

哈尔滨工业大学计算机科学与技术学院
《算法设计与分析》

课程报告

学号	2021112845
姓名	张智雄
班级	2103601
专业	人工智能
授课教师	张炜
报告日期	2022 年 12 月 11 日

Optimal Bounds for the k -cut Problem

Anupam Gupta¹ David G. Harris² Euiwoong Lee³ Jason Li¹

¹Carnegie Mellon University ²University of Maryland ³University of Michigan

J. ACM 69, 1, Article 2 (November 2021) <https://doi.org/10.1145/3478018>

1 问题定义

1.1 问题重述

k -cut 问题，即给定一个边带权图 $G = (V, E, w)$ 和一个整数 k ，删除一组最小权重的边，使得 G 至少有 k 个连通分量。我们让 λ_k 表示删除边的结果权重。

当 $k = 2$ 时，此问题便退化为求解全局最小割问题，可使用最大流算法求解。而当 $k \geq 3$ 时，最早认为此问题是一个 NP-hard 问题，即不清楚该问题是否存在确定整数 k 的多项式时间算法。

1.2 研究现状

在 1988 年 Goldschmidt 和 Hochbaum 给出了 $n^{O(k^2)}$ 时间复杂度的确定性算法；随后 Karger 提出的基于随机边收缩的算法也可以在 $\tilde{O}(mn^{2k-1})$ 时间复杂度内求解此问题；后来 Karger 和 Stein 又将此算法时间复杂度优化至 $\tilde{O}(n^{2k-2})$ 。目前已经有许多改进的确定性算法：包括 Chekuri 等人优化了 Thorup 的 tree-packing 算法，将时间复杂度降低至 $O(mn^{2k-3})$ 。直到最近，采用特别方法的随机确定性算法已经实现了 $O(n^{(2-o(1))k})$ 的时间复杂度。（本文中，指数中的 $o(1)$ 表示随着 k 增加而变为零的常量。）

此前，Gupta、Lee 和 Li 使用有界深度搜索来猜测最小 k 割图内的 2-割集，并基于图中较“小”2-割集数量的极值界限给出了一般的时间复杂度为 $O(n^{(1.98+o(1))k})$ 的算法，并通过实验验证表明 $n^{(2-o(1))k}$ 并不是正确的界限。但现阶段仍未将其扩展到一个远低于 $2k$ 的指数。

1.3 论文的研究重点

此问题的难点在于， k -cut 问题可以归约为系数为 $k - 1$ 的最大团问题 $Max-Weight_{k-1}-Clique$ 从而确定此问题的下限。据推测，当边权重为 $[1, \Omega(n^k)]$ 范围内的整数时求解 $Max-Weight_k-Clique$ 的时间复杂度为 $\Omega(n^{(1-o(1))k})$ ，而对于无权图的求解时则需要 $\Omega(n^{(\omega/3-o(1))k})$ 的时间复杂度，其中 ω 是矩阵乘法常数。而将这些界限扩展到 k -cut 问题当中，表明 $n^{(1-O(1))k}$ 可能是一般加权 k -cut 问题实例数目的最优界限。

而论文主要解决的问题是改进了 Karger 提出的收缩算法，提出了 $n^k k^{O(k^2)}$ 的最小权重 k -割图数目，改进了此前的最优界限，并且将其计算的时间复杂度降低至 $n^k (\log n)^{O(k^2)}$ 。

2 算法描述和分析

2.1 算法的概述

论文基于 Karger 最初提出的求解全局最小割的收缩算法做出改进，改变循环终止条件，并给出了对一般图的 k -cut 问题的有效边界约束。

ALGORITHM 1 Contraction Algorithm(收缩算法)

