吉林大学 JILIN UNIVERSITY

本科生毕业论文（设计）

中文题目 差分隐私下求解近似最小割问题的算法设计与实现

英文题目 **Solving the approximate minimum cut problem under 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 **XXX**

2025 年 5 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：

2023 年 5 月 29 日

中文完整题目

摘　要

Helmoheltz 方程在声波，电磁波以及弹性波领域都有着非常重要的应用，因而吸引了许多学者对此进行了很多方面的研究，在数值计算方面，如何解决问题求解区域是无界的这个难题一直是学者们非常关心的课题，Bramble 在其工作中，用完全匹配层方法（PML）将无界求解区域上的声波散射问题转化为截断的可计算区域上，这对数值计算带来了很大的方便，但同时却将原本为实系数对称问题变为复系数非对称问题，给数值求解带来新的困难。

通过查阅与学习，本文以此截断的声波散射 PML 问题为模型，将二重网格有限元法应用到该问题中，并给出了一个有效的计算格式。该方法是在粗网格有限元空间 V_H 上使用标准的有限元离散技巧去求解一个小规模的复杂的原问题，得到一个粗略的估计 $u_H \in V_H$，然后再在此基础上在细网格有限元空间 $V_h\,(h \ll H)$ 上解一个简单的只含高阶项问题，得到修正值 $u_h \in V_h$。给出了解的存在性证明，得到了与标准有限元方法一致的误差估计，通过比较验证了本算法的高效性和合理性。

关键词：

Helmoheltz 方程, 完全匹配层, 二重网格有限元方法

English Title

Author: Zhou Yuheng

Supervisor: Liu Miao

Abstract

The Helmoheltz equation has important application in acoustic, electromagnetic and elastic scattering problems, and has been investigated in many ways. In the area of numerical method, how to solve the problem that those problems are posed on infinte domains is always a popular subject. In his work, Bramble transforms the acoustic scattering problem on infinte domain to the truncated domain using perfectly matched layer method(PML). This work will be continent to simulation, however, it also brings some new problems. The original problem is symmetric with real coefficients, but now is nonsymmetric with complex coefficients.

By reading and learning, we consider the truncated acoustic scattering PML problem and develop an efficient approximate method based on two-grid method. The method is to solve a small and complex original problem by standard finit element discretization on a coarse space V_H. Based on the coarse solution $u_H \in V_H$, we solve a simple problem in the fine space v_h,which is only have the high order terms of original problem, and get a correction $u_h \in V_h$. We prove the existence and uniqueness and the error bound of optimal order of accuracy which is consistent with the standard finite element discretization. Finally, our theoretical claims are supported.

Keywords:

Helmoheltz Equation, PML Method, Two-grid Finite Element Method

目 录

第 1 章　绪论

1.1　研究背景与意义

假如你是你是一名临床医学与人工智能交叉领域的科研人员，你致力于构建一个基于患者身体指标信息的抑郁症诊断模型，从而实现抑郁症的早期精准识别。模型可以通过年龄、睡眠数据、激素水平、基因数据等信息来挖掘潜在的抑郁症特征。然而，为了提高诊断的准确性，不可避免地需要收集患者的敏感信息来生成数据集。与此同时，随着学术交流合作的日渐频繁，其它研究者可能会请求获取数据集来完成分析、验证假设。这带来了严峻的隐私保护挑战，作为信息的收集者，你有义务保证患者的敏感信息。如何衡量隐私的泄露情况，应该采用什么样的方法保护隐私成为了重要课题。

隐去部分隐私信息是一个看起来有效的方法。例如在公开数据集时，可以将姓名、生日、电话号码等标识信息隐藏。然而，这样的隐私保护方法具有局限性。攻击者可以利用一些辅助数据和推理方法，将隐去标识信息的数据重新定位到个体。假设攻击者拥有将姓名与基因对应的辅助数据集，那么通过比对基因信息，就可以将数据对应到个体。这种攻击被称为关联分析攻击，已经有研究表明，相关的攻击案例并不少见。[1]

差分隐私是一种具备严格数学证明的隐私保护模型，为解决上述问题提供了思路。它定量衡量了算法的隐私保护程度，并通过添加精心设计的噪声来确保单个数据不会显著影响输出，从而在保证算法的可用性的同时，实现对个体隐私的保护。

差分隐私与传统密码学都致力于保护隐私信息，但两者关注的方向有所不同，后者注重防止输出以外的隐私泄露，而前者假设输出本身就会包含隐私信息，希望通过设计更好的信息发布形式减少隐私泄露。

