_1, X_1), (Y_2, X_2) \ldots, (Y_\ell, X_\ell)$ 上，首先在 (Y_1, X_1)-对上应用第一个分割步骤，生成 $k_1 = O(\log |\mathbf{D}|)$ 个子问题，其输入为 $(\mathbf{D}_{(j)}, \mathbf{DC}_{(j)})$，其中 $j \in [k_1]$。然后，对每个子问题，在 (Y_2, X_2)-对上应用第二个分割步骤，生成 $O((\log |\mathbf{D}|)^2)$ 个子问题。这个迭代过程持续进行，直到序列中的每个分割步骤都应用完毕，因此总共生成 $O(poly(\log |\mathbf{D}|))$ 个子问题。

两阶段框架。 现在，我们对两阶段算法框架进行形式化刻画。设 S 为给定的空间预算。我们将为满足度约束 $\mathbf{DC} \cup \mathbf{AC}$ 的两阶段析取规则 ρ（形式如（5））及输入关系 $\mathbf{D} \cup \{Q_A\}$ 获取模型的任务表示为 $\rho(\mathbf{D} \cup \{Q_A\}, \mathbf{DC} \cup \mathbf{AC})$。该框架首先使用一系列分割步骤，将 $\rho(\mathbf{D} \cup \{Q_A\}, \mathbf{DC} \cup \mathbf{AC})$ 划分为 $O(poly(\log |\mathbf{D}|))$ 个子问题。然后，第 j 个子问题的输入为 $\mathbf{D}_{(j)}$，度约束为 $\mathbf{DC}_{(j)} \supseteq \mathbf{DC}$，记作 $\rho(\mathbf{D}_{(j)} \cup \{Q_A\}, \mathbf{DC}_{(j)} \cup \mathbf{AC})$，它要么

(1) 使用 PANDA，将形如 (6) 的预处理析取规则 ρ_S 生成为 S-目标 $(S_{(j)B})_{B \in \mathbf{BS}}$ 的模型，前提是 ρ_S 的输出大小不超过 $O(S)$；
或者 (2) 使用 PANDA，将形如 (7) 的在线析取规则 ρ_T 生成为 T-目标 $(T_{(j)B})_{B \in \mathbf{BT}}$ 的模型。

换言之，通过对约束为 $\mathbf{DC}_{(j)} \cup \mathbf{AC}$ 的子问题 $\rho(\mathbf{D}_{(j)} \cup \{Q_A\})$ 应用 PANDA，可得到以下二者之一：一个以 $\mathbf{D}_{(j)}$ 为输入关系、满足度约束 $\mathbf{DC}_{(j)}$ 的 ρ_S 模型，记为 $\rho_S(\mathbf{D}_{(j)}, \mathbf{DC}_{(j)})$；或者一个以 $\mathbf{D}_{(j)} \cup \{Q_A\}$ 为输入关系、满足度约束 $\mathbf{DC}_{(j)} \cup \mathbf{AC}$ 的 ρ_T 模型，记为 $\rho_T(\mathbf{D}_{(j)} \cup \{Q_A\}, \mathbf{DC} \cup \mathbf{AC})$。计算完所有子问题后，$\rho$ 的模型就是对所有子问题生成的 S-目标和 T-目标取并集。

两阶段框架的分析。接下来，我们分析由上述两阶段框架所能实现的固有时空权衡（在第 4.1 小节中指定，介于 S_ρ 和 T_ρ 之间）。首先，拆分步骤会在 S_ρ 和 T_ρ 上引入一个多对数因子，并产生 $O(poly(\log |\mathbf{D}|))$ 个子问题。随后，第 j 个产生的子问题 $\rho(\mathbf{D}_{(j)} \cup \{Q_A\}, \mathbf{DC}_{(j)} \cup \mathbf{AC})$ 由 PANDA 在两个阶段之一中解决。请注意，由于拆分步骤，$\mathbf{DC}_{(j)} \supseteq \mathbf{DC}$ 包含额外的度约束。为便于分析，我们通过定义集合 $\mathbf{SC}_{(j)} \text{def=} \mathbf{DC}_{(j)} \setminus\setminus \mathbf{DC}$，将这些额外约束单独列出。为应用多拟阵界，我们使用两个多拟阵，$h_S \in \Gamma_n$ 用于表示预处理阶段，$h_T \in \Gamma_n$ 用于表示在线阶段。我们定义

$$\mathbf{HDC} \stackrel{\text{def}}{=} \left\{ h : 2^{[n]} \to \mathbb{R}_+ \middle| \bigwedge_{(X,Y,N_{Y|X}) \in \mathbf{DC}} h(Y|X) \le \log N_{Y|X} \right\}$$

$$\mathbf{HSC}^{(j)} \stackrel{\text{def}}{=} \left\{ h : 2^{[n]} \to \mathbb{R}_+ \middle| \bigwedge_{(X,Y,N_{Y|X}) \in \mathbf{SC}^{(j)}} h(Y|X) \le \log N_{Y|X} \right\}$$

$$\mathbf{HAC} \stackrel{\text{def}}{=} \left\{ h : 2^{[n]} \to \mathbb{R}_+ \middle| \bigwedge_{(X,Y,N_{Y|X}) \in \mathbf{AC}} h(Y|X) \le \log N_{Y|X} \right\}$$

其中 $h(Y|X) = h(Y) - h(X)$，用于表示分别满足 DC、$\mathbf{SC}_{(j)}$ 和 AC 的集合函数。两阶段框架强制要求 h_S 符合 $\mathbf{HDC} \cap \mathbf{HSC}_{(j)}$ 和 h_T 符合 $\mathbf{HDC} \cap \mathbf{HSC}_{(j)} \cap \mathbf{HAC}$。因此，第 j 个子问题在预处理阶段占用空间 $\mathbf{e}O(S_{(j)}\rho)$，其中

$$\log S_\rho^{(j)} \stackrel{\text{def}}{=} \mathsf{LogSizeBound}_{h_S \in \Gamma_n \cap \mathbf{HDC} \cap \mathbf{HSC}^{(j)}}(\rho_S) \tag{10}$$

前提是 $S_{(j)\rho} \le S$。否则，第 j 个子问题需要耗费时间（和空间）$\mathbf{e}O(T_{(j)\rho})$，其中

$$\log T_\rho^{(j)} \stackrel{\text{def}}{=} \mathsf{LogSizeBound}_{h_T \in \Gamma_n \cap \mathbf{HDC} \cap \mathbf{HSC}^{(j)} \cap \mathbf{HAC}}(\rho_T) \tag{11}$$

通过解决所有子问题，我们得出结论：$S_\rho = \max_j S_{(j)\rho} \le S$ 和 $T_\rho = \max_j T_{(j)\rho}$。

C.3 一种朴素的两阶段算法

本节中，我们使用两阶段框架为两阶段析取规则设计两阶段算法 ρ，使其在固定空间预算 S下达到最小可能的T_ρ。

回顾一下，对于每个 $(\emptyset, Z, N_{Z|\emptyset}) \in$DC，至多存在 2^{2n} (Y, X)-对满足 $\emptyset \neq X \subset Y \subseteq Z$。因此，不同拆分步骤的总数在数据复杂度中是一个常数。为了充分利用拆分步骤的势函数，我们设计了一种朴素的两阶段算法，该算法应用包含所有不同拆分步骤的拆分步骤序列。此外，回顾一下，一个有限的拆分步骤序列会生成 $O(poly(\log |\mathbb{D}|))$ 个子问题。

朴素算法。如前所述，朴素算法首先依次应用所有互不相同的拆分步骤。直观而言，这会将 D 划分为最细粒度的部分。令 ρ (D $(j) \cup \{Q_A\}$, DC$(j) \cup$AC) 为依次应用所有互不相同的拆分步骤后产生的第 j 个子问题。回顾一下，SC(j) = DC$(j) \setminus$ DC。以下拆分性质是对 DC 依次应用所有互不相同的拆分步骤的直接结果：对于任意 $(\emptyset, Z, N_{Z|\emptyset}) \in$DC 和满足 $\emptyset \neq X \subset Y \subseteq Z$ 的 (Y, X)-对，存在一些 $(\emptyset, X, N(j)_{X|\emptyset})$、$(X, Y, N(j)_{Y|X}) \in$SC$(j)$，使得 $N(j)_{X|\emptyset} \cdot N(j)_{Y|X} \leq N_{Z|\emptyset}$。尽管每个子问题各自的 SC$(j)$ 都不同，但拆分性质对所有子问题均成立。为了编码拆分性质，我们定义拆分约束。

定义 C。 $Z \emptyset Z N_{Z|\emptyset} \in 2$ R_F $X Y|X N_{Z|\emptyset}$ $\emptyset \neq X \subset Y \subseteq$（拆分约束）。设 DC 为一组度约束。一个拆分约束是一个三元组（、、），其中、（、、）属于 DC。若一个关系守卫（、、），则称该关系守卫拆分约束（、、）。由 DC 生成的所有拆分约束的集合记为 $X Y|X N_{Z|\emptyset}$ R_F $\emptyset Z N_{Z|\emptyset}$

$$\text{SC} \stackrel{\text{def}}{=} \left\{(X, Y|X, N_{Z|\emptyset}) \mid \emptyset \neq X \subset Y \subseteq Z, (\emptyset, Z, N_{Z|\emptyset}) \in \text{DC}\right\}.$$

直观地说，每个三元组（X、$Y|X$、$N_{Z|\emptyset}$）$\in$SC 都编码了 DC 中（Y、X）对在（$\emptyset$、Z、$N_{Z|\emptyset}$）上的拆分性质。由于我们假设每个（$\emptyset$、Z、$N_{Z|\emptyset}$）$\in$DC 至少有一个守卫，因此每个（X、$Y|X$、$N_{Z|\emptyset}$）$\in$SC 都由某个 $R_F \in$D 守卫。

当其多拟阵界（如式 (10) 所示）不大于 $\log S$ 时，朴素算法会存储 ρ_S(D(j)，DC(j)) 的 S-视图。否则，它会在在线阶段将 PANDA 算法作为黑箱应用于 ρ_T(D$(j) \cup \{Q_A\}$，DC$(j) \cup$AC)，耗时如式 (11) 所示。朴素算法的分析。朴素算法充分利用了该框架的全部潜力（穷尽所有可能的拆分步骤），从而在 $S_\rho \leq S$ 时取得最佳可能的 T_ρ（相差不超过一个多对数因子）。特别地，关于最佳可能的 T_ρ，我们陈述如下定理：

定理 C.3 。对于一个两阶段析取规则 (5)，在空间预算 S 下，朴素算法可获得 $T_\rho = 2^{\text{OBJ}(S)}$，其中 e

$$\begin{aligned} \text{OBJ}(S) = \max_{\substack{h_S \in \text{HDC}, h_T \in \text{HDC} \cap \text{HAC} \\ (h_S, h_T) \in (\Gamma_n \times \Gamma_n) \cap \text{HSC}}} & \min_{B \in \text{BT}} h_T(B) \\ s.t. \quad & h_S(B) > \log S, \quad B \in \text{BS}, \end{aligned} \tag{12}$$

假设 OBJ(S) 为正且有界。

证明。合并式 (10) 和式 (11)，可得第 j 个子问题的 $\log T(j)\rho$ 可表示为

$$\begin{aligned} \log T_\rho^{(j)} = \max_{\substack{h_S \in \text{HDC} \cap \text{HSC}^{(j)} \\ h_T \in \text{HDC} \cap \text{HSC}^{(j)} \cap \text{HAC} \\ h_S, h_T \in \Gamma_n}} & \min_{B \in \text{BT}} h_T(B) \\ s.t. \quad & h_S(B) > \log S, \quad B \in \text{BS}. \end{aligned} \tag{13}$$

请注意，极大极小优化 (13) 依赖于子问题，因为它受限于 HSC(j)。为避免这种依赖，我们回顾一下，朴素算法规定了拆分性质。因此，我们定义如下（与子问题无关的）集合 SC:

$$\text{HSC} \stackrel{\text{def}}{=} \left\{ (h_S, h_T) : 2^{[n]} \times 2^{[n]} \to \mathbb{R}_+^2 \mid \bigwedge_{(X, Y|X, N_{Z|\emptyset}) \in \text{SC}} (h_S(X) + h_T(Y|X) \leq \log N_{Z|\emptyset}) \wedge (h_S(Y|X) + h_T(X) \leq \log N_{Z|\emptyset}) \right\},$$

其中 $h_S(Y|X)_{\text{def}} = h_S(Y) - h_S(X)$ $h_T(Y|X)$ def $= h_T(Y) - h_T(X)$。HSC 是满足拆分性质的集合函数对的通用集合，因此它关联 h_S和 h_T。由于每个子问题都满足拆分性质，因此 HSC(j)×HSC$(j) \subseteq$HSC。将 HSC(j)×HSC(j) 放宽为 HSC，即可得到 T_ρ 所需的上界 (12)。□

通过将h_T 始终设为 0，很容易看出，当且仅当 (12) 的可行域为空时

$$\text{LogSizeBound}_{h_S \in \Gamma_n \cap \text{HDC}}(\rho_S) \leq \log S,$$

在这种情况下，我们只需在空间 e$O(S)$ 内存储 S-视图。否则，(12) 的可行性可以得到保证。然而，朴素算法在实用性方面存在以下缺点：(1) 穷举式拆分步骤可能会导致较大的多对数因子；(2) 对于每个子问题，我们都需要从头开始，使用一个新实例运行 PANDA；(3) 得到的时空权衡难以解释。在接下来的章节中，我们将介绍两阶段 PANDA 算法，称为 2PP。该算法同样能够达到 (12) 中规定的内在权衡，同时高效地解决这两个缺点，并获得更实用、易于解释的内在权衡。

D 2PP 算法

两阶段 PANDA（2PP）算法与 PANDA 类似，建立在一类称为联合香农流不等式的不等式之上。本节首先介绍香农流不等式的背景，然后介绍我们对联合不等式的扩展。最后，我们将介绍 2PP 算法。

D.1 香农流不等式

以下不等式

$$\sum_{X\subset Y\subseteq[n]} \delta_{Y|X}\cdot h(Y|X) \stackrel{\text{def}}{=} \sum_{X\subset Y\subseteq[n]} \delta_{Y|X}\cdot(h(Y)-h(X)) \geq \sum_{\emptyset\neq Z\subseteq[n]} \lambda_{Z|\emptyset}\cdot h(Z|\emptyset), \tag{14}$$

如果对于任意多面体拟阵函数 $h\in\Gamma_n$，且 $\delta_{Y|X}$ 和 $\lambda_{Z|\emptyset}$ 均为某些非负有理系数，即 $\delta_{Y|X},\ \lambda_{Z|\emptyset}\in\mathbb{Q}_+$，该不等式成立，则称其为香农流不等式。为简洁地表示香农流不等式，我们按照 [24] 中的定义引入条件多拟阵。

条件多拟阵。令 $\mathbf{C}\subseteq 2^{[n]}\times 2^{[n]}$ 表示所有满足 $\emptyset\subseteq X\subset Y\subseteq[n]$ 的配对 $(X,\ Y)$ 的集合。向量 $\boldsymbol{h}\in\mathbb{R}^{\mathbf{C}}_+$ 的坐标以配对 $(X,\ Y)\in\mathbf{C}$ 为索引，我们将 $\boldsymbol{h}$ 的相应坐标值记为 $h(Y|X)$。当且仅当存在一个多拟阵 h，使得 $h(Y|X)=h(Y)-h(X)$ 时，向量 $\boldsymbol{h}$ 才称为条件多拟阵；并且我们称多拟阵 h 定义了条件多拟阵 $\boldsymbol{h}$。特别地，$\boldsymbol{h}=(h(Y|X))_{(X,\ Y)\in\mathbf{C}}$。

对于 (14)，若我们对任意 $\emptyset\neq X\subset Z\subseteq[n]$ 定义 $\lambda_{Z|X}=0$，则 $\{\delta_{Y|X}\}$，$\{\lambda_{Y|X}\}$ 中的每个对象都可解释为定义在 $(X,\ Y)$ 配对上的向量，其中 $\emptyset\subseteq X\subset Y\subseteq[n]$。我们分别将它们记为 $\boldsymbol{\delta}$，$\boldsymbol{\lambda}$。因此，香农流不等式 (14) 可改写为关于条件多拟阵的不等式，即 $\mathbf{C}\ \mathbb{Q}^{\mathbf{C}}_+$ 空间上的不等式：$\langle\boldsymbol{\delta},\ \boldsymbol{h}\rangle\geq\langle\boldsymbol{\lambda},\ \boldsymbol{h}\rangle$，其中 $\langle\cdot,\ \cdot\rangle$ 表示点积。

证明序列。 [24] 的一项重要贡献指出，任意香农流不等式 $\langle\boldsymbol{\delta},\boldsymbol{h}\rangle\geq\langle\boldsymbol{\lambda},\boldsymbol{h}\rangle$ 都可以仅通过应用以下 4 条规则来证明：

(R1) 次模规则	$h(I\cup J\|J)-h(I\|I\cap J)\leq 0,\quad I\perp J$
(R2) 单调性规则	$-h(Y\|\emptyset)+h(X\|\emptyset)\leq 0,\quad X\subset Y$
(R3) 复合规则	$h(Y\|\emptyset)-h(Y\|X)-h(X\|\emptyset)\leq 0,\quad X\subset Y$
(R4) 分解规则	$-h(Y\|\emptyset)+h(Y\|X)+h(X\|\emptyset)\leq 0,\quad X\subset Y$

其中，$I\perp J$ 表示 $I\not\subseteq J$ 和 $J\not\subseteq I$。(*R*1) 和 (*R*2) 直接源自多拟阵的次模性和单调性。(*R*3) 和 (*R*4) 则直接由 $h(Y|X)=h(Y)-h(X)$ 的定义得出。所有规则也可以针对所有 (X,Y) 对进行向量化，其中 $\emptyset\subseteq X\subset Y\subseteq[n]$。对于每个 $I\perp J$，我们定义一个向量 $\boldsymbol{s}_{I,J}$；对于每个 $X\subset Y$，我们定义三个向量 $\boldsymbol{m}_{X,Y},\boldsymbol{c}_{X,Y},\boldsymbol{d}_{Y,X}$，使得上述线性规则可以用点积表示：