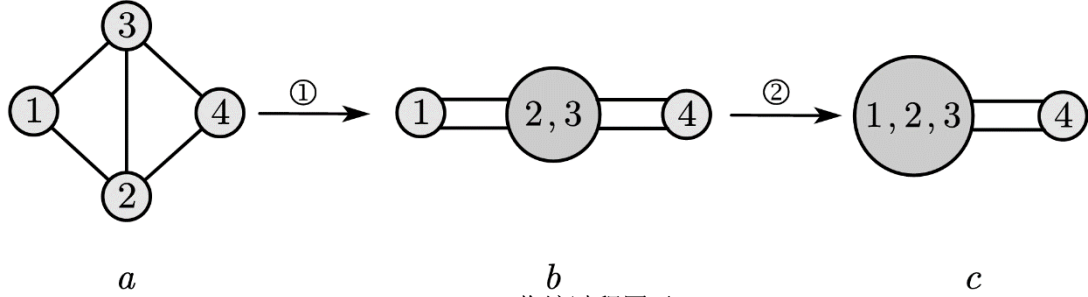
```

1: while  $|V| > \tau$  do
2:     从  $G$  中随机选择一条边  $e \in E$ ，概率与其权重成正比；
3:     连接边  $e$  的两个顶点并删除自环边；
4: end while
5: 返回一个均匀随机选择的图  $G$  的  $k$ -割图。
    
```

直观来看，**算法的思想**就是对于一个图 $G = (V, E)$ ，为找到它的权重最小的 k -割集，每次循环随机选择一条边，概率与其权重成正比。将这条边无限变短直到两关联结点重合，然后继承除被缩短的边以外其余所有边后删去自环边。重复以上过程，直到图中只剩下 τ 个结点。这 τ 个结点代表的就是这样的一个最小 k -割集。在此算法中， τ 为表示最终期望的图的大小的一个参数，最简单的想法是令 $\tau = k$ ，但是在实际分析中需要一个更大的 τ 。

下以 $\tau = 2$ ，顶点数为 4 的无向图为例简单说明算法的具体收缩过程：第①步合并顶点 2,3 后原本图 a 中连接 2,3 的边成为自环边，删除；第②步合并顶点

1,2,3 后原本图 b 中连接 1,2,3 的边成为自环边，删除。此时图中顶点数等于 2，结束循环。此时图中剩余的 2 个顶点即代表一个 2-割集。



2.2 算法的最优界限以及复杂度分析

此算法主要描述了如下定理和推论：

定理 2.1 对于任意整数 $k \geq 2$ 和实数 $\alpha \geq 1$ ，在选择适当的 $\tau = \text{poly}(\alpha, k)$ 的情况下，收缩算法以至少 $n^{-\alpha k} k^{-O(\alpha k^2)}$ 的概率输出权重为 $\alpha \lambda_k$ 的每一个 k 割图。

由于任一最小权重的 k 割图（对应 $\alpha = 1$ ）的输出概率为 $n^{-k} k^{-O(k^2)}$ ，于是可以得出推论 2.2。

推论 2.2（最小 k 割图的数量） 对于任意 $k \geq 2$ ，图 G 含有互不相同的最小权重的 k -割集的数目最多为 $n^k k^{O(k^2)}$ 个。

此算法改进了此前的最优界限 $n^{(1.98+o(1))k}$ 。并且，尽管上述收缩算法的直接实现会产生额外的 $O(n^2)$ 时间复杂度，但 Karger 和 Stein 的递归收缩算法可以在几乎与之匹配的复杂度内来列举所有的 k 割图。

定理 2.3（寻找最小 k 割图的更快速算法） 存在一算法可以在 $n^k (\log n)^{O(k^2)}$ 时间复杂度内以 $1 - 1/\text{poly}(n)$ 列举出所有的最小权重的 k 割图。

对于 $k = 2$ 的情况，此算法为传统的递归收缩算法^[2]。这一定理一定程度上降低了 Karger 此前提出的 $n^{(1.98+o(1))k}$ 的时间复杂度，甚至超过了无权图情况下 $n^{(1+o(1))k}$ 的时间复杂度。

