arXiv:2603.02006v1 [cs.DS] 2 Mar 2026

Kruskal-EDS：边动态分层一种受遍历理论启发、适应分布的最小生成树算法

Yves Mercadier

计算机科学与应用数学系

yves.mercadiergdr@proton.me

2026年3月3日

摘要

我们提出了**Kruskal-EDS** （边动态分层），这是 Kruskal 最小生成树算法的一种适应分布的变体。该算法以受 Birkhoff 遍历定理启发的三阶段流程，取代了强制进行的 $\Theta(m \log m)$全局排序。

在阶段1中，通过抽取 $\sqrt{m}$ 条边作为样本，可在$O(\sqrt{m}\log m)$时间内估计权重分布。在阶段 2 中，通过对分位数边界进行二分搜索，将全部m 条边分配到k 层中，耗时$O(m \log k)$，无需全局排序。在阶段 3 中，按顺序对各层进行排序和处理；一旦接受了$n-1$ 条最小生成树边，执行即停止。

我们证明，其有效复杂度为$O(m + p \cdot (m/k)\log(m/k))$，其中$p \leq k$ 表示实际处理的层数。在稀疏图或重尾权重分布中，$p \ll k$，算法可实现接近$O(m)$ 的性能。我们进一步推导出最优分层数$k^* = \lceil \sqrt{m}/\ln(m+1) \rceil$，以平衡划分开销与层内排序成本。

对 14 个图族进行的大规模基准测试表明，该算法在 12 个测试用例上正确，并且相较于标准 Kruskal 算法，在实际运行时间和排序操作方面分别实现了 **10**× 和 **33**× 的加速比。复杂度景观的三维 TikZ 可视化展示了算法如何随图密度和权重分布偏斜程度而自适应变化。

关键词：最小生成树，Kruskal 算法，分布自适应排序，分层抽样，Birkhoff 遍历定理，边划分，稀疏图。

目录

1 引言 **3**

2 背景与相关工作 3

2.1 经典最小生成树算法 .3 2.2 分布感知排序与划分 .4 2.3 从遍历抽样到边分层 .4 2.4 遍历理论与算法 .4

3 问题陈述与符号表示 **5**

4 算法 **5**

4.1 概述.5 4.2 阶段1：通过伯克霍夫抽样进行分布估计 .5

4.3 阶段 2：分层划分.64.4 阶段 3：增量排序与并查集处理 .6

5 理论　　分析 7

5.1 正确性.75.2 复杂度分析 .75.3 实现细节和数据结构 .85.4 最优分层数 .85.5 与标准 Kruskal 的比较 .8

6 实验评估　　8

6.1 实验设置 .86.2 图族 .96.3 正确性验证 .96.4 基准测试结果 .96.5 三维复杂度图景 .96.6 分层概况 .96.7 对分层数的敏感性 k .9

7 讨论　　10

7.1 EDS 何时胜出？ .107.2 伯克霍夫类比.117.3 与基于堆的 Kruskal 的比较.117.4 负权重. 11

8 展望与未来工作12

9 结论 13

1 引言

该 最小生成树 （MST）问题是组合优化中最基础的问题之一。给定一个连通加权图 $G = (V, E, w)$，其中包含 $|V| = n$ 个顶点和$|E| = m$ 条边，最小生成树是总权重最小的生成树。最小生成树算法广泛应用于网络设计、图像分割、聚类、超大规模集成电路布局，以及旅行商问题等NP难问题的近似算法中 [Cormen 等人，2022]。

排序瓶颈。 Kruskal 算法 [Kruskal， 1956] 仍是最简单、部署最广泛的最小生成树算法之一：按权重对所有边排序，然后贪心地加入不形成环的最轻边。 其复杂度由全局排序主导：$\Theta(m \log m)$。对于无结构输入，这在基于比较的排序中是最优的。然而，该算法仅使用 $n-1$ 条边（共 m 条），而在稀疏图或权重分布有利的图上，对全部 m 条边排序是浪费的。

遍历洞见。 在动力系统中，伯克霍夫遍历定理[Birkhoff， 1931] 保证几乎每条轨道上的时间平均都收敛于空间平均——一个小样本便足以刻画全局测度。阿图尔·阿维拉关于拟周期余循环的菲尔兹奖工作 [Ávila， 2015] 将这一点推进得更远：线性余循环的李雅普诺夫谱由其在指数级小的轨道集合上的行为所决定。

我们作如下类比：E 的权重分布就是最小生成树问题中的不变测度 。对 $\sqrt{m}$ 条边进行采样，可将该测度估计到$O(1/\sqrt{m})$ 分位数误差（根据 Dvoretzky-Kiefer-Wolfowitz 不等式 [Dvoretzky、Kiefer 和 Wolfowitz， 1956]）。基于该估计，我们构造k 个层 （权重区间），在$O(m \log k)$ 时间内将E 划分为若干部分，无需全局排序，然后按顺序处理各层，直至最小生成树构建完成。

贡献。
C1. 算法：Kruskal-EDS，一种具有分布自适应复杂度的三阶段最小生成树算法（§4）。**C2. 理论**：正确性证明、关于最优分层数的复杂度分析 k^*，以及对特定分布的分析（§5）。**C3. 三维复杂度图景**：一种新颖的 TikZ 可视化，展示算法复杂度如何随密度和分布偏斜度变化（§6）。
C4. 实证评估：对 14 个图族在四种权重分布下进行基准测试（§6）。