在一个 n 个点 m 条边的加权无向图 $G=(V,E)$ 中，割是顶点的一个二划分 $(X,V\backslash X)$，其权重为跨越这个划分的边权之和。给定一对顶点 $s,t\in V$，$s-t$ 最小割是满足 $s\in X,t\in V\backslash X$ 的权重最小的割 $(X,V\backslash X)$，也就是说，它是将 s 与 t 分隔开的权重最小的割。$s-t$ 最小割问题与 $s-t$ 最大流问题对偶，即根据最大流最小割定理，$s-t$ 最小割值等于 $s-t$ 最大流值。[2] 相类似的，全局最小割问题是求图中权值最小的割，它反映了图的连通程度，这同样是图论领域的一个基本问题。

差分隐私定量的衡量了隐私的保护程度，因此对算法的有了额外的要求，即在几乎相同的两个输入下，算法的计算过程也应当几乎相同。以最小割算法为例，这意味着在两个输入仅相差一条边的情况下，要求输出的最小割结果的概率增幅

不能超过一个极小的常数系数。设计差分隐私下的最小割算法，有助于推广最小割算法的更多实际应用，也能加深对差分隐私下算法设计方法的探索。差分隐私下的最小割算法的设计难点在于，算法需要在保持差分隐私性的同时，控制噪声带来的误差，保证输出输出的可用性。

1.2 研究现状

在过去的几十年中，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删边的求解最小割的 $O(n^4 \log n)$ 的随机算法，他们的工作同时说明了不同的最小割的数量不超过 $\frac{n(n-1)}{2}$。[3] 该算法简洁清晰，他们证明了在随机选择边收缩的情况下，指定的最小割有不可忽视的概率在算法结束时得到保留，因此重复足够多次算法的执行，就可以找到一个最小割。1996 年，Karger 等人改进了算法，将多次独立的重复执行合并成树的若干分支，从而提高了效率，得到了一个求解最小割的 $O(n^2 \log^3 n)$ 的随机算法。[4] 值得注意的是，Karger 等人的收缩算法能以高概率找到所有的最小割。

2000 年，Karger 提出了一种基于树包装的求解最小割的 $O(m \log^3 n)$ 的随机算法。[5] 这个算法也可以解决找到所有最小割的变体问题，复杂度为 $O(n^2 \log n)$。树包装是一个生成树的集合，满足图上的每条边被生成树包含的权重和不超过其边权。他们还称割与生成树为 k 关联，当且仅当割的边集与生成树边集的并集大小不超过 k。树包装可以生成一个大小为 $O(\log n)$ 的生成树集合，满足每个最小割都与其中至少 $\frac{1}{3}$ 的生成树 2 关联。如此一来，只要找到这组生成树 2 关联的所有割并判断其割值，就能算出所有的最小割。Karger 的树包装算法是目前最好的求解最小割的随机算法。

2021 年，Li 提出了一种针对 Karger 算法去随机化的 $O(m^{1+o(1)})$ 的确定性算法。[6] Li 的去随机化算法是目前最好的求解最小割的确定性算法。

1976 年，Dinitz 等人设计了一种叫作仙人掌图表示法的数据结构，以一个稀疏化图来表示所有的最小割。[7] [8] 前面提到的几个最小割算法虽然也能完成所有最小割的计算，但由于直接存储数量为 $O(n^2)$ 的最小割复杂度较高，因此找到的最小割以中间结果的形式存储在算法中，扩展能力有限。Dinitz 等人提出的仙人掌图表示法做到了用一个规模为 $O(n)$ 的图表示所有最小割。具体来说，他们为图 G 建立了一个仙人掌图 Γ 和映射 $\varphi : V_G \to V_\Gamma$，且对于任意 G 中的最小割 $(X, V\backslash X)$，其对应到 Γ 中的点集 $\varphi(X)$ 与 $\varphi(V\backslash X)$，都满足存在一个 Γ 中的最小割将其分隔开。Dinitz 等人也通过仙人掌图表示法，证明了图最小割的数量不超过 $\frac{n(n-1)}{2}$，这也是该结论最早的证明。