2.3 算法过程的具体分析与证明

假设 $G = (V, E)$ 是一个具有 n 个顶点和 m 条边的无权多重图。而加权图可通过上述无权图复制边得到，因此可视为是等价的。在此，边数 m 可能比顶点数 n 成指数级的大。而算法的计算组合边界将取决于 n ，而非直接取决于 m 。

基于 Gupta、Lee 和 Li 此前的研究，论文的证明主要包含两个部分：(I)图中的极值数的边界约束，以及(II)收缩算法的新算法分析。

首先，作者给出了一个较为粗略的极值结果。定义 $\overline{\lambda}_k = \lambda_k/k$ ，表示一个最小 k 割图中的 k 个连通分量的平均值，而 *Middle-Cuts* 表示权重在 $[\overline{\lambda}_k, 2\overline{\lambda}_k)$ 的 2-割集。但是该图可能存在权重小于 $\overline{\lambda}_k$ 的较“小”2-割集，记为 *Small-Cuts*。在此 *Small-Cuts* 的数量可忽略不计。

同时，作者在此引入了“极值约束”的概念。

定理 2.4（极值约束） 对于任意确定的整数 k ，图 G 中最多存在 $O(n)$ 数量级的 *Middle-Cuts*。

证明 在针对图的割的研究中，循环图和团是我们主要研究范围的两个极端。在循环图中，有 $\overline{\lambda}_k = 1$ ，并且没有权重小于 $2\overline{\lambda}_k$ 的割边，定理成立。但是，在循环图中存在 C_n^2 个最小 2-割集，且权重等于 $2\overline{\lambda}_k = 2$ 。而在团中，最小 k -割集剪切掉了 $k-1$ 个顶点，故 $\lambda_k = C_{n-1}^2 + (k-1)(n-k+1)$ ，而当 $n \gg k$ 时， $\overline{\lambda}_k \approx (k-1)n/k$ ，并且存在 n 个最小 2-割集，其权重为 $n-1 < 2\overline{\lambda}_k$ ，定理成立。同样地，在团中存在 C_n^2 数量的权重与 $2\overline{\lambda}_k$ 近似的最小 2-割集。

2.3.1 收缩算法的粗略分析

当我们对图 G 进行收缩时，定理 2.4 确保了最多存在 $O_k(n)$ 个在 $[\overline{\lambda}_k, 2\overline{\lambda}_k)$ 范围内的 *Middle-Cuts*，再加上数量可以忽略不计的 *Small-Cuts*。我们假设 *Middle-Cuts* 的数目为 n ，且图 G 中不存在 *Small-Cuts*（数据的精确与否对整体的分析并没有影响）。

在收缩算法中，每一个顶点都对应原图的一个 2-割集，假设图 G 不存在 *Small-Cuts*，故收缩算法中每次迭代 i 的过程中边的数量是以 $i\overline{\lambda}_k/2$ 为下界的。为了较为直观计算，我们假设图正好有这么多的边。那么在每次迭代中 i ，每个 *Middle-Cuts* 以至少 $\overline{\lambda}_k/(i\overline{\lambda}_k/2) = 2/i$ 的概率确定一条边，而后这条边从图中被删除。因此，经过 $n/2$ 次迭代后，图中的 *Middle-Cuts* 的数目近似为：

$$n \prod_{i=\frac{n}{2}}^n \left(1 - \frac{2}{i}\right) \approx \frac{n}{4}$$

由此，在产生的顶点数为 $n/2$ 的子图中，最多只有 $n/4$ 个顶点（对应 *Middle-Cuts*）的度数为 $\overline{\lambda}_k$ ，其余顶点的度数至少为 $2\overline{\lambda}_k$ 。继续收缩过程，图中

的高度数顶点会越来越多。经过 $(1 - \varepsilon)n$ 次迭代后（ ε 为一个极小常数），几乎所有 $Middle-Cuts$ 都被删除了，并且在顶点数 $i \leq \varepsilon n$ 的每个图中，其边数都十分接近于 $i\overline{\lambda}_k$ 。