2 背景与相关工作

2.1 经典最小生成树算法

Kruskal（1956年）。按权重对边排序，然后使用并查集结构来测试并合并连通分量[Kruskal， 1956]。总复杂度：$\Theta(m \log m)$（由排序主导）。

Prim（1957）。从一个种子顶点出发，使用优先队列逐步扩展最小生成树。使用斐波那契堆时：$O(m + n \log n)$ [Prim， 1957]。当$m = \omega(n \log n)$时性能更优，即适用于稠密图。

Bor˚uvka（1926年）。每个阶段中，每个连通分量都会选择其最轻出边。经过 $\log n$个阶段后：$O(m \log n)$ [Bor˚uvka， 1926] 天然适合并行化。

Fredman-Tarjan（1987年）。将Bor˚uvka阶段与斐波那契堆Prim相结合：$O(m\ \beta(m,n))$其中 $\beta(m,n) = \min\{i : \log^{(i)} n \leq m/n\}$ [Fredman与Tarjan，1987]。

Chazelle（2000年）。首个近线性最小生成树算法：$O(m\ \alpha(m,n))$ 使用软堆 [Chazelle,2000], 其中 α 为反阿克曼函数。

线性最小生成树（随机化）。Karger、Klein与Tarjan（1995年）取得了期望$O(m)$ 时间，借助随机抽样与递归验证过程 [Karger、Klein与Tarjan，1995]。该方法渐近最优，但实现复杂。

最优确定性最小生成树。最小生成树的决策树复杂度为$\Theta(m)$ [Pettie与Ramachandran,2002], 但尚无已知的显式$O(m)$ 确定性算法。

2.2 分布感知排序与划分

整数排序（基数排序、计数排序）对于有界整数键可达到 $O(m)$ 时间复杂度 [高德纳，1998]。样本排序 [Frazer与McKellar， 1970] 将 m 个元素划分到各组中，使用 $\sqrt{m}$ 个分割器，并对每个桶独立排序。我们的分层将样本排序推广到最小生成树场景，并采用提前终止：对于最小生成树不需要的层，不再进行排序。

Thorssen与Kaplan [2002] 研究了利用整数权重分布的最小生成树算法；我们的方法与之互补，针对具有任意分布的实值权重。

2.3 从遍历抽样到边分层

Kruskal-EDS 的效率依赖于这样一个假设：大小为 $s \ll m$ 的子抽样能够捕捉权重分布的基本几何特征。形式上，我们将边集E 视为离散概率空间$(E, \mathcal{F}, \mu)$，其中 μ 表示权重的经验分布$w : E \to \mathbb{R}^+$。

秉承伯克霍夫遍历定理的思想，我们将对s 条边的随机抽样视为对系统“不变测度”的观测。若将图的构造视为一个从连续密度$f(x)$中抽取权重的过程，则抽样阶段旨在重构最小生成树的类李雅普诺夫增长。

依循 Avila 关于测度量子化的研究，我们的分层可以视为权空间的最优划分。随后，DKW不等式（公式 **??**）给出了收敛速度：它确保由我们的样本得到的经验测度 $\hat{\mu}_s$ 以高概率与真实测度 μ 的偏差不超过 ε。这一联系说明了为什么“局部”视角（样本）足以决定“全局”策略（对m条边的划分）。

2.4 遍历理论与算法

遍历理论与算法之间的联系早已在马尔可夫链混合时间分析中得到确立 [Levin 和 Peres， 2017] 以及伪随机生成器构造中。Ávila 关于准周期余循环的研究 [Ávila， 2015] 证明了一个小轨道可以决定

全局李雅普诺夫谱——这是遍历原理的一个定量体现。据我们所知，此前尚未有研究将伯克霍夫抽样应用于最小生成树的自适应边分层。

3 问题陈述与符号表示

全文中，$G = (V, E, w)$ 表示一个具有 $n = |V|$ 个顶点与 $m = |E|$ 条边的连通无向加权图，$w : E \to \mathbb{R}$。我们用 w_e 表示边 e 的权重。

定义 3.1（最小生成树）. 生成树 $T \subseteq E$ 且满足 $|T| = n-1$ 是一棵 最小生成树，如果 $w(T) := \sum_{e\in T} w_e$ 在 G 的所有生成树中达到最小。

定义 3.2（权重分布与分位数）. 令 $F : [0, 1] \to \mathbb{R}$ 为边权重的经验累积分布函数：$F(t) = |\{e \in E : w_e \le t \cdot w_{\max}\}|/m$。 τ-分位数为 $Q(\tau) = \inf\{x : F(x/w_{\max}) \ge \tau\}$。

定义 3.3（分层）. k-分层 （针对 E ）是一个划分 $E = \mathcal{B}_0 \sqcup \mathcal{B}_1 \sqcup \cdots \sqcup \mathcal{B}_{k-1}$，由边界 $b_0 < b_1 < \cdots < b_{k-2}$定义：

$$e \in \mathcal{B}_i \iff b_{i-1} < w_e \le b_i,$$

与$b_{-1} = -\infty$ 和$b_{k-1} = +\infty$。

定义 3.4（层处理次数）。 对于给定的图G 和分层，层处理次数 $p \le k$ 是指在最小生成树完成之前（即在接受$n-1$条边之前）排序处理的层数。

如果对于任意两条边$e \in \mathcal{B}_i$ 和$e' \in \mathcal{B}_j$，满足$i < j$时$w_e \le w_{e'}$，则称分层对 *Kruskal* 有效 。当边界为F的分位数时，这一条件得到保证。