2009 年，Karger 基于其树包装最小割算法，提出了一个构造仙人掌图表示法

的 $O(m\log^4 n)$ 的随机算法。[9] 该算法的思想是借助树包装算法计算了所有点与边的极小最小割，再通过这部分信息设计一个递归过程完成仙人掌图表示法的构造。2024 年，He 等人将仙人掌图表示法构建算法进行优化，得到了一个 $O(m\log^3 n)$ 的随机算法，此外，他们还完成了算法的去随机化，得到了一个 $O(m\text{polylog}(n))$ 的确定性算法。[10]

差分隐私下的最小割算法也在近年来有所研究。2010 年，Gupta 等人提出了一种基于拉普拉斯机制的差分隐私最小割算法。[11] 他们的指数算法实现了 ε-差分隐私，并将得到的近似最小割与真实最小割的割值误差控制在 $O(\ln n/\varepsilon)$。此外他们还提出了 (ε,δ)-差分隐私的多项式时间复杂度算法，作为差分隐私算法的高效选择。

近几年，同样是稀疏化图的 Gomory-Hu 树在隐私化上取得了一定进展。Gomory-Hu 树以树的形式保有了全点对的 $s-t$ 最小割值，具体来说，它保证 $s-t$ 最小割的值等于 Gomory-Hu 树上 s 与 t 之间路径的边权最小值。2021 年，Li 等人提出了一个构建 $(1+\epsilon)$-近似 Gomory-Hu 树的 $\tilde{O}(m+n^{3/2}\epsilon^{-2})$ 的随机算法。[12] 这一工作基于他们此前提出的最小隔离割方法得出。[13] 2024 年，Aamand 等人对算法进行了隐私化，得到了一个加性误差为 $\tilde{O}(m/\varepsilon)$ 的构建 Gomory-Hu 树的 ε-差分隐私随机算法。[12]

2024 年，Liu 等人提出了一个针对图的隐私化算法，该算法能以 (ε,δ)-差分隐私地发布处理过后的图，并保证图上最小割的误差为 $\tilde{O}(\frac{\sqrt{nm}}{\epsilon})$。[14]

1.3 本文的主要内容

本文的目标是设计一个能够输出多个近似最小割的差分隐私算法。

TODO

第 2 章　符号表示与理论基础

2.1　图与最小割

我们用 $G=(V,E)$ 来表示一个 n 个点 m 条边无向图，其中点集是 V，边集是 E。图中连接点 u 和点 v 的边用 (u,v) 表示。若无向图带权，则我们用 w 来表示边权，也就是说，对于一条边 $(u,v)\in E$，其边权为 $w(u,v)$。对于图上中的两个点集 V_1,V_2，记连接两个点集的边集为

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

特别的，当 $V_2=V\backslash V_1$ 时，$E(V_1,V_2)$ 可以简化为 $E(V_1)$; 记连接两个点集的边权和为

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

特别的，当 $V_2=V\backslash V_1$ 时，$w(V_1,V_2)$ 可以简化为 $w(V_1)$。在本文中，如无特殊说明，图允许重边的存在，但不允许自环边的存在，图是有限图，且保证图是连通图。

接下来，我们给出图的最小割及相关概念的定义。

定义 2.1.1. 给定图 $G=(V,E)$，一个割 $R=(V_1,V_2)$ 是将顶点集合分成两个不相交的非空子集 V_1 和 V_2，即满足 $V_1\neq\emptyset,V_2\neq\emptyset,V_1\cap V_2=\emptyset,V_1\cup V_2=V$。

割定义中的点集没有先后顺序，即 $(V_1,V_2)=(V_2,V_1)$，所以只需要给出一个点集 V 的非空真子集 V_1 就可以唯一确定一个割。也就是说，割可以简化表示为 $\Delta(V_1)=R(V_1,V\backslash V_1)$。这个简化表示满足 $\Delta(V_1)=\Delta(V_2)$ 当且仅当 $V_1=V_2$ 或 $V_1=V\backslash V_2$。

定义 2.1.2. 给定图 $G=(V,E)$ 和图上的一个割 $R=(V_1,V_2)$，割的边集为 $E(V_1,V_2)$。

与割的简化表示相对应，割 $\Delta(V_1)$ 的边集也可以简化表示为 $E(V_1)$。

定义 2.1.3. 给定图 $G=(V,E)$ 和图上的一个割 $R=(V_1,V_2)$，割的容量为 $w(V_1,V_2)$。

割的容量也叫作割值。只需要将割的边集中的边权进行求和，就可以得到割的容量。因此，割的容量的另一个等价形式为 $w(V_1)$。

定义 2.1.4 (点的度数). 给定图 $G=(V,E)$，点 v 的度数为

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

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。

我们用 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，我们用 $M_G=|R_G^*|$ 表示图 G 的最小割数量。

