吉林大学 JILIN UNIVERSITY

本科生毕业论文（设计）

中文题目 差分隐私背景下求解近似最小割问题的算法研究

英文题目 **Finding approximate minimum cut in differential privacy**

学生姓名 周宇恒

学　　号 **55210916**

学　　院 计算机科学与技术学院

专　　业 理科试验班 (计算机，唐敖庆班)

指导教师 刘淼　讲师

2025 年 6 月

吉林大学学士学位论文（设计）承诺书

本人郑重承诺：所呈交的学士学位毕业论文（设计），是本人在指导教师的指导下，独立进行实验、设计、调研等工作基础上取得的成果。除文中已经注明引用的内容外，本论文（设计）不包含任何其他个人或集体已经发表或撰写的作品成果。对本人实验或设计中做出重要贡献的个人或集体，均已在文中以明确的方式注明。本人完全意识到本承诺书的法律结果由本人承担。

承诺人：

2025 年 6 月 6 日

差分隐私背景下求解近似最小割问题的算法研究

摘　要

本文聚焦差分隐私框架下的近似最小割问题展开系统性研究，提出一种高效的算法。研究以仙人掌图表示法为切入点，首先剖析仙人掌结构与最小割的内在联系，进而指出表示法的非唯一性问题，并创新性地设计标准化处理算法。该算法通过分离 p 割与 t 割、压缩冗余节点等策略，在确保了输出结果唯一性的同时，维持算法的高效性。此外，本文深入研究边相邻图定义下最小割数量的敏感度，严格证明了其上界，为后续差分隐私算法的噪声添加机制提供理论依据。

基于上述理论成果，本文提出了三种满足 (ε,δ)-差分隐私的近似最小割计算算法：1. 对原图进行隐私化处理，然后用差分隐私发布的最小割值筛选出近似最小割；2. 差分隐私地发布最小割的割值与数量，利用 k 优选择机制获取近似最小割；3. 基于算法 2，融合指数机制与 Karger 收缩算法，显著降低加性误差。其中，最优算法以 $O(\frac{n\log n}{\varepsilon})$ 的加性误差输出具有数量保证的近似最小割，实现了隐私保护和结果可用性的平衡。本研究推进了差分隐私与图算法的交叉融合，为平衡效率、可用性与隐私保护的算法设计提供了新的技术路径。未来可进一步研究最小割数量的特性，探索更高效的隐私保护策略以降低算法加性误差。

关键词：

差分隐私, 最小割问题, 仙人掌图表示法

Finding approximate minimum cut in differential privacy

Author: Zhou Yuheng

Supervisor: Liu Miao

Abstract

This thesis designs a differentially private (DP) algorithm for computing the approximate minimum cuts of a weighted graph. The research first analyzes the relationship between cactus representation and minimum cuts, identifies the non-uniqueness issue, and innovatively designs a standardization algorithm. This algorithm employs techniques such as separating p-cuts and t-cuts, compressing redundant nodes, thereby ensuring the uniqueness of the output while maintaining high computational efficiency. Additionally, the thesis explores the sensitivity of the minimum cut count and rigorously proves its upper bound. This finding provides a theoretical basis for the laplace mechanism in subsequent differential privacy algorithms.

Based on the above results, this thesis introduces three differentially private algorithms for approximating the minimum cuts of weighted graphs: The first algorithm privatizes the graph and finds approximate minimum cuts in the resulting synthetic graph; The second algorithm differentially privately releases the minimum cut value and the minimum cut count, and identifies approximate minimum cuts through top-k selection mechanism; Building on the second algorithm, the third one combines the exponential mechanism with Karger's contraction algorithm. These three algorithms are (ε, δ)-DP, and achieve an optimal additive error of $O(\frac{n \log n}{\varepsilon})$. This study strikes a balance between privacy protection and accuracy. Future work could further analyse the sensitivity of minimum cut count, and explore a more efficient machanism to reduce the additive error.

Keywords:

Differential Privacy, Minimum Cut Problem, Cactus Representation

目 录

第 1 章 绪论 1
1.1 研究背景与意义 1
1.2 研究现状 2
1.3 研究内容与组织架构 4
第 2 章 符号表示与理论基础 5
2.1 图与最小割 5
2.2 最小割数量的估计 6
2.3 仙人掌图表示法 7
2.4 差分隐私 7
第 3 章 仙人掌图表示法标准化算法 10
3.1 Dinitz 仙人掌图表示法简述 10
3.2 仙人掌图表示法的不唯一性 15
3.3 标准化仙人掌图表示法 16
第 4 章 最小割数量的敏感度分析 20
4.1 最小割数量的敏感度 20
4.2 约束条件下的敏感度 20
4.3 平均敏感度 24
第 5 章 差分隐私背景下近似最小割求解算法 26
5.1 基于差分隐私图的算法设计 26
5.2 基于 k 优选择机制的算法设计 27
5.3 加法近似参数的优化 28

第 6 章　总结与展望 31
6.1 工作总结 31
6.2 研究展望 31
参考文献 32
致　谢 34

第 1 章　绪论

1.1　研究背景与意义

随着人工智能的高速发展，数据的重要性愈发凸显，部分领域的研究涉及隐私数据的使用。以临床医学与人工智能交叉研究为例，若科研人员希望构建基于患者身体指标的抑郁症诊断模型，实现抑郁症早期精准识别。则模型可能需要诸如年龄、睡眠质量、激素水平、基因序列等数据。因此，为提升模型诊断能力，收集患者的敏感信息难以避免。此外，随着学术交流合作的不断深入，其它研究者可能申请获取数据用于分析验证，这使得隐私保护面临巨大挑战。数据收集方有责任保障患者的信息安全。所以，如何量化隐私泄露的风险、选择有效的隐私保护方法，已成为亟待解决的重要课题。

隐去隐私标识信息是一种常见的隐私保护手段。例如，在公开数据集时，通常会对姓名、生日、电话号码等可识别个人身份的信息进行隐藏处理。然而，这种保护方法存在固有缺陷。攻击者可借助辅助数据集并结合推理分析技术，重新建立匿名数据与具体个体之间的映射关系。在上文的例子中，若攻击者持有包含姓名与基因对应关系的辅助数据集，就能通过基因信息比对，实现对目标个体的精确识别。上述攻击方式被称为身份识别攻击，已经有研究表明，此类攻击在实际场景中屡见不鲜。[1]

作为应对，数据的处理和计算可以由数据拥有方（或其它受信任的第三方）来完成。具体来说，在数据请求者给出询问内容后，数据拥有方会对询问进行计算，最后返回结果。在这种模式下，数据集没有被直接公开，因此对隐私的保护进一步增强。然而，对单个数据的询问仍然可能导致数据集泄露，多个非对单的查询进行组合后也可以差分地得到个体的信息。[2] 因此，数据拥有方应当为询问设立一套标准，来控制返回结果中的隐私泄露程度。

差分隐私作为一种基于严格数学证明的隐私保护模型，对应对上述挑战提供了有效解决方案。该模型通过量化询问中算法的隐私保护程度，来要求算法添加精心设计的噪声，以确保单个数据对输出结果的影响不显著，从而在保证算法可用性的同时，实现对个体隐私的可靠保护。[3]

差分隐私与传统密码学均以隐私保护为目标，但两者侧的重点存在差异。传统密码学聚焦于防范输出结果以外的隐私泄露风险，而差分隐私则是基于输出内容本身包含隐私信息的假设，通过优化信息发布机制来降低隐私泄露概率。

本文聚焦于差分隐私下的最小割问题。在一个包含 n 个点、m 条边的加权无向图 $G = (V, E)$ 中，割指顶点的二划分 $(X, V \backslash X)$，其权重定义为跨越该划分的

边权总和。常见的最小割问题有 $s-t$ 最小割问题和全局最小割问题两种。对于给定顶点对 $s,t \in V$，$s-t$ 最小割是满足 $s \in X, t \in V \backslash X$ 条件下的权重最小的割 $(X, V \backslash X)$，即实现 s 与 t 分离的最小权重割。根据最大流最小割定理，$s-t$ 最小割问题与 $s-t$ 最大流问题存在对偶关系，两者在数值上相等。[4] 类似地，全局最小割问题旨在找到图的最小权重的割，不难说明，全局最小割的权重等于所有点对 $s.t$ 的 $s-t$ 最小割权重的最小值。最小割的权重能够衡量图的连通性，因此如何求解最小割是图论领域的一个经典问题。

由于差分隐私为算法提出了新的要求，即须控制衡量隐私保护程度的参数，因此经典的最小割求解方法在不再适用。差分隐私要求算法当得到两个近乎相同的输入时，应当有高概率相似的计算过程和结果。对应到最小割算法当中，若两个输入图仅存在一条边的差异，则其输出的每一种最小割的概率之比都接近 1。

设计满足差分隐私的最小割算法，不仅能够拓展算法在实际场景中的应用范围，还有助于加深对差分隐私框架下算法设计方法论的研究。此类算法的核心设计难点在于，如何在严格遵循差分隐私限制的同时，有效控制因添加噪声引入的误差，并同时确保算法输出结果的可用性。

1.2 研究现状

在过去的几十年间，人们提出了众多算法来解决最小割问题。

1993 年，Karger 等人提出了一种基于边收缩的随机算法，用于求解最小割问题，其时间复杂度为 $O(n^4 \log n)$。该研究同时证明，图中不同的最小割的数量上限为 $\frac{n(n-1)}{2}$。[5] 该算法构造简洁，易于理解。算法证明了在随机选择边收缩时，指定最小割有一定概率在算法终止时得以保留，通过重复执行算法，即可以高概率找到一个最小割。1996 年，Karger 等人对算法进行了改进，通过将多次独立重复执行的过程整合为树的分支结构，提升了算法效率，得到时间复杂度 $O(n^2 \log^3 n)$ 的最小割求解随机算法。[6] 此外，每个最小割被找到的概率都可以通过上述方法说明，因此算法在以高概率找到至少一个最小割的同时，也以高概率能找到所有的最小割。

2000 年，Karger 提出了一种基于树包装的随机算法，同样是用于求解最小割问题，其时间复杂度为 $O(m \log^3 n)$。[7] 当算法用于解决寻找所有最小割这一变体问题时，时间复杂度为 $O(n^2 \log n)$。树包装是一个生成树的集合，其中图的每条边被各生成树包含的权重总和不超过其自身边权。Karger 等人定义割与生成树 k 关联为，割的边集与生成树边集的并集大小不超过 k。通过树包装，他们设计了一个构建规模为 $O(\log n)$ 的生成树集合的算法，且满足每个最小割都至少与集合中 $\frac{1}{3}$ 的生成树 2 关联。在此基础上，枚举与这些生成树 2 关联的所有割，并计算其割值，

即可获取全部最小割。目前，Karger 的树包装算法仍是求解最小割问题的最优随机算法。

2021 年，Li 提出了一种求解最小割问题的确定性算法，其时间复杂度为 $O(m^{1+o(1)})$。[8] 这一工作的思路是将 Karger 的树包装算法去随机化。目前，该算法是求解最小割问题的最优确定性算法。

1976 年，Dinitz 等人提出仙人掌图表示法（cactus representation），这个数据结构以一个稀疏化图的形式表示了所有的最小割。[9] [10] 最小割的规模为 $O(n^2)$，因此直接存储的代价较高。前文提到的最小割算法能找到所有最小割，但这些最小割存储在算法的过程变量中，若要提取和利用需要额外开销，因此扩展性受限。仙人掌图表示法创新性地用一个规模为 $O(n)$ 的图实现全部最小割的表示，解决了存储开销问题并为面向图中所有最小割的算法提供了新思路。对于给定图 G，其仙人掌图表示法由仙人掌图 Γ 和映射 $\varphi : V_G \to V_\Gamma$ 构成；给定的仙人掌图和映射满足，任意 G 中的最小割 $(X, V\backslash X)$ 对应的 Γ 中的点集 $\varphi(X)$ 与 $\varphi(V\backslash X)$ 一定可被至少一个 Γ 中的最小割分隔。Dinitz 等人通过仙人掌图表示法证明了图最小割的数量不超过 $\frac{n(n-1)}{2}$，这是该结论最早的证明。