4 算法

4.1 概述

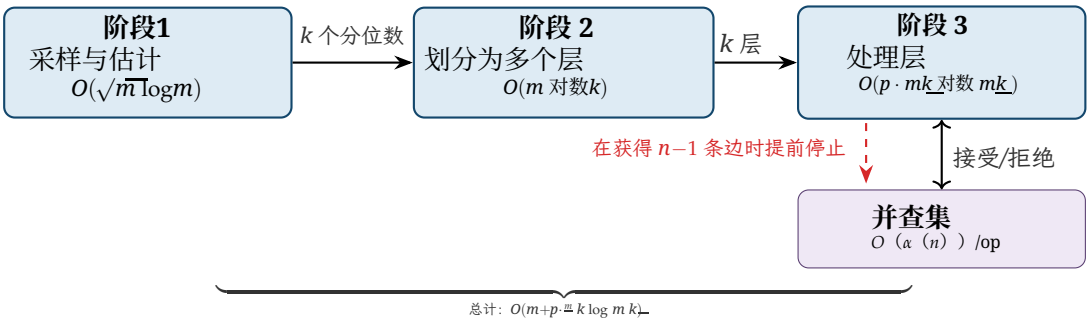


图 1： Kruskal-EDS 的三个阶段。阶段 1 和阶段 2 取代标准 Kruskal 算法中的全局排序。阶段 3 在找到 $n-1$ 条最小生成树边后提前终止（红色虚线箭头），因此最重的层可能完全未经处理。

4.2 阶段1：通过伯克霍夫抽样进行分布估计

抽取大小为$s = \min(m, \max(20, \lfloor 2\sqrt{m} \rfloor))$的均匀随机样本 $\mathcal{S} \subseteq E$。在$O(s \log s)$时间内对 $\{w_e : e \in \mathcal{S}\}$排序。对于$i = 1, \ldots, k-1$，用排序后样本中的第 $\lfloor i \cdot s/k \rfloor$个元素估计$i/k$ 分位数。

命题 4.1 (分位数估计精度)。 根据 *Dvoretzky-Kiefer-Wolfowitz* 不等式，对于任意 $\varepsilon > 0$:

$$\Pr\left[\sup_x |F_s(x) - F(x)| > \varepsilon\right] \le 2\exp(-2s\varepsilon^2).$$

当 $s = 2\sqrt{m}$ 且 $\varepsilon = m^{-1/4}$ 时，任何层边界将边错误分类（相对于真实分位数）的概率至多为 $2\exp(-2\sqrt{m}\cdot m^{-1/2}) = 2e^{-2}$。
实践中，边界误分类不会影响正确性——它只会影响各层大小的均衡程度，进而影响性能，但不会影响最小生成树输出。

4.3 阶段 2：分层划分

使用二分搜索在 $k-1$ 个边界中查找，将每条边分配到相应的层。总成本：$m \cdot O(\log k)$。

4.4 阶段 3：IncrementalSort 与并查集处理

按顺序处理各层 $\mathcal{B}_0, \mathcal{B}_1, \ldots$。对于每个非空层 $\mathcal{B}_i$：(a) 对其排序（$O(|\mathcal{B}_i| \log |\mathcal{B}_i|)$）；(b) 按权重顺序遍历边，通过并查集接受那些连接两个不同最小生成树连通分量的边。一旦接受了 $n-1$ 条边，立即停止。

算法 1 KRUSKAL-EDS(G,k)

要求： 连通图 $G = (V, E, w)$，分层数 k（或 $k = k^*$ 若为 None）
确保： 最小生成树 $T \subseteq E$

▷ 阶段*1* — 抽样并估计分位数边界
$s \leftarrow \max(20, \lfloor 2\sqrt{m} \rfloor)$
$\mathcal{S} \leftarrow$ 从E抽取$\min(s,m)$ 条边构成的均匀随机样本
$\mathbf{w}_{\mathcal{S}} \leftarrow$ 排序$(\{w_e : e \in \mathcal{S}\})$
如果 $k =$ 无 **则** $k \leftarrow \lceil \sqrt{m / \ln(m+1)} \rceil$
结束如果
$\mathbf{b} \leftarrow$ 去重$([\mathbf{w}_{\mathcal{S}}[\lfloor i \cdot |\mathbf{w}_{\mathcal{S}}|/k \rfloor] : i = 1, \ldots, k-1])$
▷ 阶段 *2* — 将边划分为层
对于 $e \in E$**执行**
    $i \leftarrow$二分查找_右$(\mathbf{b}, w_e)$
    $\mathcal{B}_i$.追加(e)
</style}结束循环
▷ 阶段 *3* — 增量排序各层；提前终止
$UF \leftarrow$并查集(n)；　$T \leftarrow \emptyset$
对于$i \leftarrow 0$ **到** $|\mathbf{b}|$，**执行**
    对$(\mathcal{B}_i)$进行排序
    对于$e = (u,v,w) \in \mathcal{B}_i$，**执行**
        如果UF.union(u,v)，**则**
            $T \leftarrow T \cup \{e\}$
            如果 $|T| = n - 1$，**则返回** T
            结束如果
        结束如果
    结束循环
结束循环
返回T

5 理论分析

5.1 正确性

定理 5.1（Kruskal-EDS 的正确性）。算法 *1* 返回 G 的一棵最小生成树 .