(R1) 次模规则	$\langle\boldsymbol{s}_{I,J},\boldsymbol{h}\rangle\leq 0,\quad I\perp J$
(R2) 单调性规则	$\langle\boldsymbol{m}_{X,Y},\boldsymbol{h}\rangle\leq 0,\quad X\subset Y$
(R3) 复合规则	$\langle\boldsymbol{c}_{X,Y},\boldsymbol{h}\rangle\leq 0,\quad X\subset Y$
(R4) 分解规则	$\langle\boldsymbol{d}_{Y,X},\boldsymbol{h}\rangle\leq 0,\quad X\subset Y$

一个香农流不等式的证明序列 $\langle\boldsymbol{\delta},\boldsymbol{h}\rangle\geq\langle\boldsymbol{\lambda},\boldsymbol{h}\rangle$ 是一个长度为 ℓ 的序列 $\text{ProofSeq}\stackrel{\text{def}}{=}(w_1\boldsymbol{f}_1,w_2\boldsymbol{f}_2,\ldots,w_\ell\boldsymbol{f}_\ell)$，满足以下所有条件：

(1) $\boldsymbol{f}_i\in\{\boldsymbol{s}_{I,J},\boldsymbol{m}_{X,Y},\boldsymbol{c}_{X,Y},\boldsymbol{d}_{Y,X}\}$ 对每个 $i\in[\ell]$，称为一个证明步骤；(2) 对每个 $i\in[\ell]$，称为证明步骤的权重 $\boldsymbol{f}_i$；(3)向量 $\boldsymbol{\delta}_0\stackrel{\text{def}}{=}\boldsymbol{\delta},\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_\ell$ 由 $\boldsymbol{\delta}_i=\boldsymbol{\delta}_{i-1}+w_i\cdot\boldsymbol{f}_i$ 定义，均为非负有理向量；(4) $\boldsymbol{\delta}_\ell\geq\boldsymbol{\lambda}$（逐元素比较）。

一个证明序列（$w_1\boldsymbol{f}_1,w_2\boldsymbol{f}_2,\ldots,w_\ell\boldsymbol{f}_\ell$）对所有多拟阵 $h\in$ 蕴含以下不等式 Γ_n，

$$\langle\boldsymbol{\delta},\boldsymbol{h}\rangle=\langle\boldsymbol{\delta}_0,\boldsymbol{h}\rangle\geq\cdots\geq\langle\boldsymbol{\delta}_\ell,\boldsymbol{h}\rangle\geq\langle\boldsymbol{\lambda},\boldsymbol{h}\rangle,$$

它为香农流不等式 $\langle\boldsymbol{\delta},\boldsymbol{h}\rangle\geq\langle\boldsymbol{\lambda},\boldsymbol{h}\rangle$ 提供了逐步证明。下述定理（也是 [35] 中的定理 2）是定理 B.12 和命题 B.13 在 [24] 中的直接结果。

定理 D.1（[24]）。对于任意满足 $\|\boldsymbol{\lambda}\|_1=1$ 的香农流不等式 $\langle\boldsymbol{\delta},\boldsymbol{h}\rangle\geq\langle\boldsymbol{\lambda},\boldsymbol{h}\rangle$，都存在一个长度为 $O(poly(2^n))$ 的证明序列。

定理 D.1 意味着，香农流不等式的证明序列长度相对于变量数量 n 呈指数增长，但在查询大小固定时，其长度为常数。

D.2 联合 Shannon 流不等式

为了阐明我们研究联合香农流不等式的动机，我们首先通过以下这类线性规划来刻画极大极小优化问题 (12) 的最优目标值 OBJ(S)

$$\mathbf{L}(\boldsymbol{\lambda}_{\mathsf{BT}}, \boldsymbol{\theta}_{\mathsf{BS}}, S) \stackrel{\text{def}}{=} \max_{\substack{h_S \in \mathsf{HDC}, h_T \in \mathsf{HDC}\cap\mathsf{HAC} \\ (h_S, h_T) \in (\Gamma_n \times \Gamma_n)\cap \mathsf{HSC}}} \sum_{B\in\mathsf{BT}} \lambda_B \cdot h_T(B) + \sum_{B\in\mathsf{BS}} \theta_B \cdot h_S(B) - (\log S)\cdot \|\boldsymbol{\theta}_{\mathsf{BS}}\|_1, \tag{15}$$

以两个向量为参数，$\boldsymbol{\lambda}_{\mathsf{BT}} \text{ def } = (\lambda_B)_{B\in\mathsf{BT}} \in \mathbb{Q}_{\mathsf{BT}+}$，其中 $\|\boldsymbol{\lambda}_{\mathsf{BT}}\|_1 = 1$，以及 $\boldsymbol{\theta}_{\mathsf{BS}} \text{ def } = (\theta_B)_{B\in\mathsf{BS}} \in \mathbb{Q}_{\mathsf{BS}+}$。等价地，任何

$$(\boldsymbol{\lambda}_{\mathsf{BT}}, \boldsymbol{\theta}_{\mathsf{BS}}) \in \left\{ (\mathbf{a}_T, \mathbf{a}_S) \mid \mathbf{a}_T \in \mathbb{Q}_+^{\mathsf{BT}}, \mathbf{a}_S \in \mathbb{Q}_+^{\mathsf{BS}}, \|\mathbf{a}_T\|_1 = 1 \right\} \tag{16}$$

都会导出一个形式为 (15) 的线性规划，其最优目标值为 $\mathbf{L}(\boldsymbol{\lambda}_{\mathsf{BT}}\ \boldsymbol{\theta}_{\mathsf{BS}} S)$。

引理 D.2。设 S 是一个 固定量。对于 $(\boldsymbol{\lambda}_{\mathsf{BT}}\ \boldsymbol{\theta}_{\mathsf{BS}})$ （如(16)所述），(12)的最优目标值 OBJ(S) 满足 ，

$$\mathrm{OBJ}(S) \le \mathbf{L}(\boldsymbol{\lambda}_{\mathsf{BT}}, \boldsymbol{\theta}_{\mathsf{BS}}, S)$$

此外，假设OBJ(S) 为正且有界，则存在一个最优的$(\boldsymbol{\lambda}^*{}_{\mathsf{BT}}\boldsymbol{\theta}^*{}_{\mathsf{BS}})$ 满足(16)，使得 t

$$\mathrm{OBJ}(S) = \mathbf{L}(\boldsymbol{\lambda}^*_{\mathsf{BT}}, \boldsymbol{\theta}^*_{\mathsf{BS}}, S).$$

我们不直接证明引理 D.2，而是证明一个稍微一般化的引理（引理 D.3），它可能本身也具有一定的研究价值。

引理 D. 3 $\boldsymbol{A} \in \mathbb{Q}^{\ell\times m}$ $\boldsymbol{b} \in \mathbb{R}^{\ell}$ $\boldsymbol{D} \in \mathbb{Q}_+^{m\times q}$ $\boldsymbol{C} \in \mathbb{Q}_+^{m\times p}$ $\boldsymbol{c}_1 \ \cdots \ \boldsymbol{c}_p$ $P =$ $\in \mathbb{R}^m \mid \boldsymbol{A}$ $\le \boldsymbol{b}$ $\ge$. 设 , , , 是一个矩阵，$\mathbf{0}$ w^* 其列为 , , , 且多面体为$\{\mathbf{x}\,\mathbf{x},\mathbf{x}\}$。设 为以下优化问题的最优目标值（假设 是固定量）

$$\begin{aligned} \max_{\mathbf{x}\in P} \quad & \min_{k\in[p]} \boldsymbol{c}_k^\top \mathbf{x} \\ s.t. \quad & \boldsymbol{D}^\top \mathbf{x} \ge \mathbf{1}_q \log S. \end{aligned} \tag{17}$$

若 w^* 为正且有界，则对于任意满足$\|\mathbf{z}\|_1 = 1$的向量$\mathbf{z}$、$\mathbf{u} \in \mathbb{Q}_+^p$，以下线性规划 ：

$$L(\mathbf{z}, \mathbf{u}) \stackrel{def}{=} \max_{\mathbf{x}\in P} \quad (\boldsymbol{C}\mathbf{z})^\top \mathbf{x} + (\boldsymbol{D}^\top \mathbf{x} - \mathbf{1}_q \log S)^\top \mathbf{u} \tag{18}$$

满足 $w^* \le L(\mathbf{z},\ \mathbf{u})$。特别地，存在一对向量 $\mathbf{z}^* \in \mathbb{Q}_+^p$、$\mathbf{u}^* \in \mathbb{Q}_+^q$，满足 $\|\mathbf{z}^*\|_1 = 1$，使得 $w^* = L(\mathbf{z}^*、\ \mathbf{u}^*)$。

证明。首先，我们将$\mathbf{u} \in \mathbb{Q}_+^q$ 引入作为（17）的拉格朗日乘子，并得到

$$\begin{aligned} w^* &\le \max_{\mathbf{x}\in P} \min_{\mathbf{u}\in\mathbb{Q}_+^q} \min_{k\in[p]} \left(\boldsymbol{c}_k^\top \mathbf{x} + (\boldsymbol{D}^\top \mathbf{x} - \mathbf{1}_q \log S)^\top \mathbf{u} \right) \\ &\le \max_{\mathbf{x}\in P} \min_{\mathbf{u}\in\mathbb{Q}_+^q, \mathbf{z}\in\mathbb{Q}_+^p, \|\mathbf{z}\|_1=1} (\boldsymbol{C}\mathbf{z})^\top \mathbf{x} + (\boldsymbol{D}^\top \mathbf{x} - \mathbf{1}_q \log S)^\top \mathbf{u} \\ &\le \min_{\mathbf{u}\in\mathbb{Q}_+^q, \mathbf{z}\in\mathbb{Q}_+^p, \|\mathbf{z}\|_1=1} \max_{\mathbf{x}\in P} (\boldsymbol{C}\mathbf{z})^\top \mathbf{x} + (\boldsymbol{D}^\top \mathbf{x} - \mathbf{1}_q \log S)^\top \mathbf{u} \\ &= \min_{\mathbf{u}\in\mathbb{Q}_+^q, \mathbf{z}\in\mathbb{Q}_+^p, \|\mathbf{z}\|_1=1} L(\mathbf{z}, \mathbf{u}) \end{aligned}$$

其中，第三个等号是由极小极大不等式得出的。因此，我们已证明，对于任意向量 $\mathbf{u} \in \mathbb{Q}_+^q$,$\mathbf{z} \in \mathbb{Q}_+^p$，若 $\|\mathbf{z}\|_1 = 1$,$L(\mathbf{z}, \mathbf{u}) \ge w^*$。接下来，我们证明存在向量 $\mathbf{u}^* \in \mathbb{Q}_+^q, \mathbf{z}^* \in \mathbb{Q}_+^p$，使得 $\|\mathbf{z}^*\|_1 = 1$ 且 $L(\mathbf{z}^*,\mathbf{u}^*) = w^*$。我们首先将 (12) 重写为以下等价线性规划：

$$\begin{aligned} \max_{\mathbf{x}, w} \quad & w \\ \text{s.t.} \quad & \boldsymbol{A}\mathbf{x} \le \boldsymbol{b} \\ & \boldsymbol{D}^\top \mathbf{x} \ge \mathbf{1}_q \log S \\ & \boldsymbol{C}^\top \mathbf{x} \ge \mathbf{1}_p w \\ & \mathbf{x} \ge \mathbf{0}, w \ge 0 \end{aligned} \tag{19}$$

设 $(w^*,\ \mathbf{x}^*)$ 是 (19) 的一个最优解。那么，(19) 的对偶问题可写为以下线性规划 ：

$$\begin{aligned} \min_{\mathbf{y},\mathbf{u},\mathbf{z}} \quad & \boldsymbol{b}^\top \mathbf{y} - (\mathbf{1}_q \log S)^\top \mathbf{u} \\ \text{s.t.} \quad & \boldsymbol{A}^\top \mathbf{y} - \boldsymbol{D}\mathbf{u} - \boldsymbol{C}\mathbf{z} \ge \mathbf{0} \\ & \mathbf{1}_p^\top \mathbf{z} \ge 1 \\ & \mathbf{y}, \mathbf{u}, \mathbf{z} \ge \mathbf{0} \end{aligned} \tag{20}$$

设 $(\mathbf{y}^*, \mathbf{u}^*, \mathbf{z}^*)$ 是（有理）对偶多面体的一个极点，且该极点取得 (20) 的最优目标值，因此 $\mathbf{z}^* \in \mathbb{Q}_+^p$, $\mathbf{u}^* \in \mathbb{Q}_+^q$。由 (19) 和 (20) 这一原始-对偶对的互补松弛条件以及假设 $w^* > 0$，可推出 $\mathbf{1}_p^\top \mathbf{z}^* = \mathbf{k}\mathbf{z}^*\mathbf{k}_1 = 1$、$(\mathbf{1}_p w^* - \boldsymbol{C}^\top \mathbf{x}^*)^\top \mathbf{z}^* = 0$，以及 $(\boldsymbol{D}^\top \mathbf{x}^* - \mathbf{1}_q \log S)^\top \mathbf{u}^* = 0$。接下来我们证明 $L(\mathbf{z}^*, \mathbf{u}^*) = w^*$。首先注意，$\mathbf{x}^*$ 是 (18) 的一个可行解，其目标值为 $(\boldsymbol{C}\mathbf{z}^*)^\top \mathbf{x}^* + (\boldsymbol{D}^\top \mathbf{x}^* - \mathbf{1}_q \log S)^\top \mathbf{u}^* = (\boldsymbol{C}\mathbf{z}^*)^\top \mathbf{x}^* = (\boldsymbol{C}^\top \mathbf{x}^*)^\top \mathbf{z}^* = (\mathbf{1}_p w^*)^\top \mathbf{z}^* = w^*$。此外，对于 (18) 的任意可行解 $\mathbf{x}$，我们有

$$
\begin{aligned}
(\boldsymbol{C}\mathbf{z}^*)^\top \mathbf{x} + (\boldsymbol{D}^\top \mathbf{x} - \mathbf{1}_q \log S)^\top \mathbf{u}^* &= (\boldsymbol{C}\mathbf{z}^*)^\top \mathbf{x} + (\boldsymbol{D}\mathbf{u}^*)^\top \mathbf{x} - (\mathbf{1}_q \log S)^\top \mathbf{u}^* && \text{重新排列} \\
&\le (\boldsymbol{A}^\top \mathbf{y}^*)^\top \mathbf{x} - (\mathbf{1}_q \log S)^\top \mathbf{u}^* && \text{对偶问题可行性和非负性} \\
&= (\boldsymbol{A}\mathbf{x})^\top \mathbf{y}^* - (\mathbf{1}_q \log S)^\top \mathbf{u}^* && \\
&\le \boldsymbol{b}^\top \mathbf{y}^* - (\mathbf{1}_q \log S)^\top \mathbf{u}^* && \text{原始可行性} \\
&= w^* && \text{强对偶性}
\end{aligned}
$$

这意味着 $L(\mathbf{z}^*, \mathbf{u}^*) \le w^*$。综合可得 $L(\mathbf{z}^*, \mathbf{u}^*) = w^*$。 □

现在，将 $\boldsymbol{\lambda}_{\text{BT}}$ 设为 $\mathbf{z}^*$，将 $\boldsymbol{\theta}_{\text{BS}}$ 设为 $\mathbf{u}^*$，引理 D.2 便可直接由引理 D.3 推出。

在本节余下部分中，我们默认假设 S 是一个固定量。此外，我们固定一对满足式 (16) 的 $(\boldsymbol{\lambda}_{\text{BT}}\ \boldsymbol{\theta}_{\text{BS}})$。现在可以重写式 (15)，列出其中所有约束，并忽略常数因子 $(\log S) \cdot \mathbf{k}\boldsymbol{\theta}_{\text{BS}}\mathbf{k}1$：

$$
\begin{aligned}
\ell(\boldsymbol{\lambda}_{\text{BT}}, \boldsymbol{\theta}_{\text{BS}}) \stackrel{\text{def}}{=} \max_{h_S, h_T} \quad & \sum_{B \in \text{BT}} \lambda_B \cdot h_T(B) + \sum_{B \in \text{BS}} \theta_B \cdot h_S(B) & & \\
\text{s.t.} \quad & h_S(Y) - h_S(X) \le n_{Y|X}, & & (X, Y, N_{Y|X}) \in \text{DC} \\
& h_T(Y) - h_T(X) \le n_{Y|X}, & & (X, Y, N_{Y|X}) \in \text{DC} \cup \text{AC} \\
& h_S(I \cup J | J) - h_S(I | I \cap J) \le 0, & & I \perp J \\
& h_T(I \cup J | J) - h_T(I | I \cap J) \le 0, & & I \perp J \\
& h_S(X) - h_S(Y) \le 0, & & \emptyset \ne X \subset Y \subseteq [n] \\
& h_T(X) - h_T(Y) \le 0, & & \emptyset \ne X \subset Y \subseteq [n] \\
& h_S(X) + h_T(Y|X) \le n_{Z|\emptyset}, & & (X, Y|X, N_{Z|\emptyset}) \in \text{SC} \\
& h_S(Y|X) + h_T(X) \le n_{Z|\emptyset}, & & (X, Y|X, N_{Z|\emptyset}) \in \text{SC} \\
& h_S(Z) \ge 0, \quad h_T(Z) \ge 0, & & \emptyset \ne Z \subseteq [n]
\end{aligned}
\tag{21}
$$