2009 年，Karger 基于树包装最小割算法，提出了一个构造仙人掌图表示法的随机算法，时间复杂度为 $O(m \log^4 n)$。[11] 该算法首先固定一根节点，并计算所有点与边的极小最小割，然后通过点的次极小最小割生成一棵树，最后通过边的极小最小割对树进行连边，形成仙人掌图，完成构造。2024 年，He 等人将仙人掌图表示法构建算法进行优化，得到了时间复杂度为 $O(m \log^3 n)$ 的随机算法，同时，通过算法去随机化处理，进一步得到了时间复杂度 $O(m\text{polylog}(n))$ 的确定性算法。[12]

近年来，差分隐私下的最小割算法研究取得进展。2010 年，Gupta 等人提出一种基于拉普拉斯机制的差分隐私最小割算法。[13] 该算法以 ε-差分隐私得到原图的一个近似最小割，且该近似最小割与真实最小割的割值误差界为 $O(\ln n/\varepsilon)$。此外，他们还设计出满足 (ε, δ)-差分隐私的多项式时间复杂度算法，为差分隐私下的最小割问题提供了高效解决方案。

Gomory-Hu 树是一种与仙人掌图表示相似的结构图，其结构性质及构造算法的隐私化研究在近年来取得重大突破。Gomory-Hu 树以树的形式存储了全点对的 $s-t$ 最小割值，原图的点与 Gomory-Hu 树上的点一一对应，且原图中 $s-t$ 最小割值等于 Gomory-Hu 树中 s 与 t 之间路径上边权的最小值。2021 年，Li 等人提出了一个时间复杂度为 $\tilde{O}(m + n^{3/2}\epsilon^{-2})$ 的随机算法，用以构建 $(1+\epsilon)$-近似 Gomory-Hu 树。[14] 该算法基于其先前提出的最小隔离割方法。[15] 2024 年，Aamand 等人对算法进行了隐私化改造，得到了一个构建 Gomory-Hu 树的 ε-差分隐私的随机算法，且

得到的最小割近似值与真实值相比的加性误差为 $\tilde{O}(m/\varepsilon)$。[14] Gomory-Hu 树隐私化方法为本文提供了灵感，即在最小割问题上，从一些特殊的结构出发完成隐私化。

2024 年，Liu 等人提出了一个图的隐私化算法，该算法能以 (ε, δ)-差分隐私地发布一个合成图，并保证合成图上割的值与其在原图中的真实割值的加性误差为 $\tilde{O}(\frac{\sqrt{nm}}{\epsilon})$。[16] 最小割是一类特殊的割，因此该算法也为差分隐私下的最小割问题提供了一个求解路径。

1.3 研究内容与组织架构

差分隐私的概念从提出至今已有二十年左右的发展历程，其中差分隐私图算法在近几年被广泛关注与研究。由于最小割不唯一，因此最小割问题有两个计算目标：其一是找到至少一个最小割，其二是找到所有最小割构成的割的集合。Karger 的边收缩算法、树包装算法以及 2021 年 Li 的确定性算法都能对这两个计算目标进行求解。而在差分隐私算法方面，现有的 2010 年 Gupta 等人提出的算法仅能输出单个最小割，尚未有算法能够找出所有最小割，存在研究空白。

本文基于已有研究成果，探索用于寻找所有最小割的算法设计。通过分析仙人掌表示及最小割数量的敏感度，综合运用拉普拉斯机制、指数机制、k 优选择机制这三种差分隐私算法，结合图论中 Karger 最小割求解算法，完成了算法的设计。

围绕上述内容，本文共分为六章，具体组织如下：

- 第一章阐明研究的背景与意义，综述领域内的研究现状，提出核心问题与创新点。
- 第二章给出图论与差分隐私的形式化表示体系，回顾重要算法和定理。
- 第三章分析了仙人掌图表示的结构与 p 割、t 割的关联，并提出了同一个图 G 对应的仙人掌图表示非唯一性问题，最后通过构造算法定义了标准形式，并给出了一个高效的仙人掌图表示标准化算法。
- 第四章分析了最小割数量的敏感度，并基于仙人掌图表示建立了精细的敏感度分析框架，并完成敏感度上界的分析。
- 第五章渐进地提出了三个求解最小割的算法，第一个算法基于隐私化图算法和割值筛选法，第二个算法基于差分隐私最小割数量和隐私化 k 优选择机制，第三个算法在第二个算法的基础上，用指数机制和 Karger 算法进行了优化，实现了较低加法误差下的差分隐私近似最小割求解。
- 第六章总结了本文的主要创新，并对未来的优化方向进行了设想。

第 2 章　符号表示与理论基础

2.1　图与最小割

设无向图 $G=(V,E)$ 包含 n 个点与 m 条边，其中 V 为顶点集，E 为边集。

图中连接顶点 u 和 v 的边记作 (u,v)；若图带权，则以 w 来表示边权，即边 $(u,v)\in E$ 的权值为 $w(u,v)$。

对于图中的两个顶点子集 $V_1,V_2\subseteq V$，定义连接二者的边集为

$$E(V_1,V_2)=\{(v_1,v_2)\in E|v_1\in V_1,v_2\in V_2\} \tag{2.1}$$

当 $V_2=V\backslash V_1$ 时，$E(V_1,V_2)$ 简记为 $E(V_1)$。相应的，连接二者的边集的边权和为

$$w(V_1,V_2)=\sum_{v_1\in V_1,v_2,\in V_2}w(v_1,v_2) \tag{2.2}$$

当 $V_2=V\backslash V_1$ 时，$w(V_1,V_2)$ 简记为 $w(V_1)$。

如无特殊说明，本文中图为有限连通图，且允许图存在重边，但不允许存在自环。

下面给出图的最小割及相关定义。

定义 2.1.1. 给定图 $G=(V,E)$，割 $R=(V_1,V_2)$ 是将顶点集划分成两个不相交的非空子集 V_1 和 V_2，即满足 $V_1\neq\emptyset,V_2\neq\emptyset,V_1\cap V_2=\emptyset,V_1\cup V_2=V$。

由于划分得到的点集没有先后顺序，因此 (V_1,V_2) 和 (V_2,V_1) 表示的是同一个割，且只需要给出 V 的非空真子集 V_1 即可唯一确定一个割。这个割可以简化表示为 $\Delta(V_1)=R(V_1,V\backslash V_1)$。不难证明，$\Delta(V_1)=\Delta(V_2)$ 当且仅当 $V_1=V_2$ 或 $V_1=V\backslash V_2$。

定义 2.1.2. 给定图 $G=(V,E)$ 及割 $R=(V_1,V_2)$，割 R 的边集为 $E(V_1,V_2)$，对应简化表示下割 $\Delta(V_1)$ 的边集为 $E(V_1)$。

定义 2.1.3. 给定图 $G=(V,E)$ 及割 $R=(V_1,V_2)$，割的容量（又称割值）为割的边集的边权和，其大小等于 $w(V_1,V_2)$，对应简化表示下割 $\Delta(V_1)$ 的割值为 $w(V_1)$。

定义 2.1.4 (点的度数). 给定图 $G=(V,E)$，顶点 v 的度数定义为

$$\deg(v)=|\{e\in E|v\in e\}| \tag{2.3}$$

定义 2.1.5. 给定图 $G=(V,E)$，一个最小割 $R=(V_1,V_2)$ 满足 R 是图 G 所有可能的割中容量最小的割。

本文用 R_G^* 表示图 G 的最小割集，$r_G^* \in R_G^*$ 表示一个最小割，$\Phi_G = w(r_G^*)$ 表示图 G 的最小割的割值。此外，本文用 $M_G = |R_G^*|$ 表示图 G 的最小割数量。

最小割并不唯一。例如，当图为一条边权均相同的链时，每一条边都对应一个最小割。

定义 2.1.6. 给定图 $G=(V,E)$，α 乘法近似、β 加法近似的最小割 R 满足 $w(R) \le \alpha \cdot \Phi_G + \beta$。

在描述近似最小割时，若 $\alpha = 1$，则无需考虑该乘法参数，若 $\beta = 0$，则无需考虑该加法参数。

定义 2.1.7 ($S-T$ 最小割)**.** 给定图 $G=(V,E)$ 和图上互不相交的两个非空点集 $S,T \subset V$，$S-T$ 最小割是满足 $S \subseteq V_1, T \subseteq V_2$ 的割 (V_1,V_2) 中权值最小的割。

记 $S-T$ 最小割的割值为 $\Phi(S,T)$。当 S 和 T 均为只包含一个点的集合时，可以得到 $s-t$ 最小割的定义。

2.2 最小割数量的估计

最小割可以由 Karger 的树包装算法快速求得。

定理 2.2.1 (最小割算法)**.** [7] 存在一个随机算法，能在 $O(m \log^3 n)$ 的时间复杂度内以高概率找到一个最小割。

Karger 的收缩算法也是一个高效的求解最小割的随机算法，通过这个算法可以对最小割的数量进行估计。此外，Karger 的收缩算法还能较好的解决近似最小割的求解问题。

定义 2.2.1 (点收缩)**.** 给定图 $G=(V,E)$ 和点集的一个子集 $X \subseteq V$，点收缩收缩过程为，在图 G 中新建一个点 x，对于点 $y \in V \backslash X$，其向 x 连一条边权为 $\sum_{x' \in X} w(y,x')$ 的边（若边权为 0 则不连边），并将 X 及与其相连的边全部删除。

定义 2.2.2 (边收缩)**.** 给定图 $G=(V,E)$ 和图上的一条边 $(u,v) \in E$，边 (u,v) 的收缩定义为对 $\{u,v\}$ 这一点集执行点收缩。

Karger 的收缩算法的思路如下：以均匀分布随机选择图的一条边，并对这条边进行边收缩，重复该步骤直到图中的顶点数量等于一个预先设定的参数 k 为止。算法执行完时，如果一个割的割边集中没有边被收缩，那么称这个割是有效的。

定理 2.2.2. *[5]* 一个给定的最小割，在算法进行到图被收缩至 k 个顶点时，有效的概率是 $\Omega((n/k)^{-2})$。

当 $k=2$ 时，给定的最小割仍然有效的概率为 $\Omega((n/2)^{-2})$，且由于 $k=2$，算法终止时有且仅有一个割有效。由概率分布的累计不能超过 1，可以得到一个最小割数量的上界。

定理 2.2.3. 给定图 $G=(V,E)$，图中最小割数量至多为 n^2。

定理2.2.2也可以推广到近似最小割的情况。

定理 2.2.4. *[5]* 一个给定的 α 乘法近似最小割，在算法进行到图被收缩至 $k(k \geq \lfloor 2\alpha \rfloor)$ 个顶点时，有效的概率是 $\Omega((n/k)^{-2\alpha})$。

定理 2.2.5. *[17]* 给定图 $G=(V,E)$，图中 α 乘法近似最小割的数量至多为 $n^{2\alpha}$。

2.3 仙人掌图表示法

仙人掌图表示法是最早由 Dinitz 等人提出的结构图，该结构图保留了原图所有的最小割信息，且结构图是仙人掌图。

定义 2.3.1. 图 G 为仙人掌图当且仅当，对于任意边 $e \in V_G$ 都满足 e 至多属于一个简单环。

定理 2.3.1 (仙人掌图表示法). *[9]* 给定带权图 G，存在一个仙人掌图 Γ 和映射 $\varphi : V_G \to V_\Gamma$，满足：