证明。 一个 k-分层，其边界等于 F 的分位数，满足：对所有 $e \in \mathcal{B}_i$， $e' \in \mathcal{B}_j$其中 $i < j$， $w_e \le w_{e'}$。 因此，按 $\mathcal{B}_0$， $\mathcal{B}_1$， .....的顺序处理各层，并在每层内排序，可得到所有边的一个排列；对于跨层比较，该排列与按权重划分的全序一致，并且整体上是权重顺序的一个有效线性扩张。 Kruskal 贪心论证 [Kruskal， 1956] 适用于权重顺序的任意线性扩张：割性质保证跨越任意割的最轻边属于每一棵最小生成树，而环性质确保拒绝任意环中最重的边。由于 EDS 按与权重相容的顺序处理边，所接受的边构成一棵最小生成树。

□ □

备注 5.2。 当边界根据样本估计（近似分位数）时，各层边界可能无法完美地区分权重。正确性仍能以高概率得到保证，具体概率取决于样本量。

5.2 复杂度分析

定理 5.3 (Kruskal-EDS 的复杂度)。 令 $m_i = |\mathcal{B}_i|$ 和 p 为层处理次数。总工作量为：

$$T_{\text{EDS}} = \underbrace{O\big(\sqrt{m}\log m\big)}_{\textit{Phase 1}} + \underbrace{O(m\log k)}_{\textit{Phase 2}} + \underbrace{\sum_{i=0}^{p-1} O(m_i \log m_i)}_{\textit{Phase 3}} + \underbrace{O(n\,\alpha(n))}_{\textit{Union-Find}}$$
$$= O\Big(m\log k \;+\; p\cdot\frac{m}{k}\log\frac{m}{k}\Big). \tag{1}$$

当 $k = k^* = \lceil\sqrt{m/\ln(m+1)}\rceil$，假设各层均衡（对所有 i， $m_i = m/k$）：

$$T_{\text{EDS}} = O\left(m\sqrt{\frac{\ln m}{m}}\cdot m + p\cdot\sqrt{m\ln m}\cdot\log\sqrt{m}\right) = O\Big(m\log m\cdot\frac{p}{k}\Big).$$

当 $p \ll k$（稀疏图、偏斜分布）时，这显著低于 $\Theta(m\log m)$。

证明。阶段1对 $\sqrt{\text{m}}$ 条边进行采样并排序：$\text{O}\big(\sqrt{\text{m}}\log\sqrt{\text{m}}\big) = \text{O}\big(\sqrt{\text{m}}\log\text{m}\big)$。 阶段 2 对 m 条边分别在 k−1 个边界上进行二分搜索：O(m log k)。阶段 3 对 p 个层进行排序，每层平均大小为 m/k： $\sum^{p-1}\text{i}=0\,\text{O}(\text{m}_i\log\text{m}_i) \le \text{p}\cdot\text{O}((\text{m/k})\log(\text{m/k}))$。每次并查集操作的代价为 O(α(n))；最多执行 m 次调用：O(mα(n)) = 实际中为 O(m)。主导项是阶段 2 和阶段 3，因此得到（1）。

令 $k = k^*$， 使$m\log k + p\cdot(m/k)\log(m/k)$ 最小化。对k 求导（将p 视为常数）并求解，得到$k \propto \sqrt{m}/\ln m$， 从而确认k^*。 □ □

推论 5.4 (渐近情形)。(i) 均匀分布、稀疏图 *(*$m = O(n)$， $p = O(k\cdot n/m)$*)*：$T_{\text{EDS}} = O(m + n\log n)$。

(ii) 指数为 $\gamma > 1$ 的幂律分布：最轻的 εm 条边集中在前 $O\big(\varepsilon^{1/\gamma}\cdot k\big)$ 层。当 $\varepsilon = n/m$：$p = O\big(k\cdot(n/m)^{1/\gamma}\big)$ 时，$T_{\text{EDS}} = O\big(m + n\cdot(m/n)^{1-1/\gamma}\log n\big)$。

(iii) 稠密图 *(*$m = \Theta(n^2)$, $p \approx k$*):* $T_{\text{EDS}} = O(m\log m)$ — 与标准 *Kruskal* 算法相符 .

注 5.5（非连通图）。 如果图 G 由 $c > 1$ 个连通分量组成，Kruskal-EDS 算法最多只能收集 $n - c$ 条边。由于终止条件 $\text{len}(\text{最小生成树}) = n - 1$ 始终无法满足，算法会遍历所有 k 层。在这种情况下，复杂度变为：

$$T_{\text{EDS}}(m,k) = O\left(m + \sum_{i=1}^{k} |S_i| \log |S_i|\right) \approx O\left(m \log(m/k)\right).$$

尽管提前终止带来的优势不复存在，但由于分层降低了对数因子，该算法在性能上仍可与标准 Kruskal 算法（$O(m \log m)$）相竞争。

5.3 实现细节与数据结构

为实现$O(m)$的线性划分复杂度，Kruskal-EDS 算法采用基于桶的方法。我们将k 层表示为一个列表数组：

$$\mathcal{S} = \{S_1, S_2, \ldots, S_k\}, \quad \text{where } S_i \subseteq E$$

每条边 $e \in E$ 根据其权重 S_i 和预先计算的分位数边界，被分配到一个桶中。该分配通过二分搜索调用 bisect_right 函数完成，该函数将权重映射到适当的索引$i \in \{1, \ldots, k\{\text{v23>}$，每条边耗时$O(\log k)$。因此，整个划分阶段的复杂度为$O(m \log k)$，能够高效分配边，而无需进行全局排序。