回顾一下，我们隐含地有 $h_S(\emptyset) = h_T(\emptyset) = 0$，并且 $n_{Y|X} = \log N_{Y|X}$。接下来写出式 (21) 的对偶线性规划。对于每个 $(X, Y, N_{Y|X}) \in \text{DC}$，我们为 $h_S(Y) - h_S(X) \le n_{Y|X}$ 关联一个对偶变量 $(\delta_S)_{Y|X}$；对于每个 $(X, Y, N_{Y|X}) \in \text{DC} \cup \text{AC}$，则为 $h_T(Y) - h_T(X) \le n_{Y|X}$ 关联一个对偶变量 $(\delta_T)_{Y|X}$。对于每个 $I \perp J$，其中 $I, J \subseteq [n]$，我们为 h_S 的次模约束关联一个对偶变量 $(\sigma_S)_{I,J}$，并为 h_T 的次模约束关联一个 $(\sigma_T)_{I,J}$。对于每个 $\emptyset \ne X \subset Y \subseteq [n]$，我们为 h_S 的单调性约束关联一个对偶变量 $(\mu_S)_{X,Y}$，并为 h_T 的单调性约束关联一个对偶变量 $(\mu_T)_{X,Y}$。最后，对于每个 $(X, Y|X, N_{Z|\emptyset}) \in \text{SC}$，我们为 $h_S(X) + h_T(Y) - h_T(X) \le N_{Z|\emptyset}$ 关联一个对偶变量 $\gamma_{X,Y|X}$，并为 $h_S(Y) - h_S(X) + h_T(X) \le N_{Z|\emptyset}$ 关联一个对偶变量 $\gamma_{Y|X,X}$。此外，我们以显然的方式将向量 $(\lambda_B)_{B \in \text{BT}}$ 和 $(\theta_B)_{B \in \text{BS}}$ 扩展到每个 $Z \in 2^{[n]}$：

$$
\lambda_Z \stackrel{\text{def}}{=} \begin{cases} \lambda_B & \text{when } Z = B \in \text{BT} \\ 0 & \text{otherwise} \end{cases} \qquad \theta_Z \stackrel{\text{def}}{=} \begin{cases} \theta_B & \text{when } Z = B \in \text{BS} \\ 0 & \text{otherwise} \end{cases}
$$

滥用记号时，只要 $(X, Y) \in \text{DC}$ 成立，我们就写作 $(X, Y) \in \text{DC}$；只要 $(X, Y, N_{Y|X}) \in \text{DC}$ 成立。同样，只要 $(X, Y, N_{Z|\emptyset}) \in \text{SC}$ 成立，我们就写作 $(X, Y|X) \in \text{SC}$。注意，借助最佳约束假设，我们总能从给定的 (Y, X)-对中恢复唯一的 $N_{Y|X}$ 或 $N_{Z|\emptyset}$。(21) 的对偶问题现在可以写为

$$
\begin{aligned}
\min \quad & \sum_{(X,Y) \in \text{DC}} n_{Y|X} \cdot (\delta_S)_{Y|X} + \sum_{(X,Y) \in \text{DC} \cup \text{AC}} n_{Y|X} \cdot (\delta_T)_{Y|X} \\
& + \sum_{(X, Y|X) \in \text{SC}} n_{Z|\emptyset} \cdot (\gamma_{X,Y|X} + \gamma_{Y|X,X}) \\
\text{s.t.} \quad & \text{inflow}_S(Z) \ge \theta_Z & & \emptyset \ne Z \subseteq [n] \\
& \text{inflow}_T(Z) \ge \lambda_Z & & \emptyset \ne Z \subseteq [n] \\
& (\delta_S)_{Y|X}, (\mu_S)_{X,Y}, (\sigma_S)_{I,J} \ge 0 & & \\
& (\delta_T)_{Y|X}, (\mu_T)_{X,Y}, (\sigma_T)_{I,J} \ge 0 & & \\
& \gamma_{X,Y|X}, \gamma_{Y|X,X} \ge 0 & &
\end{aligned}
\tag{22}
$$

其中，对每个 $\emptyset \neq Z \subseteq [n]$,

$$\begin{aligned}\text{inflow}_S(Z) \stackrel{\text{def}}{=} &\left(\sum_{X:(X,Z)\in \text{DC}} (\delta_S)_{Z|X} - \sum_{Y:(Z,Y)\in \text{DC}} (\delta_S)_{Y|Z}\right) + \left(-\sum_{X:X\subset Z} (\mu_S)_{X,Z} + \sum_{Y:Z\subset Y} (\mu_S)_{Z,Y}\right)\\ &+\left(\sum_{\substack{I\perp J\\ I\cap J=Z}} (\sigma_S)_{I,J} + \sum_{\substack{I\perp J\\ I\cup J=Z}} (\sigma_S)_{I,J} - \sum_{J:J\perp Z} (\sigma_S)_{Z,J}\right)\\ &+\left(\sum_{\substack{Z:(Z,Y|Z)\in \text{SC}\\ Z\subset Y}} \gamma_{Z,Y|Z} - \sum_{\substack{Z:(Z,Y|Z)\in \text{SC}\\ Z\subset Y}} \gamma_{Y|Z,Z} + \sum_{\substack{Z:(X,Z|X)\in \text{SC}\\ X\subset Z}} \gamma_{Z|X,X}\right)\end{aligned}$$

$$\begin{aligned}\text{inflow}_T(Z) \stackrel{\text{def}}{=} &\left(\sum_{X:(X,Z)\in \text{DC}\cup\text{AC}} (\delta_T)_{Z|X} - \sum_{Y:(Z,Y)\in \text{DC}\cup\text{AC}} (\delta_T)_{Y|Z}\right) + \left(-\sum_{X:X\subset Z} (\mu_T)_{X,Z} + \sum_{Y:Z\subset Y} (\mu_T)_{Z,Y}\right)\\ &+\left(\sum_{\substack{I\perp J\\ I\cap J=Z}} (\sigma_T)_{I,J} + \sum_{\substack{I\perp J\\ I\cup J=Z}} (\sigma_T)_{I,J} - \sum_{J:J\perp Z} (\sigma_T)_{Z,J}\right)\\ &+\left(\sum_{\substack{Z:(Z,Y|Z)\in \text{SC}\\ Z\subset Y}} \gamma_{Y|Z,Z} - \sum_{\substack{Z:(Z,Y|Z)\in \text{SC}\\ Z\subset Y}} \gamma_{Z,Y|Z} + \sum_{\substack{Z:(X,Z|X)\in \text{SC}\\ X\subset Z}} \gamma_{X,Z|X}\right)\end{aligned}$$

接下来，我们介绍联合香农流不等式。

定义 *D.4*（联合香农流不等式）。该不等式

$$\begin{aligned}\sum_{(X,Y)\in \text{DC}} h_S(Y|X)\cdot(\delta_S)_{Y|X} + \sum_{(X,Y)\in \text{DC}\cup\text{AC}} h_T(Y|X)\cdot(\delta_T)_{Y|X} + \sum_{(X,Y|X)\in \text{SC}} (h_S(X)+h_T(Y|X))\cdot\gamma_{X,Y|X}\\ + \sum_{(X,Y|X)\in \text{SC}} (h_S(Y|X)+h_T(X))\cdot\gamma_{Y|X,X} \geq \sum_{B\in \text{BS}} \theta_B\cdot h_S(B) + \sum_{B\in \text{BT}} \lambda_B\cdot h_T(B),\end{aligned} \tag{23}$$

被称为一个联合香农流不等式如果它对所有$(h_S, h_T) \in \Gamma_n \times \Gamma_n$且所有系数都是非负有理数。

通过将任一多拟阵指定为始终为 0，上述不等式可推出以下两个香农流不等式，称为参与型香农流不等式。

$$\sum_{(X,Y)\in \text{DC}} h_S(Y|X)\cdot(\delta_S)_{Y|X} + \sum_{(X,Y|X)\in \text{SC}} h_S(X)\cdot\gamma_{X,Y|X} + \sum_{(X,Y|X)\in \text{SC}} h_S(Y|X)\cdot\gamma_{Y|X,X} \geq \sum_{B\in \text{BS}} \theta_B\cdot h_S(B) \tag{24}$$

$$\sum_{(X,Y)\in \text{DC}\cup\text{AC}} h_T(Y|X)\cdot(\delta_T)_{Y|X} + \sum_{(X,Y|X)\in \text{SC}} h_T(Y|X)\cdot\gamma_{X,Y|X} + \sum_{(X,Y|X)\in \text{SC}} h_T(X)\cdot\gamma_{Y|X,X} \geq \sum_{B\in \text{BT}} \lambda_B\cdot h_T(B) \tag{25}$$

将联合香农流不等式写成关于条件多拟阵的不等式会很方便。多拟阵 h_S定义了一个条件多拟阵 $\boldsymbol{h}_S$，而多拟阵 h_T定义了一个条件多拟阵 $\boldsymbol{h}_T$。更确切地说，我们定义向量 $\boldsymbol{\lambda}$，$\boldsymbol{\theta} \in \mathbb{Q}^{\mathcal{C}+}$（扩展到 (X, Y) 对，其中 $\emptyset \subseteq X \subset Y \subseteq [n]$），并按如下方式为坐标赋值：

$$\lambda(Y|X) \stackrel{\text{def}}{=} \begin{cases}\lambda_B & \text{when } Y=B, X=\emptyset\\ 0 & \text{otherwise}\end{cases} \qquad \theta(Y|X) \stackrel{\text{def}}{=} \begin{cases}\theta_B & \text{when } Y=B, X=\emptyset\\ 0 & \text{otherwise}\end{cases}$$

类似地，我们定义向量 $\boldsymbol{\delta}_S, \boldsymbol{\delta}_T \in \mathbb{Q}^{\mathcal{C}+}$，其坐标值为：

$$\delta_S(Y|X) \stackrel{\text{def}}{=} \begin{cases}(\delta_S)_{Y|X} & \text{when } (X,Y)\in \text{DC}\\ 0 & \text{otherwise}\end{cases} \qquad \delta_T(Y|X) \stackrel{\text{def}}{=} \begin{cases}(\delta_T)_{Y|X} & \text{when } (X,Y)\in \text{DC}\cup\text{AC}\\ 0 & \text{otherwise}\end{cases}$$

最后，系数 $\{\gamma_{X,Y|X} \mid (X,Y|X) \in \mathrm{SC}\} \cup \{\gamma_{Y|X,X} \mid (X,Y|X) \in \mathrm{SC}\}$ 会同时对不等式（24）和（25）的系数产生贡献。我们将一对向量 $\boldsymbol{\gamma}_S, \boldsymbol{\gamma}_T \in \mathbb{Q}C+$ 定义如下：

$$\gamma_S(Y|X) \stackrel{\text{def}}{=} \begin{cases} \gamma_{U,V|U} & \text{when } Y = U, X = \emptyset, (U,V|U) \in \mathrm{SC} \\ \gamma_{V|U,U} & \text{when } Y = V, X = U, (U,V|U) \in \mathrm{SC} \\ 0 & \text{otherwise} \end{cases}$$

$$\gamma_T(Y|X) \stackrel{\text{def}}{=} \begin{cases} \gamma_{U,V|U} & \text{when } Y = V, X = U, (U,V|U) \in \mathrm{SC} \\ \gamma_{V|U,U} & \text{when } Y = U, X = \emptyset, (U,V|U) \in \mathrm{SC} \\ 0 & \text{otherwise} \end{cases}$$

用于跟踪对 $\boldsymbol{h}_S$ 的贡献，$\boldsymbol{h}_T$，分别。令 $\boldsymbol{g}_S \mathrm{def}= \boldsymbol{\delta}_S + \boldsymbol{\gamma}_S$ 和 $\boldsymbol{g}_T \mathrm{def}= \boldsymbol{\delta}_T + \boldsymbol{\gamma}_T$，则可以用点积重写不等式 (24) 和 (25)：

$$\langle \boldsymbol{g}_S, \boldsymbol{h}_S \rangle \stackrel{\text{def}}{=} \langle \boldsymbol{\delta}_S, \boldsymbol{h}_S \rangle + \langle \boldsymbol{\gamma}_S, \boldsymbol{h}_S \rangle \geq \langle \boldsymbol{\theta}, \boldsymbol{h}_S \rangle \tag{26}$$

$$\langle \boldsymbol{g}_T, \boldsymbol{h}_T \rangle \stackrel{\text{def}}{=} \langle \boldsymbol{\delta}_T, \boldsymbol{h}_T \rangle + \langle \boldsymbol{\gamma}_T, \boldsymbol{h}_T \rangle \geq \langle \boldsymbol{\lambda}, \boldsymbol{h}_T \rangle \tag{27}$$

此外，联合香农流不等式 (23) 可以写为

$$\langle \boldsymbol{g}_S, \boldsymbol{h}_S \rangle + \langle \boldsymbol{g}_T, \boldsymbol{h}_T \rangle \geq \langle \boldsymbol{\theta}, \boldsymbol{h}_S \rangle + \langle \boldsymbol{\lambda}, \boldsymbol{h}_T \rangle \tag{28}$$

定理 D.5。 不等式 (28) 是联合香农流不等式，当且仅当存在非负有理数向量 $\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T$，使对偶问题 (22) 的所有约束均得到满足。特别地，我们称 $(\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T)$ 为该联合香农流不等式的见证。

证明。 [24] 中的命题 5.6 指出：(26) 是香农流不等式，当且仅当存在一个 $(\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S) \geq \mathbf{0}$（在[24]中称为见证），使得 $(\boldsymbol{g}_S, \boldsymbol{\sigma}_S, \boldsymbol{\mu}_S)$ 满足以下约束集：

$$\{\mathsf{inflow}_S(Z) \geq \theta_Z \mid \emptyset \neq Z \subseteq [n]\};$$

同样，(27) 是香农流不等式 不等式，当且仅当存在一个（见证）$(\boldsymbol{\sigma}_T, \boldsymbol{\mu}_T) \geq \mathbf{0}$ 使得 $(\boldsymbol{g}_T, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T)$ 满足这些 t of 约束：

$$\{\mathsf{inflow}_T(Z) \geq \lambda_Z \mid \emptyset \neq Z \subseteq [n]\}.$$

回顾 (22) 的表述，这两组约束恰好构成了 (22) 中的所有约束。因此，(28) 是联合香农流不等式，当且仅当存在一个 $(\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T) \geq \mathbf{0}$ 使得 $(\boldsymbol{g}_S, \boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{g}_T, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T)$ 满足对偶问题 (22) 的所有约束。□

定理 D.5 意味着，对偶问题 (22) 的一个可行解，特别是其中的分量

$$\{(\delta_S)_{Y|X} \mid (X,Y) \in \mathrm{DC}\} \cup \{(\delta_T)_{Y|X} \mid (X,Y) \in \mathrm{DC}\} \cup \{\gamma_{X,Y|X} \mid (X,Y|X) \in \mathrm{SC}\} \cup \{\gamma_{Y|X,X} \mid (X,Y|X) \in \mathrm{SC}\}$$

与 $(\boldsymbol{\lambda}_{\mathrm{BT}}\ \theta_{\mathrm{BS}})$ 结合，定义了一个联合香农流不等式 (23)。根据定理 D.5，对偶问题的分量 $(\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T)$ 是该联合香农流不等式的一个见证。注意，联合香农流不等式意味着

$$\langle \boldsymbol{\theta}, \boldsymbol{h}_S \rangle + \langle \boldsymbol{\lambda}, \boldsymbol{h}_T \rangle \leq \sum_{(X,Y) \in \mathrm{DC}} n_{Y|X} \cdot (\delta_S)_{Y|X} + \sum_{(X,Y) \in \mathrm{DC} \cup \mathrm{AC}} n_{Y|X} \cdot (\delta_T)_{Y|X} + \sum_{(X,Y|X) \in \mathrm{SC}} n_{Z|\emptyset} \cdot (\gamma_{X,Y|X} + \gamma_{Y|X,X})$$

其中，右侧恰好是 $\ell(\boldsymbol{\lambda}_{\mathrm{BT}}\ \boldsymbol{\theta}_{\mathrm{BS}})$，这是通过取对偶问题 (22) 的最优解得到的（根据强对偶性）。我们已经证明：对于任意满足 (16) 的 $(\boldsymbol{\lambda}_{\mathrm{BT}}\ \boldsymbol{\theta}_{\mathrm{BS}})$，都可以构造一个联合香农流不等式，

$$\langle \boldsymbol{g}_S, \boldsymbol{h}_S \rangle + \langle \boldsymbol{g}_T, \boldsymbol{h}_T \rangle \geq \langle \boldsymbol{\theta}, \boldsymbol{h}_S \rangle + \langle \boldsymbol{\lambda}, \boldsymbol{h}_T \rangle,$$

它有一个见证 $(\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T)$，其蕴含的上界与 $\ell(\boldsymbol{\lambda}_{\mathrm{BT}}\ \boldsymbol{\theta}_{\mathrm{BS}})$ 一致。

D.3 PANDA 算法的简要回顾与增强

本节简要回顾 PANDA 算法 . 设 ρ 是度约束 DC 下的析取规则 (4)。在较高层次上 el, PANDA 执行以下操作：

步骤 1. 找到一个非负有理数向量 $\boldsymbol{\lambda}_{\mathrm{BT}} = (\lambda_B)_{B \in \mathrm{BT}}$，使得 $\mathrm{k}\boldsymbol{\lambda}_{\mathrm{BT}}\mathrm{k}_1 = 1$；将其扩展为一个向量（定义在条件多拟阵上）$\boldsymbol{\lambda} \in \mathbb{Q}C+$和对数大小界$\Gamma_n \cap \mathrm{HDC}(\rho) = \max_{h \in \Gamma_n \cap \mathrm{HDC}} \mathrm{h}\boldsymbol{\lambda}, \boldsymbol{h}\mathrm{i}$，其中右侧是一个线性规划；步骤 2. 求出该线性规划的一个最优对偶解 $(\boldsymbol{\delta}, \boldsymbol{\sigma}, \boldsymbol{\mu})$，$\max_{h \in \Gamma_n \cap \mathrm{HDC}} \mathrm{h}\boldsymbol{\lambda}, \boldsymbol{h}\mathrm{i}$，使得

$$\mathrm{OBJ} \stackrel{\text{def}}{=} \sum_{(X,Y) \in \mathrm{DC}} \log N_{Y|X} \cdot \delta_{Y|X} = \mathrm{LogSizeBound}_{\Gamma_n \cap \mathrm{HDC}}(\rho)$$

且 $\mathrm{h}\boldsymbol{\delta}, \boldsymbol{h}\mathrm{i} \geq \mathrm{h}\boldsymbol{\lambda}, \boldsymbol{h}\mathrm{i}$ 构成一个香农流不等式；