现在推广至 k -割集的情形，考虑任意一个最小 k -割集 K ，它以恒定的概率在前 $(1 - \varepsilon)n$ 次迭代中保留下来。在迭代过程中，当得到的结果图顶点数 $i \leq \varepsilon n$ 时， K 被选中的概率近似为 $\lambda_k / i\overline{\lambda}_k = k/i$ 。而在收缩算法的整个运行过程中，得到最终顶点数为 $\tau = \text{poly}(k)$ 的图时， K 被保留下来的概率近似为：

$$\text{constant} \cdot \prod_{i=\tau}^{\varepsilon n} \left(1 - \frac{k}{i}\right) \approx \Theta(n^{-k})$$

但上述的证明仍是粗略的、不够严密的，首先是对许多界限对参数大小作出了强加的假设，例如只保证了图中边数的下限，并且不应该强行假设每次迭代中的变化是相同的；其次，收缩算法是一个随机的过程，并不能直接假设相关的数量等于其期望值。

2.3.2 收缩算法的精细分析

要对此收缩算法进行较为精确地分析，首先需要引入有关割的简单概念。对于一个整数 $k \geq 3$ ，一个 k -割集 K 是将 V 划分为 k 个非子集， ∂K 表示跨越 K -割图不同部分的边集。定义 K 的权重是边集 ∂K 的势， λ_k 是任何 k -割集的最小权重，并有 $\overline{\lambda}_k := \lambda_k / k$ 。

一个2-割集 $\{C, V \setminus C\}$ 称为割，用 C 表示。割的岸表示 C 或 $V \setminus C$ 中较小的集合。（如果它们的大小相同，则任意选择一个），而岸的大小是岸的势。

对于顶点集 A, B ，我们让 $E(A, B)$ 表示连接 A 和 B 两个集合的边的集合，表示为 $\partial S = E(C, V \setminus C)$ 。同时，定义 $Small-Cuts$ 满足 $|\partial C| < \frac{k-1}{k} \overline{\lambda}_k$ ， $Middle-Cuts$ 满足 $\frac{k-1}{k} \overline{\lambda}_k \leq |\partial C| < 2\overline{\lambda}_k$ 。

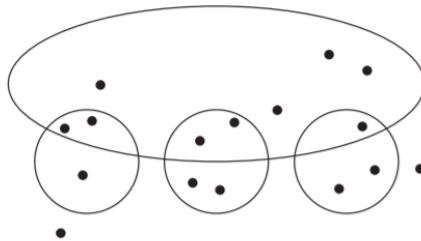


Figure 2 含有 8 个原子的 Venn 图

如上图，给定顶点集 F_1, F_2, \dots, F_t ， $Venn(F_1, \dots, F_t)$ 表示它们的Venn图。Venn图的一个非空区域，即一个非空集合，可以表示为 $G_1 \cap \dots \cap G_t$ ，其中每个集合 G_i 要么是 F_i ，要么是它的补集 $V \setminus F_i$ 。

F_1, F_2, \dots, F_t 生成一个 ℓ -割集 $K = \{A_1, \dots, A_t\}$ ，其中 A_1, \dots, A_t 是 $Venn(F_1, \dots, F_t)$ 的原子。 K 的权重最多为 $|\partial F_1| + \dots + |\partial F_t|$ 。由此可引出以下结论：

定理 2.5 如果 $n \geq k$ ，则 $m \geq \frac{nk}{2(k-1)} \bar{\lambda}_k$ 。

证明 将所有顶点按度数升序排序，即 $\deg(v_1) \leq \deg(v_2) \leq \dots \leq \deg(v_t)$ ，则由点集 $\{v_1\}, \{v_2\}, \dots, \{v_{k-1}\}$ 生成的 k -割集的权重最多为 $\deg(v_1) + \dots + \deg(v_{k-1})$ 。又由 λ_k 是最小 k -割集权重，因此有 $\deg(v_1) + \dots + \deg(v_{k-1}) \geq \lambda_k$ ，因此当 $i \geq k$ 时，有 $\deg(v_i) \geq \deg(v_{k-1}) \geq \frac{\deg(v_1) + \dots + \deg(v_{k-1})}{k-1} \geq \frac{\lambda_k}{k-1}$ 。将所有顶点度数相加，由边与顶点度数的关系可知：