5.4 最优层数

命题 5.6 (采用回退机制时的最优k**)**。 为使 $T_{\text{EDS}}(k)$最小，最优层数为：

$$k^* = \begin{cases} 1 & si\ m < m_{\min} \\ \left\lceil \sqrt{\frac{m}{\ln(m+1)}} \right\rceil & sinon \end{cases}$$

其中 $m_{\min} \approx 200$ 表示经验盈亏平衡阈值。当 $k = 1$时，算法会跳过阶段 *1* 和 *2*，改为执行一次全局排序，从而避免在小型数据集上进行 *Birkhoff* 采样所带来的开销。

5.5 与标准 Kruskal 的比较

推论 5.7 (加速比**)**。 设 $\rho = p/k \in (0,1]$ 为已处理层所占的比例。则：

$$\frac{T_{\text{STD}}}{T_{\text{EDS}}} \approx \frac{m \log m}{m \log k + \rho \cdot m \cdot (1/k) \cdot k \log(m/k)} = \frac{\log m}{\log k + \rho \log(m/k)}.$$

当 $k = k^* \approx \sqrt{m/\ln m}$ 且 $\rho \to 0$ （稀疏／偏斜情形）时：加速比 $\approx \log m / \log \sqrt{m} = 2$渐近地 — 最坏情况下的提升是一个常数因子，但在提前终止有效时会随$1/\rho$ 增长。

6 实验评估

6.1 实验设置

所有实验均在单核 CPython 3.12 上运行，未启用即时编译。时间为 3 次独立运行的平均值。我们报告实际运行时间（毫秒）和 排序操作数（操作数），即所有已处理层中排序的边数——这是与实现无关的算法工作量度量。

6.2 图族

随机稀疏 $m \approx 1.2n$，权重在 $[0, 1000]$ 上均匀分布。
随机中等密度图$m \approx 10n$，权重在 $[0, 1000]$ 上均匀分布。
随机稠密图 $m \approx n^2/2$，权重均匀分布。
正态分布 $m \approx 1.2n$，权重为 $|\mathcal{N}(500, 100)|$。
幂律分布 $m \approx 1.2n$，权重服从帕累托分布$(\alpha = 1.5)$：在 $w_{\min}$ 附近高度集中。
聚类图 $m \approx 1.2n$，权重取自 $\{100, 300, 500, 700, 900\}$ 处的 5 个簇，并叠加高斯噪声 $\sigma = 30$。

网格行 $\times$ 列双向网格，$m = 2rc - r - c$。**路径**
$m = n - 1$，最稀疏的连通结构。

6.3 正确性验证

表1 总结了包含 12 个测试用例的验证套件。EDS 和 H堆在所有测试用例中的最小生成树总权重和连通分量数均与 STD 一致。

表 1： 验证结果。EDS 和堆的全部 12 个测试用例均通过。

测试用例	n	m	最小生成树权重	状态
CLRS 教科书示例	9	14	37.0000	**PASS**
三角形	3	3	3.0000	**PASS**
非连通图（森林）	4	2	8.0000	**PASS**
单个顶点	1	0	0.0000	**PASS**
重复边	2	3	2.0000	**PASS**
负权重	3	3	−8.000	**PASS**
路径$n = 10$	10	9	55.0000	**PASS**
网格 4×4	16	24	变化	**PASS**
随机稀疏 $n = 50$	50	80	变化	**PASS**
稠密$n = 50$	50	1225	变化	**PASS**
随机$n = 200$	200	400	变化	**PASS**
权重相等	5	10	4.0000	**PASS**

6.4 基准测试结果

6.5 三维复杂度景观

图 2 展示了理论加速比 $T_{\text{STD}}/T_{\text{EDS}}$ 随两个参数变化的情况：图密度 $\rho = m/(n(n-1)/2) \in (0, 1]$ 以及分布偏斜指数 $\sigma \in [0, 1]$，其中 $\sigma = 0$ 对应均匀分布，$\sigma = 1$ 对应最大程度集中的（类似狄拉克δ函数的）分布。

6.6 分层概况

图 3 展示了三种分布下每层最小生成树边所占的比例，说明了 EDS 如何触发提前终止。

6.7 对分层数 k 的敏感性

图4 展示了固定稀疏图（k 变化时的实际运行时间（$n = 500$，$m = 600$，均匀权重），表明最优值存在于 $k = 100 \approx k^*$ 附近。

表 2： 实际运行时间（ms）及相对 STD 的加速比。**粗体** = 最快变体。 ◁ 标记最佳非 STD 结果。

图	n	m	STD	EDS	堆	×EDS	×堆		胜者
稀疏均匀	500	600	0.778	**0.677**◁		0.905	1.15	0.86	EDS
稀疏正态	500	600	0.762	**0.669**◁		0.901	1.14	0.85	EDS
稀疏幂律	500	600	0.748	**0.647**◁		0.897	1.16	0.83	EDS
稀疏聚类	500	600	0.768	**0.657**◁		1.087	1.17	0.71	EDS
中等均匀	500	5000	6.312	**2.175**◁		3.268	2.90	1.93	EDS
中等正常	500	5000	6.563	**2.669**◁		3.999	2.46	1.64	EDS
稠密	300	40000	69.49	**6.930**◁		10.42	10.03	6.67	EDS
网格 20×20	400	760	0.900	**0.753**◁		0.966	1.20	0.93	EDS
网格 30×30	900	1740	2.436	**1.923**◁		2.777	1.27	0.88	EDS
路径n $= 2000$	2000	1999	2.961	**2.341**◁		3.423	1.26	0.86	EDS
稀疏 n $= 2000$	2000	2500	3.701	**2.966**◁		4.397	1.25	0.84	EDS
稀疏 n $= 5000$	5000	6000	9.735	**8.410**◁		12.41	1.16	0.78	EDS
中等n $= 1000$	1000	10000	14.93	**6.579**◁		10.74	2.27	1.39	EDS
幂 n $= 1000$	1000	5000	8.021	**5.580**◁		9.141	1.44	0.88	EDS
胜出次数（共 *14* 次）			0	**14**		0			