第 3 步：构造一个长度为 $O(poly(2^n))$ 的证明序列 ProofSeq，用于香农流不等式 $\mathrm{h}\boldsymbol{\delta},\ \boldsymbol{h}\mathrm{i} \geq \mathrm{h}\boldsymbol{\lambda},\ \boldsymbol{h}\mathrm{i}$（参见定理 D.1）；第 4 步：运行一个 PANDA 实例，记作 PANDA(D，DC，$(\boldsymbol{\lambda},\ \boldsymbol{\delta})$，ProofSeq)，它将 ProofSeq 中的每个证明步骤解释为对输入关系执行的关系运算，其中每个运算的运行时间保证为 $\mathrm{e}O(2^{\mathrm{OBJ}})$。总体而言，该 PANDA 实例的运行时间为 $\mathrm{e}O(2^{\mathrm{OBJ}})$，并计算出一个大小为 $\mathrm{e}O(2^{\mathrm{OBJ}})$ 的模型。

特别地，PANDA 实例在其输入上维护以下 4 个不变量，即 (D，DC，$(\boldsymbol{\lambda},\ \boldsymbol{\delta})$，ProofSeq)：

PANDA 不变量。

(1) 度支持不变量：对于每个 $\delta_{Y|X} > 0$，存在 $Z \subseteq X,\ W \subseteq Y$ 使得 $W - Z = Y - X$ 且 $(Z,\ W,\ N_{W|Z}) \in \mathrm{DC}$。该度约束被称为支持正 $\delta_{Y|X}$。如果有多个 $(Z,\ W,\ N_{W|Z}) \in \mathrm{DC}$ 支持 $\delta_{Y|X}$，我们选择其中 $N_{W|Z}$ 最小的一个，并将其称为 $\delta_{Y|X}$ 的支持约束。(2) $0 < \mathrm{k}\boldsymbol{\lambda}\mathrm{k}_1 \leq 1$

(3) Shannon 流不等式与支持性的度约束满足 $\acute{\mathrm{I}}_{(X,Y)}\ n(\delta_{Y|X}) \leq \mathrm{k}\boldsymbol{\lambda}\mathrm{k}1 \cdot \mathrm{OBJ}$，其中

$$n(\delta_{Y|X}) \stackrel{\text{def}}{=} \begin{cases} \delta_{Y|X} \cdot n_{W|Z} & \text{if } \delta_{Y|X} > 0 \text{ and} \\ & (Z, W, N_{W|Z}) \text{ supports it} \\ 0 & \text{if } \delta_{Y|X} = 0. \end{cases}$$

并将量 $\acute{\mathrm{I}}_{(X,Y)}\ n(\delta_{Y|X})$ 称为势函数。

(4) 对于每个 $\delta_{Y|\emptyset} > 0$，支撑度约束 $(\emptyset, Y, N_{Y|\emptyset})$ 满足 $n_{Y|\emptyset} \leq \mathrm{OBJ}$。

对PANDA的轻微增强。

就我们的目的而言，我们对 PANDA 做了轻微增强，使其能够接受非最优证明序列。更准确地说，假设我们得到的不是步骤 1 和 2，而是向量 $(\boldsymbol{\lambda}_{\mathrm{BT}}\ \boldsymbol{\delta}_{\mathrm{DC}})$ 和见证 $(\boldsymbol{\sigma}, \boldsymbol{\mu})$，并且将两个向量分别扩展为 $\boldsymbol{\lambda}, \boldsymbol{\delta} \in \mathbb{Q}_{\mathrm{C}+}$ 和 $\mathrm{h}\boldsymbol{\delta}, \boldsymbol{h}\mathrm{i}$ 后，$\geq \mathrm{h}\boldsymbol{\lambda}, \boldsymbol{h}\mathrm{i}$ 构成一个带有见证 $(\boldsymbol{\sigma}, \boldsymbol{\mu})$ 的香农流不等式（不一定是最优的）。注意，所蕴含的上界 OBJ 满足

$$\mathsf{LogSizeBound}_{\Gamma_n \cap \mathsf{HDC}}(\rho) \leq \mathrm{OBJ} \stackrel{\text{def}}{=} \sum_{(X,Y) \in \mathrm{DC}} \log N_{Y|X} \cdot \delta_{Y|X},$$

但不一定与 $\mathsf{LogSizeBound}_{\Gamma_n \cap \mathsf{HDC}}(\rho)$ 一致。在这种情况下，按照步骤 3 和 4，我们证明，以给定香农流不等式的非最优证明序列为输入的 PANDA 实例，其运行时间符合该非最优香农流不等式所预示的界，即 $\mathrm{e}O(2^{\mathrm{OBJ}})$。

增强如下。我们注意到，在 PANDA [24], 正确性证明中，只有不变量 (4)（位于 PANDA 实例的开头）依赖于证明序列的最优性。然而，正如命题 6.2 中所论证的，如果一开始对于某个 $\delta_{Y|\emptyset} > 0, n_{Y|\emptyset} > \mathrm{OBJ}$，那么我们可以将原始的香农流不等式替换为新的香农流不等式 $\mathrm{h}\boldsymbol{\delta}\prime, \boldsymbol{h}\mathrm{i} \geq \mathrm{h}\boldsymbol{\lambda}\prime, \boldsymbol{h}\mathrm{i}$，以及见证 $(\boldsymbol{\sigma}\prime, \boldsymbol{\mu}\prime\})$，使得不变量 (1)-(3) 成立，并且证明序列的长度至少减少 1。我们可以反复迭代这一替换步骤，直到不变量 (4) 得到满足。如果不变量 (4) 始终未得到满足，那么最终将得到一个长度为 0 的证明序列；此时，香农流不等式变成一个平凡的不等式 $\mathrm{h}\boldsymbol{\delta}_0, \boldsymbol{h}\mathrm{i} \geq \mathrm{h}\boldsymbol{\lambda}_0, \boldsymbol{h}\mathrm{i}$，对于某个 $\boldsymbol{\lambda}_0, \boldsymbol{\delta}_0$，其中 $0 < \mathrm{k}\boldsymbol{\lambda}_0\mathrm{k}_1 \leq 1$（根据不变量 (2)）且 $\boldsymbol{\delta}_0 \geq \boldsymbol{\lambda}_0$（逐元素比较）。这意味着，任何满足 $R_F \text{where } (\lambda_0)_{F|\emptyset} > 0$ 的输入关系都可以作为模型，并且存在一个 $n_{F|\emptyset} \leq \mathrm{OBJ}$ 可以被指定为该输出模型（因此模型大小为 $\mathrm{e}O(2^{\mathrm{OBJ}})$），否则，

$$\sum_{F:(\lambda_0)_{F|\emptyset}>0} \frac{(\lambda_0)_{F|\emptyset}}{\|\boldsymbol{\lambda}_0\|_1} \cdot n_{F|\emptyset} > \sum_{F:(\lambda_0)_{F|\emptyset}>0} \frac{(\lambda_0)_{F|\emptyset}}{\|\boldsymbol{\lambda}_0\|_1} \cdot \mathrm{OBJ} = \mathrm{OBJ},$$

这与不变量 (3) 矛盾。对于 2PP 算法，我们隐含地假设 PANDA 配备了这一小幅增强。

D.4 算法

虽然 2PP 遵循与朴素算法相似的结构（并使用两阶段算法框架），但它受联合 Shannon 流不等式的指导，仅执行必要的拆分步骤和 PANDA 实例，从而提供了更高的实用性和可解释性。特别地，我们将为 2PP 证明以下定理。

定理 D.6. 令 ρ 为一种形式为(5)、满足度约束DC （由输入关系守卫）和AC （由访问请求守卫）的两阶段析取规则。令 $(\boldsymbol{\lambda}_{\mathrm{BT}}\ \boldsymbol{\theta}_{\mathrm{BS}}) \in \left\{(\mathbf{a}_T, \mathbf{a}_S)\ |\mathbf{a}_T \in \mathbb{Q}_{\mathrm{BT}+}, \mathbf{a}_S \in \mathbb{Q}_{\mathrm{BS}+}, \|\mathbf{a}_T\|_1 = 1\right\}$。2PP 算法分两个阶段获得 ρ 的模型，并实现以下（平滑的）内在权衡：

$$S_\rho^{\|\boldsymbol{\theta}_{\mathrm{BS}}\|_1} \cdot T_\rho \cong 2^{\ell(\boldsymbol{\lambda}_{\mathrm{BT}}, \boldsymbol{\theta}_{\mathrm{BS}})} \tag{29}$$

其中 $\cong$ 在右侧隐藏了一个多对数因子。

注意，内在权衡也可以等价地写为

$$S_\rho^{\|\boldsymbol{\theta}_{\mathrm{BS}}\|_1} \cdot T_\rho \cong \prod_{(X,Y) \in \mathrm{DC}} N_{Y|X}^{(\delta_S)_{Y|X}} \cdot \prod_{(X,Y) \in \mathrm{DC} \cup \mathrm{AC}} N_{Y|X}^{(\delta_S)_{Y|X}} \cdot \prod_{(X,Y|X) \in \mathrm{SC}} N_{Z|\emptyset}^{\gamma_{X,Y|X} + \gamma_{Y|X,X}} \tag{30}$$

算法1： 2PP-预处理(D，DC，AC，(θ，eg_S)，ProofSeq(S))

输入：数据库实例 D，以及由 D 守卫的度约束 DC 输入：由任意可能的访问请求 Q_Ae 守卫的度约束 AC 输入：参与的香农流不等式 he$\boldsymbol{g}_S, \boldsymbol{h}_S$i ≥ h$\boldsymbol{\theta}, \boldsymbol{h}_S$i 及其证明序列 ProofSeq($S$) 1 令 SC 为由 DC 张成的拆分约束集合 2 SC$^+$ ← $\{(Y,X) \mid (X, Y|X, N_{Z|\emptyset}) \in \text{SC} \gamma_{X,Y|X} > 0 \vee \gamma_{Y|X,X} > 0\}$3 对每个 $(Y,X) \in$SC$^+$ 应用一系列拆分步骤，并生成 k 个子问题，其输入为 (D (j) , DC^j)，$j \in [k]$// $k = O(poly(\log |\text{D}|))$ 4 J ← ∅5 对所有 $j \in [k]$ 执行 6 创建一个 PANDA 实例 PANDA(D (j), DC(j), (e$\boldsymbol{\theta}$, e$\boldsymbol{g}_S$), ProofSeq(S)) 7 若势函数满足 (32)，则 8 ($S(j)$ B)$B\in$BS ← PANDA(D (j), DC(j), (e$\boldsymbol{\theta}$, e$\boldsymbol{g}_S$), ProofSeq(S)) 9 否则 10 终止 PANDA(D (j), DC(j), (e$\boldsymbol{\theta}$, e$\boldsymbol{g}_S$), ProofSeq(S)) 11 将 (D (j), DC(j)) 插入 J 12 返回 $\left(\text{J}, (\text{Đ}_{j} S(j)\ B)_{B\in\text{BS}}\right)$

具体而言，给定固定的 $S$for S_ρ，我们可以利用引理 D.2（通过互补松弛）构造出 ($\boldsymbol{\lambda}^*$BT $\boldsymbol{\theta}^*$BS)，使得 OBJ(S) =L($\boldsymbol{\lambda}^*$BT $\boldsymbol{\theta}^*$BS S)。由于 $\ell(\boldsymbol{\lambda}^*$BT $\boldsymbol{\theta}^*$BS) −(log S) · k$\boldsymbol{\theta}^*$BSk$_1$ =L($\boldsymbol{\lambda}^*$BT $\boldsymbol{\theta}^*$BS S)，定理 D.6 得出了式 (12) 所述的最佳内在权衡。在本节余下部分，我们将介绍 2PP 算法，并将定理 D.6 的完整证明留到下一节。

准备阶段。与 PANDA 类似，2PP 也有一个准备阶段，用于构造 2PP 实例所需的输入。首先，我们根据给定的 ($\boldsymbol{\lambda}$BT $\boldsymbol{\theta}$BS) 构造一个联合香农流不等式，h$\boldsymbol{g}_S, \boldsymbol{h}_S$i +h$\boldsymbol{g}_T, \boldsymbol{h}_T$i ≥h$\boldsymbol{\theta}, \boldsymbol{h}_S$i +h$\boldsymbol{\lambda}, \boldsymbol{h}_T$i，以及该不等式的一个见证 ($\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T$)。回想一下，这个联合香农流不等式蕴含的上界与 $\ell(\boldsymbol{\lambda}$BT $\boldsymbol{\theta}$BS) 一致。

其次，我们为这个联合香农流不等式构造一个证明序列。其思路是为其中参与的两个香农流不等式分别构造两个并行的证明序列，再将它们拼接起来。回想一下，这两个参与型香农流不等式为

$$\langle \boldsymbol{g}_S, \boldsymbol{h}_S \rangle \geq \langle \boldsymbol{\theta}, \boldsymbol{h}_S \rangle$$
$$\langle \boldsymbol{g}_T, \boldsymbol{h}_T \rangle \geq \langle \boldsymbol{\lambda}, \boldsymbol{h}_T \rangle$$

根据定理 D.5 的证明，($\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S$) 是 h$\boldsymbol{g}_S, \boldsymbol{h}_S$i ≥h$\boldsymbol{\theta}, \boldsymbol{h}_S$i 的一个见证，而 ($\boldsymbol{\sigma}_T, \boldsymbol{\mu}_T$) 是 h$\boldsymbol{g}_T, \boldsymbol{h}_T$i ≥h$\boldsymbol{\lambda}, \boldsymbol{h}_T$i 的一个见证。我们将香农流不等式 h$\boldsymbol{g}_S, \boldsymbol{h}_S$i ≥h$\boldsymbol{\theta}, \boldsymbol{h}_S$i 归一化为 he$\boldsymbol{g}_S, \boldsymbol{h}_S$i ≥he$\boldsymbol{\theta}, \boldsymbol{h}_S$i，其中 e$\boldsymbol{g}_S$def= $\boldsymbol{g}_S$/k$\boldsymbol{\theta}$k1，$\boldsymbol{\theta}$def= $\boldsymbol{\theta}$/k$\boldsymbol{\theta}$k1，并采用新的见证 (e$\boldsymbol{\sigma}_S$, e$\boldsymbol{\mu}_S$) def =($\boldsymbol{\sigma}_S$/k$\boldsymbol{\theta}$k1, $\boldsymbol{\mu}_S$/k$\boldsymbol{\theta}$k1)。根据定理 D.1，我们可以为香农流不等式 ehe$\boldsymbol{g}_S, \boldsymbol{h}_S$i ≥h$\boldsymbol{\theta}, \boldsymbol{h}_S$i 构造一个证明序列 ProofSeq($S$)，并为 h$\boldsymbol{g}_T, \boldsymbol{h}_T$i ≥h$\boldsymbol{\lambda}, \boldsymbol{h}_T$i 构造一个证明序列 ProofSeq($T$)；这两个证明序列的长度均为 $O(poly(2^n))$，其中 n 是变量的数量。我们称 ProofSeq(S) 和 ProofSeq(T) 为该联合香农流不等式的参与型证明序列。简要来说，在准备阶段，2PP：

(1)（见定理D.5）构造一个联合香农流不等式h$\boldsymbol{g}_S, \boldsymbol{h}_S$i +h$\boldsymbol{g}_T, \boldsymbol{h}_T$i ≥h$\boldsymbol{\theta}, \boldsymbol{h}_S$i +h$\boldsymbol{\lambda}, \boldsymbol{h}_T$i，并带有一个见证（$\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T$）；以及
(2)（见定理 D.1）为参与的香农流不等式 he$\boldsymbol{g}_S, \boldsymbol{h}_S$i ≥he$\boldsymbol{\theta}, \boldsymbol{h}_S$i 构造一个 ProofSeq($S$)，并为参与的香农流不等式 h$\boldsymbol{g}_T, \boldsymbol{h}_T$i ≥ h$\boldsymbol{\lambda}, \boldsymbol{h}_T$i 构造一个 ProofSeq($T$)

现在，我们将详细介绍 2PP 的两个阶段。具体而言，我们将 2PP-预处理称为 2PP 的预处理阶段，并将2PP-Online 称为 2PP的在线阶段。2PP-预处理和2PP-Online 的概要分别见算法 1 和算法 2 的方框中。

预处理阶段。我们将此阶段称为 2PP-预处理，其概要见算法 1 的方框。我们首先遍历联合香农流不等式 h$\boldsymbol{g}_S, \boldsymbol{h}_S$i +h$\boldsymbol{g}_T, \boldsymbol{h}_T$i ≥ h$\boldsymbol{\theta}, \boldsymbol{h}_S$i +h$\boldsymbol{\lambda}, \boldsymbol{h}_T$i 和 ($\boldsymbol{\sigma}_S, \boldsymbol{\mu}_S, \boldsymbol{\sigma}_T, \boldsymbol{\mu}_T$)，并应用一系列拆分步骤；对于每个满足 $(X, Y|X, N_{Z|\emptyset}) \in$SC 且满足 $\gamma_{X,Y|X} > 0$或 $\gamma_{Y|X, X} > 0$的 (Y,X)，该序列都包含一个 (Y,X)-对。这一拆分步骤序列会生成 $poly(\log |\text{D}|)$ 个子问题。令 ρ(D (j) ∪{Q_A}, DC(j) ∪ AC) 为拆分步骤序列之后的第 j 个子问题，其度束 DC(j) 由 D (j) 守卫。接下来，我们为它创建一个 PANDA 实例 PANDA(D (j), DC(j), (e$\boldsymbol{\theta}$,e$\boldsymbol{g}_S$), ProofSeq(S))，并考察其初始势

$$\sum_{(X,Y)} n^{(j)}_{W|Z} \cdot (\widetilde{g}_S)_{Y|X} \stackrel{\text{def}}{=} \sum_{(X,Y)\in\text{DC}} n^{(j)}_{W|Z} \cdot \frac{(\delta_S)_{Y|X}}{\|\boldsymbol{\theta}\|_1} + \sum_{(X,Y|X)\in\text{SC}} n^{(j)}_{X|\emptyset} \cdot \frac{\gamma_{X,Y|X}}{\|\boldsymbol{\theta}\|_1} + \sum_{(X,Y|X)\in\text{SC}} n^{(j)}_{W|Z} \cdot \frac{\gamma_{Y|X,X}}{\|\boldsymbol{\theta}\|_1} \tag{31}$$

