arXiv:2603.03273v1 [cs.DS] 3 Mar 2026

超图边着色聚类的改进组合算法

Seongjune Han[1], Nate Veldt[2]

[1]德克萨斯农工大学数学系，美国德克萨斯州[2]德克萨斯农工大学计算机科学与工程系，美国德克萨斯州

摘要

许多复杂系统和数据集具有不同类别之间的多元交互特征，可建模为边着色超图。我们聚焦于使用 NP难的边着色聚类问题对此类数据集进行聚类，其目标是为节点分配颜色，使节点颜色尽可能与边颜色相匹配。以往研究的一个重点是为该问题开发具有组合性且易于扩展的近似算法。本文提出了首个近似因子优于 2 的组合近似算法。

1 引言

图是表示数据集或复杂现实世界系统中实体两两关系的强大模型。然而，许多系统和数据集涉及多个实体之间同时发生的多方交互（例如群组邮件、共同撰写的在线文章，以及电商平台上共同评测的零售商品集合），而图无法充分刻画这类交互。此外，这些多方交互通常与某种类型或类别相关联（例如邮件主题、文章发表的场所、商品评测者的 IP 地址）。分析这类数据集的一项基本任务是进行类别聚类，旨在识别倾向于参与某一特定类别多方交互的对象群体。该问题的一个形式化目标是超图中的最小边着色聚类（MinECC） [1]。对于此问题，输入由边染色超图表示，其中超边表示多方交互，颜色表示交互类型。目标是为节点着色，使未满足的边的权重最小化。未满足的边是指至少包含一个未被分配为该边颜色的节点的边。尽管该问题是 NP难的，但已经提出了许多近似算法。研究者已大量关注可实现为纯组合算法（即不依赖问题的全局凸松弛）的快速算法 [18,14, 1, 9]。针对 MinECC的更快算法可以作为基本原语，服务于该问题所激发并应用于的各种任务，包括社区发现 [1],团队组建 [2, 10], 和资源分配 [3]。

相关工作。 关于边着色聚类的早期研究主要集中于图，并包括最大化满足边数量的近似算法研究 [3], 以及固定参数可处理性结果 [6]。Amburg 等人 [1]将研究扩展到了超图。他们通过对线性规划（LP）松弛进行舍入，提出了一种近似因子为 $\min\left(2-\frac{1}{k}, 2-\frac{1}{r+1}\right)$ 的近似算法，其中 k 是颜色数量， r 是最大

超边大小。此外，他们还引入了近似因子随其线性增长的组合算法 r。Amburg 等人 [2]随后将这一工作拓展到多样化群体发现和团队组建等问题，扩大了 MinECC应用范围。Veldt [18] 提出了一种基于线性规划舍入的 $\min\left(2-\frac{2}{k}, 2-\frac{2}{r+1}\right)$-近似算法，以及组合式 2-近似算法，其运行时间与超图规模呈线性关系。此后不久，Crane 等人 [9] 将 ECC推广到允许预算数量的重叠聚类分配或节点删除，并给出了包括贪心组合式 r-近似算法在内的其他结果。随后，Lee 等人 [14] 使用原始-对偶方法，为重叠目标开发了组合方法，其解质量优于贪心算法，且运行速度快于线性规划舍入方法。尽管此前的大量工作都集中于快速的组合算法，但目前尚无近似比优于 2 的组合算法。此外，在所有线性时间算法中，对于无权情形存在确定性 2-近似算法，对于加权情形存在随机化 2-近似算法，但尚无针对加权情形的确定性 2-近似算法 [18]。

我们的贡献。 我们的主要贡献是为 MinECC提出一种 $(2-2/k)$-近似算法，该算法可以完全采用组合方法实现，其理论运行时间相对于超图规模近乎线性。当 $r \geq k-1$ 时，这与 Veldt [18] 提出的最佳理论近似算法相当，而且无论在理论上还是实践中，运行速度都更快。这项贡献的关键在于，我们展示了如何通过在辅助图中求解一个最小 s-t割问题，快速且以组合方式求解一种特定类型的 MinECC 线性规划松弛。作为次要贡献，我们开发了一种确定性的 2-近似算法，其运行时间为线性，并适用于加权超图。为此，我们结合了其他线性时间 MinECC 算法 [18]的核心思想，以及一种针对加权最小顶点覆盖 [4]的快速 2-近似算法。

2 预备知识

MinECC 的一个实例由边染色超图 $H=(V,E,C,\ell)$ 给出，其中 V 是节点集，E 是边集，$C=[k]=\{1,2,3,\cdots,k\}$ 是边颜色集合，$\ell\colon E\to C$ 是一个将边映射到 C 中颜色的映射。令 r 表示最大超边大小，$\mu=\sum_{e\in E}|e|$ 表示超图的大小。令 E_i 表示颜色为 i 的边集。令 $E(u)$ 表示与节点 $u\in V$ 相邻的边集。对于加权实例，$w(e)\geq 0$ 表示 $e\in E$ 的非负权重。定义 $w(A)=\sum_{e\in A}w(e)$。在 MinECC 中，目标是为节点着色，使其颜色尽可能与包含该节点的边的颜色相匹配。具体来说，如果超边 $e\in E$ 中的所有节点都被赋予颜色 $\ell(e)$，则该超边为满足；否则为未满足。 MinECC旨在为节点着色，使未满足边的总权重最小化。当存在节点 v 属于边 e、$f\in E$ 的交集，且 $\ell(e)\neq\ell(f)$ 时，我们无法为 v 着色以同时满足这两条边。我们将这样的边对 (e,f) 称为坏边对，并将所有坏边对组成的集合记为 $\mathcal{B}$。令 $C(u)=\{j\in C\colon \exists e\ni u \text{ with } j=\ell(e)\}$ 表示与节点 u 相邻的颜色列表。 MinECC 的规范线性规划松弛（$\mathrm{LP_{ECC}}$）定义如下：

$$
\begin{array}{lllll}
\min & \sum_{e\in E} w(e) z_e & & & \\
\text{s.t.} & \forall e\in E & z_e \geq x_u^{\ell(e)} & \forall u\in e & \\
& \forall u\in V & \sum_{i\in C} x_u^i = k-1 & & \\
& \forall u\in V & 0\leq x_u^i\leq 1 & \forall i\in C & \\
& \forall e\in E & 0\leq z_e\leq 1. & &
\end{array}
\qquad (\mathrm{LP_{ECC}})
$$