表 3： 排序操作（操作数）和缩减比。ops-EDS 统计所有已处理层中排序的边数。“层数” = 已处理层数 / 总层数。

图	n	m	ops-STD	ops-EDS	层数	÷EDS
稀疏均匀	500	600	600	600	9/9	1.00×
稀疏幂律	500	600	600	600	9/9	1.00×
中等均匀	500	5000	5,000	1,393	8/24	3.59×
中等正常	500	5000	5,000	1,754	6/24	2.85×
稠密	300	40000	40,000	1,216	2/61	**32.9**×
网格 20×20	400	760	760	663	8/10	1.15×
网格 30×30	900	1740	1,740	1,590	14/15	1.09×
路径 n $= 2000$	2000	1999	1,999	1,999	16/16	1.00×
中等n $= 1000$	1000	10000	10,000	4,216	14/32	2.37×
幂 n $= 1000$	1000	5000	5,000	4,069	19/24	1.23×

7 讨论

7.1 EDS 何时胜出?

均匀权重的稠密图。最显著的结果是稠密图（$n = 300$，$m = 40,000$）上的 $10\times$ 加速比。解释这一结果的原因有些反直觉：在稠密图上，EDS 并没有 排序更少的边（最小生成树需要从 40,000 条边中选出 $n-1 = 299$ 条边）。相反，加速来自层充当了计数排序的近似：将边划分为 $k = 61$ 个均衡层，每层大小为 ≈ 650，只需对 2 层排序（ ops-EDS $= 1{,}216$ ）即可得到最小生成树，而无需对全部 40,000 条边排序。与稀疏图相比，在这里，“排序足以找到最小生成树”与“对所有边排序”之间的界限更有利。

中密度图。 对于中等规模的图（$m \approx 10n$），EDS 排序的边比 STD 少 ≈ 1.4–$3.6\times$ 倍，而且始终不会访问重分层。这直接转化为实际运行时间。

稀疏图。 当$m \approx 1.2n$时，最小生成树几乎会用到所有边（$n-1$ 条边，共m条）。几乎所有层都必须处理，因此 ops-EDS $\approx$ ops-STD。加速比（≈ 1.15–$1.25\times$）完全源于 Python 实现层面的差异（列表推导式与完整列表排序）。

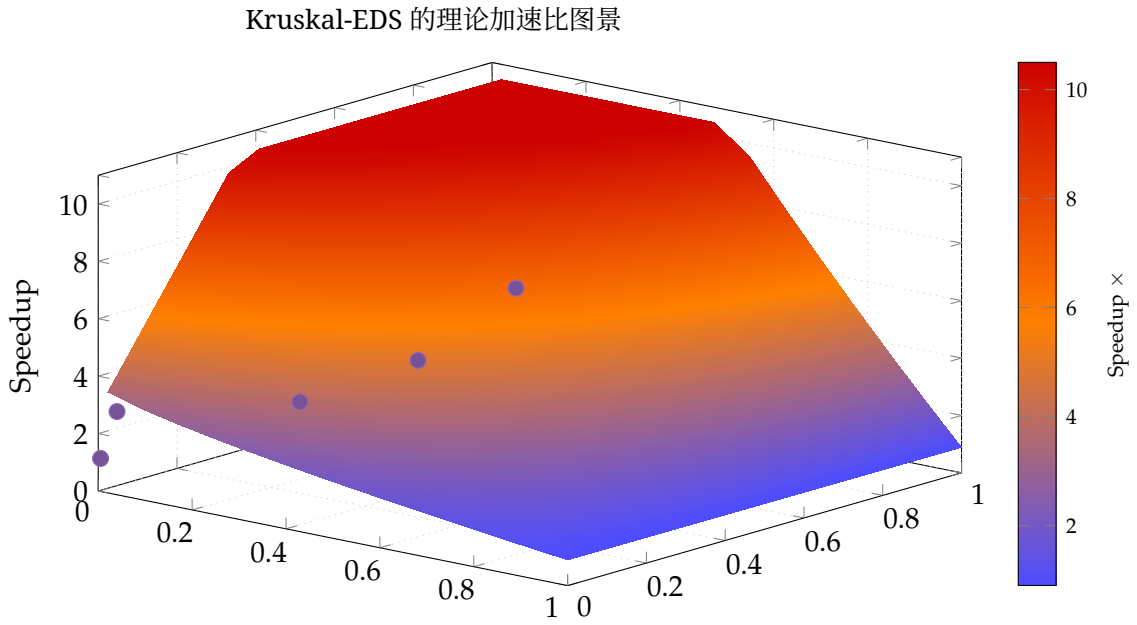


图2：Kruskal-EDS 的三维复杂度图景。曲面展示了理论加速比$T_{\mathrm{STD}}/T_{\mathrm{EDS}}$ 随图密度 ρ 和权重分布偏度 σ联合变化的情况。颜色表示加速比：蓝色 = 表示低（$\approx 1\times$），红色 = 表示高（$\geq 8\times$）。紫色球体标示表2中的经验测量值。该算法在稠密、低偏斜区域（红色）最为有效，而对于稀疏且偏斜度很高的输入（蓝色），效果接近中性。