$$\begin{aligned} 2m &= [\deg(v_1) + \dots + \deg(v_{k-1})] + [\deg(v_k) + \dots + \deg(v_n)] \\ &\geq \lambda_k + (n - k + 1) \cdot \lambda_k / (k - 1) = n\lambda_k / (k - 1) = nk\bar{\lambda}_k / (k - 1) \end{aligned}$$

定理 2.6 图 G 中Small-Cuts的数量一定少于 2^{k-2} 。

证明 采用反证法，假设存在 2^{k-2} 个或更多的Small-Cuts，我们将能构造一个权重小于 λ_k 的 k -割集，与 λ_k 的定义相矛盾。

定理 2.7 假设 $T_1, T_2, \dots, T_{r-1}, T_r$ 为Middle-Cuts，其中 $r = \lceil k/2 \rceil$ 。则要么不等式 $|Venn(T_1, T_2, \dots, T_{r-1})| < 2(r-1)$ ，要么 $|Venn(T_1, \dots, T_r)| < 2r$ （或同时成立）。

证明 采用反证法，对 k 分两种情况讨论。当 k 为偶数时，则 $r = k/2$ ，我们假设 $|Venn(T_1, \dots, T_r)| = t \geq k$ ，则 T_1, \dots, T_r 可以生成一个 t -割集 K ，由于 T_1, \dots, T_r 均为Middle-Cuts，可推出 K 的权值小于 $r \cdot 2\bar{\lambda}_k = \lambda_k$ ，与 λ_k 的定义矛盾；当 k 为奇数时， $r = (k+1)/2$ ，假设有不等式 $|Venn(T_1, \dots, T_r)| = t \geq k+1$ 以及 $|Venn(T_1, \dots, T_{r-1})| = t' \geq k-1$ 成立。则 T_1, \dots, T_{r-1} 可以生成一个 t' -割集 K' ，由于 T_1, \dots, T_{r-1} 均为Middle-Cuts，因此可以推知 K 的权值小于 $(r-1) \cdot 2\bar{\lambda}_k = [(k-1)/k] \lambda_k$ ，如果 $t' \geq k$ ，则与 λ_k 的定义产生矛盾，因此一定有 $t' = k-1$ ，由此又能生成一个权重小于 $(r-1) \cdot 2\bar{\lambda}_k + \bar{\lambda}_k = \lambda_k$ 的 k -割集 K'' ，与 λ_k 矛盾。由此可证得定理成立。

结合以上定理，引入改进的向日葵(Sunflower)定理^[3]可推得如下结论：

推论 2.8 图中存在 $k^{O(k)}n$ 数量的 *Middle-Cuts*.

推论 2.9 假设所有 *Middle-Cuts* 中均存在势最大为 k 的岸，那图中最多存在 $k^{O(k)}n$ 数量的 *Middle-Cuts*.

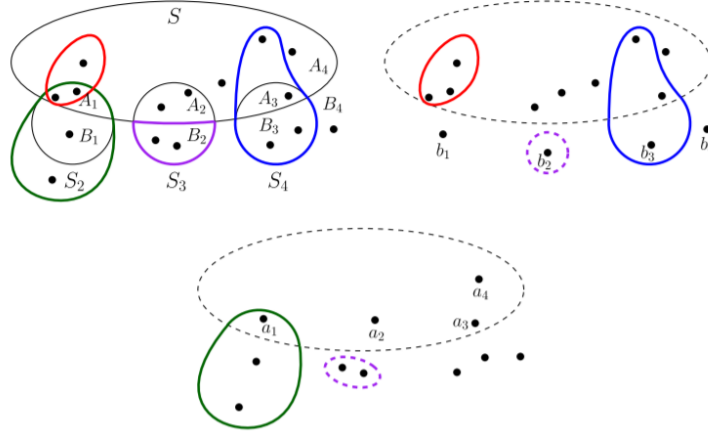


Figure 3 构建图 H_1 (右)和 H_2 (中)，给定 *Middle-Cuts* S_1, S_2, S_3, S_4 (左)。
每组颜色代表一个在 H_1 或 H_2 中保留的 *Middle-Cuts*,