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

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。

我们记 $S-T$ 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。Karger 给出了一个随机化的求解最小割的高效算法，他也通过这个算法对最小割的数量进行了估计。

定义 2.2.1. 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. 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 算法的思路如下：以均匀分布来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，那么我们说这个割是有效的。

Karger 提出了下面的定理来估计最小割数量。

定理 2.2.1. *[3]* 在算法进行到图被收缩至 k 个顶点时，一个给定的最小割有效的概率是 $\Omega((n/k)^{-2})$。

当 $k=2$ 时，给定的最小割仍然有效的概率为 $\Omega((n/2)^{-2})$，且在 k 的这个取值下，有且仅有一个割有效，因此，我们可以得到一个对最小割数量的估计。

定理 2.2.2. 给定图 $G=(V,E)$，图中最小割数量至多为 n^2。

Karger 还将其算法进行了推广，由此可以得到一个对近似最小割数量的估计。

定理 2.2.3. *[15]* 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)**.** *[7]* 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，我们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). *[16]* 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

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

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). *[17] [18]* 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

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

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，我们记该分布为 Lap(b)。

定理 2.4.2 (拉普拉斯机制). *[19]* 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

$$f(G) + (X_1, \ldots, X_d)$$

满足 ε 差分隐私。

定理 2.4.3 (差分隐私图). *[14]* 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，满足对于合成图中任意不想交的点集 $S,T\subseteq V_G$，满足

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

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。

TODO：指数机制

TODO：top-k 选择

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[7] 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，为差分隐私下的最小割问题的分析提供理论基础。

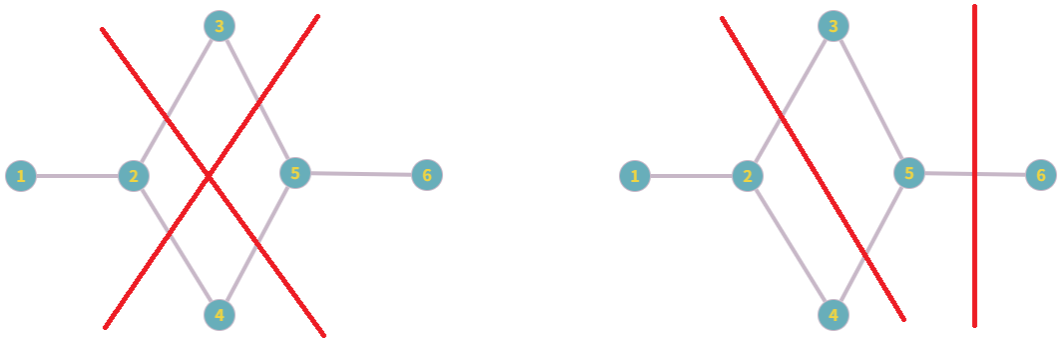
3.1　Dinitz 仙人掌图表示法简述

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。首先，我们给出最小割在仙人掌图表示法中的表示形式。

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

图3-1给出了割相交和平行的两个例子，其中的红色直线代表一个割。

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 $r*(G) \in R*(G)$，在最小割图中新建一个与之相对应的点。

- 若两个最小割 $r_1*(G), r_2*(G)$ 相交，则在最小割图中对应的点之间连一条边。

引理 3.1.1 (割的次模性). [20] 给定图 $G=(V,E)$ 和图的两个割 $\Delta(X), \Delta(Y)$，有

$$w(X)+w(Y) \geq w(X \cup Y)+w(X \cap Y)$$

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 根据割的次模性，我们可以得到

$$2\Phi_G \leq w(X \cup Y)+w(X \cap Y) \leq w(X)+w(Y)=2\Phi_G$$

因此，$w(X \cup Y)=w(X \cap Y)=\Phi_G$，第一条结论得证。

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

$$w(X \cap Y)+w(X \cap (V \backslash Y))=w(Y)+2w(X \cap Y, X \cap (V \backslash Y))$$

因此，$w(X \cap Y, X \cap (V \backslash Y))=\frac{\Phi_G}{2}$，第二条结论得证。

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

$$w(X \cap Y, X \cap (V \backslash Y))+w(X \cap Y,(V \backslash X) \cap Y)+w(X \cap Y,(V \backslash X) \cap (V \backslash Y))=w(X \cap Y)=\Phi_G$$