当变量 x_u^i, z_e 为二元变量时，该松弛即成为一个能够精确刻画 MinECC 目标的二元线性规划。节点 u 和颜色 i 对应的变量 x_u^i 可以解释为距离。若变量为二元变量，则 $x_u^i = 0$ 表示 u 被赋予颜色 i，而变量 z_e 则表示边 e 是否满足（$z_e = 0$）或未满足（$z_e = 1$）。我们将 LP_{ECC} 的最优解值记为 $\text{OPT}(\text{LP}_{\text{ECC}})$，其他线性规划也采用类似记法。

顶点覆盖归约。 Veldt [18] 表明 MinECC 可以近似保持地归约为 Vertex Cover，以此为基础开发快速的 2-近似算法。要理解这一归约，可以先注意到，MinECC 也可以等价地视为一项边删除任务，其目标是删除一个权重最小的边集（在无权情形下则是边数最少的边集），使得剩余超图中的每个节点所关联的边至多具有一种颜色。这又等价于删除一个权重最小的边集，使剩余图中不再存在坏边对；随后，我们便可以根据剩余边为节点着色。这类似于 Vertex Cover，即找出一个权重最小的节点集以覆盖所有边的任务。具体而言，将MinECC 归约为 Vertex Cover时，超图中的每条边 $e \in E$ 都对应于一个 v_e 中的节点Vertex Cover 实例，而两个节点（v_e、v_f）在（e、f）$\in \mathcal{B}$时构成一条边（见图 1）。这一归约导出了一种针对$_{\text{VC}}$ 的替代型线性规划松弛 LP MinECC:

$$\begin{array}{lll} \min & \sum_{e\in E} w(e)x_e & \\ \text{s.t.} & x_e + x_f \geq 1 & \forall (e,f) \in \mathcal{B} \\ & x_e \geq 0 & \forall e \in E. \end{array} \qquad (\text{LP}_{\text{VC}})$$

该线性规划相当于将 MinECC 缩减为 Vertex Cover ，然后对 Vertex Cover采用标准的线性规划松弛。约束 $x_e + x_f \geq 1$ 体现了这样一个事实：每个坏边对中至少有一条边必须被删除。关于 Vertex Cover 的已有研究中有一条标准结论：该线性规划总是存在一个半整数最优解，即 $x_e \in \{0, 1, 1/2\}$ 对每个$e \in E$ [15]均成立。

3 确定性 2-近似算法

Veldt [18] 提出了一种随机化 2-近似算法，运行时间为 $O(\mu)$，且也适用于加权超图。该算法通过隐式应用 Pitt Vertex Cover提出的 [16]随机化 2-近似算法 MinECC，而不构造缩减后的图。为了获得一种适用于加权 MinECC超图且具有线性时间复杂度的确定性 2-近似算法，我们采用类似的方法，但改为隐式实现一种已有的 Vertex Cover确定性算法。具体而言，算法1 是 Bar-Yehuda 和 Even 针对节点加权 Vertex Cover图提出的确定性 2-近似算法 [4, 5]。该算法采用贪心策略，遍历各条边，并将实例中两个相邻节点的权重分别降低其当前权重的较小值。遍历所有边后，算法选取所有权重变为零的节点。该算法的运行时间与 Vertex Cover实例中的边数呈线性关系。然而，如果我们显式地将 Vertex Cover缩减为一个MinECC 实例，再应用算法 Vertex Cover 1，则运行时间将随超图规模呈超线性增长。

在算法 2 中，我们给出了算法 LocalRatioECC 的伪代码。该算法在不构造约化的 Vertex Cover实例的情况下，隐式地将算法 1 应用于 MinECC 实例 $H = (V, E, C, \ell)$。对于此算法，我们假设边 E 已按颜色升序排列。对于每个节点 $v \in V$，我们考察其关联边 $E(v)$，其中 $v(1)$ $v(2) \ldots, v(d_v)$ 表示这些边的索引，d_v 表示 v的度。算法 1 遍历 Ver-tex Cover 实例中的边，这相当于遍历 MinECC中的坏边对

算法 1 LocalRatioVC

输入： 一个图 $G = (V, E)$
输出： 一个顶点覆盖 C
$C \leftarrow \emptyset$
对 $v \in V$ **执行**
$r(v) = w(v)$
结束循环
对于 $(u, v) \in E$ **执行**
$M \leftarrow$ 最小值$\{r(u), r(v)\}$
$r(u) \leftarrow r(u) - M$
$r(v) \leftarrow r(v) - M$
结束循环
$C \leftarrow \{v \in V \mid r(v) = 0\}$