算法 2： 2PP-Online(J，$\{Q_A\}$，AC，$(\boldsymbol{\lambda}, \boldsymbol{g}_T)$，ProofSeq($T$))

输入：一个索引 J，包含 $O(poly(\log |\mathbb{D}|))$ 个条目，其中每个条目包含输入关系 D (j) 和由 D (j) 守卫的度约束 DC(j) 输入：一个访问请求 Q_A，以及由 Q_A 守卫的度约束 AC 输入：参与其中的香农流不等式 $\mathrm{h}\boldsymbol{g}_T, \boldsymbol{h}_T\mathrm{i} \geq \mathrm{h}\boldsymbol{\lambda}, \boldsymbol{h}_T\mathrm{i}$ 及其证明序列 ProofSeq(T) 1 对所有 (D (j), DC(j)) ∈ J 执行 2 创建一个 PANDA 实例 PANDA(D (j) ∪ $\{Q_A\}$, DC(j) ∪ AC, $(\boldsymbol{\lambda}, \boldsymbol{g}_T)$, ProofSeq($T$)) 3 $(T(j)_B)_{B\in\mathrm{BT}} \leftarrow$ PANDA(D (j) ∪ $\{Q_A\}$, DC(j) ∪ AC, $(\boldsymbol{\lambda}, \boldsymbol{g}_T)$, ProofSeq($T$)) 4 返回 $(\text{Đ}_j T(j)_B)_{B\in\mathrm{BT}}$

其中 $n_{(j)W|Z} \stackrel{\text{def}}{=} \log N_{(j)W|Z}$ and $(Z, W, N_{(j)W|Z}) \in \mathrm{DC}(j)$ </style}是支持正值 $(\mathrm{e}g_S)_{Y|X}$ 的约束。如果势函数满足

$$\sum_{(X,Y)} n^{(j)}_{W|Z} \cdot (\widetilde{g}_S)_{Y|X} \leq \log S, \tag{32}$$

然后，2PP 允许这个 PANDA 实例运行，并存储其输出 $(S_{(j)B})_{B\in\mathrm{BS}}$。否则，2PP 中止这个 PANDA 实例，并使用索引 J 记录第 j 个子问题的输入 (D $_{(j)}$，DC $_{(j)}$)。预处理阶段存储的数据结构是用于跟踪已中止实例的输入（及其度约束）的索引 J，以及来自所有成功 PANDA 实例的一组表 $S_B = \text{Đ}_j S_{(j)B}$，$B \in \mathrm{BS}$。

在线阶段。我们将此阶段称为 2PP-Online，其流程概述于算法 2 的方框中。该算法扫描预处理阶段构建的索引 J，并针对每个 $(\mathbb{D}_{(j)}, \mathrm{DC}_{(j)}) \in \mathrm{J}$，创建并运行一个 PANDA 实例 PANDA($\mathbb{D}_{(j)} \cup \{Q_A\}$，$\mathrm{DC}_{(j)} \cup \mathrm{AC}$，$(\boldsymbol{\lambda}, \boldsymbol{g}_T)$，ProofSeq($T$))。每个 PANDA 实例终止时，2PP 会收集输出 $(T(j)_B)_{B\in\mathrm{BT}}$。在线阶段的总体输出是在线创建并运行的所有 PANDA 实例所产生的表集合 $T_B = \text{Đ}_j T_{(j)B}$，$B \in \mathrm{BT}$。

D.5 2PP算法的分析

本节余下部分将正式证明 2PP 算法的定理 D.6。

定理 D.6 的证明。回想一下，2PP 初始阶段的拆分步骤会产生 $O(poly(\log |\mathbb{D}|))$ 个子问题。我们通过分析第 j 个子问题的空间和时间开销来进行证明。

首先，在预处理阶段，如果PANDA实例PANDA(D (j) , DC(j), $(\mathrm{e}\theta, \mathrm{e}g_S)$,ProofSeq($S$)) 的势函数不大于$\log S$，那么根据 PANDA的不变量 (4)，输出表 $(S(j)_B)_{B\in\mathrm{BT}}$（以及该PANDA 实例生成的每个中间视图）的大小为$\mathrm{e}O(S)$。否则，第 j 个子问题需要占用空间 $O(|\mathbb{D}|)$，用于将其输入 (D (j) , DC(j))存储在索引中。因此，该数据结构的总体空间消耗为 $O(poly(\log |\mathbb{D}|)) \cdot \mathrm{e}O(|\mathbb{D}| + S) = \mathrm{e}O(S)$。

接下来，我们需要证明，对于委派到在线阶段的任意子问题，由 2PP 创建的 PANDA 实例 PANDA($\mathbb{D}_{(j)} \cup \{Q_A\}, \mathrm{DC}_{(j)} \cup \mathrm{AC}, (\boldsymbol{\lambda}, \boldsymbol{g}_T), \mathrm{ProofSeq}(T)$)，会在 $\mathrm{e}O(T_\rho)$ 时间内终止，其中

$$\log T_\rho = \ell(\boldsymbol{\lambda}_{\mathrm{BT}}, \boldsymbol{\theta}_{\mathrm{BS}}) - \log S \cdot \|\boldsymbol{\theta}_{\mathrm{BS}}\|_1.$$

回顾 $n_{Y|X} \stackrel{\text{def}}{=} \log N_{Y|X}$，根据强对偶性，

$$\ell(\boldsymbol{\lambda}_{\mathrm{BT}}, \boldsymbol{\theta}_{\mathrm{BS}}) = \sum_{(X,Y)\in\mathrm{DC}} n_{Y|X} \cdot (\delta_S)_{Y|X} + \sum_{(X,Y)\in\mathrm{DC}\cup\mathrm{AC}} n_{Y|X} \cdot (\delta_S)_{Y|X} + \sum_{(X,Y|X)\in\mathrm{SC}} n_{Z|\emptyset} \cdot (\gamma_{X,Y|X} + \gamma_{Y|X,X}).$$

为证明这一点，我们考察第 j 个子问题对应的两个 PANDA 实例：$\rho(\mathbb{D}_{(j)} \cup \{Q_A\}$，$\mathrm{DC}_{(j)} \cup \mathrm{AC})$，即

$$\textit{(preprocessing instance)} \qquad \mathrm{PANDA}(\mathcal{D}^{(j)}, \mathrm{DC}^{(j)}, (\widetilde{\boldsymbol{\theta}}, \widetilde{\boldsymbol{g}}_S), \mathrm{ProofSeq}(S))$$

$$\textit{(online instance)} \qquad \mathrm{PANDA}(\mathcal{D}^{(j)} \cup \{Q_A\}, \mathrm{DC}^{(j)} \cup \mathrm{AC}, (\boldsymbol{\lambda}, \boldsymbol{g}_T), \mathrm{ProofSeq}(T))$$

预处理实例的势函数如式（31）所示，而在线实例的势函数如下：

$$\sum_{(X,Y)} n^{(j)}_{W|Z} \cdot (g_T)_{Y|X} \stackrel{\text{def}}{=} \sum_{(X,Y)\in\mathrm{DC}\cup\mathrm{AC}} n^{(j)}_{W|Z} \cdot (\delta_T)_{Y|X} + \sum_{(X,Y|X)\in\mathrm{SC}} n^{(j)}_{W|Z} \cdot \gamma_{X,Y|X} + \sum_{(X,Y|X)\in\mathrm{SC}} n^{(j)}_{X|\emptyset} \cdot \gamma_{Y|X,X}$$

其中 $n_{(j)W|Z} \stackrel{\text{def}}{=} \log N_{(j)W|Z}$ and $(Z, W, N_{(j)W|Z}) \in \mathrm{DC}(j)$ 是支持正 $(g_T)_{Y|X}$ 的约束。现在我们有，

$$\sum_{(X,Y)} n^{(j)}_{W|Z} \cdot (g_T)_{Y|X} + \|\boldsymbol{\theta}\|_1 \cdot \sum_{(X,Y)} n^{(j)}_{W|Z} \cdot (\widetilde{g}_S)_{Y|X} = \sum_{(X,Y)\in\mathrm{DC}} n^{(j)}_{W|Z} \cdot (\delta_S)_{Y|X} + \sum_{(X,Y)\in\mathrm{DC}\cup\mathrm{AC}} n^{(j)}_{W|Z} \cdot (\delta_T)_{Y|X}$$

$$+\sum_{(X,Y|X)\in \mathsf{SC}} (n^{(j)}_{X|\emptyset}+n^{(j)}_{W|Z})\cdot \gamma_{X,Y|X}+\sum_{(X,Y|X)\in \mathsf{SC}} (n^{(j)}_{W|Z}+n^{(j)}_{X|\emptyset})\cdot \gamma_{Y|X,X}$$

回顾一下，对于满足 $(X,Y|X,N_{Z|\emptyset}) \in \mathsf{SC}$ and $\gamma_{X,Y|X}>0 \vee \gamma_{Y|X,X}>0$的每个 (Y,X)-pair，2PP 都会执行一个分割步骤。因此，对于每个这样的 (X,Y)-pair，都存在某些 $(\emptyset,X,N(j)_{X|\emptyset})$、$(X,Y,N(j)_{Y|X}) \in \mathsf{DC}(j)$，使得 $N(j)_{X|\emptyset}\cdot N(j)_{Y|X} \le N_{Z|\emptyset}$。这意味着 $n(j)_{X|\emptyset}+n(j)_{W|Z} \le \log N_{Z|\emptyset}$。此外，对于每个 $(X,Y) \in \mathsf{DC}$，其中$(\delta_S)_{Y|X}>0$ or $(\delta_T)_{Y|X}>0$，有 $n(j)_{W|Z} \le \log N_{Y|X}$。因此，我们得到

$$\sum_{(X,Y)} n^{(j)}_{W|Z}\cdot (g_T)_{Y|X}+\|\boldsymbol{\theta}\|_1\cdot \sum_{(X,Y)} n^{(j)}_{W|Z}\cdot (\widetilde{g}_S)_{Y|X} \le \ell(\boldsymbol{\lambda}_{\mathsf{BT}},\boldsymbol{\theta}_{\mathsf{BS}})$$

这意味着，对于第 j 个子问题（因此对于任意子问题），要么在预处理阶段，

$$\sum_{(X,Y)} n^{(j)}_{W|Z}\cdot (g_T)_{Y|X} \le \log S,$$

或在在线阶段

$$\begin{aligned}\sum_{(X,Y)} n^{(j)}_{W|Z}\cdot (g_T)_{Y|X} &\le \ell(\boldsymbol{\lambda}_{\mathsf{BT}},\boldsymbol{\theta}_{\mathsf{BS}})-\|\boldsymbol{\theta}\|_1\cdot \sum_{(X,Y)} n^{(j)}_{W|Z}\cdot (\widetilde{g}_S)_{Y|X}\\ &\le \ell(\boldsymbol{\lambda}_{\mathsf{BT}},\boldsymbol{\theta}_{\mathsf{BS}})-\|\boldsymbol{\theta}_{\mathsf{BS}}\|_1\cdot \log S\\ &= \log T_\rho.\end{aligned}$$

因此，所有在线 PANDA 实例都会在式 (30) 所预测的时间内终止。

□

E 第6节缺失的详细内容

E.1 通过分数边覆盖实现的权衡（第6.2节）

以下引理是 Shearer 引理（ [29] 中的引理 D.1）的一般化。

引理 E.1。 设 $\mathcal{H}$ = （[n]、E） 是一个超图，且$\hat{\mathbf{u}}$ 是 [n] \ $A \subseteq [n]$ 的分数边覆盖。则，

$$\sum_{F\in\mathcal{E}} \hat{u}_F \cdot h(F \mid A \cap F) + h(A) \geq h([n])$$

证明。不妨设 $A = \{1、\ldots\ldots、\ \ell - 1\}$。于是我们可以写为：

$$\begin{aligned}
h([n]) &= \sum_{j=\ell}^{n} h(j \mid i : i < j) + h(A) \\
&\leq \sum_{j=\ell}^{n} \sum_{F\in\mathcal{E}: j\in F} \hat{u}_F \cdot h(j \mid i : i < j, i \in F) + h(A) \\
&= \sum_{F\in\mathcal{E}} \hat{u}_F \sum_{j\in F\setminus A} h(j \mid i : i < j, i \in F) + h(A) \\
&= \sum_{F\in\mathcal{E}} \hat{u}_F \left(h(F) - \sum_{j\in F, j<\ell} h(j \mid i : i < j, i \in F) \right) + h(A) \\
&= \sum_{F\in\mathcal{E}} \hat{u}_F \left(h(F) - \sum_{j\in F\cap A} h(j \mid i : i < j, i \in F \cap A) \right) + h(A) \\
&= \sum_{F\in\mathcal{E}} \hat{u}_F \left(h(F) - h(A \cap F) \right) + h(A) \\
&= \sum_{F\in\mathcal{E}} \hat{u}_F \cdot h(F \mid A \cap F) + h(A)
\end{aligned}$$

至此，证明完成。 □

定理 6.1 的证明。为得到所需的权衡，我们考虑两个 PMTD。第一个 PMTD P_1 有一个节点 t，其中包含 $\chi_1(t) = [n]$ 和 $M_1 = \emptyset$；第二个 PMTD P_2 也有一个包 t，其中包含 $\chi_2(t) = [n]$ 和 $M_2 = \{t\}$。 P_1 包含 T-视图 $T_{[n]}(\mathbf{x}^{[n]})$，而 P_2 包含 S-视图 $S_A(\mathbf{x}_A)$。这两个 PMTD 对应以下物化策略：直接存储访问请求的答案，或者从头计算该访问请求。因此，我们只需考虑一条析取规则（我们用 $\mathbf{x}$ 表示元组 $\mathbf{x}_{12\ldots n}$）：

$$T_{[n]}(\mathbf{x}) \vee S_A(\mathbf{x}_A) \leftarrow Q_A(\mathbf{x}_A) \wedge \bigwedge_{F\in\mathcal{E}} R_F(\mathbf{x}_F).$$

定义 $\alpha = \alpha(\mathbf{u}\ A)$ 和 $\hat{\mathbf{u}} = \mathbf{u}/\alpha$。现在我们可以写出如下证明：

$$\begin{aligned}
\sum_{F\in\mathcal{E}} u_F \cdot \log N_{F|\emptyset} + \alpha \cdot \log |Q_A| &\geq \sum_{F\in\mathcal{E}} u_F \cdot \{h_T(F \mid A \cap F) + h_S(A \cap F)\} + \alpha \cdot h_T(A) \\
&= \sum_{F\in\mathcal{E}} u_F \cdot h_T(F \mid A \cap F) + \alpha \cdot h_T(A) + \sum_{F\in\mathcal{E}} u_F \cdot h_S(A \cap F) \\
&\geq \sum_{F\in\mathcal{E}} u_F \cdot h_T(F \mid A \cap F) + \alpha \cdot h_T(A) + h_S(A) && \textit{(Shearer's Lemma)} \\
&= \alpha \sum_{F\in\mathcal{E}} \hat{u}_F \cdot h_T(F \mid A \cap F) + \alpha \cdot h_T(A) + h_S(A) && \textit{(Lemma E.1)} \\
&\geq \alpha \cdot h_T([n]) + h_S(A)
\end{aligned}$$

第二个不等式是将 Shearer 引理直接应用于 H 的子超图（$A, \{A \cap F \mid F \in$ E$\}$）得到的，因为 $\mathbf{u}$ 是 A 的分数边覆盖。最后一个不等式直接由引理 E.1 得出。根据定理 D.6，我们得到所需的权衡。□

E.2 通过树分解实现的权衡（第 6.3 节）

令 $\varphi(\mathbf{x}_A \mid \mathbf{x}_A)$ 为一个 CQAP。根据第 6.3 节，令 P 为由诱导出的 PMTD 集合生成的所有两阶段析取规则的集合。我们首先通过证明以下结论来展开分析：对于该集合中的任意析取规则 ρ_a，都存在另一个析取规则 ρ_b，就其内在权衡而言不比 ρ_a 容易。有趣的是，尽管选择了一个（可能）更难的规则，我们仍能恢复许多最先进的权衡。我们先陈述两个关键观察。

观察 E.1。对于任意两阶段析取规则 ρ_a 和 ρ_b，ρ_a 被称为不比 ρ_b 更难（或等价地，ρ_b 不比 ρ_a 更容易），如果 ρ_b 的 S-目标是 ρ_a 的 S-目标的子集，并且 ρ_b 的 T-目标是 ρ_a 的 T-目标的子集。

换句话说，观察E.1 表明，向析取规则的头部添加更多目标只会使其模型评估变得更容易，因为我们始终可以忽略新目标。下一个引理利用了 PMTD 中 T-视图的结构。

引理 E.2。设 $\varphi(\mathbf{x}_A \mid \mathbf{x}_A)$ 是给定的合取查询。设 $(\mathcal{T}\ \chi, r)$ 是固定的自由连通树分解。设 $\mathcal{P}$ 是由 $(\mathcal{T}\ \chi, r)$ 诱导出的 *PMTD* 的集合。设 ρ_a 是由 $\mathcal{P}$ 生成的两阶段析取规则。那么，存在一个两阶段析取规则 ρ_b，其难度不低于 ρ_a，且对于 ρ_b 的任意两个 T-目标，它们在 $\mathcal{T}$ 中对应的节点不同时位于某条根到叶路径上。

证明。令 $\mathrm{BT}(\rho)$ 为一个集合，使得对于每个 $B \in \mathrm{BT}(\rho)$，存在一个由两阶段析取规则 ρ 选取的 T-目标 T_B（同样地，我们为 S-视图定义 $\mathrm{BS}(\rho)$）。注意，根据视图的定义，节点的一个 T-视图意味着其祖先节点都与 T-视图相关联。