因此，$w(X \cap Y,(V \backslash X) \cap (V \backslash Y))=0$，第三条结论得证。 □

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，我们就称它为 p 割；否则，称它为 t 割。

图3-2给出了图中的所有 p 割（用蓝色直线表示）和 t 割（用红色直线表示）。

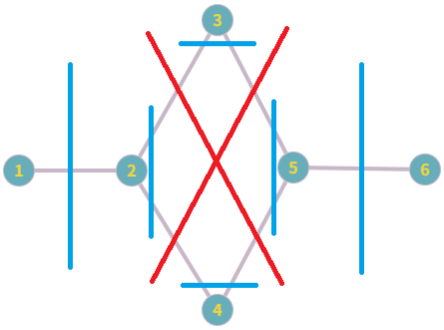


图 3-2 p 割和 t 割例

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 $X\cap Y'=\emptyset$。我们称点 v 处于 R 和 R' 之间当且仅当 $v\in X'\cap Y$。我们称割 $R''=(X'',Y'')$ 处于 R 和 R' 之间当且仅当 $X\subset X''$ 且 $Y'\subset Y''$。

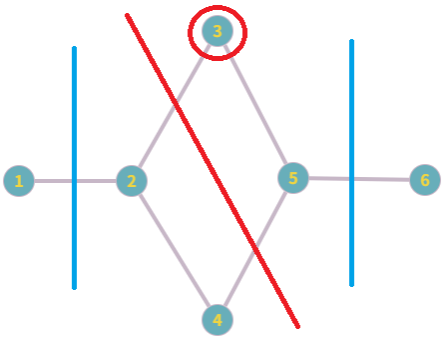


图 3-3 处于两个 p 割之间的点和割例

图3-3中红色点以及红色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'，我们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 $R=(X,V\backslash X)$，我们记 $S(X)$ 为所有与 R 相邻且分隔开 X 中顶点的割的集合。

定义 3.1.6 (p 束). 给定一个 p 割集合 S，我们称其为 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，我们用 X_S 表示，该 p 束由 R 的 X_S 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 当且仅当其在 S 中任意两个 p 割之间，S 此类顶点的集合记作 $V(S)$。t 割属于 p 束 S 当且仅当其在 S 中任意两个 p 割之间。特别的，对于叶 p 束 S，顶点 v 属于 S 当且仅当 $v \in X_S$。

这里需要特别说明的是，t 割不会属于任何叶 p 束，这是由割的次模性推论得到的。

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，$S \cap S'$ 中的元素 R 是唯一的，且 S, S' 分别位于 R 的两侧。

定理 3.1.3 (树表示法)**.** [7] 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)**.** [7] 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。

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，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。因此 $\phi = \phi'$。

图 G 的 p 束集合有限且唯一确定，而根据定理3.1.4可得 Λ 中的点与 G 的 p 束一一对应，且边与 p 束的相邻关系一一对应，因此 Λ 与 Λ' 相同。综上，(Λ, ϕ) 和 (Λ', ϕ') 是同一个树表示法。 □

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 $P' \preceq P$。

通俗来讲，一组割会将图的点集划分成若干个划分块，每个划分块就是一个原子。

定理 3.1.6 (最小割和 p 割对应的原子集等价). [7] 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。

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). [7] 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 $V(S) = \emptyset$，这是由 $\hat{\Phi_G}$ 环仅由 x_R 即 S 外部的顶点构成这一结果得到的；第二个结论是 S 内的 t 割恰好是将环分成两部分的割（需满足每一部分至少有两个点），且这些 t 割在最小割图中恰好构成一个联通块。

通过上述定义与定理可以发现，仙人掌图表示法用非环边表示 p 割，用环边二元组表示 t 割。

3.2 仙人掌图表示法的不唯一性

考虑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，通俗的说，Γ 中的点对应着 G 中的 0 个、1 个或多个点。首先，我们形式化的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