- 对于点 $v_1, v_2 \in V_G$，$\varphi(v_1)=\varphi(v_2)$ 当且仅当图 G 不存在最小割 $R=(V_1,V_2)$ 使得 $v_1 \in V_1, v_2 \in V_2$；
- 对于图 G 的任意一个最小割 $R=(V_1,V_2)$，都满足 $(\varphi(V_1), V_\Gamma \backslash \varphi(V_1))$ 是图 Γ 的一个最小割。

2.4 差分隐私

差分隐私是一种针对敏感输入数据集计算的隐私定义，它聚焦于对个体隐私的保护。通俗来说，差分隐私要求在两个几乎相同的输入数据下，算法的计算过程应当同样保持几乎一致。当输入数据仅改变一个个体或者说一个元素时，任何输出结果的概率增幅不能超过一个很小的常数 e^{ε}。图论算法中，输入的元素单位为边，而边权可以视作叠加边的数量，因此，图的差分隐私算法需要考察两个仅相差一条边的图的输出情况。接下来给出差分隐私在图论中的形式化定义。

定义 2.4.1 (边相邻). 称图 G, G' 边相邻当且仅当满足以下条件：

- 顶点集相等：$V_G = V_{G'}$；
- 存在唯一边 $(u,v) \in V^2$，使得 $|w(u,v) - w_{G'}(u,v)| = 1$；
- 对于任意其余边 $(u',v') \in V^2 \backslash \{(u,v)\}$，满足 $w(u,v) = w_{G'}(u,v)$。

定义 2.4.2 (差分隐私). *[18]* 图算法 A 是 (ε, δ) 差分隐私的，当且仅当对于任意的边相邻的图输入 G, G' 和输出值域的子集 O，有

$$\mathbb{P}[A(G) \in O] \leq e^{\varepsilon}\mathbb{P}[A(G') \in O] + \delta \tag{2.4}$$

特别地，如果 $\delta = 0$，算法满足 ε 差分隐私。

当 $\delta = 0$ 时，差分隐私也被称为纯差分隐私。当 $\delta \neq 0$ 时，差分隐私也被称为近似差分隐私。近似差分隐私不能严格限定概率增幅，但是当 δ 设定为一个极小的值时，仍然是一个有效的结果。

定理 2.4.1 (基本组合). *[19] [20]* 设 $\varepsilon_1, \ldots, \varepsilon_t > 0$ 且 $\delta_1, \ldots, \delta_t > 0$。若运行 t 个算法，其中第 i 个算法是 $(\varepsilon_i, \delta_i)$ 差分隐私的，那么整个算法是 $(\varepsilon_1 + \ldots + \varepsilon_t, \delta_1 + \ldots + \delta_t)$ 差分隐私的。

基本组合定理表明，一个差分隐私序列仍然具备差分隐私性。这使得在设计算法时，可以将目标拆解成若干个差分隐私步骤，从而降低了设计难度。

定义 2.4.3 (拉普拉斯分布). 如果随机变量的概率密度函数分布为

$$f(x|\mu, b) = \frac{1}{2b} exp\left(-\frac{|x-\mu|}{b}\right) \tag{2.5}$$

那么它就是拉普拉斯分布。

拉普拉斯分布的函数关于 $x = \mu$ 轴对称，且对称轴的两侧分别是一个指数分布。拉普拉斯分布的期望为 μ，方差为 $2b^2$。特别的，当 $\mu = 0$ 时，记该分布为 Lap(b)。

定理 2.4.2 (拉普拉斯机制). *[21]* 给定一个将图 G 映射到 $\mathbb{R}^d$ 的函数 f，其满足对于任意两个边相邻图 G 和 G'，有 $\|f(G) - f(G')\|_1 \leq \Delta$，则发布加入独立同分布随机噪声 $X_i \sim Lap(\Delta/\varepsilon)$ 的结果

$$f(G) + (X_1, \ldots, X_d) \tag{2.6}$$

满足 ε 差分隐私。

定理 2.4.3 (差分隐私图)**.** [16] 令 $\varepsilon \in \left(\frac{1}{n}, \frac{1}{2}\right)$ 和 $0 < \delta < \frac{1}{2}$ 为差分隐私参数。对于任意有 n 个点和 m 条边（$m \geq n$）的带权图 G，存在一个 (ε, δ)-差分隐私算法，能以至少 $1 - o(1)$ 的概率输出一个合成图 $\hat{G}$，满足对于合成图中任意不相交的点集 $S, T \subseteq V_G$，都有

$$|w_G(S,T) - w_{\hat{G}}(S,T)| = O\left(\frac{\sqrt{nm}}{\varepsilon}\log^3\left(\frac{n}{\delta}\right)\right) \tag{2.7}$$

拉普拉斯机制给出了一种差分隐私的方法，同时说明了敏感度 Δ 与误差之间的关联。定理中噪声拉普拉斯函数的系数表明，敏感度越大，噪声的标准差也随之线性变大。

定义 2.4.4 (k 优选择问题)**.** 给定 m 个数构成的序列 $(x_1, x_2, \ldots, x_m)$，k 优选择问题要求输出值前 k 小的数的下标 $i_1, i_2, \ldots, _k$ 以及每个数的值。当 $k = 1$ 时，问题也被称为最优选择问题。

在差分隐私下，k 优选择问题的两个输入 $(x_1, x_2, \ldots, x_m)$ 和 $(x'_1, x'_2, \ldots, x'_m)$ 相邻当且仅当 $\|x - x'\|_\infty \leq 1$。

定理 2.4.4 (指数机制)**.** [22] 对于最优选择问题，按分布

$$Pr[y = i] = \frac{exp(-\frac{\varepsilon}{2}x_i)}{\sum_{j\in[m]} exp(-\frac{\varepsilon}{2}x_j)} \tag{2.8}$$

输出下标 y，则有

$$Pr[x_i > x_{min} + \frac{2\ln(\frac{m}{m_{min}})}{\varepsilon} + \frac{2t}{\varepsilon}] \leq e^{-t} \tag{2.9}$$

其中 x_{min} 为 x 的最小取值，m_{min} 为满足最小取值的下标数量。选择合适的 t，该方法能以高概率得到一个 $O(\frac{2\log m}{\varepsilon})$ 近似的最小值，且算法是 ε-差分隐私的。

对指数机制进行扩展，可以得到 k 优选择机制。

定理 2.4.5 (k 优选择机制)**.** [23] 令 $\varepsilon \leq 0.2$ 和 $\delta < 0.05$ 为差分隐私参数。对于任意 $m \geq 2$，存在一个 (ε, δ)-差分隐私算法，输出一组 k 优选择问题的答案下标，并能以高概率实现，输出每个数的 $O(\frac{\sqrt{k\log(m/\delta)}}{\varepsilon})$ 近似值，并保证对于每个整数 $i \in [1, k]$，第 i 优值是真实值的 $O(\frac{\sqrt{k\log(m/\delta)}}{\varepsilon})$ 近似。

第 3 章　仙人掌图表示法标准化算法

仙人掌图表示法保留了原图所有的最小割信息，因此如果能差分隐私地输出图的仙人掌图表示法，那么也就完成了差分隐私下近似最小割的求解。然而，Dinitz 的论文[9] 中提供的仙人掌图表示法定义存在一定局限性，为隐私化带来了障碍。本章将从仙人掌图表示法的定义出发，给出一个仙人掌图表示法的标准化算法，该算法是差分隐私下仙人掌图表示求解算法的前置基础。同时，借助对仙人掌图表示法的分析，本文也将在后续章节中完成对最小割性质的分析。

3.1　Dinitz 仙人掌图表示法简述

仙人掌图表示法的标准化算法，是在 Dinitz 仙人掌图表示法[9] 的基础上完成设计的。本节将简述仙人掌图表示法的构建与图本身的关联。根据定义2.3.1，给定任意带权图 G，存在仙人掌图 Γ 和映射 φ 作为其仙人掌图表示法。

仙人掌图表示法能够表示的割分为两类，其对应原图 G 中的 p 割与 t 割，这种对应得益于割本身具有的次模性。此处将首先介绍次模性和其推论，它们是后文分析中的重要工具。

引理 3.1.1 (割的次模性)**.** *[24] 给定图 $G=(V,E)$ 和图的两个割 $\Delta(X),\Delta(Y)$，有*

$$w(X)+w(Y)\geq w(X\cup Y)+w(X\cap Y) \tag{3.1}$$

引理 3.1.2 (次模性推论)**.** *给定图 G 和图中两个相交的割 $\Delta(X),\Delta(Y)$，则有如下结论：*

- $w(X\cap Y)=\Phi_G$
- $w(X\cap Y, X\cap(V\backslash Y))=\frac{\Phi_G}{2}$
- $w(X\cap Y,(V\backslash X)\cap(V\backslash Y))=0$

证明　首先证明第一条结论。根据割的次模性，可以得到

$$2\Phi_G\leq w(X\cup Y)+w(X\cap Y)\leq w(X)+w(Y)=2\Phi_G \tag{3.2}$$

因此，$w(X\cup Y)=w(X\cap Y)=\Phi_G$。

接下来证明第二条结论。根据第一条结论的对称性，$w(X\cap Y)=w(X\cap$

$(V\backslash Y)) = \Phi_G$；由于 $\Delta(Y)$ 是最小割，所以 $w(Y) = \Phi_G$。根据 w 的定义，有

$$w(X \cap Y) + w(X \cap (V\backslash Y)) = w(Y) + 2w(X \cap Y, X \cap (V\backslash Y)) \tag{3.3}$$

因此，$w(X \cap Y, X \cap (V\backslash Y)) = \frac{\Phi_G}{2}$。

最后证明第三条结论。根据第二条结论的对称性 $w(X\cap Y, X\cap(V\backslash Y))+w(X\cap Y,(V\backslash X)\cap Y) = \frac{\Phi_G}{2}$。根据 w 的定义，有

$$w(X\cap Y, X\cap(V\backslash Y))+w(X\cap Y,(V\backslash X)\cap Y)+w(X\cap Y,(V\backslash X)\cap(V\backslash Y)) = w(X\cap Y) = \Phi_G \tag{3.4}$$

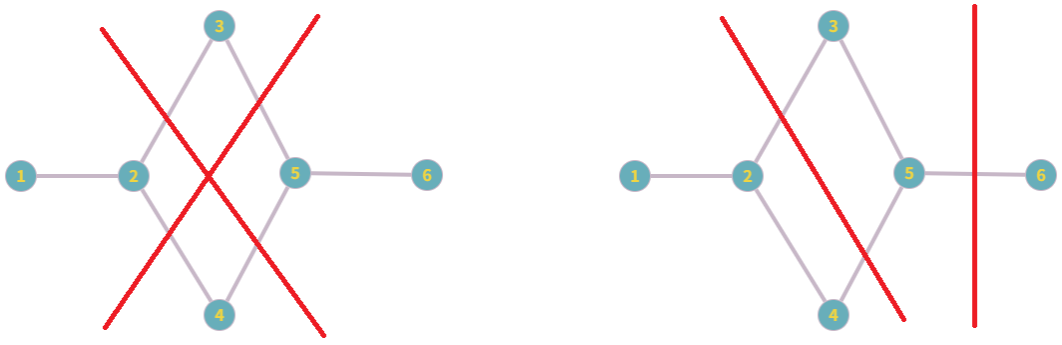
因此，$w(X \cap Y, (V\backslash X) \cap (V\backslash Y)) = 0$。 □