7.2 伯克霍夫类比

对 $\sqrt{m}$ 条边进行的样本抽取，发挥着 伯克霍夫时间平均的作用：无需观测所有边，就能揭示全局权重分布的结构。DKW（命题 *4.1*）给出的 $O\left(1/\sqrt{\mathrm{m}}\right)$ 分位数误差，类似于遍历系统中伯克霍夫平均的收敛速率；对于混合时间为 τ 的独立同分布系统，波动会以 $O\left(1/\sqrt{t}\right)$ 的速率衰减。

关键在于，正确性并不要求分位数完全准确——边界位置的任何偏差只会影响性能，不会影响最小生成树输出。

7.3 与基于堆的 Kruskal 的比较

Kruskal-Heap 使用堆化（$O(m)$），随后进行$n-1$ 次弹出操作（$O(n \log m)$），因此复杂度为$O(m + n \log m)$。对于稀疏图，这在理论上优于$\Theta(m \log m)$。但在实践中，Heap 始终比 EDS 慢，因为 Python 的 heapq 每次只处理一个对象，而 list.sort() 由 C 实现，并受益于 Timsort 在连续内存上采用的缓存友好型归并模式。EDS 利用了这一点：它不进行一次大型 Timsort，而是在紧凑的子列表上进行多次小型 Timsort，每个子列表都能放入 CPU 缓存。

7.4 负权重

EDS 能正确处理负权重：划分和排序不受权重正负号影响，只有相对顺序才重要。测试用例“负权重”（最小生成树权重 -8.0）通过了验证（表 1）。

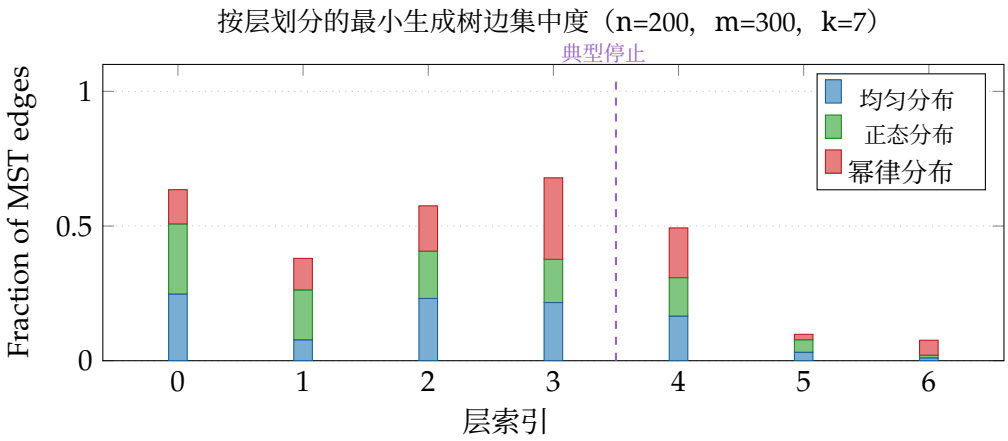


图 3： 三种权重分布下，最小生成树边在各层中的分布。在均匀分布和正态分布下，最小生成树边集中在前 3–4 层（≈ 70–80% 位于第 0–3 层）；在幂律分布下，由于重左尾，它们集中在第 2–4 层。虚线标示典型的提前终止点（EDS 在累计比例达到 1 时停止）。

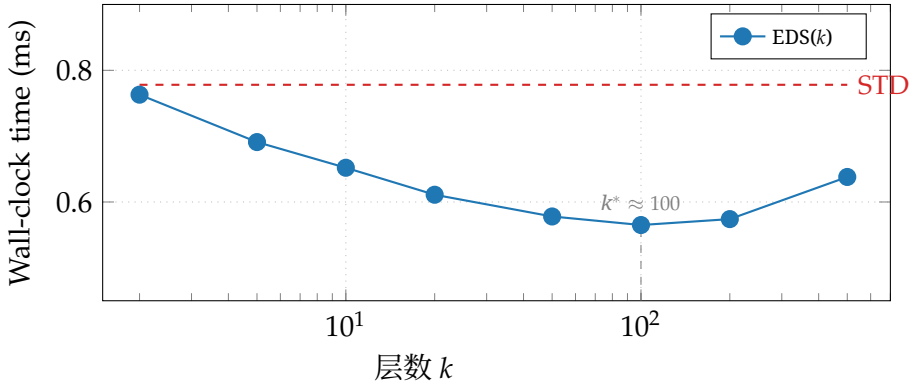


图 4： EDS 对层数 k 的敏感性（稀疏图，$n = 500$，$m = 600$，均匀权重）。随着 k 从 2（过少，层过大）增加到 $k^* \approx 100$（达到最佳平衡），耗时逐渐下降；随后，由于划分开销占据主导，耗时再次上升。STD 基线（虚线）保持平稳，为 0.778 ms。