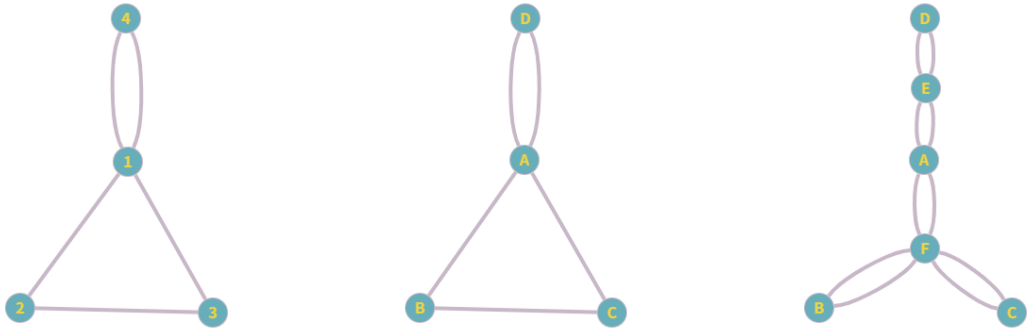
$$CutSet(\Gamma, \varphi) = \{(X, Y) \in R_\Gamma^* | (\bigcup_{x \in X} \varphi^{-1}(x), \bigcup_{y \in Y} \varphi^{-1}(y))\}$$

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

这个定义从侧面表明，一个仙人掌图表示法对应了原图的一个割集。但是，一个原图的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 想要证明这一点，我们只需要给出一个图 G 以及其两个不相同的仙人掌图表示法 $(\Gamma, \varphi), (\Gamma', \varphi')$。



(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

我们给出了一个 $n = 4$ 的例子，G, Γ, Γ' 的结构如图所示，且映射满足

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

在这个例子中，原图的最小割有 $(\{1,4\},\{2,3\}), (\{1,2,4\},\{3\}), (\{1,3,4\},\{2\}), (\{1,2,3\},\{4\})$。仙人掌图表示法 (Γ, φ) 使用了与图 G 相同的结构，因此其最小割与原图的最小割一一对应。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。 □

虽然图的仙人掌图表示法不具有唯一性，但其仙人掌图表示法相互等价，这是由表示的最小割集的唯一性得出的。因此，如果在仙人掌图表示法的生成算法中加入仙人掌图表示法的标准化算法 ALG_{std}，将输出的仙人掌图表示法转化为其等价类的标准元，那么就可以确保算法输出的唯一性。

特别的，仙人掌图表示法的标准化在差分隐私下尤为重要。对于最小割集相同的边相邻图，算法得到的仙人掌图表示法可能因边集的差异而不同，而在标准化处理后，输出将正确地被判定为相同。

3.3 标准化仙人掌图表示法

本节中，仙人掌图表示法的标准化共分为两部分。第一步是通过给出构造方法定义仙人掌图表示法的标准元。第二步是给出将现有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 割且方法唯一。根据定理3.1.7，p 束的结构图可以表示该 p 束的所有 t 割且方法唯一。因此，如果将树表示法和每个 p 束的结构图进行合成得到的图也是唯一的，且恰好能表示所有的最小割，我们将这个图设置为该仙人掌图表示法的标准元。算法1给出了 ALG_{gen} 的具体实现。我们称 $ALG_gen(G)$ 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 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，原本连向 v_S 的代表 p 割 R 的边重新连向 G_S 中的点 x_R。
6: **return** (Γ, φ)

定理 3.3.1. *给定图 G, G'，其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 (Γ, φ) 和 (Γ', φ') 等价，则根据定义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 Γ 的最小割集和 Γ' 的最小割集。$ALG_{gen}(G)$ 的仅与 G 的最小割集有关，因此 $ALG_{gen}(G) = ALG_{gen}(G')$。

□

算法 $ALG_{gen}(G)$ 给出了仙人掌图表示法的标准元，但难以直接用于仙人掌图表示法构造算法中。因为算法没有提供一个高效的计算方法，而朴素的 p 割，t 割，p 束计算需要较高复杂度，继而成为算法的效率瓶颈。

我们发现，如果我们首先利用现有工作生成一个仙人掌图表示法，然后使用仙人掌图表示法标准化算法 ALG_{std}，将仙人掌图表示法转化为其等价类内的标准元，那么就能同时实现高效和标准化。与此同时，ALG_{std} 可以通过输入仙人掌图表示法本身的性质，来得到一个较好的复杂度。

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：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
对于所有二元环，将环上的两条边合并为一条边。 ▷ 二元环表示 p 割
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
        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**
            if e 是环 C 上的边 **then**
                将 e 的 k 一端替换成 k'
            else if e 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。 ▷ 三元环只表示 p 割
for Γ 中度数为 2 的点 v **do**
    找到与 v 相连的边 $(v, u_1), (v, u_2)$
    if 若 v 不处于任何一个简单环上且 $\varphi^{-1}(v) = \emptyset$ **then**
        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')