接下来将给出若干定义来解释 p 割和 t 割这两个概念。在仙人掌图表示法的最初构造方法中，求解 p 割与 t 割是流程的第一步。

定义 3.1.1 (割的平行与相交). 设 $R = (X, Y)$ 和 $R' = (X', Y')$ 是图中的不同割，它们的相对位置存在两种可能情况：

- 集合 $X \cap X'$、$X \cap Y'$、$Y \cap X'$、$Y \cap Y'$ 均非空；
- 这些集合中存在空集。

在第一种情况下，割 R 和 R' 被称为相交的一对割，在第二种情况下，它们被称为平行的一对割。



(a) 相交的一对割　　(b) 平行的一对割

图 3-1　割的平行与相交例

如图3-1所示，灰色的直线代表图中节点的连边，红色直线代表割，左右两图分别对应相交和平行的割的例子。

定义 3.1.2 (最小割图). 给定图 G 和其最小割集 $R^*(G)$，最小割图按如下方式生成：

- 对于每个最小割 $r \in R^*(G)$，在最小割图中新建一个与之相对应的点。
- 若两个最小割 r_1, r_2 相交，则在最小割图中对应的点之间连一条边。

定义 3.1.3 (p 割与 t 割). 如果图 G 的一个最小割 R 与其他任何最小割都平行，就称它为 p 割；否则，称它为 t 割。

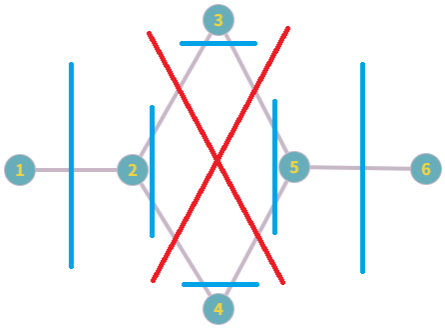


图 3-2 p 割和 t 割例

如图3-2所示，蓝色直线为图中的 p 割，红色直线为图中的 t 割。

仙人掌图表示法的构造分为两步，第一步是构建能表示所有 p 割的树表示法，第二步是用环状结构图替代 p 束顶点。

定义 3.1.4 (处于两个 p 割之间的点和割). 给定两个 p 割 $R = (X, Y)$ 和 $R' = (X', Y')$，不妨假设 $X \cap Y' = \emptyset$。称点 v 处于 R 和 R' 之间当且仅当 $v \in X' \cap Y$。称割 $R'' = (X'', Y'')$ 处于 R 和 R' 之间当且仅当 $X \subset X''$ 且 $Y' \subset Y''$。

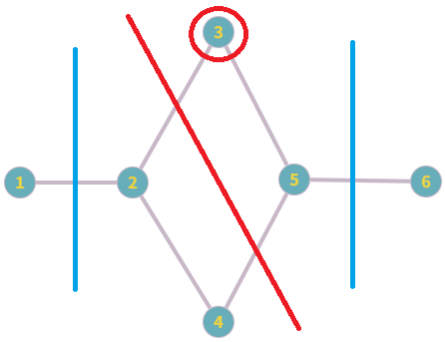


图 3-3 处于两个 p 割之间的点和割例

如图3-3所示，红色圈中顶点以及红色直线代表的割处于两个蓝色直线表示的 p 割之间。

定义 3.1.5 (相邻的 p 割)**.** 给定两个 p 割 R 和 R'，称它们相邻当且仅当不存在处于它们之间的 p 割。

对于一个 p 割 $R=(X,V\backslash X)$，记 $S(X)$ 为所有与 R 相邻且分隔开 X 中顶点的 p 割的集合。需要注意，$S(X)$ 与 $S(V\backslash X)$ 对应该 p 割的两侧。

定义 3.1.6 (p 束)**.** 给定一个 p 割集合 S，称其为 p 束当且仅当存在一个 p 割 $R=(X,V\backslash X)$ 满足 $S=S(X)\cup\{R\}$。特别的，当 S 中仅包含一个 p 割 $R=(X,V\backslash X)$ 时，该 p 束为叶 p 束，用 S_X 表示，记该 p 束由 R 的 X 一侧得到。

图的所有 p 束可以通过枚举所有 p 割以及其两侧得到，也就是说，图的 p 束集合是有限且唯一确定的。

定义 3.1.7 (p 束内部的顶点和 t 割)**.** 顶点 v 属于 p 束 S 当且仅当对于任意一对 p 束中的 p 割都满足该顶点在这两个割之间，此类顶点的集合记作 $V(S)$。t 割属于 p 束 S 当且仅当对于任意一对 p 束中的 p 割都满足该 t 割在这两个割之间。特别的，对于叶 p 束 S_X，顶点 v 属于 S 当且仅当 $v\in X$，t 割不会属于叶 p 束。

需要特别说明，根据割的次模性推论，一组在最小割图中连通的 t 割对应至少 4 个 p 割，因此若 t 割属于叶 p 束 S_X，则与 S_X 只包含一个 p 割这一结论矛盾。

定义 3.1.8 (相邻的 p 束)**.** 两个 p 束 S,S' 相邻当且仅当 $S\cap S'$ 非空。

当 S,S' 相邻时，S,S' 分别位于 R 的两侧，因此 $S\cap S'$ 中的元素是唯一的。在这些概念的基础上，Dinitz 等人提出了可以表示所有 p 割的树表示法。

定理 3.1.3 (树表示法)**.** [9] 给定带权图 G，存在一个棵树 Λ 和映射 $\phi:V_G\to V_\Lambda$，满足：

- 对于点 $v_1,v_2\in V_G$，$\phi(v_1)=\phi(v_2)$ 当且仅当图 G 不存在 p 割 $R=(V_1,V_2)$ 使得 $v_1\in V_1,v_2\in V_2$；
- 图 G 的最小割 $R=(V_1,V_2)$ 与图 Λ 的最小割 $(\phi(V_1),V_\Lambda\backslash\phi(V_1))$ 一一对应。

定理 3.1.4 (树表示法的性质)**.** [9] 图 G 的树表示法 Λ 有以下两个性质：

- Λ 上每一条边的边权都等于最小割的割值。
- Λ 中的点与 p 束一一对应，边与 p 束的相邻关系一一对应。

为了辅助仙人掌图表示法的标准化算法，接下来的引理将说明树表示法的唯一性。

引理 3.1.5 (树表示法的唯一性). 给定带权图 G，其树表示法 (Λ, ϕ) 唯一。

证明 使用反证法,不妨假设图 G 有两个不相同的树表示法 (Λ, ϕ) 和 (Λ', ϕ')。首先证明 $\phi = \phi'$。根据定理3.1.3的第一条性质，若存在 $v_1, v_2 \in V_G$ 满足 $\phi(v_1) = \phi(v_2)$ 但 $\phi(v_1) \neq \phi(v_2)$，则将两点分隔开的 p 割的存在性出现矛盾。

接下来证明 $\Lambda = \Lambda'$。图 G 的 p 束集合有限且唯一确定，定理3.1.4表明树表示法的树结构 Λ 中，顶点与 G 的 p 束一一对应，边与 p 束的相邻关系一一对应。 □

在树表示法中，顶点与 p 束一一对应，每个 t 割都唯一属于一个 p 束。接下来将分析 t 割在仙人掌图表示法中的表示形式。

定义 3.1.9 (原子). 给定图 $G = (V, E)$ 和 G 中割的集合 $\mathcal{C}$。$\mathcal{C}$ 的原子是一个 V 的划分 P 的所有划分块，其中 P 满足

- 对于任意割 $(X, V \backslash X) \in \mathcal{C}$ 以及任意原子 $A \in P$，满足 $A \subseteq X$ 或 $A \subseteq V \backslash X$。
- P 是满足条件的最粗划分，也就是说对于任何满足条件的划分 P'，都有 $P' \preceq P$。

给定的一组割会将图的点集划分成若干块，每个划分块对应一个原子。

定理 3.1.6 (最小割和 p 割对应的原子集等价). [9] 给定图 G，由所有最小割构成的割集得到的原子集和由所有 p 割构成的割集得到的原子集等价。

这一定理表明，一个 p 束内不会同时包含顶点和 t 割。

定义 3.1.10 (p 束结构图 G_S). 定义 G_S 为图 G 中 p 束 S 的结构图，其生成方式如下：

- 将 G_S 初始化为 G。
- 枚举 S 中的 p 割 R，并对被该割与 S 分隔开的点集执行点收缩（同时记收缩得到的点为 x_R）。

定义 3.1.11 ($\hat{c}$ 环). 所有边的权重都为 $\hat{c}/2$ 的环被称为 $\hat{c}$ 环。

定理 3.1.7 (含 t 割的 p 束的结构图为环). [9] 如果一个 p 束 S 存在内部的 t 割，那么图 G_S 是以顶点 x_R（$R \in S$）构成的 $\hat{\Phi_G}$ 环。

定理3.1.7说明了当 p 束 S 内存在 t 割的情况，其核心结论主要有两点。第一个结论是当 S 内存在 t 割时，则 $V(S) = \emptyset$，这对应了 $\hat{\Phi_G}$ 环仅由 x_R 即 S 外部的顶

点构成这一结果；第二个结论是 S 内的 t 割恰好是将环分成两部分的割（每一部分至少包含两个顶点），且这些 t 割在最小割图中恰好构成一个联通块。

Dinitz 等人在树表示法中用环替换含 t 割的 p 束对应的顶点，得到了仙人掌图表示法的构造算法。因此，在仙人掌图表示法中，非环边代表 p 割，t 割仅由环边二元组表示。

3.2 仙人掌图表示法的不唯一性

仙人掌图表示法由仙人掌图 Γ 和映射 φ 共同构成。首先，本节将定义仙人掌图表示法对应的割集，这一概念展示了，在已知仙人掌图 Γ 和映射 φ 的情况下，如何还原出原图的最小割集合。

还原的过程需要用到仙人掌图表示法中映射 φ 的逆映射 φ^{-1}，该逆映射是从 V_Γ 到 $\mathcal{P}(V_G)$ 的映射。分析 φ 的性质可得，Γ 中的点对应着 G 中的 0 个、1 个或多个顶点。下面将给出仙人掌图表示法对应割集的表示形式，并由此定义仙人掌图表示法的等价性。

定义 3.2.1. 给定点集 V，仙人掌图表示法由仙人掌图 Γ 和映射 $\varphi : V \to V_\Gamma$ 构成，其对应的割集为

$$CutSet(\Gamma,\varphi) = \{(X,Y) \in R_\Gamma^* | (\bigcup_{x\in X} \varphi^{-1}(x), \bigcup_{y\in Y} \varphi^{-1}(y))\} \tag{3.5}$$

两个仙人掌图表示法 $(\Gamma,\varphi),(\Gamma',\varphi')$ 等价当且仅当 $CutSet(\Gamma,\varphi) = CutSet(\Gamma',\varphi')$。

一个仙人掌图表示法唯一对应了原图的最小割集，但原图的最小割集可能对应多个仙人掌图表示法。

引理 3.2.1. 图的仙人掌图表示法不具有唯一性。

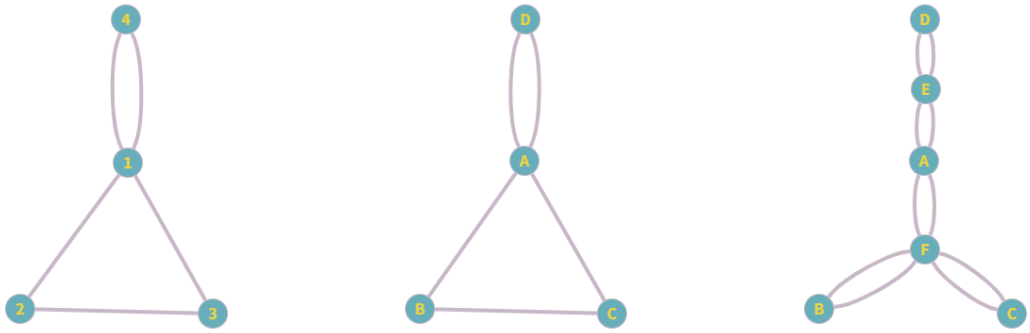
证明 证明将通过给出一个图 G 以及其两个不相同的仙人掌图表示法 $(\Gamma,\varphi),(\Gamma',\varphi')$ 的构造来完成。