实例。为了遍历所有坏边对，我们访问每个节点 $v \in V$并考虑属于 $E(v$的边对 f 和 b，它们分别初始化为 1 和 d_v。每一步中， f 会递增，或 b 会递减。当边对 $(e_{v(f)}, e_{v(b)})$ 构成坏边对时，我们按照算法 1中的局部比率更新，将两条边的权重均减去它们当前权重中的较小值。因此，两条边中至少有一条的权重会变为零。由于算法最终会收集所有权重变为零的边，而这些边恰好就是待删除的边，因此我们可以立即删除权重变为零的边。此方法通过跳过涉及已删除边的边对，避免显式遍历所有坏边对，从而保证总运行时间为 $O(\mu$。算法 2 采用与 Veldt [18]相同的坏边对遍历策略。不同之处在于，访问坏边对 $(e_{v(f)}, e_{v(b)})$ 时，该算法通过隐式实现 LocalRatioVC，以确定性方式删除至少一条超边，而不是像 Pitt 算法那样随机删除一条超边。我们用以下定理概述这一结果。

定理 3.1。 算法 *2* 是针对 MinECC 的 *2*-近似算法。其运行时间为 $O(\sum_{v \in V} d_v) = O(\mu)$ 。

证明。 由于算法 2 是 LocalRatioVC 的隐式实现，且从MinECC 到 Vertex Cover 的归约保持近似性，因此算法 2 是一个 2-近似算法。当它访问共享节点 v的坏边对时，需要执行 $O(d_v)$ 的工作量，因为每一步都会删除至少一条包含 v 的超边。因此，其总体运行时间为 $O(\sum_{v \in V} d_v) = O(\mu)$。 □

算法 2 LocalRatioECC

输入： 加权实例 $H=(V,\ E,\ C,\ \ell)$
输出： $D \subseteq E$ 条待删除边
$D \leftarrow \emptyset$
对 $v \in V$ **执行**
$E(v)=[e_{v(1)}\ \ e_{v(2)}\ \ \cdots\ \ e_{v(d_v)}]$
$f=1, b=d_v$
当 $e_{v(b)} \in D$ 且 $b>f$ **时**
$b \leftarrow b-1$
结束循环
当 $e_{v(f)} \in D$ 且 $b>f$ **时**
$f \leftarrow f+1$
结束循环
当 $\ell(e_{v(f)}) \neq \ell(e_{v(b)})$ **时**
如果 $w(e_{v(f)}) < w(e_{v(b)})$ **则**
$D \leftarrow D \cup \{e_{v(f)}\}$
$w(e_{v(b)}) \leftarrow w(e_{v(b)}) - w(e_{v(f)})$
当 $e_{v(f)} \in D$ 且 $b>f$ **时**
$f \leftarrow f+1$
结束循环
否则，如果 $w(e_{v(f)}) = w(e_{v(b)})$ **则**
$D \leftarrow D \cup \{e_{v(f)}, e_{v(b)}\}$
当 $e_{v(f)} \in D$ 且 $b>f$ **时**
$f \leftarrow f+1$
结束 while
当 $e_{v(b)} \in D$ 且 $b>f$ **执行**
$b \leftarrow b-1$
结束 while
else
$D \leftarrow D \cup \{e_{v(b)}\}$
$w(e_{v(f)}) \leftarrow w(e_{v(f)}) - w(e_{v(b)})$
当 $e_{v(b)} \in D$ 且 $b>f$ **执行**
$b \leftarrow b-1$
结束 while
结束 if
结束 while
结束 for
返回 D

4 更快的（$2-2/k$）-近似

本节详细介绍我们针对 $2-2/k$提出的新型（ MinECC ）近似算法，并且该算法可以完全组合化。我们将通过将问题归约为一个 k-可着色 Vertex Cover 实例，再应用该问题的近似算法来获得近似保证 [11]。这相当于对 $\mathrm{LP}_{\mathrm{VC}}$应用舍入算法。更重要的贡献是一种全新且更高效的方法，可以组合化地求解 $\mathrm{LP}_{\mathrm{VC}}$ ，而无需实际构造Vertex Cover 实例或显式建立 LP。

算法 3 近似线性规划舍入算法

输入： LP 的最优半整数解 $X = \{x_e \mid e \in E\}$ $_{\text{VC}}$

输出： 待删除边的集合 $Y \subseteq E$

$E^1 = \{e \in E\colon\ x_e = 1\}$

$E^{1/2} = \{e \in E\colon\ x_e = 1/2\}$

$E^0 = \{e \in E\colon\ x_e = 0\}$

$i^* \leftarrow \text{argmax}_{i \in C}(w(E_i \cap E^{1/2}))$

返回 $Y\colon = E^1 \cup (E^{1/2} - E_{i^*} \cap E^{1/2})$

4.1 线性规划舍入算法

从 MinECC 到 Vertex Cover 的归约会自然生成一个可 k-着色的图：它将超边颜色与归约后 Vertex Cover 实例中节点的颜色相关联。这是因为，根据设计，颜色相同的超边无法构成坏边对。这一观察很有用，因为 Hochbaum [11] 先前展示了如何对 Vertex Cover 的线性规划松弛进行舍入，从而为 $2-2/k$-可着色图得到一个 (k)-近似解。归约时，我们无需显式构造一个 Vertex Cover实例；从 MinECC进行归约时，可以直接采用 Hochbaum 的方法（即算法 3），将线性规划 $_{\text{VC}}$的半整数解舍入。

定理 4.1。算法 *3* 是一个 $(2-\frac{2}{k})$-近似算法，适用于 MinECC。

证明。给定超图 $H = (V,\ E,\ C,\ \ell)$ 以及 LP_{VC} 的最优半整数解 $\{x_e^* \mid e \in E\}$，首先我们来证明 $\text{OPT}(\text{LP}_{\text{VC}})\colon = \sum_{e \in E} w(e)x_e^* = w(E^1) + \frac{1}{2}w(E^{1/2})$。

$$\begin{aligned}\sum_{e \in E} w(e)x_e^* &= \sum_{e \in E^1} w(e)x_e^* + \sum_{e \in E^{1/2}} w(e)x_e^* + \sum_{e \in E^0} w(e)x_e^* \\ &= \sum_{e \in E^1} w(e) \cdot 1 + \sum_{e \in E^{1/2}} w(e) \cdot \frac{1}{2} + \sum_{e \in E^0} w(e) \cdot 0 \\ &= w(E^1) + \frac{1}{2}w(E^{1/2}).\end{aligned}$$