回顾引理3.2.1中的例子，仙人掌图表示法主要需要解决的是三元环和链两种情况。算法2给出了 ALG_{std} 的具体实现。

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，我们分析仙人掌图表示中，单个环表示的最小割的数量和类型。根据定理3.1.7

首先，我们需要证明 $ALG_{std}((\Gamma, \varphi))$ 算法得到的环与 $ALG_{gen}(G)$ 的环一一对应。在仙人掌图表示中，非环边代表的最小割一定是 p 割，环中代表的最小割可能有 p 割也可能有 t 割。根据算法1可知，$ALG_{gen}(G)$ 的环仅用于表示 t 割。对于

一个 $ALG_{gen}(G)$ 里的简单环 C，若其表示的 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) 中一定由一个相同的环 C' 表示。对于一个仙人掌图表示 (Γ,φ) 中的环 C'，环上相邻的两条边可以表示一个 p 割，这些 p 割在算法中通过 (k',k'') 这条非环边重新表示了；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，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，我们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。(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。由于代表同一个 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} 通过收缩这样的点，就可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。

最终，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

在求解近似最小割的过程中，需要尽可能找到多的解，然而差分隐私的算法要求一条边的存在与否对输出的影响不能过大。因此，需要首先对最小割数量进行敏感性进行定量分析，才能设计出恰当的噪声添加值。

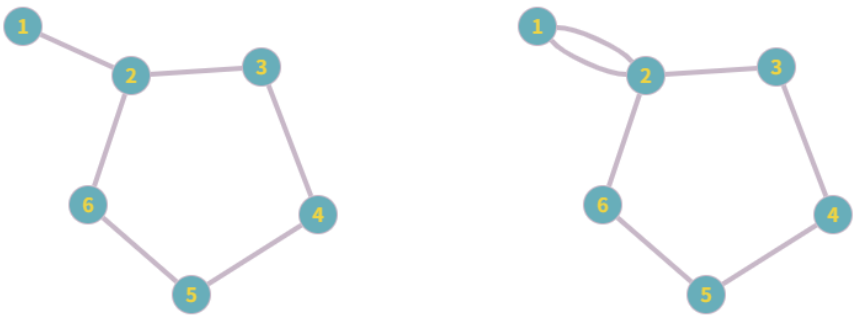
假设现在有两个边相邻的图 G, G'，其中 G' 由在 G 中加入一条边权为 1 的边 (u, v) 得到。不妨假设最小割数量计算函数的输入与输出都以适当的二进制形式进行编码，可以得到最小割数量的敏感度为 $d = |M_G - M_{G'}|$。

我们知道，对于任意图 G，最小割数量满足 $1 \leq M_G \leq n^2$，因此 $0 \leq d \leq n^2$。

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 我们不妨将点集中的点编号，用 v_1 至 v_n 表示。下面给出构造方法：图 G 由如下方法生成，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' 由图 G 的基础上，增加一条连接 v_1 和 v_2 的边得到。

这里的连边均为 1，因此图 G 的最小割值为 1，唯一的方案是 $(\{v_1\}, V \backslash v_1)$，因此 $M_G = 1$。图 G' 的最小割值为 2，此时对于任意 $2 \leq i \leq j \leq n$，$(\{v_i, \ldots, v_j\}, V \backslash \{v_i, \ldots, v_j\})$ 都是一个最小割，因此 $M_{G'} = \frac{n^2-n}{2}$。在这种构造方法下，$d = \Omega(n^2)$。 □



(a) 图 G　　(b) 图 G'

图 4-1　$n = 6$ 时的构造示例

图4-1给出了一种 $n = 6$ 时的构造示例。引理表明，边相邻图的最小割数量的敏感度在一些情况下特别高，高敏感度意味着需要添加较大噪声，进而使得差分隐私下发布的最小割数量可用性较低。因此在设计算法时，需要引入一些额外的约束条件。

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，可以加入边相邻图的最小割值相同这一额外限制条件，也就是说，图 G, G' 满足 $\Phi_G = \Phi_{G'}$。在本章的后文中，如未特殊说明，图 G, G' 将默认满足这一性质。

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对问题分析的精细性。所以，对于一个图 G 以及其标准仙人掌图表示法 (Γ, φ)，我们引入以下参数来描述图的结构：