图 3 – 4 给出了 G,Γ,Γ' 的结构，其中 $n = 4$。仙人掌图表示法的映射部分 φ,φ' 取值为

$$\varphi = \varphi' = \{(1,A),(2,B),(3,C),(4,D)\} \tag{3.6}$$

在该构造下，图 G 的最小割集为

$$\{(\{1,4\},\{2,3\}),(\{1,2,4\},\{3\}),(\{1,3,4\},\{2\}),(\{1,2,3\},\{4\})\} \tag{3.7}$$



(a) 图 G (b) 图 Γ (c) 图 Γ'

图 3-4 同一个图的两种仙人掌图表示法

仙人掌图表示法 (Γ, φ) 中，Γ 与 G 结构相同，因此最小割一一对应。仙人掌图表示法 (Γ', φ') 进行了两个改动：第一个改动是通过加入节点 F，使 A, B, C 构成的三元环变成三个二元环，三元环所表示的最小割分别转而由这三个二元环所表示，所以该改动不影响割集，以割 $(\{1,3,4\},\{2\})$ 为例，其由 Γ 中的三元环边 $(A,B),(B,C)$ 共同表示，而到了 Γ' 中，其转而由两条 (B,F) 边构成的二元环表示；第二个改动是在 A 和 D 之间加入 E，这使得 Γ' 中的最小割数量增加，即 $(\{A,B,C,F\},\{D\})$ 扩展成了 $(\{A,B,C,F\},\{D,E\})$ 和 $(\{A,B,C,E,F\},\{D\})$ 两个最小割，然而由于 $\varphi^{-1}(E)=\emptyset$，因此这两个 Γ' 中的割对应 G 的同一个割，改动不影响割集。综上，$(\Gamma,\varphi),(\Gamma',\varphi')$ 都是 G 的仙人掌图表示法，图的仙人掌图表示法不具有唯一性。 □

图的最小割集是唯一的，所以同一个图的仙人掌图表示法对应的割集相同，即仙人掌图表示法等价。也就是说，图的仙人掌图表示法不具有唯一性，但具有等价性。

3.3 标准化仙人掌图表示法

本节将设计仙人掌图表示法等价类的标准元，并给出仙人掌图表示法的标准化算法。仙人掌图表示法的生成算法结束后，将输出通过标准化算法转化为等价类的标准元，那么就可以确保对于相同的原图，可以得到具有唯一性的仙人掌图表示法。

对于最小割集相同的边相邻图，常规算法可能受边集差异的影响，得到两个等价但不相等的仙人掌图表示法。在差分隐私场景下，算法需要确保等价的仙人掌图表示法被判定为相同，本节提出的标准化算法可以解决这一问题。

本节中仙人掌图表示法的标准化共分为两部分：第一部分通过构造来定义仙人掌图表示法的标准元，第二部分给出将仙人掌图表示法转化为其标准元的高效算法。

根据引理3.1.5，树表示法 Λ 可以表示所有 p 割，且其形态具有唯一性。根据定理3.1.7，p 束的结构图可以表示该 p 束的所有 t 割且其形态具有唯一性。因此，如果将树表示法和每个 p 束的结构图用一种确定性方法进行合成，得到的仙人掌图表示法也是唯一的，且恰好能表示所有的最小割。本文设计了一种能使合成产生的仙人掌图表示法拥有较好性质的方法，并将用该方法生成的图设置为该仙人掌图表示法的标准元。由图 G 生成仙人掌图表示法标准元的具体算法 ALG_{gen} 在算法1中给出，同时，称 $ALG_{gen}(G)$ 为图 G 的标准仙人掌图表示法。

算法 1 图 G 的仙人掌图表示法构造算法 ALG_{gen}

输入：图 G
输出：仙人掌图表示法 (Γ, φ)
1: 计算图 G 的所有 p 割，t 割。
2: 计算图 G 中的所有 p 束。
3: 为每个 p 束 S 新建一个 Γ 中的点 v_S，并更新 S 内的原图顶点到 v_S 的映射 φ。
4: 若两个 p 束 S, S' 相邻，则为其在 Γ 中的点 $v_S, v_{S'}$ 连一条边，得到图 G 的树表示法。
5: 若 p 束 S 中有 t 割，则将 v_S 替换为 G_S。若一条边连向 v_S，设该边代表的 p 割为 R，则请其重新连向 G_S 中的点 x_R。
6: **return** (Γ, φ)

接下来的定理将说明，若两个图的仙人掌图表示法等价，则其标准仙人掌图表示法也相同。

定理 3.3.1. 给定图 G, G'，已知它们的仙人掌图表示法分别为 (Γ, φ) 和 (Γ', φ')。若 (Γ, φ) 和 (Γ', φ') 等价，则 $ALG_{gen}(G) = ALG_{gen}(G')$。

证明 条件中提到，(Γ, φ) 和 (Γ', φ') 等价，根据定义3.2.1，有 $CutSet(\Gamma, \varphi) = CutSet(\Gamma', \varphi')$。由仙人掌图表示法的定义，$CutSet(\Gamma, \varphi)$ 和 $CutSet(\Gamma', \varphi')$ 分别对应 G 的最小割集和 G' 的最小割集。$ALG_{gen}(G)$ 的步骤仅用到了 G 的最小割集的信息，因此若最小割集相同，算法输出也相同，即 $ALG_{gen}(G) = ALG_{gen}(G')$。 □

对于一个仙人掌图表示法,其标准元为其原图 G 的标准仙人掌图表示法 $ALG_{gen}(G)$。算法1以构造的形式定义了标准仙人掌图表示法及标准元，但不能直接用于标准元的求解中。主要问题有两方面，一方面是其用到了原图 G 的信息，因此在仅有仙人掌图表示法时不能直接使用；另一方面是，p 割、t 割、p 束的计算复杂度较高，易成为算法的效率瓶颈。

因此，本节接下来将给出一仙人掌图表示法的标准化算法 ALG_{std}，其可以将现有工作生成的仙人掌图表示法直接转化为等价类内的标准元，而不需要获取原

算法 2 仙人掌图表示法标准化算法 ALG_{std}

输入：仙人掌图表示法 (Γ, φ)
输出：标准仙人掌图表示法 (Γ', φ')

设仙人掌图表示法 (Γ, φ) 的最小割的割值为 c；
对所有二元环，将两条环边合并为一条边，边权求和； ▷ 等价转换
for Γ 中的简单环 C **do**
    for C 中的节点 k **do**
        在 Γ 中新建顶点 k', k'' 来替换 k， ▷ 分离简单环中的 p 割
        令 $\varphi^{-1}(k'') = \varphi^{-1}(k), \varphi^{-1}(k') = \emptyset$，
        在 k', k'' 之间连一条边权为 c 的边；
        for 与 k 相连的边 e **do** ▷ 边的重连
            if e 是环 C 上的边 **then**
                将 e 的 k 一端替换成 k'；
            else if e 不是环 C 上的边 **then**
                将 e 的 k 一端替换成 k''；
            end if
        end for
    end for
end for
对 Γ 中的所有三元环执行点收缩； ▷ p 割已被分离，三元环为冗余结构
for Γ 中度数为 2 的点 v **do** ▷ 删除冗余空节点
    if 若 v 不处于任何一个简单环上且 $\varphi^{-1}(v) = \emptyset$ **then**
        找到与 v 相连的边 $(v, u_1), (v, u_2)$；
        删除点 v 以及与其相连的边；
        加入边 (u_1, u_2)；
    end if
end for
return (Γ', φ')

图信息。与此同时，ALG_{std} 高效利用了输入仙人掌图表示法的信息，复杂度相比 ALG_{gen} 显著降低。其具体实现如算法2所示。

回顾引理3.2.1中给出的仙人掌图表示法不唯一性的例子，可知产生的差异有三元环的两种表达形式，以及冗余链等情况。算法2通过分离环上的 p 割以及冗余结构的处理，确保了到标准仙人掌图表示法的转换。

定理 3.3.2. *给定图 G 和其仙人掌图表示法 (Γ, φ),则 $ALG_{std}((\Gamma, \varphi)) = ALG_{gen}(G)$*

证明 首先，需要证明 $ALG_{std}((\Gamma, \varphi))$ 算法得到的环与 $ALG_{gen}(G)$ 的环一一对应。在仙人掌图表示法中，非环边代表的最小割一定是 p 割，环中的一对边代表的最小割可能是 p 割也可能是 t 割。根据算法1，$ALG_{gen}(G)$ 的 p 割全部对应了树表示法的边，环仅用于表示 t 割，也就是说，环上相邻边构成的 p 割不予考虑。对于一个 $ALG_{gen}(G)$ 中的简单环 C，其表示的 t 割在最小割图中恰好构成一个连通块，因此这些割在 (Γ, φ) 中一定对应一个相同大小的环 C'。需要特别说明的是，对于这个仙人掌图表示 (Γ, φ) 中的环 C'，环上相邻的两条边表示一个 p 割，这些 p 割在算法中通过 (k', k'') 这条非环边重新表示，因此此时环不再具有表示 p 割的作用。若环 C' 不表示任何 t 割，那么其一定是一个二元环或三元环，同时由于其不具有表示 p 割的作用，这些环是冗余结构，在算法中被消除。综上，$ALG_{std}((\Gamma, \varphi))$ 算法得到的环与 $ALG_{gen}(G)$ 的环一一对应。

接下来证明将所有环缩成点后，$ALG_{std}((\Gamma, \varphi))$ 的树结构和 $ALG_{gen}(G)$ 的树结构相同。(Γ, φ) 的 p 割由非环边和环共同表示，而在处理环的过程中，所有环表示的 p 割转而以 (k', k'') 的非环边形式表示。因此，处理完环之后，(Γ, φ) 的树结构和 $ALG_{gen}(G)$ 都能表示所有 p 割。此时，(Γ, φ) 的树结构表示了所有 p 割，但不一定是树表示法，因为树表示法的边与 p 割一一对应，但是树结构可以存在多条边对应同一个 p 割的情况。代表同一个 p 割的两条边之间的树上路径的点 v 都满足 $\varphi^{-1} = \emptyset$，因此，ALG_{std} 可以通过收缩所有这样的冗余顶点，将树结构转化为树表示法。根据定理3.1.3，树表示法具有唯一性。综上，$ALG_{std}((\Gamma, \varphi))$ 的树结构和 $ALG_{gen}(G)$ 的树结构相同。

结合以上两个结果，可以得出 $ALG_{std}((\Gamma, \varphi)) = ALG_{gen}(G)$。 □

第 4 章　最小割数量的敏感度分析

4.1　最小割数量的敏感度

与此前工作不同的是，本文的近似最小割算法需要找到尽可多的解，因此算法需要考虑输出的最小割数量，并确保隐私得到保护。前文提到，图的差分隐私要求一条边的存在与否对输出的影响不能过大，因此，图的最小割数量的敏感度的定量分析是必要的。敏感度越大，需要添加的噪声也越大，所以本掌将结合仙人掌图表示，将敏感度通过增加条件来限制在一个相对较小的值。

对于两个边相邻的图 G 和 G'，不妨令 G' 的构造由在 G 中加入一条边权为 1 的边 (u,v) 完成。在最小割数量计算函数的输入与输出都以适当的二进制形式进行编码的情况下，最小割数量的敏感度为 $d=|M_G-M_{G'}|$。

根据定理 2.2.3，任意图 G 的最小割数量满足 $1\leq M_G\leq n^2$，因此可以得到一个平凡的结论 $0\leq d\leq n^2$。在不添加额外条件的情况下，该敏感度范围的最大值是可以取到的。