如图，对图 G 按照不同的收缩方式可以得到 H_1 和 H_2 两幅图，由上述推论我们可以归纳得到原图 G 中的 *Middle-Cuts* 数目最多为：

$$c_k(n_1 - k) + c_k(n_2 - k) = c_k[(n - |S| + i) + (|S| + j) - 2k] = c_k(n + i + j - 2k)$$

其中， $c_k = k^{O(k)}$ 。又由推论可知 $i + j \leq k$ ，故 *Middle-Cuts* 数目最多为 $c_k(n - k)$ ，这与推论 2.8 中的结果也是相吻合的。

下一步是求解一个给定的 k -割集在完整收缩过程中被保留的概率下界，按照 Harris 和 Srinivasan 的研究方法^[4]，可以得到如下推论：

推论 2.10 对任意参数 $\alpha \geq 1$ 和任意 $|\partial K| \leq \alpha \lambda_k$ 的 k -割集 K ，参数 $\tau = \lceil 4\alpha k \rceil$ 时，收缩算法选择 K 的概率至少为 $n^{-2\alpha(k-1)}k^{-O(\alpha k)}$ 。

有了关于割集的保留概率的简单界限，接下来具体分析收缩过程中的动态变化。对于给定边集 J ，令 $\alpha = |J|/\lambda_k$ ，引入参数 $\varepsilon \in [0, 1/k)$ ，并定义 *Good-Cuts* C 满足 $|\partial C \setminus J| \geq (1 - \varepsilon) \frac{k}{k-1} \bar{\lambda}_k$ 。定义一个关键统计量：

$$R_i = \sum_{j=i+1}^n \frac{\bar{\lambda}_k}{|E(G_j)|}$$

在此， R_i 作为收缩算法中概率的一个线性近似，其中 G_j 为第 j 次迭代过程中得到的子图，同时还定义了两个相关参数：

$$\delta := \frac{1 - \varepsilon k}{k - 1}, \quad \beta := k + 2\alpha k / \varepsilon$$

于是我们可以对收缩算法的单次迭代中的边数下限做出估计：

定理 2.11 令 s 为图 G 中 *Good-Cuts* 的数量，如果 $n \geq \beta$ ，则有

$$m \geq s \cdot \frac{k}{k-1} \bar{\lambda}_k / 2 + (n - s - \beta) \cdot \bar{\lambda}_k$$

证明 图 G 中最多只有 $k - 2$ 个顶点对应 *Small-Cuts*，且最多有 $s + 2\alpha k/\varepsilon$ 个顶点对应 *Middle-Cuts*，而剩余的顶点度数至少为 $2\lambda_k$ ，由顶点与度的关系可得：

$$2m \geq (s + 2\alpha k/\varepsilon) \cdot \frac{k}{k-1} \bar{\lambda}_k + (n - s - \frac{2\alpha k}{\varepsilon} - k + 2) \cdot 2\bar{\lambda}_k \geq s \cdot \frac{k}{k-1} \bar{\lambda}_k / 2 + (n - s - \beta) \cdot \bar{\lambda}_k$$

推论 2.12 如果图 G 存在 *Good-Cuts*，则 $m \geq (n - \beta) \bar{\lambda}_k - \min\{s, n - \beta\} \bar{\lambda}_k / 2$ 。