我们按如下方式构造两阶段析取规则 ρ_b：对于每个 PMTD，如果 ρ_a 选择一个 S-视图，ρ_b 就遵循 ρ_a 的选择；如果 ρ_a 选择一个 T-视图，ρ_b 就查看从根到这个 T-视图（由 ρ_a 选择）的路径，并选择第一个满足 t 使得 $v(t) \in \mathrm{BT}(\rho_a)$ 的节点，即 T 在该分支上选择的最高层 ρ_a-视图。很容易看出，按照这种构造，T 中不存在一条根到叶路径，使得对应于 ρ_b 的两个不同 T-目标的两个包都位于其上。此外，$\mathrm{BT}(\rho_b) \subseteq \mathrm{BT}(\rho_a)$，且 $\mathrm{BS}(\rho_b) = \mathrm{BS}(\rho_a)$。因此，规则 ρ_b 不比规则 ρ_a 容易。□

现在我们可以展示任意树分解的一项主要性质了。

引理 E.3. 设 $\varphi(\mathbf{x}_A \mid \mathbf{x}_A)$ 是一个给定的 *CQAP*。设 $(\mathcal{T}\ \chi, r)$ 是一个固定的自由连通树分解。设 $\mathcal{P}$ 是由 $(\mathcal{T}\ \chi, r)$ 诱导出的 *PMTD* 集合。由 $\mathcal{P}$ 生成的任意 2 阶段析取规则 ρ_a 都不比以下形式的两阶段析取规则更容易

$$T(\mathbf{x}_{\chi(t_\ell)}) \vee \bigvee_{j \in [\ell]} S(\mathbf{x}_{A_j}) \leftarrow Q_A(\mathbf{x}_A) \wedge \bigwedge_{F \in \mathcal{E}} R_F(\mathbf{x}_F), \tag{33}$$

其中 $t_1 = r, t_2, \ldots, t_\ell$ 是树 $\mathcal{T}$ 中构成一条路径的节点，该路径从根节点 r 开始；并且

$$A_j \stackrel{def}{=} \begin{cases} A & \text{if } j = 1 \\ \chi(t_j) \cap \chi(t_{j-1}) & \text{if } j = 2, \ldots, \ell \end{cases}$$

证明。我们首先将 ρ_a 和 $(\mathrm{T}\ \chi, r)$ 作为输入，应用引理 E.2，得到规则 ρ_b，该规则满足：ρ_b 的任何 T-目标都不是另一个此类目标的祖先。我们固定 ρ_b 的 T-目标，即固定 $\mathrm{BT}(\rho_b) = \{B_1, \ldots, B_k\}$。我们的目标是证明，$\rho_b$ 必须包含一个 S-目标子集，其对应的 T 中的节点构成一条从根开始、并以某个节点 t 为终点的路径，使得 $\chi(t) \in \mathrm{BT}(\rho_b)$。下文中，我们将使用函数 $\chi^{-1}(B)$ 来恢复节点 $t \in V(\mathrm{T})$，使得 $\chi(t) = B$，并使用 $parent(t)$ 表示非根节点 t 的父节点。

我们通过允许对手在 PMTD中选择目标，同时自适应地选择对手必须从中进行选择的 PMTD，来证明该性质成立。我们可以选择 PMTD 的顺序，因为为了构造一个两阶段析取规则，必须从P中的每个 PMTD 选择一个视图，而我们只能控制检查它们的顺序。

考虑 PMTD P_1，其中 S-目标的集合恰好是 $M_1 = \{\chi^{-1}(B_1) \ldots, \chi^{-1}(B_k)\}$。我们允许对手从 P_1 中选择一个目标。我们声称，对手必须选择一个与 M_1 中某个节点对应的 S-视图。事实上，对手不能从 P_1 中选择 T-视图，因为 ρ_b 的 T-目标已经确定且无法更改。假设对手选择了与节点 $\chi^{-1}(B_i)$ 关联的 S-视图。对于这个 S-视图（因为其父节点与一个 T-视图相关联），

$$v(\chi^{-1}(B_i)) = \chi(\chi^{-1}(B_i)) \cap \chi(parent(\chi^{-1}(B_i))) = B_i \cap \chi(parent(\chi^{-1}(B_i)). \tag{34}$$

现在我们选择 PMTD P_2 where $M_2 = M_1 \cup \mathrm{parent}(\chi^{-1}(B_i))$，并将其交给对手。再次强调，对手不能选择 T-视图，因为这会改变 $\mathrm{BT}(\rho_b)$，因此必须选择与 M_2 中某个节点关联的 S-视图。假设对手选择了与节点 $\chi^{-1}(B_{i'}) \in M_2$ 关联的 S-视图。我们生成下一个 PMTD P_3 where $M_3 = M_2 \cup \mathrm{parent}(\chi^{-1}(B_{i'}))$。一般而言，在 q 轮之后生成的 PMTD P_{q+1} 满足 $M_{q+1} = M_q \cup \mathrm{parent}(\chi^{-1}(B_q))$，其中 $\chi^{-1}(B_q) \in M_q$ 是最后一轮中被选作 S-视图的节点（对于具有物化集 M_q 的 PMTD P_q）。当对手选择与根对应的 S-视图时，该过程终止。并且可以保证该过程会在有限步之后终止，因为我们始终通过沿树向上移动来向物化集中添加节点，而树的大小有限。在每一步中，对手都会为一个包选择一个 S-视图，并且只能在所有分支上逐渐接近根。因此，当对手到达根时，被选中的 S-视图中必定存在一个子集，它们对应于一条从根 $t_1 = r$ 到 T 中某个节点 t_ℓ 的路径，使得 $\chi(t_\ell) \in \mathrm{BT}(\rho_b)$，这就是所需的性质。另外请注意，这些 S-视图满足 (34)，因此证明完成。□

引理 E.3 告诉我们，只需分析形式如 (33) 所示的两阶段析取规则。现在我们继续为 (33) 寻找证明序列。设 $t_1 = r, t_2, \ldots, t_\ell$ 这些包分别对应分数边覆盖 $\mathrm{u}(1) \ldots, \mathrm{u}(\ell)$，其中 $\mathrm{u}(j) = (u(j)\, F)_{F \in \mathrm{E}}$，适用于 $j \in [\ell]$。我们将 α_j 定义为每个包相对于 A_j 中变量的松弛量（即 $\alpha_j = \alpha(\mathrm{u}(j)\, A_j)$），并引入因子 $\beta_j = \beta\bigstar/\alpha_j$，其中 $\beta\bigstar = \acute{\mathrm{I}}_{\, j \in [\ell]}\, \alpha_j$。接下来，我们对每个 ℓ 个包应用定理 6.1，将针对第 j 个包得到的证明序列乘以 β_j，然后按如下方式对各项求和：

$$
\begin{aligned}
\sum_{j\in[\ell]} \beta_j \sum_{F\in\mathcal{E}} u_F^{(j)} \cdot \log|R_F| + \beta^\star \cdot \log|Q_A| &\geq \sum_{j\in[\ell]} \beta_j \sum_{F\in\mathcal{E}} u_F^{(j)} \cdot (h_T(F \mid A_j \cap F) + h_S(A_j \cap F)) + \beta^\star \cdot h_T(A_1) \\
&= \sum_{j\in[\ell]} \beta_j \sum_{F\in\mathcal{E}} u_F^{(j)} \cdot h_T(F \mid A_j \cap F) + \beta^\star \cdot h_T(A_1) + \sum_{j\in[\ell]} \beta_j \sum_{F\in\mathcal{E}} u_F \cdot h_S(A_j \cap F) \\
&\geq \sum_{j\in[\ell]} \beta_j \cdot \alpha_j \sum_{F\in\mathcal{E}} \hat{u}_F^{(j)} \cdot h_T(F \mid A_j \cap F) + \beta^\star \cdot h_T(A_1) + \sum_{j\in[\ell]} \beta_j \cdot h_S(A_j) \\
&= \sum_{j\in[\ell]} \beta^\star \sum_{F\in\mathcal{E}} \hat{u}_F^{(j)} \cdot h_T(F \mid A_j \cap F) + \beta^\star \cdot h_T(A_1) + \sum_{j\in[\ell]} \beta_j \cdot h_S(A_j) \\
&= \sum_{j\in\{2,\ldots,\ell\}} \beta^\star \sum_{F\in\mathcal{E}} \hat{u}_F^{(j)} \cdot h_T(F \mid A_j \cap F) + \beta^\star \left(\sum_{F\in\mathcal{E}} \hat{u}_F^{(1)} \cdot h_T(F \mid A_1 \cap F) + h_T(A_1) \right) + \sum_{j\in[\ell]} \beta_j \cdot h_S(A_j) \\
&\geq \sum_{j\in\{2,\ldots,\ell\}} \beta^\star \sum_{F\in\mathcal{E}} \hat{u}_F^{(j)} \cdot h_T(F \mid A_j \cap F) + \beta^\star h_T(\chi(t_1)) + \sum_{j\in[\ell]} \beta_j \cdot h_S(A_j) \\
&\geq \sum_{j\in\{2,\ldots,\ell\}} \beta^\star \sum_{F\in\mathcal{E}} \hat{u}_F^{(j)} \cdot h_T(F \mid A_j \cap F) + \beta^\star h_T(A_2) + \sum_{j\in[\ell]} \beta_j \cdot h_S(A_j) \\
&\ldots \\
&\geq \beta^\star h_T(\chi(t_\ell)) + \sum_{j\in[\ell]} \beta_j \cdot h_S(A_j),
\end{aligned}
$$

其中，第二个不等式是 Shearer 引理的直接应用，第三个不等式则直接由引理 E.1 得出。由此证明序列，我们得到以下内在权衡，

$$|Q_A|^{\beta^\star} \cdot |\mathcal{D}|^{\sum_{j\in[\ell]} \beta_j \cdot u_j^*} \cong S^{\sum_{j\in[\ell]} \beta_j} \cdot T^{\beta^\star}$$

其中 $u_j^* = \acute{\mathrm{I}}_{\, F\in\mathcal{E}}\, u_{(j)\, F}$。等价地，我们得到

$$|Q_A| \cdot |\mathcal{D}|^{\sum_{j\in[\ell]} u_j^*/\alpha_j} \cong S^{\sum_{j\in[\ell]} 1/\alpha_j} \cdot T. \tag{35}$$

在上述权衡中，对于给定的 S，（假设 $|Q_A| = 1$），我们得到

$$
\begin{aligned}
\log T &= \sum_{j\in[\ell]} \frac{u_j^*}{\alpha_j} \cdot \log|\mathcal{D}| - \sum_{j\in[\ell]} (1/\alpha_j) \log S \\
&= \sum_{j\in[\ell]} (1/\alpha_j) \cdot (u_j^* \log|\mathcal{D}| - \log S)
\end{aligned}
$$

一个观察是，如果路径 $t_1, \ldots, t_\ell$ 上的某个包 t_j 的 AGM 界不大于 S，那么 t_j 对应的 S-视图 $S(\mathbf{x}^{A_j})$ 的物化可完全作为式 (33) 的模型进行物化。否则，对于每个 $t_j, j \in [\ell]$，对任意 $j \in [\ell]$，$(u_j^* \log|\mathrm{D}| - \log S)$ 均为非负，因此上述 $\log T$ 表达式会随 ℓ 增大而单调增加。因此，代价最高的权衡对应于式 (33) 中从根开始并到达叶节点的析取规则。要得到最终的时空权衡，我们只需取 T 中所有根到叶路径上的最差权衡。

在结束本节之前，我们将证明，对于固定树分解 $(\mathrm{T}\ \chi)$ 的所有根到叶路径而言，我们在 (35) 中得到的权衡能够在不引入额外超参数的情况下，复现（并可能改进）[13], 的定理 13。具体而言，在 (35) 中，作者为每个 $t \in V(\mathrm{T})$ 设置一个超参数 $\delta(t)$，并令在线回答时间为 $T = |\mathrm{D}|^{\sum_{j\in[\ell]} \delta(t_j)}$，其中 $t_1, \ldots, t_\ell$ 是 T 中使 $\acute{\mathrm{I}}_{\, j\in[\ell]}\, \delta(t_j)$ 最大的根到叶路径。假设我们针对这条根到叶路径 $t_1, \ldots, t_\ell$ 构造一个形式如 (35) 的两阶段析取规则，那么对于任意分数边覆盖 $u_j^*, j \in [\ell]$，都有（通过改写 (35)）

$$\log_{|\mathcal{D}|} S = \frac{1}{\sum_{j\in[\ell]} (1/\alpha_j)} \cdot \left(\sum_{j\in[\ell]} \frac{u_j^*}{\alpha_j} - \log_{|\mathcal{D}|} T \right)$$

$$= \frac{1}{\sum_{j\in[\ell]}(1/\alpha_j)} \cdot \left(\sum_{j\in[\ell]} \frac{u_j^*}{\alpha_j} - \sum_{j\in[\ell]} \delta(t_j) \right)$$
$$= \frac{1}{\sum_{j\in[\ell]}(1/\alpha_j)} \cdot \sum_{j\in[\ell]} \frac{1}{\alpha_j}(u_j^* - \delta(t_j)\cdot\alpha_j)$$
$$\le \max_{j\in[\ell]} \left(\sum_{F\in\mathcal{E}} u_F^{(j)} - \delta(t_j)\cdot\alpha_j \right).$$

最后一个不等式成立，是因为对于所有满足 $\sum_i w_i = 1$ 的 $w_i \ge 0$，都有 $\sum_{i=1}^n \gamma_i w_i \le \max_i \gamma_i$。令 $w_i = \frac{1}{\alpha_i} \cdot \frac{1}{\sum_{j\in[\ell]}(1/\alpha_j)}$，即可得到所需表达式。因此，很容易看出，我们在 (35) 中针对固定树分解的所有根到叶路径得到的权衡，确实复现了 [13] 中的结果。

E.3 补充示例

在本节中，我们通过我们的框架给出若干时空权衡的具体示例。在所有示例中，我们假设数据库大小为 $|\mathbb{D}|$。此外，为简洁起见，我们使用视图的有序元组（T-视图和 S-视图）来表示具有路径型结构的 PMTD（元组的第一个元素表示根）。为了便于从证明序列中解读权衡，我们始终在证明序列左侧标出其对应的联合香农流不等式所隐含的上界。对于该隐含上界，我们用 $n_F \text{def=} \log |R_F|$ 表示关系 R_F的对数大小，用 $w_A \text{def=} \log |Q_A|$ 表示访问请求 Q_A的对数大小。

示例 *E.4*（三角形查询）。我们考虑一个具有空 访问模式的三角形查询，即 $A = \emptyset$:

$$\varphi(x_1, x_3 \mid \emptyset) \leftarrow R(x_1, x_2) \wedge R(x_2, x_3) \wedge R(x_3, x_1).$$

我们考虑两个 PMTD，它们的袋均为 $\{x_1, x_2, x_3\}$。在第一个 PMTD 中，该袋未被物化，因此与之关联一个 T 视图 T_{123}。在第二个 PMTD 中，该袋已被物化，并具有一个 S 视图 S_{13}。因此，这两个 PMTD 可以表示为

$$(T_{123}), \quad (S_{13}).$$

因此，我们得到如下析取规则（不含原子 Q_A）：

$$T_{123} \vee S_{13} \leftarrow R(x_1, x_2) \wedge R(x_2, x_3) \wedge R(x_3, x_1).$$

其中一个（空）证明序列就是 $\log |\mathcal{D}| \ge h_S(13)$，这表明我们可以在线性空间中存储所有至少参与一个三角形的 (x_1, x_3) 对。

示例 *E.5*（正方形查询）。在此示例中，我们采用如下 CQAP:

$$\varphi(x_1, x_3 \mid x_1, x_3) \leftarrow R(x_1, x_2) \wedge R(x_2, x_3) \wedge R(x_3, x_4) \wedge R(x_4, x_1).$$

这对应以下任务：给定图中的两个顶点，判断它们是否位于正方形的两个对角顶点上。我们考虑两个 PMTD。第一个 PMTD 有一个根袋 $\{1, 3, 4\}$，与一个 T 视图 T_{134} 相关联；还有一个袋 $\{1, 3, 2\}$，与一个 T 视图 T_{132} 相关联。第二个 PMTD 只有一个袋 $\{1, 2, 3, 4\}$，与一个 S 视图 S_{13} 相关联。这两个 PMTD 可以表示为

$$(T_{134}, T_{132}), \quad (S_{13})$$

这进而生成两条析取规则:

$$T_{134} \vee S_{13} \leftarrow Q_{13}(x_1, x_3) \wedge R(x_1, x_2) \wedge R(x_2, x_3) \wedge R(x_3, x_4) \wedge R(x_4, x_1)$$
$$T_{132} \vee S_{13} \leftarrow Q_{13}(x_1, x_3) \wedge R(x_1, x_2) \wedge R(x_2, x_3) \wedge R(x_3, x_4) \wedge R(x_4, x_1).$$

我们可以为第一条规则构造如下证明序列:

$$\begin{aligned} n_{14} + n_{34} + 2\cdot w_{13} &\ge h_S(1) + h_T(4|1) + h_S(3) + h_T(4|3) + 2\cdot h_T(13) \\ &\ge h_S(13) + h_T(4|1) + h_T(4|3) + 2\cdot h_T(13) \\ &\ge h_S(13) + h_T(4|13) + h_T(13) + h_T(4|13) + h_T(13) \\ &= h_S(13) + 2\cdot h_T(134) \end{aligned}$$

对于第二条规则，我们对称地构造 $n_{12} + n_{32} + 2\cdot w_{13} \ge h_S(13) + 2\cdot h_T(132)$ 的证明。因此，我们得到 $S\cdot T^2 \cong |\mathcal{D}|^2 \cdot |Q_A|^2$ 的权衡。当 $|Q_A| = 1$ 时，此权衡重现了 [12] 中例15得到的改进结果。