引理 4.1.1. *对于任意 $n\geq 3$，存在图 G,G' 的构造方法，使得最小割数量的敏感度为 $\Omega(n^2)$。*

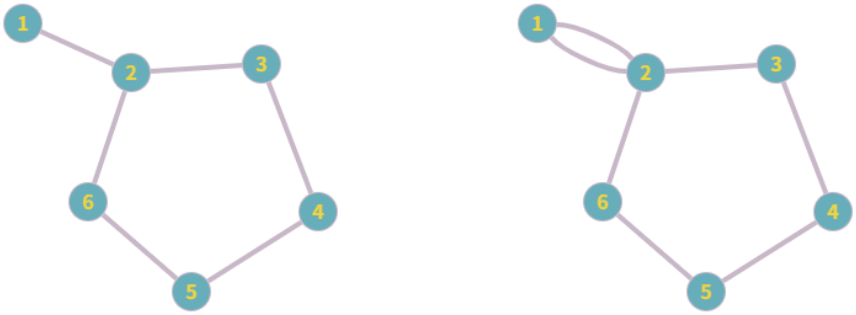
证明 为便于描述，首先将 G,G' 中的顶点编号，用 v_1 至 v_n 表示。令后文连接顶点所用的边的权重均为 x。

接下来给出 G,G' 的具体构造方法。在图 G 中，连接 v_1 与 v_2，v_2 与 v_n，并对于所有整数 $2\leq i<n$，连接 v_i 和 v_{i+1}；图 G' 由在图 G 的基础上，增加一条连接 v_1 和 v_2 的边得到。图 G 由一个由 $v_2,\ldots,v_n$ 组成的 $n-1$ 元环和一条边 (v_1,v_2) 构成，因此最小割值为 $\Phi_G=x$，唯一的最小割是 $(\{v_1\},V\backslash v_1)$，也就是是说，最小割数量 $M_G=1$。图 G 由一个由 $v_2,\ldots,v_n$ 组成的 $n-1$ 元环和 v_1,v_2 组成的 2 元环构成，因此最小割值 $\Phi'_G=2x$，此时对于任意 $2\leq i\leq j\leq n$，$(\{v_i,\ldots,v_j\},V\backslash\{v_i,\ldots,v_j\})$ 都是最小割的解，因此 $M_{G'}=\frac{(n-1)(n-2)}{2}$。在这种构造方法下，$d=\Theta(n^2)$。　□

图4-1给出了一种 $n=6$ 时的构造示例。引理4.1.1表明，边相邻图的最小割数量的敏感度较高这意味着算法需要添加较大噪声，进而使得差分隐私下发布的最小割数量的可用性较低。下一节，算法将引入额外约束条件并进行分析。

4.2　约束条件下的敏感度

给定图 G，在加入一条单位边后，最小割的割值可能增加，也可能保持不变。引理4.1.1指出，若图有一个割值为 x 的割和较多割值为 $x+1$ 的割，那么加边导致



(a) 图 G (b) 图 G'

图 4-1 $n = 6$ 时的构造示例

最小割的割值提高 1 后，会使最小割的数量出现较大的增幅。因此，为了得到可用的结果，本节将加入额外限制条件，使得边相邻图 G, G' 的最小割割值相同，即图 G, G' 满足 $\Phi_G = \Phi_{G'}$。本节后续的分析将基于这一限制。

标准仙人掌图表示法包含了图所有最小割的信息,因此其结构参数有助于加强对最小割数量敏感度分析的精细性。对于图 G 以及其标准仙人掌图表示法 (Γ, φ)，本节引入以下参数来描述其结构：

- α_g：标准仙人掌图表示法的点数，即 $|V_\Gamma|$。
- α_p：标准仙人掌图表示中所有点 v 对应的 $|\varphi^{-1}(v)|$ 的最大值
- α_c：标准仙人掌图表示中环的数量。
- α_r：标准仙人掌图表示中环长度的最大值。
- α_d：标准仙人掌图表示中树结构的直径，也就是所有图上简单路径中，非环边数量的最大值。

前文提到，G' 由在图 G 的基础上，增加一条边得到，记这条边为 (u, v)。边 (u, v) 在标准仙人掌图表示法 (Γ, φ) 中的位置与最小割数量变化量有关，接下来给出具体分析。

若 $\varphi(u) = \varphi(v)$，那么说明 u, v 之间不被任何最小割分隔开，在仙人掌图表示法中，它们已经被视为连通性很高的两个点，因此加入边 (u, v) 对最小割数量没有任何影响。

若 $\varphi(u) \neq \varphi(v)$，记 $U = \varphi(u)$,$V = \varphi(V)$。标准仙人掌图表示法中的最小割分为两部分，其中 p 割对应树表示的树边，t 割对应环上不相邻的边二元组。U 到 V 连边后，树表示中 U 到 V 路径上的所有 p 割都将不再是最小割；路径上所

有 p 束对应的环代表的 t 割都会变少，具体来说，对于一个 p 束 S 的环 G_S，令 $U \in x_{R'}, V \in x_{R''}$，所有将 $x_{R'}$ 与 $x_{R''}$ 分开的 t 割都将不再是最小割。

接下来对环上的情况进行定量分析。令 $f(x)$ 为长度为 x 的环表示的 t 割数量，则

$$f(x) = \begin{cases} \frac{x(x-3)}{2} & x \geq 3 \\ 0 & 1 \leq x \leq 2 \end{cases} \tag{4.1}$$

假设环 G_S 的环长为 l，$x_{R'}$ 与 $x_{R''}$ 在环上的距离为 t（满足 $t \leq l-t$），那么最小割的减少量为

$$g(l,t) = f(l) - f(t) - f(l-t) - [t \geq 3] - [l-t \geq 3] \tag{4.2}$$

令 $G(l) = \max_{t=1}^{l-1} g(l,t)$，其含义为长度为 l 的环对应的树结构顶点被路径 (U,V) 经过时，最小割减少量的最大值。不妨对 l,t 的值进行讨论来得到该函数的取值：当 $1 \leq l \leq 3$ 时，环上不包含 t 割，因此 $G(l) = 0$；当 $l = 4$ 时，取 $t = 2$ 为极值，$G(4) = 2$；当 $l \geq 5$ 时，$l - t \geq t$ 使得只需要考虑 $[t \geq 3]$ 的两种取值，即 $t = 1, t = 2, t \geq 3$ 这三种情况。

- 当 $t = 1$ 时，$g(l,1) \leq g(l,2)$，一定不优。
- 当 $t = 2$ 时，$g(l,2) = f(l) - f(l-2) - 1 = 2l - 6$；
- 当 $t \geq 3$ 时，$g(l,t) = f(l) - f(t) - f(l-t) - 2 = -(t - \frac{l}{2})^2 + \frac{l^2}{4} - 2$。

当 $l = 5$ 时，$t \leq 2$，因此 $G(5) = g(5,2) = 4$。当 $l \geq 6$ 时，$g(l,t)$ 在 $t \geq 3$ 范围的极大值在 $t = \lfloor \frac{l}{2} \rfloor$ 时取到，且有 $g(l,t) \geq g(l,2)$。综上，可以得到

$$G(l) = \max\{0, \lfloor \frac{l^2}{4} - 2 \rfloor\} \tag{4.3}$$

上述分析说明了加入一条边 (u,v) 在树结构图以及环上对最小割数量的影响。这里还需要特别说明的是，加入 (u,v) 使最小割的割值增加一的情况一定满足如下条件：

- $\varphi(u) \neq \varphi(v)$。
- 标准仙人掌图表示法是一条链。
- $\varphi(u), \varphi(v)$ 分别是链的两个端点。

综上，在加入边不影响最小割值的情况下，最小割数量变化范围的表达式如下。

定理 4.2.1. *给定边相邻图 G, G'，其中 G' 由在 G 中加入一条边权为 1 的边 (u, v) 得到。那么有*

$$M_G - \alpha_d - \min\{\alpha_d, \alpha_c, \frac{\alpha_g}{\alpha_r}\} \cdot \max\{0, \lfloor \frac{\alpha_r^2}{4} - 2 \rfloor\} \leq M_{G'} \leq M_G \tag{4.4}$$

这里链上的变化数量受到树结构图的直径约束，环上的变化数量受到路径经过环的数量以及每个环的环长约束。

最小割数量的变化还可以由 M_G 本身的值来进行更为简洁的估计。前文提到，对于一个长度 $l \geq 4$ 的环 G_S，其表示的 t 割有 $f(l) = \frac{l(l-3)}{2}$ 个，边 (u, v) 经过它是会使其最小割数量减少至多 $G(l) = \lfloor \frac{l^2}{4} - 2 \rfloor$。此外，该 p 束 S 连接的不涉及 $x_{R'}, x_{R''}$ 的至少 $l-2$ 条边对应的 p 割在加边后仍然为最小割。因此，与该 p 束 S 相关的最小割的数量为 $\frac{l^2-l-4}{2}$，减少量至多为 $\lfloor \frac{l^2}{4} - 2 \rfloor$。

定理 4.2.2. *对于任意加边 (u, v)，存在一种最小割分配方案，满足每个最小割至多分配至一个 p 束中，使得每个 p 束 S 损失的最小割数量不超过其分配量与 $|G_S|$ 和的一半。*

证明 按上文方法分配最小割后，p 束 S 分配到的最小割数量为 $\frac{l^2-l-4}{2}$，$G_S = l$ 其损失的最小割数量为 $\lfloor \frac{l^2}{4} - 2 \rfloor$。有

$$\frac{l^2 - l - 4}{2} + l = \frac{l^2 + l - 4}{2} \geq \frac{l^2}{2} - 4 \geq 2\lfloor \frac{l^2}{4} - 2 \rfloor \tag{4.5}$$

□

通过该方案，可以给出一个基于 M_G 的估计。

定理 4.2.3. *给定边相邻图 G, G'，其中 G' 由在 G 中加入一条边权为 1 的边 (u, v) 得到。那么有*

$$\frac{M_G}{2} - 1.5n \leq M_{G'} \leq M_G \tag{4.6}$$

该定理给出了最小割数量敏感度的上界 $\frac{M_G}{2} + 1.5n$。接下来将给出一个构造来说明这个上界可以近似地达到，该构造下的最小割数量的敏感度与定理中最坏情况下的敏感度仅相差一个常乘法系数，图 G 构造方法如下：将 $(1-\alpha)n$ 个点用边权 c 连成一条链，设链的两端分别为 v_1, v_2；将 $\alpha n \geq 4$ 的点用边权 $\frac{c}{2}$ 连成一个环，设环的一个对角线连接的两个点为 v_3, v_4；最后将 v_2, v_3 用边权为 c 的边相连，完成构造。

在该构造下，$M_G = (1-\alpha)n + \frac{\alpha n(\alpha n-3)}{2}$。敏感度最高的加边是 (v_1, v_4)，敏感度 $M_G - M_{G'} = \lfloor \frac{\alpha^2 n^2}{4} \rfloor - 2 + (1-\alpha)n$。取 $\alpha = \frac{1}{2}$，可得 $M_G = \frac{n(n-2)}{8}$，$M_G - M_{G'} = \lfloor \frac{n^2}{16} \rfloor - 2 + \frac{1}{2}n$。在 $n \geq 10$ 时有

$$\frac{M_G}{2} + \frac{3}{2}n = \frac{n^2}{16} + \frac{11}{8}n \leq 4(M_G - M_{G'}) \tag{4.7}$$

4.3 平均敏感度

前面的分析说明了，给定图 G，在最坏情况下，加入一条边带来的最小割数量变化较大，也就是说敏感度较高。这一节将通过平均敏感度，评估随机加边条件下，最小割数量变化较大的概率。

通常情况下，平均敏感度定义中的边相邻图会以图 G 与在其边集中删除一条边得到的图 G' 的形式给出。[25] 但本节将采用加边的形式，这能确保仙人掌图表示的结构易于分析。以删边的形式定义平均敏感度，可以估计一个图算法在规模较大的子图上的输出和完整图上的输出差异；以加边的形式定义平均敏感度会在这一功能上有所欠缺，但能与前文内容相结合，来给出最小割数量变化值的期望。