- 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。

在 G 中加入一条边权为 1 的边 (u, v) 得到 G'，接下来我们分析这条边 (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的关系。首先，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 $\varphi(u) \neq \varphi(v)$，我们不妨令 $U = \varphi(u)$,$V = \varphi(V)$。根据标准仙人掌图表示法的构造过程可知，其表示最小割的结构主要有树表示和每个 p 束对应的结构环两部分。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 束对应的结构环代表的 t 割都会变少。具体来说，对于一个 p 束 S 的结构环 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

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

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

$$g(l,t) = f(l) - f(t) - f(l-t) - [t \ge 3] - [l-t \ge 3]$$

令 $G(l) = \max_{t=1}^{l-1} g(l,t)$，我们不妨对 l,t 的值进行讨论来得到该函数的取值：当 $1 \le l \le 3$ 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 $l \ge 5$ 时，由于 $l-t \ge t$，且 f 为单调函数因此我们只需要讨论 $t=2, t \ge 3$ 这两种情况。

- 当 $t=2$ 时，$g(l,2) = f(l) - f(l-2) - 1 = 2l-6$；
- 当 $t \ge 3$ 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，$t \le 2$，因此 $G(5) = g(5,2) = 4$。当 $l \ge 6$ 时，$g(l,t)$ 的极小值在 $t = \lfloor \frac{l}{2} \rfloor$ 时取到，此时 $g(l,t) \ge g(l,2)$。综上，可以得到

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

除此之外，可以发现，加入 (u,v) 使最小割的割值增加一的情况只有在 $\varphi(u) \ne \varphi(v)$，标准仙人掌图表示法中的树表示是一条链，且 $\varphi(u), \varphi(v)$ 分别是链的两个端点时出现。

接下来，我们给出最小割数量变化范围的表达式。

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*

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

最小割数量的变化还可以由 M_G 本身的值进行估计。对于一个长度 $l \ge 4$ 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 $G(l) = \lfloor \frac{l^2}{4} - 2 \rfloor$。此外不难证明，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

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

□

因此我们可以给出基于 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*

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

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的达到，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 $(1-\alpha)n$ 的点用边权 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。

首先，$M_G = n + \frac{\alpha n(\alpha n-3)}{2}$。敏感度最高的加边是 (v_1, v_4)，敏感度 $M_G - M_{G'} = \lfloor \frac{\alpha^2 n^2}{4} \rfloor + (1-\alpha)n$。不失一般性地取 $\alpha = \frac{1}{2}$，可得 $M_G = \frac{n(n+2)}{8}$，$M_G - M_{G'} = \lfloor \frac{n^2}{16} \rfloor + \frac{1}{2}n$。此时有

$$\frac{M_G}{2} + \frac{3}{2}n = \frac{n^2}{16} + \frac{13}{8}n \leq 4(M_G - M_{G'})$$

4.3 平均敏感度

前面的分析表明，在最坏情况下，最小割数量的敏感度较高。这一小节将通过平均敏感度分析造成高敏感度的情况的频次。

在平均敏感性的通常定义中，其边相邻的图以删边的形式给出。[21] 然而在前文描述边相邻图时，由于删边会较大的改变仙人掌表示的结构，因此用加边的形式描述了 G, G' 间的关系。这一部分将沿用加边的方法来定义平均敏感度。通过删边形式的平均敏感度，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，改为加边形式会弱化这一功能。加边形式的平均敏感度能对应前文的分析，并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*

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

考虑这样一种 G 的构造，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 $W(\frac{1}{3}n)$，接下来将 $\frac{1}{3}n$ 个点与 u 用极大的边权相连，将 $\frac{1}{3}n$ 个点与 v 用极大的边权相连。此时最小割数量函数 M 的平均敏感度满足

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

最高敏感度 W 是一个 n 的不超过二次的一个多项式。综上，存在一个常数 β 使得在该构造下

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

分析表明，在最坏情况下，导致高敏感度的加边出现频率的期望较高。

第 5 章　差分隐私下近似最小割求解算法

第 6 章　总结与展望

本文总结未来工作展望

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

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

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

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

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] LIU J, UPADHYAY J, ZOU Z. Almost linear time differentially private release of synthetic graphs[J]. arXiv preprint arXiv:2406.02156, 2024.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

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

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

致 谢

虽然还有一些工作没有完成，但是大学四年的生活在一点点走向尽头。我还清晰地记得自己初入校门时的激动，那是一段无需口罩的防护、可以直接看见彼此笑脸的时光。四年的大学生活没有想象中的轻松，在结束之际，向陪伴我走过困苦时期的各位表示感谢：