E.4 k-可达性的权衡（第 6.4 节）

示例 $E.6$（2-可达性）。考虑具有以下访问模式的 2-可达性合取查询（针对 $|Q_{13}|=1$ 进行优化）：

$$\phi_2(x_1,x_3 \mid x_1,x_3) \leftarrow R_1(x_1,x_2) \wedge R_2(x_2,x_3).$$

从仅包含一个包 {1,2,3} 的树分解出发，我们构造所有非冗余且非支配的 PMTD 的集合，即

$$(T_{123}), \quad (S_{13}).$$

它们只生成一条两阶段析取规则，

$$T_{123} \vee S_{13} \leftarrow Q_{13}(x_1,x_3), R_1(x_1,x_2), R_2(x_2,x_3),$$

对此，我们可以构造如下证明序列：

$$\begin{aligned} n_{12}+n_{23}+2\cdot w_{13} &\geq h_S(1)+h_T(2|1)+h_S(3)+h_T(2|3)+2\cdot h_T(13)\\ &\geq h_S(13)+h_T(2|1)+h_T(2|3)+2\cdot h_T(13)\\ &\geq h_S(13)+h_T(2|13)+h_T(2|13)+2\cdot h_T(13)\\ &\geq h_S(13)+2\cdot h_T(123) \qquad (S\cdot T^2 \cong |\mathbb{D}|^2\cdot |Q_{13}|^2) \end{aligned}$$

讨论。假设 $w_{13}=\log |\mathbb{D}|, (1,13,N_{13|1}) \in \mathrm{AC}$ 且 $w_{3|1} \stackrel{\text{def}}{=} \log N_{13|1}$，那么 $n_{12}+w_{3|1} \geq h_T(12)+h_T(3|12) = h_T(123)$ 是一条（可能理想的）证明序列，这意味着我们可以在时间 $T=|\mathbb{D}|\cdot N_{13|1}$ 内回答任意访问请求，而无需进行任何物化。注意，如果对于某个 $\epsilon>0$，$S=|\mathbb{D}|, N_{13|1}=|\mathbb{D}|^{1/2-\epsilon}$，那么它严格优于上述证明序列（这意味着 $T=|\mathbb{D}|^{3/2}$）。

例 $E.7$（3-可达性）。我们研究 3-可达性 CQAP（针对 $|Q_A|=1$ 进行优化），即

$$\phi_3(x_1,x_4 \mid x_1,x_4) \leftarrow R(x_1,x_2) \wedge R(x_2,x_3) \wedge R(x_3,x_4).$$

使用所有非冗余且非支配的 PMTD 集合（如图 3 所示），我们生成所有两阶段析取规则（丢弃冗余规则/目标后）及其对应的证明序列：

(1) ρ_1: $T_{134} \vee T_{124} \vee S_{14} \leftarrow Q_{14}(x_1,x_4) \wedge R(x_1,x_2) \wedge R(x_2,x_3) \wedge R(x_3,x_4)$

$$\begin{aligned} n_{12}+n_{34}+2w_{14} &\geq h_S(1)+h_S(4)+h_T(2|1)+h_T(3|4)+2h_T(14)\\ &\geq h_S(14)+h_T(2|1)+h_T(3|4)+2h_T(14)\\ &\geq h_S(14)+h_T(2|14)+h_T(3|14)+2h_T(14)\\ &\geq h_S(14)+h_T(124)+h_T(134) \qquad (S\cdot T^2 \cong |\mathbb{D}|^2\cdot |Q_A|^2) \end{aligned}$$

(2) ρ_2: $T_{123} \vee S_{13} \vee T_{124} \vee S_{14} \leftarrow Q_{14}(x_1,x_4) \wedge R(x_1,x_2) \wedge R(x_2,x_3) \wedge R(x_3,x_4)$

$$\begin{aligned} 2\cdot n_{12}+n_{23}+n_{34}+3\cdot w_{14} &= 2(h_S(1)+h_T(2|1))+h_S(3)+h_T(2|3)+h_S(4)+h_T(3|4)+3\cdot h_T(14)\\ &\geq h_S(14)+h_S(13)+2h_T(2|14)+h_T(3|41)+h_T(2|314)+3\cdot h_T(14)\\ &= h_S(14)+h_S(13)+2h_T(124)+h_T(1234)\\ &\geq h_S(14)+h_S(13)+3\cdot h_T(124) \qquad (S^2\cdot T^3 \cong |\mathbb{D}|^4\cdot |Q_A|^3) \end{aligned}$$

(3) ρ_3: $T_{134} \vee T_{234} \vee S_{24} \vee S_{14} \leftarrow Q_{14}(x_1,x_4) \wedge R(x_1,x_2) \wedge R(x_2,x_3) \wedge R(x_3,x_4)$
ρ_3 的证明序列在此省略，因为它与规则 ρ_2 对称。

(4) ρ_4: $T_{123} \vee S_{13} \vee T_{234} \vee S_{24} \vee S_{14} \leftarrow Q_{14}(x_1,x_4) \wedge R(x_1,x_2) \wedge R(x_2,x_3) \wedge R(x_3,x_4)$
对于 ρ_4，我们展示如下两个互不支配的证明序列：

$$\begin{aligned} n_{12}+n_{34}+w_{14} &\geq h_S(1)+h_S(4)+h_T(2|1)+h_T(3|4)+h_T(14)\\ &\geq h_S(14)+h_T(2|14)+h_T(3|214)+h_T(14)\\ &\geq h_S(14)+h_T(1234)\\ &\geq h_S(14)+h_T(123) \qquad (S\cdot T \cong |\mathbb{D}|^2\cdot |Q_A|) \end{aligned}$$

$$\begin{aligned} 2n_{23}+2n_{12}+2n_{34}+w_{14} &\geq 2\cdot h_S(23)+h_S(12)+h_S(34)+h_S(1)+h_T(2|1)+h_S(4)+h_T(3|4)+h_T(14)\\ &= h_S(2)+h_S(3|2)+h_S(3)+h_S(2|3)+h_S(12)+h_S(34)+h_S(1)+h_T(2|1)+h_S(4)+h_T(3|4)+h_T(14)\\ &\geq h_S(123)+h_S(234)+h_S(2)+h_S(3)+h_S(1)+h_T(2|1)+h_S(4)+h_T(3|4)+h_T(14)\\ &\geq h_S(123)+h_S(234)+h_S(24)+h_S(13)+h_T(2|1)+h_T(3|4)+h_T(14)\\ &\geq 2\cdot h_S(24)+2\cdot h_S(13)+h_T(2|14)+h_T(3|124)+h_T(14)\\ &\geq 2\cdot h_S(24)+2\cdot h_S(13)+h_T(1234) \end{aligned}$$

$$\geq 2 \cdot h_S(24) + 2 \cdot h_S(13) + h_T(123) \qquad (S^4 \cdot T \cong |\mathcal{D}|^6 \cdot |Q_A|)$$

最后，对于上述每条规则，都存在一个证明序列，对应于在在线阶段从头开始执行广度优先搜索（BFS），其时间为 $O(|\mathbb{D}|)$。以 ρ_1 为例，我们得到

$$\begin{aligned} n_{23} + w_{14} &\geq h_T(23) + h_T(14) \\ &\geq h_T(1234) \\ &\geq h_T(134) \qquad (T = |\mathcal{D}| \cdot |Q_A|) \end{aligned}$$

相应的权衡曲线图见图 4a。

示例 *E.8*（4-可达性）。我们研究以下用于 4-可达性的 CQAP（针对 $|Q_A| = 1$ 进行优化）：

$$\phi_4(x_1, x_5 \mid x_1, x_5) \leftarrow R_{12}(x_1, x_2) \wedge R_{23}(x_2, x_3) \wedge R_{34}(x_3, x_5) \wedge R_{45}(x_4, x_5)$$

我们固定以下一组非冗余且非支配的 PMTD（总计 11 个），其中使用视图有序元组表示具有路径型结构的 PMTD（元组的第一项表示根）。请注意，纳入更多 PMTD 可能会得到更优的权衡。

$$(T_{1235}, T_{345}), \quad (T_{1235}, S_{35}), \quad (T_{1345}, T_{123}), \quad (T_{1345}, S_{13}), \quad (T_{1245}, T_{234}), \quad (T_{1245}, S_{24})$$
$$(T_{125}, T_{2345}), \quad (T_{125}, S_{25}), \quad (T_{145}, T_{1234}), \quad (T_{145}, S_{14}), \quad (S_{15})$$

尽管由上述 PMTD 集生成了 210 条两阶段析取规则，但与 3-可达性类似，我们可以舍弃目标严格多于其他规则的规则。首先请注意，任何规则都必须以 S_{15} 为目标。对于任何选择 T_{1245} 为目标（或 T_{125}、T_{145}）的析取规则，即 ρ_1：$T_{2345} \vee S_{15}$，我们总有以下证明序列：

$$\begin{aligned} n_{12} + n_{45} + w_{15} &\geq h_S(1) + h_T(2|1) + h_S(5) + h_T(4|5) \\ &\geq h_S(15) + h_T(2|15) + h_T(4|125) + h_T(15) \\ &= h_S(15) + h_T(1245) \qquad (S \cdot T \cong |\mathcal{D}|^2 \cdot |Q_A|) \end{aligned}$$

否则，对于最后 7 个 PMTD，析取规则必须选择 $\{T_{234}, S_{24}, T_{2345}, S_{25}, T_{1234}, S_{14}, S_{15}\}$，或者只选择 $\{T_{234}, S_{24}, S_{25}, S_{14}, S_{15}\}$，这是因为存在 T_{234}，可以移除冗余项 T_{2345}。现在，析取规则从前 4 个 PMTD 中选取目标，我们可以分成以下几种情况，并舍弃冗余目标：

(1) ρ_2： $T_{1235} \vee T_{1345} \vee (T_{234} \vee S_{24} \vee S_{25} \vee S_{14} \vee S_{15})$

$$\begin{aligned} n_{12} + n_{23} + n_{34} + n_{45} + 2w_{15} &\geq h_S(1) + h_T(2|1) + h_S(2) + h_T(3|2) + h_S(4) + h_T(3|4) + h_S(5) + h_T(4|5) + 2h_T(15) \\ &\geq h_S(15) + h_S(24) + h_T(2|15) + h_T(3|125) + h_T(3|145) + h_T(4|15) + 2h_T(15) \\ &= h_S(15) + h_S(24) + h_T(1235) + h_T(1345) \qquad (S^2 \cdot T^2 \cong |\mathbb{D}|^4 \cdot |Q_A|^2) \end{aligned}$$

(2) ρ_3： $T_{345} \vee S_{35} \vee T_{123} \vee S_{13} \vee (T_{234} \vee S_{24} \vee S_{25} \vee S_{14} \vee S_{15})$

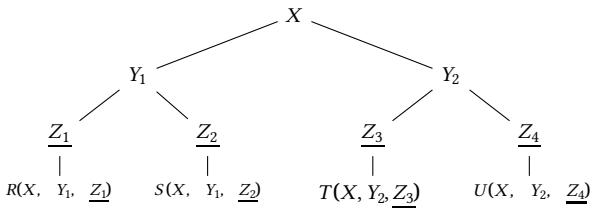
ρ_3 的证明序列在此省略，因为 ρ_3 不比 ρ_2 更难，因为 $\{3,4,5\} \subseteq \{1,3,4,5\}$ 和 $\{1,2,3\} \subseteq \{1,2,3,5\}$

(3) ρ_4： $T_{345} \vee S_{35} \vee T_{1345} \vee (T_{234} \vee S_{24} \vee S_{25} \vee S_{14} \vee S_{15}) = T_{345} \vee S_{35} \vee (T_{234} \vee S_{24} \vee S_{25} \vee S_{14} \vee S_{15})$

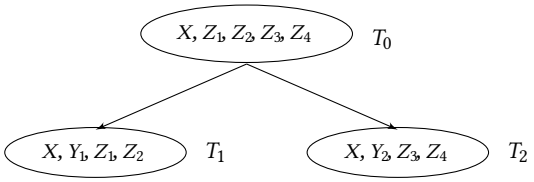
对于 ρ_4，我们展示如下两个互不支配的证明序列：

$$\begin{aligned} &2n_{23} + 2n_{12} + 5n_{34} + 3n_{45} + 5w_{15} \\ &\geq 2(h_S(2) + h_T(3|2)) + 2(h_S(1) + h_T(2|1)) + 2(h_S(3) + h_T(4|3)) + 3(h_S(4) + h_T(3|4)) \\ &\quad + 3(h_S(5) + h_T(4|5)) + 5h_T(15) = 2(h_T(2|1) + h_T(3|2) + h_T(4|3)) + 3(h_T(4|5) + h_T(\\ &3|4)) + 5h_T(15) + 2(h_S(3) + h_S(5)) + (h_S(2) + h_S(5)) + (h_S(2) + h_S(4)) + 2(h_S(1) \\ &\quad + h_S(4)) \geq 2(h_T(2|15) + h_T(3|125) + h_T(4|1235)) + 3(h_T(4|5) + h_T(3|45)) + 2h_T(\\ &15) + 3h_T(5) + 2(h_S(3) + h_S(5|3)) + (h_S(2) + h_S(5|2)) + (h_S(2) + h_S(4|2)) + 2(h_S \\ &(1) + h_S(4|1)) = 2h_T(12345) + 3h_T(345) + 2h_S(35) + h_S(25) + h_S(24) + 2h_S(14) \\ &\geq 5h_T(345) + 2h_S(35) + h_S(25) + h_S(24) + 2h_S(14) \\ &\qquad (S^6 \cdot T^5 \cong |\mathbb{D}|^{12} \cdot |Q_A|^5) \end{aligned}$$

$$\begin{aligned} &3n_{23} + 3n_{34} + 3n_{45} + n_{12} + 2n_{34} + n_{23} + 3w_{15} \\ &\geq 3(h_S(3) + h_S(2|3)) + 3h_S(34) + 3(h_S(5) + h_T(4|5)) + (h_S(1) + h_T(2|1)) + 2(h_S(4) + h_T(3|4)) + (h_S(2) + h_T(3|2)) + 3h_T(15 \quad) \\ &\geq 3h_S(234) + 3h_S(35) + 3h_T(4|5) + (h_S(1) + h_T(2|1)) + 2(h_S(4) + h_T(3|4)) + (h_S(2) + h_T(3|2)) + 3h_T \\ &(15) \geq 3h_S(24) + 3h_S(35) + h_S(14) + h_S(24) + 3h_T(4|5) + h_T(2|1) + 2h_T(3|4) + h_T(3|2) + 3h_T(15) \geq 3h_S \\ &(24) + 3h_S(35) + h_S(14) + h_S(24) + 3h_T(4|5) + h_T(2|1) + 2h_T(3|45) + h_T(3|2) + 2h_T(5) + h_T(15) \end{aligned}$$



(a) 由完全二叉树表示的层次查询。树的叶节点构成 $\mathbf{x}_A$。每条根到叶路径构成一个关系（已标记）。



(b) 左侧层次查询的树分解。

图 6：层次合取查询与访问模式及其树分解示例。

$$\geq 3h_S(24) + 3h_S(35) + h_S(14) + h_S(24) + 2h_T(345) + h_T(4|15) + h_T(2|145) + h_T(3|1245) + h_T(15)$$
$$\geq 3h_S(24) + 3h_S(35) + h_S(14) + h_S(24) + 2h_T(345) + h_T(12345)$$
$$\geq 3h_S(24) + 3h_S(35) + h_S(14) + h_S(24) + 3h_T(345) \qquad (S^8 \cdot T^3 \cong |\mathcal{D}|^{13} \cdot |Q_A|^3)$$

(4) ρ_5： $T_{1235} \vee T_{123} \vee S_{13} \vee (T_{234} \vee S_{24} \vee S_{25} \vee S_{14} \vee S_{15})$

此处省略 ρ_5 的证明序列，因为它与规则 ρ_4 对称。

与 3-可达性类似，上述每条规则都有一个证明序列，对应于在线阶段的广度优先搜索。相应的权衡曲线图见图 4b。

F 布尔层次查询的空间—时间权衡

本节展示我们的框架在 CQ 布尔层次查询中的一些应用，这些查询是定义在主体为层次查询的查询之上的 CQAP。若一个查询中的任意两个变量，其原子集合要么不相交，要么其中一个包含于另一个，则称该查询为层次的。另一种解释是，每个层次查询都存在一种规范排序[3]，它是一棵有根树，其中查询中每个原子的变量都位于树中的同一条根到叶路径上，且每个原子都是其最低层变量的子节点。例如，图 6a 所示的查询就是层次查询。变量 Y_1 出现在原子 $\{R, S\}$ 中，而这些原子是变量 X 出现的原子集合 $\{R, S, T, U\}$ 的子集。请注意，每条根到叶路径都构成一个关系。

层次查询是一类有趣的查询，能够刻画多种问题的可处理性，例如概率数据库中的查询求值 [10], 动态查询求值 [5, 18], 等。[20]的开创性工作为枚举任意（不一定是全的）层次查询的结果，提供了预处理时间与延迟保证之间的权衡（即 $A = \emptyset$ 在其设定下的情况）。然而， [20] 并未涉及这样一种设定：可以为层次查询中的某些变量指定访问模式，而这正是我们工作的重点。尽管设定有所不同，对 [20] 中主算法的改编仍能为空间使用量与回答时间之间的权衡提供一个非平凡的基准，我们将在本节中重点介绍。改编后的算法本身也可能具有独立的研究价值。我们首先介绍用于静态枚举层次查询的主算法。

给定一个层次查询 φ（$\mathbf{y}$）， [20] 中的预处理阶段会取自由变量[4]$\mathbf{y}$，并构造一个偏斜感知视图树列表以及重/轻指示视图。从根开始，对于任何违反自由连通性质的绑定变量，会采用两种求值策略。第一种策略会物化查询结果的一个子集：该子集是针对变量顺序中 $\text{anc}(W) \cup \{W\}$[5] 上的轻值获得的。它还会聚合掉以 W 为根的子树中的绑定变量。由于轻值的度有界，这种物化的开销很小。第二种策略会计算针对 $\text{anc}(W) \cup \{W\}$ 上那些在至少一个关系中为重（即度较高）的值所获得的其余查询结果的紧凑表示。该策略将 W 视为自由变量，并递归处理位于 W 下方的其他绑定变量。