定义 4.3.1. 图算法 A 的平均敏感度为

$$\mathbb{E}_{e \in V^2, \Phi(G)=\Phi((V,E\cup\{e\}))}[d_{Ham}(A(G), A((V, E \cup \{e\})))] \tag{4.8}$$

接下来，本文将给出一种 G 的构造来说明，随机加边条件下，最小割数量的变化的平均值可以取到一个较高的值。记 $W(n)$ 为规模为 n 的图的最高敏感度，具体构造步骤如下：首先生成一个规模为 $\frac{1}{3}n$ 的图 G_t，使得 G_t 在加入边 (u, v) 时得到该规模下最高的敏感度 $W(\frac{1}{3}n)$；接下来令 $\frac{1}{3}n$ 个点与 u 用极大的边权相连，令另外 $\frac{1}{3}n$ 个点与 v 用极大的边权相连。以该方法构造出的图 G 中，u, v 分别与 $\frac{1}{3}n$ 个点紧密相连，这意味着当选取的边的两端分别位于这两个点集时，其产生的最小割数量的变化等价与边 (u, v)。

因此，在该构造下最小割数量函数 M 的平均敏感度满足

$$\mathbb{E}_{e \in V^2, \Phi(G)=\Phi(G+e)}[d_{Ham}(M(G), M(G+e))] \geq \frac{\frac{1}{3}n(\frac{1}{3}n-1)}{n(n-1)} W(\frac{1}{3}n) \tag{4.9}$$

最高敏感度 W 是一个 n 的二次多项式。综上，存在一个常数 β 使得在该构造下

$$\mathbb{E}_{e \in V^2, \Phi(G)=\Phi(G+e)}[d_{Ham}(M(G), M(G+e))] \geq \beta W(n) \tag{4.10}$$

分析表明，随机加边条件下，最小割数量的变化可以取到一个较高的期望。

第 5 章　差分隐私背景下近似最小割求解算法

本章将详细阐述差分隐私背景下近似最小割求解算法的具体设计方案。该算法不仅需满足差分隐私的要求，同时要有效控制近似最小割与真实最小割之间的加法误差；此外，算法还需尽可能多地输出有效的近似最小割的解，并兼顾算法的运行效率。形式化的来说，算法的输入为图 G，给定隐私参数 ε, δ 和误差 Δ，算法应当能 (ε, δ)-差分隐私地输出一个尽可能大的割集 R，使得对于任意割 $r \in R$，满足 $w(r) \leq \Phi_G + \Delta$。

5.1　基于差分隐私图的算法设计

算法设计的难点在于，输入的数据需要隐私化处理后才能使用。对图直接进行差分隐私是一种可行的方案，且在隐私化后的图上进行计算可以使用非差分隐私算法。定理2.4.3展示了 Liu 等人设计的差分隐私图算法。回顾该算法内容，对于一个输入 G，算法可以以高概率 (ε, δ)-差分隐私地输出一个合成图 $\hat{G}$。对于合成图中任意不相交的点集 $S, T \subseteq V_G$，均满足如下关系

$$|w_G(S,T) - w_{\hat{G}}(S,T)| = O\left(\frac{\sqrt{nm}}{\varepsilon}\log^3\left(\frac{n}{\delta}\right)\right) \tag{5.1}$$

这表明,对于原图中的任意最小割 R,其在合成图 $\hat{G}$ 中的割值为 $\Phi_G + O\left(\frac{\sqrt{nm}}{\varepsilon}\log^3\left(\frac{n}{\delta}\right)\right)$。由此可见，合成图 $\hat{G}$ 作为图 G 隐私处理后的结果，能够确保最小割有较好的近似，也就是说，该方法能确保合成图中的近似最小割数量不少于原图的最小割数量。

在完成图的隐私化步骤后,算法可以枚举合成图中的所有割,并将割值与 $\Phi_G + \Delta$ 比较。因此，算法需要差分隐私地发布最小割割值 Φ_G，即得到一个误差较小的近似值 $\hat{\Phi}_G$，这一任务可以由拉普拉斯机制完成。

差分隐私地计算最小割值的方法如下：首先，使用定理2.2.1中的 Karger 最小割算法，可以以高概率求出图 G 的最小割值 Φ_G。在边相邻的两个图 G, G' 中，恰好存在一条边边权相差 1，其它边边权均相等，因此，图 G 中的一个最小割在图 G' 中的割值最多增加一。根据定理2.4.2中的拉普拉斯机制，将最小割值作为函数 f，则其对应的敏感度 $\Delta = 1$，因此对最小割的真实加入噪声后，算法能以 ε-差分隐私的发布最小割值的近似值 $\hat{\Phi}_G = \Phi_G + X$，其中 $X \sim Lap(1/\varepsilon)$。

使用拉普拉斯机制带来的误差为 $O(\frac{1}{\varepsilon})$。具体分析如下:拉普拉斯分布 $Lap(1/\varepsilon)$ 的概率密度函数为

$$f(x) = \frac{\varepsilon}{2}e^{-\varepsilon\|x\|} \tag{5.2}$$

对 x 取绝对值后，函数变成指数分布的形式，其概率密度函数为

$$f(x)=\varepsilon e^{-\varepsilon x},(x\geq 0) \tag{5.3}$$

绝对值的累计分布函数为

$$P(|X|\leq t)=1-e^{-\varepsilon x},(t\geq 0) \tag{5.4}$$

由该式可得，有至少 $1-\alpha$ 的概率使得 $|X|\leq \frac{1}{\varepsilon}\ln\left(\frac{1}{\alpha}\right)$。因此，算法有高概率满足 $\hat{\Phi}(G)=\Phi(G)+O(\frac{1}{\varepsilon})$。

在差分隐私地求得最小割的近似值后，算法可以通过枚举割并对割值进行判断来求得解。具体来说，算法将 β 定为一个足够大的常数，并枚举 V_G 的所有子集 X，判断 $\Delta(X)$ 在合成图 $\hat{G}$ 中的割值是否满足 $w_{\hat{G}}(X)\leq\hat{\Phi}_G+\beta\frac{\sqrt{nm}}{\varepsilon}\log^3\left(\frac{n}{\delta}\right)$，满足条件的 $\Delta(X)$ 被视为近似最小割并加入输出的割集 R 中。需要注意的是，输出的割集 R 会包含图 G 中的所有最小割和部分近似最小割，因此割集 R 的大小可能超过 n^2。

使用定理2.4.1这一基本组合定理，可以为算法中差分隐私步骤的 ϵ 和 δ 赋合适的常系数，不难说明，上述算法是 (ε,δ)-差分隐私的，且能以高概率输出至少 M_G 个 $O\left(\frac{\sqrt{nm}\log^3\left(\frac{n}{\delta}\right)}{\varepsilon}\right)$ 近似的最小割。

5.2 基于 k 优选择机制的算法设计

除了对图直接进行隐私化处理外，算法也可以在对原图进行若干计算后再进行差分隐私。定理2.4.5中提到的 k 优选择机制可以 (ε,δ)-差分隐私地在 m 个值中取 k 个 $O(\frac{\sqrt{k\log(m/\delta)}}{\varepsilon})$ 近似最小值。因此，算法可以差分隐私的获取最小割值 $\hat{\Phi}_G$ 和最小割的数量 $\hat{M}_G$，并用 k 优选择机制在所有 2^n 个割中选择权值前 $\hat{M}_G$ 小的割。通过该方法，算法可以获得 $\hat{M}_G$ 个割，再进行筛选处理可以得到差分隐私背景下的近似最小割。

回顾最小割数量的敏感度，根据定理4.2.3，给定边相邻图 G,G'，其中 G' 由在 G 中加入一条边权为 1 的边得到，则有 $0\leq M_G-M_{G'}\leq\frac{M_G}{2}+1.5n$。由于边相邻图的最小割数量较大，因此需要取对数处理。不妨令 $a=\log_2(M_G+3n)$，其敏感度为 1。接下来，使用拉普拉斯机制差分隐私的输出 a 的值 $\hat{a}$，并令差分隐私的最小割数量为 $\hat{M}_G=min\{max\{0,2^{\hat{a}}-3n\},n^2\}$。

得到估计值 $\hat{\Phi}_G$ 和 $\hat{M}_G$ 后，算法可以枚举 V_G 的所有子集 X，并计算割 $\Delta(X)$ 在原图 G 中的割值。对所有 2^n 个割值使用定理2.4.5中的 k 优选择机制，算法可以

得到 $\hat{M}_G$ 个割以及每个割的 $O(\frac{n\sqrt{n\log(1/\delta)}}{\varepsilon})$ 近似值。由于对最小割数量的估计存在误差，因此原图的第 $\hat{M}_G$ 小割可能并非最小割，因此，筛选处理在本节的算法仍然是必要的。与上一节类似，算法将 β 定为一个足够大的常数，并枚举所有这 $\hat{M}_G$ 个割，并分别判断其割值是否小于 $\hat{\Phi}_G + \beta\frac{n\sqrt{n\log(1/\delta)}}{\varepsilon}$，满足条件的割视被视为近似最小割并加入输出的割集 R 中。

相比于基于差分隐私图的算法，该算法加入了对最小割数量的估计，因此适用于要求输出割数量与原图 G 中最小割数量相仿时的场景。然而，本文中给出的差分隐私的最小割数量估计算法误差较大，算法可能出现输出割集大小远小于原图 G 中最小割数量的问题，这与目标相悖。

为了解决 $\hat{M}_G$ 过小的情况，算法需要进行一些修改。对于图 G 的 M_G 个最小割中的每个割，其在 k 优选择机制中的的割值为 $\Phi_G + O(\frac{\sqrt{kn\log(1/\delta)}}{\varepsilon})$。$\hat{M}_G$ 的规模为 $O(n^2)$，因此代入 k 的取值后，可以保证原图 G 的最小割在筛选步骤中不会被排除在外。因此当 $\hat{M}_G \leq M_G$ 时，k 优选择机制得到的割一定不会在筛选步骤中被排除在外。由于筛选机制的存在，可以令 $k = n^2 \geq M_G$，这样能保证输出的近似最小割数量不少于原图最小割的数量。

修改后的基于 k 优选择机制的算法的表现并不比基于差分隐私图的算法更好，但其提供了一种对原图进行进一步处理以实现算法优化的可能性。

5.3 加法近似参数的优化

当 $m = \Theta(n^2)$ 时，基于差分隐私图的算法与基于 k 优选择机制的算法，其加法近似参数均为 $\tilde{O}(\frac{n\sqrt{n}}{\varepsilon})$。本节提出一种融合指数机制与 Karger 收缩算法的方法，旨在降低加法近似参数。

回顾 k 优选择机制，算法从 m 个数中选择了 k 个最小值，其加法近似参数 $O(\frac{\sqrt{k\log(m/\delta)}}{\varepsilon})$ 同时由 m, k 决定。而受到最小割数量的限制，在最坏情况下，k 的取值为 $\Theta(n^2)$。因此要想优化加法近似参数，需要从降低 m 入手。

在上一节中，算法枚举了点集的所有子集来寻找割，这是因为在没有给定更多条件的情况下，图中所有割都是潜在的 k 小割，因此 $m = 2^n$。算法需要缩小潜在 k 小割的范围来使 m 值降低。

由于 k 的值为 $O(n^2)$，因此 k 小割可以视为原图的一个近似最小割。前文提到，当 α 为一常数时，Karger 收缩算法可以在多项式复杂度内找到所有 α 乘法近似最小割。具体来说，根据定理2.2.4，一个给定的 α 乘法近似最小割在收缩至 $\lfloor 2\alpha \rfloor$ 个顶点时，其有效的概率为 $\Omega(n^{-2\alpha})$。因此，执行 $n^{2\alpha+1}$ 次 Karger 收缩算法，则该 α 乘法近似最小割以高概率在至少一次算法执行中有效。因此，以高概率所有 α 乘