其次，根据 i^* 的定义，我们知道

$$w(E_{i^*} \cap E^{1/2}) \geq \frac{1}{k}w(E^{1/2}).$$

接下来请注意，如果 (e, f) 是坏边对，其中 $e \in E^0$，那么为了满足$x_e + x_f \geq 1$，f必须属于 E^1。因此，$E^0 \cup (E_{i^*} \cap E^{1/2})$ 是一组可满足边。现在，考虑其补集 $E^1 \cup (E^{1/2} - E_{i^*} \cap E^{1/2})$。那么，其权重（假设 $k \geq 2$）为

$$\begin{aligned}w(E^1 \cup (E^{1/2} - E_{i^*} \cap E^{1/2})) &= w(E^1) + w(E^{1/2}) - w(E_{i^*} \cap E^{1/2}) \\ &\leq w(E^1) + w(E^{1/2}) - \frac{1}{k}w(E^{1/2}) \\ &\leq \left(2 - \frac{2}{k}\right)\left(w(E^1) + \frac{1}{2}w(E^{1/2})\right) \\ &= \left(2 - \frac{2}{k}\right)\text{OPT}(\text{LP}_{\text{VC}}) \\ &\leq \left(2 - \frac{2}{k}\right)\text{OPT}(\text{MinECC}),\end{aligned}$$

其中 OPT (MinECC) 是 MinECC 中 H的最优解的权重总和。 □

Hochbaum [11] 证明， Vertex Cover 线性规划松弛可以通过将其归约为辅助图中的最小 s-t 割来求解，其中辅助图中的边数约为 Vertex Cover 实例中边数的两倍。将这种方法应用于一个 MinECC 实例 $H=(V,E,C,\ell)$，会得到一个辅助图中含有 $2|E|+2$ 个节点和 $2|E|+2|\mathcal{B}|=O(|E|^2)$ 条边的 s-t 割问题。理论上，可以使用 van den Brand等人 [17]的算法，在 $\tilde{O}(|E|^2)$ 时间内求解由此得到的最小 s-t 割问题。这已经快于通过舍入 $\mathrm{LP}_{\mathrm{ECC}}$ [18]得到的最佳近似；当 $r \geq k-1$时，该方法可达到相同的 $(2-2/k)$-近似。对于具有 N个变量、形式为 $\min_{Ax=b;x\geq 0} c^T x$ 的线性规划，目前已知的最佳理论求解运行时间为 $\Omega(N^\omega)$ 时间 [13, 8]。将 $\mathrm{LP}_{\mathrm{ECC}}$ 转换为这种形式后，变量数为 $N=|E|+k|V|+\mu$。由于 $\omega \geq 2$，运行时间因此为 $\Omega((k|V|+\mu)^2)$。

关键挑战。尽管理论上更快，但显式地化简为 Vertex Cover 并应用 Hochbaum 的结果 [11] 并不实用，因为化简后的 Vertex Cover 实例会变得非常稠密。对于许多真实的 MinECC 数据集（见表 1）， $|\mathcal{B}|$ 会非常庞大。即使显式地求出$\mathcal{B}$ 也代价高昂；即便 r 和k 均为常数，其所需时间相对于超图规模仍然是超线性的。与此同时，尽管 $\mathrm{LP}_{\mathrm{ECC}}$ 的理论运行时间很差，但在实践中，它的约束往往远少于 $\mathrm{LP}_{\mathrm{VC}}$ （见表 1），并且已有研究表明，对于许多真实场景中的实例，它在实践中很容易求解 [1]。我们的主要贡献是一种求解 $\mathrm{LP}_{\mathrm{VC}}$的新方法，它完全避免处理坏边对，因此无论在理论上还是实践中都能获得更好的运行时间。

4.2 颜色对二元线性规划

为避免上述挑战，我们提出一个与 $\mathrm{LP}_{\mathrm{VC}}$ 等价的二元线性规划，但它的约束少得多，并且可以化简为一个稀疏得多的最小 s-t 割问题。这个二元线性规划的构想融合了 $\mathrm{LP}_{\mathrm{ECC}}$ 和 $\mathrm{LP}_{\mathrm{VC}}$中的概念。我们首先定义一个新的线性规划——颜色对线性规划（$\mathrm{LP}_{\mathrm{CP}}$）——它合并了 $\mathrm{LP}_{\mathrm{ECC}}$和 $\mathrm{LP}_{\mathrm{VC}}$中的约束：

$$
\begin{aligned}
\min \quad & \sum_{e\in E} w(e) z_e \\
\text{s.t.} \quad & \forall e \in E, \quad z_e \geq x_u^{\ell(e)} \qquad \forall u \in e \\
& \forall u \in V, \quad x_u^i + x_u^j \geq 1 \quad \forall (i,j) \in \tbinom{C(u)}{2} \\
& \forall u \in V, \quad 0 \leq x_u^i \leq 1 \quad \forall i \in C(u) \\
& \forall e \in E, \quad 0 \leq z_e \leq 1.
\end{aligned}
\tag{2}
$$

在此松弛中，约束（$x_u^i + x_u^j \geq 1$）体现了节点 u 不能同时具有颜色 i 和颜色 j这一事实。此约束类似于 $\mathrm{LP}_{\mathrm{VC}}$ 中的坏边对约束 $x_e + x_f \geq 1$，但我们不需要 $\Omega(|\mathcal{B}|)$ 个此类约束。此约束放宽了 $\mathrm{LP}_{\mathrm{ECC}}$ 中的约束 $\sum_{c\in C} x_u^c = k-1$。因此，得到的松弛更宽松。可以证明 $\mathrm{LP}_{\mathrm{CP}}$ 等价于