示例 F.1。对于图 6a 中的查询，针对轻度数阈值 N^ϵ，分别为 $\{X\}$、$\{X, Y_1\}$、$\{X, Y_2\}$ 构造以下视图： $V_X(Z_1, Z_2, Z_3, Z_4) V_{Y_1}(X, Z_1, Z_2)$ $V_{Y_2}(X, Z_3, Z_4)$（假设关系大小为 N）。指示视图为 $H_X(X)$ 和 $L_X(X)$，用于存储哪些 X 值是重的、哪些是轻的。递归地继续处理时，我们将 A 视为自由变量，并构造指示视图 $H_{Y_1}(X, Y_1)$ 和 $L_{Y_1}(X, Y_1)$（右子树同理）。重指示视图的大小至多为 $O(N^{1-\epsilon})$。

枚举阶段随后将视图作为输入，交由所谓的 UNION 和 PRODUCT 算法；该算法组合 $O(1)$ 个视图树的输出，并利用这样一个观察：重绑定变量本身构成一个层次结构。

例 *F.2.* 继续这个例子，在枚举阶段， $V_X(Z_1, Z_2, Z_3, Z_4)$ 的输出可通过常数延迟获得。对于 $H_X(X)$ 中每个重的 X，我们递归处理，并使用视图 $V_{Y_1}(X, Z_1, Z_2)$（其输出可通过常数延迟访问），或者

[3] [20] 中的原始定义也包含一个依赖函数，但由于我们在分解构造中不使用它，因此将其省略。[4] [20] 中的术语将头部中的变量称为自由变量，并将查询中的所有其他变量称为约束变量（不要与$\mathbf{x}_A$混淆，本文中我们将其称为约束变量）。[5]anc(W)定义为从 W 到根的路径上、排除 W 后的变量。

对于所有轻的 $\{X, Y_1\}$，其延迟为常数；或者利用以下观察：左子树仅剩的情形是 $\{X, Y_1\}$ 为重，这使我们能够枚举子查询 $Q_{Z_1}(X, Y_1, Z_1) = R(X, Y_1, Z_1)$ 和 $Q_{Z_2}(X, Y_1, Z_2) = S(X, Y_1, Z_2)$ 的 (Z_1, Z_2) 个答案；对于给定的 $\{X, Y_1\}$ 赋值，这两个子查询都支持常数延迟枚举。总体延迟为 $O(N^{1-\epsilon})$。

定理 F.3（[20]）。给定一个静态宽度为w的层次查询、大小为 N的数据库以及 $\epsilon \in [0, 1]$，可在经过 $O(N^{1+(\mathrm{w}-1)\epsilon})$ 的预处理时间和空间后，以 $O(N^{1-\epsilon})$ 的延迟枚举查询结果。

静态宽度 w 是分数超树宽的一种推广，它将查询中自由变量的结构纳入考量。对于图 6a 中的查询，$\mathrm{w} = 4$。

改编算法。为了将用于枚举层次查询结果的算法改编为适用于我们的设定，我们注意到两点。首先，我们应用 [20] 的预处理阶段，将自由变量设为 $\mathbf{x}_A$，并构建数据结构。然后，我们可以为基本关系和视图树构建带索引的表示，其中索引变量集合为 $\mathbf{x}_A$。换言之，对于任意视图 V，以及每个 $v \in \Pi_{\mathrm{vars}(V)\cap A}(V)$，我们将 $V \ltimes v$ 存储在哈希表 H_V 中，并以 v 作为键。其次，给定一个固定取值（例如 z）作为 $\mathbf{x}_A$，我们向枚举算法提供带索引的基本关系、视图指示器以及视图（即 $H_V[\Pi_{\mathrm{vars}(V)\cap A}(z)]$），该算法以 $O(N^{1-\epsilon})$ 的延迟枚举查询结果。由于作为输入提供的每个基本关系 $R' = R \ltimes z$ 对于 $\mathrm{vars}(R) \cap A$（即 $\Pi_{\mathrm{vars}(R)\cap A}(z)$）只有一个可能的取值，因此枚举算法唯一能输出的答案就是 z，否则便声明不存在答案。由此，我们能够成功回答布尔层次合取查询与访问模式。设 w 表示由 [20] 得到的层次查询的静态宽度，其中自由变量设为 $\mathbf{x}_A$。我们可以得到以下权衡。

定理 F.4. 给定一个布尔型层次合取查询与访问模式（即 $H = A$）、一个大小为 N 的数据库、作为静态宽度的 w（自由变量为 $\mathbf{x}_A$），以及 $\epsilon \in [0, 1]$，该查询可在 $O(N^{1-\epsilon})$ 时间内回答，所用的预处理数据结构占用 $S = O(N^{1+(\mathrm{w}-1)\epsilon})$ 空间。

示例 *F.5.* 将定理 F.4 应用于该 CQAP（我们用 $\mathbf{Z}$ 作为集合 $\{Z_1, Z_2, Z_3, Z_4\}$ 的简写）

$$\varphi(\mathbf{Z} \mid \mathbf{Z}) \leftarrow Q_A(\mathbf{Z}) \wedge R(X, Y_1, Z_1) \wedge S(X, Y_1, Z_2) \wedge T(X, Y_2, Z_3) \wedge U(X, Y_2, Z_4)$$

对应于图 6a，得到权衡 $S \cdot T^3 \cong |\mathbb{D}|^4$，适用于该查询的任意实例化。这里，$\mathbf{Z}$ 是访问模式。

接下来，我们展示如何通过我们的框架得出示例 F.5 中的权衡，并进一步改进，使其也将访问请求 Q_A 纳入考量。图 6b 展示了我们针对该查询所考虑的（自由连通）树分解。由此（自由连通）树分解诱导出的 PMTD 集合（共 5 个）包含 5 个 PMTD；这是通过选择以下其中一个作为物化集 M 得到的：$\{T_0\}$、$\{T_1\}$、$\{T_2\}$、$\{T_1, T_2\}$ 或 $\emptyset$。令

$$\mathrm{body} = Q_A(\mathbf{Z}) \wedge R(X, Y_1, Z_1) \wedge S(X, Y_1, Z_2) \wedge T(X, Y_2, Z_3) \wedge U(X, Y_2, Z_4),$$

我们从 5 个 PMTD 中生成以下两阶段析取规则：

$$\begin{aligned}
T_0(\mathbf{Z}, X) \vee S_{1234}(\mathbf{Z}) &\leftarrow \mathrm{body}\\
T_1(Z_1, Z_2, Y_1, X) \vee S_{12}(Z_1, Z_2, X) \vee S_{1234}(\mathbf{Z}) &\leftarrow \mathrm{body}\\
T_1(Z_1, Z_2, Y_1, X) \vee T_2(Z_3, Z_4, Y_2, X) \vee S_{34}(Z_3, Z_4, X) \vee S_{1234}(\mathbf{Z}) &\leftarrow \mathrm{body}\\
T_1(Z_1, Z_2, Y_1, X) \vee T_2(Z_3, Z_4, Y_2, X) \vee S_{12}(Z_1, Z_2, X) \vee S_{1234}(\mathbf{Z}) &\leftarrow \mathrm{body}
\end{aligned}$$

现在，我们为这四条规则分别构造证明序列。对于第一条规则，我们得到如下所示的权衡 $S \cdot T^3 \cong |\mathbb{D}|^4 \cdot |Q_A|^3$

$$\begin{aligned}
4\log|\mathcal{D}| + 3\log|Q_A(\mathbf{Z})| &\geq 3h_T(X) + h_S(Z_1Y_1X|X) + h_S(Z_2Y_1X|X) + h_S(Z_3Y_2X|X) + h_S(Z_4Y_2X) + 3h_T(\mathbf{Z})\\
&\geq 3h_T(X) + h_S(Z_1Y_1X|X) + h_S(Z_2Y_1X|X) + h_S(Z_4Z_3Y_2X|Z_4X) + h_S(Z_4X) + 3h_T(\mathbf{Z})\\
&= 3h_T(X) + h_S(Z_1Y_1X|X) + h_S(Z_2Y_1X|X) + h_S(Z_4Z_3Y_2X) + 3h_T(\mathbf{Z})\\
&\geq 3h_T(X) + h_S(Z_1Y_1X|X) + h_S(Z_4Z_3Z_2Y_1X|Z_4Z_3X) + h_S(Z_4Z_3X) + 3h_T(\mathbf{Z})\\
&= 3h_T(X) + h_S(Z_1Y_1X|X) + h_S(Z_4Z_3Z_2Y_1X) + 3h_T(\mathbf{Z})\\
&\geq 3h_T(X) + h_S(Z_4Z_3Z_2Z_1Y_1X|Z_4Z_3Z_2X) + h_S(Z_4Z_3Z_2X) + 3h_T(\mathbf{Z})\\
&= 3h_T(X) + h_S(Z_4Z_3Z_2Z_1Y_1X) + 3h_T(\mathbf{Z})\\
&\geq 3h_T(X\mathbf{Z}) + h_S(\mathbf{Z}) \qquad (S \cdot T^3 \cong |\mathcal{D}|^4 \cdot |Q_A|^3)
\end{aligned}$$

对于其余规则，我们得到权衡 $S \cdot T \cong |\mathbb{D}|^2 \cdot |Q_A|$。下面给出第二条规则的证明序列。第三条和第四条规则的证明序列非常相似，因此省略。

$$\begin{aligned}
2\log|\mathcal{D}| + \log|Q_A(\mathbf{Z})| &\geq h_T(Y_1X) + h_S(Z_1Y_1X|Y_1X) + h_S(Z_2Y_1X) + h_T(Z_2Z_1)\\
&\geq h_T(Y_1X) + h_S(Z_2Z_1Y_1X|Z_2Y_1X) + h_S(Z_2Y_1X) + h_T(Z_2Z_1)\\
&= h_T(Y_1X) + h_S(Z_2Z_1Y_1X) + h_T(Z_2Z_1)\\
&\geq h_T(Z_2Z_1Y_1X) + h_S(Z_2Z_1X) \qquad (S \cdot T \cong |\mathbb{D}|^2 \cdot |Q_A|)
\end{aligned}$$

总体而言，我们得到权衡 $S\cdot T^3 \cong |\mathbb{D}|^4\cdot |Q_A|^3$，因为 $S\cdot T \cong |\mathbb{D}|^2\cdot |Q_A|$ 被其支配。

改进的权衡。现在，我们给出另一种证明序列，可以改进当前示例查询的权衡 $S\cdot T^3 \cong |\mathbb{D}|^4\cdot |Q_A|^3$。关键思路是根据绑定变量而非自由变量进行分桶，这一思路也被 [11] 使用。我们沿用之前相同的析取规则。对于第一条规则，我们可以如下获得改进的权衡：

$$\begin{aligned}
4\log|\mathcal{D}| + 4\log|Q_A(\mathbf{Z})| &= h_T(Y_1XZ_1|Z_1) + h_S(Z_1) + h_T(Y_1XZ_2|Z_2) + h_S(Z_2) + h_T(Y_2XZ_3|Z_3)\\
&\quad + h_S(Z_3) + h_T(Y_2XZ_4|Z_4) + h_S(Z_4) + 4h_T(\mathbf{Z})\\
&\geq h_S(\mathbf{Z}) + h_T(Y_1XZ_1|Z_1) + h_T(\mathbf{Z}) + h_T(Y_1XZ_2|Z_2) + h_T(\mathbf{Z})\\
&\quad + h_T(Y_2XZ_3|Z_3) + h_T(\mathbf{Z}) + h_T(Y_2XZ_4|Z_4) + h_T(\mathbf{Z})\\
&\geq h_S(\mathbf{Z}) + 4h_T(X\mathbf{Z}) \qquad (S\cdot T^4 \cong |\mathcal{D}|^4\cdot |Q_A|^4)
\end{aligned} \tag{36}$$

上述证明序列 gener 体现了权衡 $S\cdot T^4 \cong |\mathbb{D}|^4\cdot |Q_A|^4$，对于 $|Q_A|=1$ 而言是明显的改进。对于其余规则，我们保 ep 上述推导出的权衡，即 $S\cdot T \cong |\mathbb{D}|^2\cdot |Q_A|^2$.

注意，权衡 $S\cdot T^3 \cong |\mathbb{D}|^4\cdot |Q_A|^3$ 支配第一条规则中的 $S\cdot T^4 \cong |\mathbb{D}|^4\cdot |Q_A|^4$，以及其余规则中的 $S\cdot T \cong |\mathbb{D}|^2\cdot |Q_A|^2$。因此，在所有情形下我们都得到了严格改进的权衡。

在我们的框架中刻画定理 ***F.4***。在结束本节之前，我们提出一种通用策略，用于刻画定理 F.4对一部分层次查询所带来的权衡。具体而言，我们证明，对于任何仅在叶变量中包含$\mathbf{x}_A$ 的布尔层次合取查询与访问模式，都存在一个能够恢复定理 F.4 所得权衡的证明序列。我们将使用$\mathbf{Z}$ 表示叶变量。回顾一下，每个层次查询都具有一个规范排序6，它是一棵有根树，其中查询中每个原子的变量都位于树中同一条根到叶路径上，并且每个原子都是其最低变量的子节点。

我们先描述将要使用的查询分解。考虑该层次查询的规范变量顺序。分解的根袋 T_0 由$\mathbf{x}_A$和变量顺序中的根变量（设为 X）组成。对于 X的每个子节点 Y_i，我们为 T_i添加一个子袋，其中包含以 Y_i为根的子树中的$\mathbf{x}_A$的子集以及 $\text{anc}(Y_i)\cup Y_i$。我们按照自顶向下的方式遍历变量顺序，并处理所有非绑定变量，继续执行这一过程。不难看出，得到的树确实是一个有效的分解。

示例 *F.6.* 图 6b 展示了根据图 6a 中的规范排序生成的查询分解。根袋包含所有绑定变量和 X。X有两个子节点 Y_1，Y_2，因此分解中的根节点有两个子节点。左子节点包含 X，Y_1，以 Y_1为根的子树包含 Z_1，Z_2这些绑定变量，它们被添加到分解中的左侧节点。类似地，右侧节点包含 X，Y_2，Z_3，Z_4。

与贯穿全文的示例类似，我们现在构造由该分解诱导的 PMTD 集合，并生成相应的两阶段析取规则。容易看出，每条析取规则都包含 $S(\mathbf{Z})$。

令 $\mathbf{U}_1 \subseteq \mathbf{Z}$和 $\mathbf{w}_1 = |\mathbf{U}_1|$。对于任意析取规则，它都必须包含形如 $T(\mathbf{U}_1 p)$ 和 $S(\mathbf{Z})$ 的项。这里，p表示所选 T-视图对应的包中除有界变量以外的变量集合。请注意，诱导 PMTD 中的 $M=\emptyset$ 会迫使任意析取规则的头部至少包含一个 T-视图。考虑某个 $v^\star \in U_1$。

$$\begin{aligned}
\mathbf{w}_1\log|\mathcal{D}| + (\mathbf{w}_1-1)\log|Q_A(\mathbf{Z})| &= h_S(\text{anc}(v^\star)\cup v^\star) + \sum_{v\in\mathbf{Z}\setminus v^\star}(h_T(p) + h_S(\text{anc}(v)\cup v \mid p)) + (\mathbf{w}_1-1)h_T(\mathbf{Z})\\
&\geq (\mathbf{w}_1-1)h_T(\mathbf{U}_1 p) + h_S(\text{anc}(v^\star)\cup v^\star) + \sum_{v\in\mathbf{Z}\setminus v^\star} h_S(\text{anc}(v)\cup v \mid p)\\
&= (\mathbf{w}_1-1)h_T(\mathbf{U}_1 p) + h_S(\text{anc}(v_1)\cup v_1 \mid p) + \cdots + h_S(\text{anc}(v_k)\cup v_k\cup v^\star \mid v^\star\cup p) + h_S(\text{anc}(v^\star)\cup v^\star)\\
&\geq (\mathbf{w}_1-1)h_T(\mathbf{U}_1 p) + h_S(\text{anc}(v_1)\cup v_1 \mid p) + \cdots + h_S(\text{anc}(v_k)\cup v_k\cup v^\star \mid v^\star\cup p) + h_S(p\cup v^\star)\\
&\geq (\mathbf{w}_1-1)h_T(\mathbf{U}_1 p) + h_S(\text{anc}(v_1)\cup v_1 \mid p) + \cdots + h_S(\text{anc}(v_k)\cup v_k\cup v^\star)\\
&\geq (\mathbf{w}_1-1)h_T(\mathbf{U}_1 p) + h_S(\text{anc}(v_1)\cup v_1\cdots\cup v_k\cup v^\star)\\
&\geq (\mathbf{w}_1-1)h_T(\mathbf{U}_1 p) + h_S(\mathbf{Z}) \qquad (S\cdot T^{\mathbf{w}_1-1} \cong |\mathcal{D}|^{\mathbf{w}_1}\cdot |Q_A|^{\mathbf{w}_1-1})
\end{aligned}$$

当 w1 尽可能大时，权衡最昂贵。因此，对于 $\mathbf{w}_1=\mathbf{w}$（对应于 $U_1=\mathbf{Z}$），我们得到权衡 $S\cdot T\mathbf{w}^{-1} \cong |\mathbb{D}|\mathbf{w}\cdot |Q_A(\mathbf{Z})|\mathbf{w}^{-1}$。然而，对于给出主导权衡的同一析取规则 $T_0(\mathbf{Z}\ A)\vee S_{\mathbf{x}_A}(\mathbf{Z})$，我们还可以得到另一种证明序列，从而实现类似于 (36) 的改进。

$$\mathbf{w}\log|\mathcal{D}| + \mathbf{w}\log|Q_A(\mathbf{Z})| = h_S(\mathbf{Z}) + \sum_{i=1}^{\mathbf{w}} h_T(\text{anc}(Z_i)\cup Z_i \mid Z_i) + \mathbf{w}\ h_T(\mathbf{Z})$$

[6] [20] 中的原始定义也包含一个依赖函数，但由于分解构造中不会用到它，因此我们省略了该函数。

$$= h_S(\mathbf{Z}) + \sum_{i=1}^{\mathrm{w}} (h_T(X \cup Z_i \mid Z_i) + h_T(\mathbf{Z}))$$

$$\geq h_S(\mathbf{Z}) + \mathrm{w} \cdot h_T(X\mathbf{Z}) \qquad (S \cdot T\mathrm{w} \cong |\mathbb{D}|\mathrm{w} \cdot |Q_A|\mathrm{w})$$