法近似最小割都被找到。Karger 收缩算法本身为非差分隐私算法，因此在处理其输入时需要考虑隐私性。

设调用 k 优选择机制时算法的加法近似参数为 Δ'，则应有 $\Delta' \leq \Delta$。若如此，则对于图 G 的任意最小割，在调用 k 优选择机制时割值均小于 $\Phi_G + \Delta$，则可保证找到的近似最小割数量不少于原图的最小割数量。

Karger 收缩算法可以找到割值不超过 $\alpha\Phi_G$ 的所有割，而本文提出的算法需要寻找割值不超过 $\Phi_G + \Delta'$ 的最小割。前者采用乘法近似参数，后者采用加法近似参数，参数形式的差异要求对算法进行额外分析。联立两式得 $\alpha = 1 + \frac{\Delta'}{\Phi_G}$，由于 Karger 收缩算法求得的近似最小割数量与 α 有关，因此 α 的值需取为常数。也就是说，算法需要对 G 进行一定处理来保证 $\frac{\Delta'}{\Phi_G}$ 存在一个常数上界。

对 G 的处理应该在引入较少误差 Δ' 的同时，又能提高最小割的割值。具体方法为在图 $G = (V, E)$ 中差分隐私地加边加边。算法使用指数分布来差分隐私地选择加边的边集。首先给定参数 T，算法按如下方法构造一个边集 H：

- 将图 G 的 n 个顶点按任意顺序排列成一个环。
- 对环上任意相邻两点，在 H 中加入 $T/2$ 条连接这两个点的边权为 1 的边。

接下来，令 $H_0 \subset H_1, \ldots, \subset H_{|H|}$ 为任意大小严格递增的边集序列，且 $H_{|H|} = H$。对于每个下标 $i \in [1, |H|]$，令其权值为 $t_i = |\Phi_{(V,E\cup H_i)} - T|$。使用指数机制选择下标 i，则有高概率得到一个解，满足

$$Pr[|\Phi_{(V,E\cup H_i)} - T| > t_{min} + \frac{2\ln(nT)}{\varepsilon} + \frac{2t}{\varepsilon}] \leq e^{-t} \tag{5.5}$$

容易证明，$t_{min} = \max\{0, \Phi_G - T\}$。令 $T = \frac{20\ln n}{\varepsilon}$，令 $t = 2\ln n$。整理，记

$$t_{range} = \frac{2\ln(nT)}{\varepsilon} + \frac{2t}{\varepsilon} = \frac{6\ln n + 2\ln T}{\varepsilon} = \frac{6\ln n + 2\ln(20\ln n) - 2\ln\varepsilon}{\varepsilon} \tag{5.6}$$

当 $\varepsilon \in [\frac{1}{n}, \frac{1}{2}]$,$n \geq 200$ 时，有 $t_{range} \leq \frac{10\ln n}{\varepsilon}$。

记 $\hat{G} = (V, E \cup H_i)$，讨论 Φ_G 与 T 的关系并化简，可以得出算法以 $1 - \frac{1}{n^2}$ 的概率满足

$$10\ln n/\varepsilon < \Phi_{\hat{G}} < \Phi_G + 30\ln n/\varepsilon \tag{5.7}$$

在上述生成 $\hat{G}$ 的过程中，加入边的数量为 $O(\frac{n\ln n}{\varepsilon})$，且有 $1 \leq \frac{\Phi_G}{\Phi_{\hat{G}}} \leq 4$，且边集的选取满足差分隐私。因此，只需要在 $\hat{G}$ 中运行 n^9 次 Karger 收缩算法，就能以高概率找到 G 的所有近似最小割。根据定理2.2.5，α 乘法近似最小割的数量至多为

$n^{2\alpha}$，所以找到割的数量是 $O(n^8)$ 的。即 $m = O(n^8)$，得到 k 优选择机制的加法近似参数为 $O(\frac{n\sqrt{\log(n^8/\delta)}}{\varepsilon})$。算法的总加法近似参数因为加边操作而限制为 $O(\frac{n\ln n}{\varepsilon})$。

与上一节类似地，由于本文对最小割数量的估计方法误差较大，本节的算法也需要采取一定修改。算法定 k 的值为 n^2，并按如上步骤找到权值 k 小的割。接下来，差分隐私的发布最小割的值 $\hat{\Phi}_G$，然后将 β 定为一个足够大的常数，并对这 n^2 个割分别判断其割值是否小于 $\hat{\Phi}_G + \beta\frac{n\ln n}{\varepsilon}$，满足条件的割被视为近似最小割并加入输出的割集 R 中。

定理 5.3.1. 给定 $\varepsilon \in [\frac{1}{n}, \frac{1}{2}]$ 和 δ 作为隐私参数。对于给定任意 $n \geq 200$ 的图 G，存在一个 (ϵ, δ)-差分隐私算法，能够至少输出 M_G 个与最小割的割值 $O(\frac{n\ln n}{\varepsilon})$ 近似的割。

第 6 章　总结与展望

6.1　工作总结

最小割问题不仅在理论计算机领域有着重要的学术研究价值，同时在通信网络、芯片电路、系统设计、生物信息等领域也有着广泛的应用价值。差分隐私使算法在更多数据敏感的场景得到应用成为了可能。本论文的主要工作为：

- 定义了标准仙人掌图表示法，并给出了一个高效的标准化个仙人掌图表示法的算法。
- 定量分析了最小割数量的敏感度。
- 给出了一个加性误差为 $O(\frac{n\log n}{\varepsilon})$ 的差分隐私近似最小割算法。

6.2　研究展望

未来工作可从以下方向展开：

- 在理解仙人掌图表示法的基础上分析多边形图表示法，以分析近似最小割本身的性质。
- 设计能差分隐私的输出仙人掌图表示的算法。
- 对本文中的算法进行进一步改进，以获取更优的加性误差结果。

参考文献

[1] NARAYANAN A, HUEY J, FELTEN E W. A precautionary approach to big data privacy[J]. Data protection on the move: Current developments in ICT and privacy/data protection, 2016 : 357 – 385.

[2] DINUR I, NISSIM K. Revealing information while preserving privacy[C] // Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. 2003 : 202 – 210.

[3] VADHAN S. The complexity of differential privacy[J]. Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich, 2017 : 347 – 450.

[4] FORD JR L R, FULKERSON D R. Maximal flow through a network[J]. Canadian journal of Mathematics, 1956, 8 : 399 – 404.

[5] KARGER D R. Global Min-cuts in RNC, and Other Ramifications of a Simple Min-Cut Algorithm.[C] // Soda : Vol 93. 1993 : 21 – 30.

[6] KARGER D R, STEIN C. A new approach to the minimum cut problem[J]. Journal of the ACM (JACM), 1996, 43(4) : 601 – 640.

[7] KARGER D R. Minimum cuts in near-linear time[J]. Journal of the ACM (JACM), 2000, 47(1) : 46 – 76.

[8] LI J. Deterministic mincut in almost-linear time[C] // Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing. 2021 : 384 – 395.

[9] DINITZ E A, KARZANOV A V, LOMONOSOV M V. On the structure of the system of minimum edge cuts of a graph[J]. Issledovaniya po Diskretnoi Optimizatsii, 1976 : 290 – 306.

[10] FLEINER T, FRANK A. A quick proof for the cactus representation of mincuts[J]. EGRES Quick Proof, 2009, 3 : 2009.

[11] KARGER D R, PANIGRAHI D. A near-linear time algorithm for constructing a cactus representation of minimum cuts[C] // Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms. 2009 : 246 – 255.

[12] HE Z, HUANG S-E, SARANURAK T. Cactus representation of minimum cuts: Derandomize and speed up[C] // Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). 2024 : 1503 – 1541.

[13] GUPTA A, LIGETT K, MCSHERRY F, et al. Differentially private combinatorial optimization[C] // Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms. 2010 : 1106 – 1125.

[14] LI J, PANIGRAHI D. Approximate gomory–hu tree is faster than n–1 max-flows[C] // Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing. 2021 : 1738 – 1748.

[15] LI J, PANIGRAHI D. Deterministic min-cut in poly-logarithmic max-flows[C] // 2020 IEEE 61st

Annual Symposium on Foundations of Computer Science (FOCS). 2020 : 85 – 92.

[16] LIU J, UPADHYAY J, ZOU Z. Optimal bounds on private graph approximation[C] // Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). 2024 : 1019 – 1049.

[17] KARGER D R. Random sampling in cut, flow, and network design problems[C] // Proceedings of the twenty-sixth annual ACM symposium on Theory of computing. 1994 : 648 – 657.

[18] DWORK C. Differential privacy[C] // International colloquium on automata, languages, and programming. 2006 : 1 – 12.

[19] DWORK C, MCSHERRY F, NISSIM K, et al. Calibrating noise to sensitivity in private data analysis[C] // Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. 2006 : 265 – 284.

[20] DWORK C, LEI J. Differential privacy and robust statistics[C] // Proceedings of the forty-first annual ACM symposium on Theory of computing. 2009 : 371 – 380.

[21] DWORK C, ROTH A, OTHERS. The algorithmic foundations of differential privacy[J]. Foundations and Trends® in Theoretical Computer Science, 2014, 9(3–4) : 211 – 407.

[22] MCSHERRY F, TALWAR K. Mechanism design via differential privacy[C] // 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). 2007 : 94 – 103.

[23] QIAO G, SU W, ZHANG L. Oneshot differentially private top-k selection[C] // International Conference on Machine Learning. 2021 : 8672 – 8681.

[24] CUNNINGHAM W H. Minimum cuts, modular functions, and matroid polyhedra[J]. Networks, 1985, 15(2) : 205 – 215.

[25] VARMA N, YOSHIDA Y. Average sensitivity of graph algorithms[J]. SIAM Journal on Computing, 2023, 52(4) : 1039 – 1081.

致 谢

“人比山高，脚比路长”——这句校歌歌词寄托了吉大人求实创新、立志图强的期许，初闻校歌悠扬旋律时的憧憬犹在昨日，转眼间，大学四年已行至尾声。值此毕业论文完成之时，谨以最诚挚的谢意，献给所有在这段独特旅程中给予我鼓励和支持的师长、伙伴与家人。

首先，由衷感谢我的导师刘淼老师。在论文材料提交等环节，得益于老师的帮助，相关工作方得以顺利完成。同时，这四年来，老师作为教练在算法竞赛上的支持，使得我所在的队伍取得了优异成绩，这份师恩铭记于心。

尤为重要的是，我要向我的校外导师刘景铖老师表达最深切的感激。老师在我本科阶段的尾声，点燃了我对学术研究的全新热情，并为我未来的持续探索指明了方向。选题阶段，老师鼓励我自主调研，挖掘选题方向；算法设计时，老师凭借深厚的学术经验，为我提供了宝贵的改进建议；论文写作中，老师耐心的指导，令我受益匪浅。这些指导是我未来研究生涯以及人生路上的宝贵财富。

特别感谢刘华虓老师在研究项目中的指导，让我得以初窥门径，积累了宝贵的工作经验。同时，也要感谢计算机科学与技术学院的全体老师以及其他任课老师。你们渊博的学识和鞭辟入里的教学，极大地夯实了我的专业基础，为论文撰写提供了不竭的源泉。

感谢我的同学们，四载同窗，我们共同学习、成长、进步，携手克服重重困难，成为彼此青春岁月里最坚定的支持者。

最后，以最深沉的谢意献给我的家人。是你们的理解、支持与付出，筑成了我的坚实后盾，让我得以全身心地遨游于知识的海洋。你们的爱是我勇往直前的永恒动力。

未来征程，道阻且长。我将以坚毅为桨，勇气为帆，长风破浪会有时，直挂云帆济沧海！