以下颜色对二元线性规划（$\mathrm{BLP_{CP}}$）：

$$
\begin{aligned}
\min \quad & \frac{1}{2}\sum_{e\in E} w(e)(b_e - a_e + 1) \\
\text{s.t.} \quad & \forall e \in E: \quad a_e \le a_u^{\ell(e)} && \forall u \in e \\
& \forall e \in E: \quad b_u^{\ell(e)} \le b_e && \forall u \in e \\
& \forall u \in V: \quad a_u^i \le b_u^j && \forall (i,j) \in \tbinom{C(u)}{2} \\
& \forall u \in V: \quad a_u^j \le b_u^i && \forall (i,j) \in \tbinom{C(u)}{2} \\
& \forall u \in V: \quad a_u^i, b_u^i \in \{0,1\} && \forall i \in C(u) \\
& \forall e \in E: \quad a_e, b_e \in \{0,1\}.
\end{aligned}
\tag{$\mathrm{BLP_{CP}}$}
$$

我们证明，该 BLP 等价于 $\mathrm{LP_{VC}}$。鉴于其中并未明确提及任何坏边对，这一结论或许令人意外。需要指出的是，要看出 $\mathrm{BLP_{CP}}$ 等价于 $\mathrm{LP_{VC}}$，并非一定要考虑或证明关于 $\mathrm{LP_{CP}}$ 的任何内容。不过，这可以作为一个有益的概念性铺垫。具体而言，不难看出，$\mathrm{LP_{CP}}$是 $\mathrm{LP_{ECC}}$的一个轻微松弛。它的不同之处在于，它始终允许半整数解，并且等价于 $\mathrm{BLP_{CP}}$ 和 $\mathrm{LP_{VC}}$。我们的分析无需证明所有这些等价关系，只需直接证明 $\mathrm{BLP_{CP}}$ 等价于 $\mathrm{LP_{VC}}$ ，并且可以通过归约到最小 s-t割问题来求解。

定理 4.2。 设 $\{a_e, b_e, a_u^i, b_u^i \mid e \in E, u \in V, i \in C\}$ 是 BLP_{CP}的最优变量集合，并对 $e \in E$ 定义：

$$x_e = \frac{1}{2}(b_e - a_e + 1). \tag{1}$$

那么， $\{x_e\}$ 是 LP_{VC} 的一组最优变量。

证明。我们首先证明 $\{x_e\}$ 按公式（1定义的变量是 $\mathrm{LP_{VC}}$的可行解。如果（e， f）是一个坏边对，则存在某个 $u \in e \cap f$ 使得以下不等式成立：

$$a_e \le a_u^{\ell(e)} \le b_u^{\ell(f)} \le b_f \quad \text{and} \quad a_f \le a_u^{\ell(f)} \le b_u^{\ell(e)} \le b_e.$$

这意味着 $x_e + x_f = \frac{1}{2}(b_e + b_f - a_e - a_f + 2) \ge 1$。因此，我们得到了 $\mathrm{LP_{VC}}$ 的一个可行解，且其目标值相同，所以 $\mathrm{OPT(BLP_{CP})} \ge \mathrm{OPT(LP_{VC})}$。

现在，我们证明 $\mathrm{OPT(BLP_{CP})} \le \mathrm{OPT(LP_{VC})}$。令 $\{x_e\}$ 为 $\mathrm{LP_{VC}}$ 的一个最优解，并按如下方式定义 a_e, b_e：

$$(a_e, b_e) := \begin{cases} (1,0) & \text{if } x_e = 0 \\ (0,1) & \text{if } x_e = 1 \\ (1,1) & \text{if } x_e = 1/2. \end{cases}$$

注意， $x_e = (b_e - a_e + 1)/2$，这意味着 $\mathrm{LP_{VC}}$ 和 $\mathrm{BLP_{CP}}$ 的目标函数相等。我们还按如下方式定义 a_u^i 和 b_u^i：

$$a_u^i := \max_{e \in E_i, u \in e} a_e \quad \text{and} \quad b_u^i := \min_{e \in E_i, u \in e} b_e.$$

这意味着对所有 $u \in e$都有 $a_e \le a_u^{\ell(e)}$和 $b_u^{\ell(e)} \le b_e$。现在，假设对 $i \ne j$有 $a_u^i > b_u^j$。那么，$a_u^i = 1$ 和 $b_u^j = 0$。这告诉我们存在某个 $e \in E_i$，使得 $u \in e$ 和 $a_e = 1$。根据 a_e的定义，有 $a_e = 1$ 意味着 $x_e \le \frac{1}{2}$。类似地，存在某个$f \in E_j$，使得 $u \in f$和 $b_f = 0$。根据 b_e的定义，有 $b_f = 0$意味着 $x_f = 0$。由于 $u \in e \cap f$和 $i \ne j$，(e, f)是一个坏边对，但 $x_e + x_f \le \frac{1}{2}$，这与$\mathrm{LP_{VC}}$中的约束 $x_e + x_f \ge 1$ 相矛盾。因此，$a_u^i \le b_u^j$ 成立。用类似的论证，我们也可以证明约束 $a_u^j \le b_u^i$ 同样成立。 □