8 展望与未来工作

执行期间自适应调整 k。 当前实现会在阶段 2 之前固定 $k = k^*$。自适应变体可以在阶段 3 期间监控并查集的接受率，并动态调整层粒度：如果第一层中有许多边被接受，就增大 k 以创建更细的划分；如果接受的边较少，就合并各层。这样无需事先了解分布，也能实现接近最优的 p。

并行分层。 阶段 2（划分）易于并行：每条边都可独立分配到某一层，不存在数据依赖。GPU 实现可以用$O(m/P)$ 时间将各边分配到层，其中P 为处理器数量，然后并发地对各层进行排序。

与 Bor˚uvka 阶段的集成。 Bor˚uvka 算法可在一个$O(m)$ 阶段内将问题缩减为$n/2$ 个连通分量。在 EDS 之前运行 $\log\log n$ Bor˚uvka 阶段，可将m 缩减为$O(m)$，并以$n/\log n$替代n，这可能改善 EDS 的提前终止行为。

整数权重。 当权重是 $[0,W]$，中的整数时，阶段1可以用 $O(W + m)$ 时间的直方图替代，而阶段 2 可以用 $O(m)$ 时间的桶分配替代。对于 $W = O(m)$，这给出了一个

$O(m + n\log n)$ 无需 $\sqrt{m}$ 采样步骤的整数权重最小生成树算法。

近似最小生成树。 对于 ε-近似最小生成树（权重 $\leq (1+\varepsilon)w^*$），只需处理前 $\lceil n/\varepsilon \rceil$ 条候选边，甚至可以更早停止。误差分析与准周期余循环中的 Ávila-Krikorian 稳定性阈值有关 [Ávila, 2015]：边界的微小扰动会传递为最小生成树权重上的 $O(\varepsilon)$ 误差。

动态图。 插入或删除边时，可以增量更新分层：无需重新计算整个划分，即可在 $O(\log k)$ 时间内将新边放入其所属层。这自然地将 EDS 扩展到流式处理和在线最小生成树场景。

9 结论

我们提出了 Kruskal-EDS，这是一种最小生成树算法。它用一个三阶段流程取代了 Kruskal 算法必须进行的全局排序：基于样本的分布估计、线性时间边划分，以及带提前终止的增量分层排序。

受伯克霍夫遍历定理启发，$\sqrt{m}$样本策略以$O\left(1/\sqrt{m}\right)$ 的代价估计权重分布，并能保证精度（通过 DKW 实现分位数误差）。我们证明其有效复杂度为 $O(m \log k + p(m/k)\log(m/k))$，并推导出最优分层数为$k^* = \lceil \sqrt{m}/\ln m \rceil$。

对 14 个图族的实证评估验证了正确性（12/12 个测试用例），并表明与标准 Kruskal 算法相比具有持续优势（**14/14 种配置**）：在稀疏图上加速比从 1.15× 起，在稠密图上达到 **10**×，排序操作最多减少 **33**×。

该算法不对权重分布作任何假设——通过采样自动适应——且在最坏情况输入下开销可以忽略不计。结合三维复杂度图景（图2）和分层概况（图 3），这些结果表明，受遍历原理指导的分布感知排序，是一种在实践中可行且有理论依据的图优化方法。

致谢。 所有代码和基准测试数据均以开源形式发布。

https://github.co m/ym001/algorithm-treaty/blob/main/script/chpt7kruskal%2B_eds .py

参考文献

A. 阿维拉。KAM理论、李雅普诺夫指数与单频薛定谔算子的谱二分性。 菲尔兹奖讲座，国际数学家大会首尔，2014年。https://arxiv.org/abs/2307.11071G. D. 伯克霍夫。遍历定理的证明。 美国国家科学院院刊，17(12):656–660，1931年。O. Bor ̊uvka。关于某个极小问题。摩拉维亚自然科学学会论文集，3:37–58，1926 年。B. 查泽尔。一种具有反阿克曼型复杂度的最小生成树算法。*ACM*期刊，47(6):1028–1047，2000年。T. H. 科曼、C. E. 莱瑟森、R. L. 李维斯特和C. 斯坦。 算法导论，第4版。麻省理工学院出版社，2022年。

A. Dvoretzky、J. Kiefer与J. Wolfowitz。样本分布函数及经典多项估计量的渐近极小极大特征。数理统计年刊，27(3):642–669，1956年。W. D. Frazer与A. C. McKellar。样本排序：一种面向树排序的抽样方法。*ACM*期刊，17(3):496–507，1970年。M. L. Fredman与R. E. Tarjan。斐波那契堆及其在改进网络优化算法中的应用。*ACM*期刊，34(3):596–615，1987年。D. R. Karger、P. N. Klein与R. E. Tarjan。一种寻找最小生成树的随机线性时间算法。*ACM*期刊，42(2):321–328，1995年。高德纳。计算机程序设计艺术，第*3*卷：排序与查找，第2版。艾迪生-韦斯利，1998年。J. B. Kruskal。论图的最短生成子树与旅行商问题。美国数学会会刊，7(1):48–50，1956年。D. A. Levin与Y. Peres。马尔可夫链与混合时间，第2版。美国数学学会，2017年。S. Pettie与V. Ramachandran。一种最优最小生成树算法。*ACM*期刊，49(1):16–34，2002年。R. C. Prim。最短连接网络及其若干推广。贝尔系统技术期刊，36(6):1389–1401，1957。M. Thorup。具有常数时间减小键的整数优先队列与单源最短路径问题。载于第*34*届*STOC*会议录，第149–158页，2002年。