接下来推导对随机变量 R_i 的关键约束，对于 $p \geq j$ 和 $s \geq 0$ ，定义函数

$$f(j, s, p) = \log(p/j) + \frac{\log(1 + (s/p)(1 + 1/\delta)(1 - (j/p)^\delta))}{1 + \delta}$$

利用此函数的非负性、单调不减以及其下凹的性质，借助连续的微分方程系统得以得出如下推论：

推论 2.13 如果图 G 中 *Good-Cuts* 的个数为 s ，并且包含 n 个顶点，则在 J 的收缩过程中的某一阶段 i 满足 $\beta \leq i \leq n$ ，有 $E[R_i] \leq f(i - \beta, s, n - \beta)$ 。

将上述定理以及相关推论整合起来即可得到算法描述中提及的定理 2.1、推论 2.2、定理 2.3，即可证明 k -cut 问题的最佳界限为 $n^k k^{O(k^2)}$ ，且能够在 $n^k (\log n)^{O(k^2)}$ 时间复杂度内解决。

3 讨论和分析

3.1 总结讨论

在这篇论文中，作者通过巧妙地引入一系列启发式的最坏情况假设，模糊了期望与实际取值的差异，并将离散的随机过程放宽为连续的微分方程系统得出了对收缩算法的过程具体细节的猜测。而后利用其公式良好的凹凸性和单调性，使用归纳法归纳证明这个公式在最坏情况下成立。

并且论文在对收缩算法进行分析时主要考虑边是以条为单位收缩的，此外还有令每条边以某个给定概率独立收缩的分析策略^[5]，这相当于一次性考虑执行了多步的收缩算法。在某些方面，这种独立收缩的分析策略更加简明直接，在保留边之间的许多独立性的同时集中分析一些相关参数。但缺点是缺乏对具体收缩边的数量的精确控制，且当顶点数较小时，会引入及更大的误差。

3.2 算法的不足与改进

在 $Max-Weight_k-Clique$ 问题需要 $n^{(1-o(1))k}$ 时间复杂度的假设下, 论文给出的时间复杂度为 $n^k(\log n)^{O(k^2)}$ 的算法几乎是最优的。但对于任意常数 $c < 1$ 的单位加权图, 实现时间复杂度为 $O(n^{ck})$ 仍是一个未解决的难题。

同时, 由于收缩算法本身是一个近似算法, 其随机性使得其具体细节是难以准确确定的, 文章通过一系列的约束条件以及模糊假设将时间复杂度优化逼近到一个较低水平, 但仍有提升的空间。而对于下一步的改进以及深入研究, 可能的方向我个人认为有三个:

a) 删除或减弱一部分约束条件以及假设, 利用更深入精细的数学分析对算法细节进行更精确的研究以便于进一步的算法优化;

b) 改用性质更优秀的启发函数, 利用其函数性质对收缩中边的变化进行更为准确的计算, 将范围进一步缩小;

c) 修改线性近似统计量 R_i 的表达式, 进一步提升其拟合程度, 进而提高计算的准确性以及可靠性。

关于 $k-cut$ 问题的研究未来还有许多工作去探索, 而解决此问题对计算机视觉等领域的发展以及技术突破具有重要意义, 以上为此论文的全部概述, 期待未来能在此问题上再有重大突破。

参考文献

- [1]Gupta A, Harris D G, Lee E, et al. Optimal Bounds for the k-cut Problem[J]. ACM Journal of the ACM (JACM), 2021, 69(1): 1-18.
- [2]Karger D R, Stein C. A new approach to the minimum cut problem[J]. Journal of the ACM (JACM), 1996, 43(4): 601-640.
- [3]Alweiss R, Lovett S, Wu K, et al. Improved bounds for the sunflower lemma[C]//Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing. 2020: 624-630.
- [4]Harris D G, Srinivasan A. Improved bounds and algorithms for graph cuts and network reliability[J]. Random Structures & Algorithms, 2018, 52(1): 74-135.
- [5]Gupta A, Lee E, Li J. The Karger-Stein algorithm is optimal for k-cut[C]//Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing. 2020: 473-484.