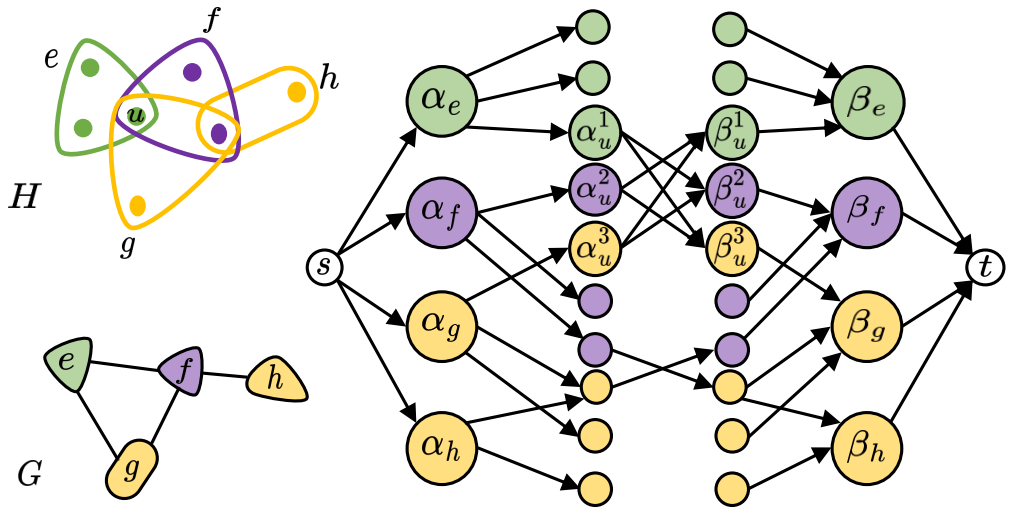


图 1：左上方展示了一个 MinECC 实例 H，它可以归约为左下方的 k-可着色Vertex Cover 实例 G 。右侧展示了归约流网络，其最小 s-t割与 BLP 的最优解一致$_{\text{CP}}$ （见第 4.3 节）。

在BLP$_{\text{CP}}$中，约束的数量为 $2\sum_{u\in V}|C(u)|+2\mu+2|E|+2\sum_{u\in V}\binom{|C(u)|}{2}=O(\mu+k^2|V|)$ 。这并不依赖于 $|\mathcal{B}|$ ，并使我们能够避免第4.1节中讨论的问题。如果k 是常数，则该值关于超图规模 μ是线性的。

4.3 归约为流网络

除了约束数量远少于 LP$_{\text{VC}}$ 外，BLP$_{\text{CP}}$ 还可以归约为一个新的最小s-t割问题，其节点数和边数少于通过显式归约为Vertex Cover 并应用 Hochbaum [11] 的结果所得到的图。形式上，对于实例 $H=(V,\ E,\ C,\ \ell)$，我们采用以下流网络构造：

1. 引入源点和汇点 s 和 t，并初始化空节点集 A_E、B_E、A_V 和 B_V。2. 对于 $e\in E$，添加节点 $\alpha_e\in A_E$ 和 $\beta_e\in B_E$。3. 对于 $u\in V$和 $i\in C(u)$，添加节点 $\alpha_u^i\in A_V$和 $\beta_u^i\in B_V$。4. 对于 $u\in V$和 $(i,j)\in\binom{C(u)}{2}$，添加有向边 (α_u^i,β_u^j) 和 (α_u^j,β_u^i)，权重为 ∞。5. 对于 $e\in E$ 和每个 $u\in e$，引入有向边 $(\alpha_e,\alpha_u^{\ell(e)})$ 和 $(\beta_u^{\ell(e)},\beta_e)$，权重为 ∞。6. 对于每个 $e\in E$，引入有向边 (s,α_e) 和 (β_e,t)，权重为 $\frac{1}{2}w(e)$。

在形成该图之后，我们在其上求解最大 s-t流问题，并利用它通过标准技术计算最小 s-t割。这将节点划分为两个集合， S 和 $\bar{S}$。图中的每个节点对应于BLP$_{\text{CP}}$中的一个二元变量：节点 α_e 对应于变量a_e，节点 β_e 对应于 b_e，类似地我们有映射 $\alpha_u^i\leftrightarrow a_u^i$和 $\beta_u^i\leftrightarrow b_u^i$。若某变量对应的节点位于 S中，则将该变量设为1，否则该变量为0。

现在，我们说明该构造如何精确刻画 BLP$_{\text{CP}}$ 中的约束。如果存在一条从 α 到 β、权重为 $-\infty$ 的有向边，那么 $\beta\in S\implies\alpha\in S$，并且 $\alpha\in S\implies\beta\in S$，以避免

表 1：六个基准 MinECC数据集的统计信息。

	超图统计							约束		
数据集	$\|V\|$	$\|E\|$	k	r	μ	$\|\mathcal{B}\|$	$(2-2/k)$	$\mathrm{LP}_{\mathrm{ECC}}$	$\mathrm{LP}_{\mathrm{VC}}$	$\mathrm{LP}_{\mathrm{CP}}$
Brain	638	21,180	2	2	42,360	490,163	1	42,998	490,163	42,998
MAG-10	80,198	51,889	10	25	180,726	473,780	1.8	260,924	473,780	910,795
烹饪	6,714	39,774	20	65	428,275	312,098,857	1.9	434,989	312,098,857	657,292
DAWN	2,109	87,104	10	22	343,211	340,602,559	1.8	345,320	340,602,559	374,775
沃尔玛	88,837	65,898	44	25	452,208	27,283,314	1.955	541,045	27,283,314	4,680,597
Trivago	207,974	247,362	55	85	757,946	2,449,951	1.964	965,920	2,449,951	12,210,823

表2：三种算法的比较。比率列表示各算法的目标分数除以 LP下界。我们展示了 5 次运行的平均值和标准差。

	$\mathrm{LP}_{\mathrm{ECC}}$-轮			VC-Flow			ColorPair-Flow		
数据集	比率	运行时间 (s)	内存	比率	运行时间 (s)	内存	比率	运行时间 (s)	内存
Brain	1	0.191±0.009	54MB	1.002	5.443±0.196	1.1GB	1	0.526±0.179	209MB
MAG-10	1	5.383±0.378	861MB	1.193	12.413±0.163	1.2GB	1.193	4.454±0.142	821MB
烹饪	1	18.465±1.647	540MB	N/A	超时	-	1.605	1.599±0.038	1.1GB
DAWN	1	2.888±0.121	398MB	N/A	超时	-	1	5.582±0.177	1.2GB
沃尔玛	1	631.559±4.443	3.5GB	N/A	已超时	-	1.654	6.794±0.042	2.4GB
Trivago	1.001	51.124±7.399	9.5GB	1.272	299.06±2.25	6.1GB	1.272	54.646±0.159	4.1GB

切断以及 ∞-权重边。如果 α 和 β 分别与变量 a 和 b相关联，则这等价于形式为 $a \le b$ 的约束，该约束强制要求 $a=1 \implies b=1$ 和 $b=0 \implies a=0$。

最后注意，最小割值恰好等于 $\mathrm{OPT}(\mathrm{BLP}_{\mathrm{CP}})$。考虑一条边 $e \in E$。如果 $\alpha_e \in S, \beta_e \in S$，那么 (s,α_e) 和 (β_e,t) 都不会被切断。如果 $\alpha_e \in S, \beta_e \in S$，那么 (s,α_e) 和 (β_e,t) 都会被切断。这会使割值增加 $2 \times \frac{1}{2}w(e) = w(e)$。如果 $\alpha_e, \beta_e \in \bar{S}$或 $\alpha_e, \beta_e \in S$，那么相应地，(s,α_e) 或 (β_e,t) 会被切断。在这两种情况下，都会使割值增加 $\frac{1}{2}w(e)$。考虑上述情形对变量 a_e 和 b_e的含义时，我们会得到与 $\mathrm{BLP}_{\mathrm{CP}}$ 目标函数相同的贡献。

改进的运行时间。我们现在分析求解新的最小 s-t割问题的运行时间。归约流网络中的节点数为$N = 2+2|E|+2\sum_{u\in V}|C(u)| \le 2+2|E|+2k|V|$，边数为

$$M = 2|E| + 2\sum_{e\in E}|e| + 2\sum_{u\in V}\binom{|C(u)|}{2} \le 2|E| + 2\mu + (k^2-k)|V|.$$

给定一个包含 N 个节点和 M 条边的有向图，van den Brand等人的算法 [17] 可在 $\tilde{O}(M+N^{1.5})$ 时间内找到最小 s-t 割。Chen 等人 [7] 可在 $M^{1+o(1)}$ 时间内解决该问题。由此可知，$\mathrm{BLP}_{\mathrm{CP}}$ 可在 $(\mu+k^2|V|)^{1+o(1)}$ 时间或 $\tilde{O}(\mu+k^2|V|+(|E|+k|V|)^{1.5})$ 时间内求解。这比显式归约到 Vertex Cover 并采用Hochbaum的方法 [11] （见第4.1节）所得的运行时间快得多。特别是，当 k 为常数时，我们目前最优的理论运行时间相对于超图规模几乎是线性的。

5 实验结果

我们将新提出的（$2-2/k$）近似算法称为 MinECC ColorPair-Flow，因为它相当于求解一个由颜色对 BLP 导出的最大 s-t 流问题。该算法使用定理4.2来求得 LP_{VC}的一个解，随后使用算法3 删除某些边，使节点能够以一种令所有剩余边都满足条件的方式着色。我们使用 Julia 实现了该算法，并在一套标准基准 MinECC 数据集上进行了实验。有关这些超图的数据集统计信息及先前研究，请参见表1。 [1, 18]我们在一台配备 1TB 内存的研究服务器上进行了实验。代码可在以下地址获取：https://github.com/shan-mcs/minecc-combinatorial/。我们使用 Julia 实现的推送-重标记算法求解最大流，该算法最初用于 Veldt 等人的研究 [19] ，后经 Huang 等人改进 [12]。表2 中的结果表明，该方法通常能找到远优于最坏情况保证（$2-2/k$）的解，有时甚至能找到最优解。

为了进行比较，我们实现了 Hochbaum 的工作所引出的最小 s-t 割构造方法 [11] ，用于 Vertex Cover (VC-Flow；见第4节）。该方法面临显著的内存和运行时间问题，因为它需要显式枚举并存储所有坏边对 $\mathcal{B}$，而其数量可能极其庞大。果不其然，该方法通常会超时（我们将上限设为 15 分钟）。我们还尝试使用 Gurobi 优化软件直接求解 LP_{VC} ，但由于同样的原因，也遇到了内存和运行时间问题。这一比较凸显了我们方法的优势。LP_{VC} 是一个值得考虑的线性规划松弛，因为它具有半整数解，且特别适用于组合方法。然而，直接求解它的方法完全不可行，因为 $|\mathcal{B}|$即使对于中等规模的超图也可能非常庞大。我们强调，这种方法在 MinECC 上的糟糕表现绝非 Hochbaum 工作的局限所致 [11], ，该工作关注的是 Vertex Cover Vertex Cover。相反，糟糕表现源于显式地将 MinECC 约化为 Vertex Cover时遇到的问题。我们的方法绕过了这一问题，无需枚举 $\mathcal{B}$即可求解线性规划。

我们还使用 Gurobi 求解 LP_{ECC} ，并通过一种标准方法对其进行舍入：将节点 u 赋予颜色 $c = \arg\max_i x_u^i$ [1, 18]。这样得到的解质量更高（因为 LP_{ECC} 提供了更紧的下界），而且在某些情况下甚至比 ColorPair-Flow 稍快。不过，这种方法的运行时间可能异常长。在沃尔玛数据集上，LP_{ECC}-round 耗时超过 10 分钟，而 ColorPair- Flow 不到 10 秒。目前还不清楚超图的哪些特征会导致运行时间特别长。尤其是，沃尔玛数据集在 $|V|$、 $|E|$、r、 k、 μ 或 $|\mathcal{B}|$方面都不是最高的。相比之下，ColorPair-Flow 在运行时间和解质量之间取得了较好的平衡，既能生成优质解，又能确保运行时间快且稳定、可预测。我们还跟踪了各方法分配的内存。对于大型数据集，ColorPair-Flow 所需内存相对较少。对于沃尔玛和 Trivago 数据集，ColorPair-Flow 分别分配了 2.4GB 和 4.1GB，而基于 LP_{ECC}的方法则分配了 3.5GB 和 9.5GB。

6 结论

我们提出了一种用于 MinECC 的组合式（$2-2/k$）近似算法，其理论运行时间远短于求解并舍入规范线性规划松弛。在实践中，规范线性规划仍能产生极佳的结果，而且通常速度很快，但在某些情况下也可能非常慢。一个值得探索的方向是使用更快的最大流求解器，进一步缩短 ColorPair-Flow 的实际运行时间。

致谢

两位作者均获得美国空军科学研究办公室（AFOSR）资助，资助编号为 FA9550-25-1-0151。此外，Nate Veldt 还获得美国陆军研究办公室（ARO）资助，资助编号为 W911NF24-1-0156。本文所述观点和结论均为作者所有，不应被解读为代表美国陆军研究办公室或美国政府的官方政策，无论这些政策是明示还是暗示的。

参考文献

[1] Ilya Amburg、Nate Veldt 和 Austin Benson。带有类别边标签的图与超图聚类。载于*2020*年*ACM*万维网会议论文集（*WWW 2020*），2020年。[2] Ilya Amburg、Nate Veldt 和 Austin R. Benson。通过超图聚类发现多元且经验丰富的群体。载于*2022*年*SIAM*国际数据挖掘会议论文集（*SDM 2022*），2022年。[3] E. Angel、E. Bampis、A. Kononov、D. Paparas、E. Pountourakis 和 V. Zissimopoulos。k边着色图上的聚类。离散应用数学，211：15–22，2016年。[4] R Bar-Yehuda 和 S Even。加权顶点覆盖问题的线性时间近似算法。算法期刊，2(2)：198–203，1981年。[5] R. Bar-Yehuda 和 S. Even。用于近似加权顶点覆盖问题的局部比率定理。109：27–45，1985年。[6] Leizhen Cai 和 On Yin Leung。交错路径与着色聚类。*arXiv*预印本：*1807.10531*，2018年。[7] Li Chen、Rasmus Kyng、Yang P Liu、Richard Peng、Maximilian Probst Gutenberg 和 Sushant Sachdeva。近线性时间内的最大流与最小费用流。载于*2022*年电气与电子工程师协会第*63*届计算机科学基础年度研讨会论文集（*FOCS 2022*），2022年。[8] Michael B Cohen、Yin Tat Lee 和 Zhao Song。以当前矩阵乘法所需时间求解线性规划。*ACM*学报（*JACM*），68(1)：1–39，2021年。[9] Alex Crane、Brian Lavallee、Blair D. Sullivan 和 Nate Veldt。超图中的重叠式鲁棒边着色聚类。载于*2024*年*ACM*网络搜索与数据挖掘国际会议论文集，WSDM 2024，2024年。[10] Alex Crane、Thomas Stanley、Blair D Sullivan 和 Nate Veldt。超图中的边着色聚类：超越最小化未满足边。载于国际机器学习会议，ICML 2025，第11435–11464页，2025年。[11] Dorit S Hochbaum。稳定集、顶点覆盖和集合装填问题的有效界。离散应用数学，6(3)：243–254，1983年。[12] Yufan Huang、David F Gleich 和 Nate Veldt。最密子超图：负超模函数与强局部化方法。载于*2024*年*ACM*万维网会议论文集，WWW 2024，第881–892页，2024年。

[13] Shunhua Jiang、Zhao Song、Omri Weinstein 和 Hengjie Zhang。求解一般线性规划的更快算法。载于第*53*届*ACM SIGACT*计算理论年度研讨会论文集，STOC 2021，2021年。[14] Changyeol Lee、Yongho Shin 和 Hyung-Chan An。边着色超图重叠式鲁棒聚类的改进算法：一种基于线性规划的组合方法。载于第*39*届神经信息处理系统年度会议论文集，NeurIPS 2025，2025年。[15] George L Nemhauser 和 Leslie E Trotter。顶点打包：结构性质与算法。《数学规划》，8(1)：232–248，1975年。[16] Leonard Brian Pitt。顶点覆盖的简单概率近似算法。耶鲁大学计算机科学系，1985年。[17] Jan van den Brand、Yin Tat Lee、Yang P Liu、Thatchaphol Saranurak、Aaron Sidford、Zhao Song 和 Di Wang。稠密实例中近线性时间求解最小费用流、马尔可夫决策过程和 ℓ_1-回归。载于第*53*届*ACM SIGACT*计算理论年度研讨会论文集，STOC 2021，2021年。[18] Nate Veldt。边着色超图聚类的最优线性规划舍入与线性时间近似算法。载于国际机器学习会议（*ICML 2023*），2023年。[19] Nate Veldt、Christine Klymko 和 David F Gleich。具有更佳种子集包含性质的基于流的局部图聚类。载于*2019*年*SIAM*国际数据挖掘会议论文集，SDM 2019，第378–386页，2019年。