通过边阻断精确操控团数

周毅[a]，*，蒋浩宇[a]，朱成昊[a]，安德烈·罗西[b]

[a]计算机科学与工程学院，电子科技大学，成都，中国[b]*LAMSADE UMR CNRS 7243* 巴黎综合理工大学–*PSL*，巴黎，法国

arXiv:2601.01869v1 [cs.DS] 5 Jan 2026

摘要

边干预团问题（EICP）旨在从图中移除至多 k条边，从而最小化剩余图中最大团的规模。该问题揭示了图操控中的一个基本问题：哪些边对于保持大团在结构上至关重要？这一问题还受到蛋白质功能维护和图像匹配等实际应用的推动。EICP 在计算上具有挑战性，属于超出 NP 的复杂性类。现有方法依赖通用的混合整数双层规划求解器，或将问题重新表述为单层混合整数线性规划。然而，当图的规模和阻断预算k 增长时，这些方法仍然无法扩展。为克服这一问题，我们研究了新的混合整数线性规划形式，将该问题重新表述为一系列参数化边阻断团问题（EBCP）。这一视角将原问题分解为更简单的子问题，并能够对与团相关的不等式进行更紧的建模。此外，我们提出了一种两阶段精确算法 RLCM：首先应用问题专用约简技术来缩减图，然后利用定制的分支定界框架求解缩减后的问题。在最大团基准图、大型真实世界稀疏网络和随机图上的大量计算实验表明，RLCM 始终优于现有方法。

关键词： 组合优化、阻断问题、最大团、混合整数线性规划

*通讯作者。电子邮箱：zhou.yi@uestc.edu.cn（周毅）、FirstSSAT@outlook.com（蒋浩宇）、axs7384@gmail.com（朱成昊）、andre.rossi@dauphine.psl.eu（安德烈·罗西）

1. 引言

最大团问题 （MCP）是一个教科书式的组合优化问题，在理论和实践中都得到了广泛研究。给定一个图 G，团是顶点的一个子集，其中任意两个不同的顶点都相邻。MCP旨在确定给定图 G 中最大团的大小，通常称为团数 $\omega(G)$。从理论角度来看，MCP是NP-困难的，难以近似 [15]，并且是W[1]-困难的 [8]。尽管存在这些困难性结果，过去几十年来人们仍开发了大量精确算法和启发式算法，其中许多算法在实际实例 [28, 3, 25, 16] 上表现良好。

近年来，人们越来越关注通过边修改来改变图的团数的问题 [33, 11, 29, 12, 2]。这类问题的一个基本变体是，考虑通过添加或删除指定数量的边来增加或减小团数。由于图的团数只能通过添加边来增加、通过删除边来减小，因此一个自然的问题是：应添加哪些 k 边才能使团数最大化，或应删除哪些 k 边才能使团数最小化。

前一个问题可以重新表述为：寻找最大的顶点集，使其诱导子图与完全图至多相差k 条缺失边。文献中将此问题称为最大k-缺陷团问题（*MDCP*） [21, 14]。据我们所知，近年来 MDCP 得到了广泛研究，并且可以在大型真实世界图上高效求解。 [5, 13, 4, 6, 18]。相比之下，选择至多 k 条边进行删除，从而使所得图的团数最小化的问题，称为边阻断团问题（*EICP*） [12]。因此，MDCP 和 EICP 似乎是对偶优化问题，如下表所示。

问题	定义	操纵
MDCP	$\max_{F\subseteq V\times V,\lvert F\rvert\leq k} \omega((V,\ E\cup F))$	边添加
EICP	$\min_{F\subseteq V\times V,\lvert F\rvert\leq k} \omega((V, E\setminus F))$	边删除

然而，从建模角度来看，EICP 的复杂性本质上更高。具体而言，EICP 可以表述为一个双层优化

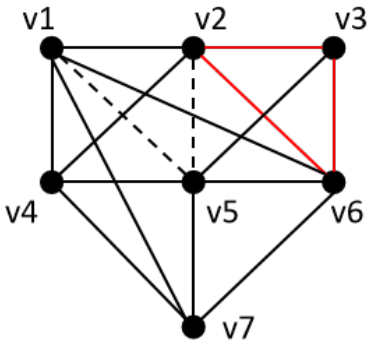


图1：EICP 示例。在原图中，团数 $\omega(G)$ 等于 4。如果 $k = 2$，则最优策略是删除边（v_1、v_5）和（v_2、v_5）（以虚线标出），从而将 $\omega(G)$ 降至 3（以红色边突出显示）。

问题 [7]，其中嵌套了两个对抗性优化问题：上层最小化 $\omega(G)$，而下层则在所得图中计算最大团。这与 MDCP 形成鲜明对比，后者可以表述为单层优化问题。图1给出了一个示例。除了在图论方面的研究价值外，EICP 也受到一系列现实应用需求的驱动。下面介绍两个具有代表性的示例。

蛋白质功能维护。 在蛋白质结构分析中，团通常对应高度保守的功能区域。删除团内的边可以理解为破坏特定的分子相互作用，从而提供一种评估部分交互作用 [23, 1] 下蛋白质功能韧性的方法。

图像匹配。 图像匹配问题可以建模为关联图中的团查找问题，其中最大团代表一致的特征对应关系 [26, 24]。求解 EICP 可以识别关键匹配，并评估边删除情况下匹配的稳健性。

尽管 EICP 从多个角度来看都很重要，但与 MDCP 相比，文献对这一问题的关注要少得多。或许目前求解 EICP 最有效且具备实际可行性的精确方法由 [12], 提出，其中结合了混合整数线性规划公式化与上界技术。然而，由于缺乏公开可用的完整源代码等原因，该方法在大型真实世界图上的表现尚未得到系统评估。

此外，在实际时间限制下，随着阻断预算 k 在关键范围内增加，该方法能够求解的实例数量会显著减少。 [33] 提出了一种替代方法，将 EICP 转化为节点阻断团问题 （NICP），其定义为：确定至多给定数量的顶点，使删除这些顶点后剩余图的团数最小化。这种转化使得现有 NICP 求解器（例如 [11, 20] 开发的求解器）得以应用。然而，该转化会使边数从原图中的 $|E|$ 增加到转化后图中的 $\Theta(|E|^2)$，从而带来巨大的计算开销，尤其是在大型真实世界图上。因此，这种方法目前主要具有理论研究价值。另一个与 EICP 密切相关的问题是边阻断团问题 （EBCP） [19], ，其目标是尽可能少地删除边，使剩余图的团数不超过给定阈值。显然，可以通过在不同阈值下迭代检验 EBCP 来求解 EICP。本文将进一步探讨这种联系，并展示如何在算法层面加以利用。最后，如前所述，EICP 属于更广泛的双层阻断问题。原则上，也可以应用通用的混合整数双层优化求解器 [9, 10] 。我们通过实验评估了这一方案，并得出结论：它并非求解 EICP 的最高效方法。

1.1. 我们的贡献

尽管 EICP 具有重要意义，但高效的算法仍然有限。本文通过以下贡献填补了这一空缺。

- 我们将 EICP 重新表述为一系列参数化边阻断团问题（EBCP），其目标是移除尽可能少的边，使团数降至给定阈值 p 以下。我们为 EBCP 提出了新的混合整数线性规划公式，包括一种基于极大团的公式化，并说明如何将其相关不等式嵌入分支割框架，作为有效割。

- 基于这一重新表述，我们提出了 RLCM，这是一种两阶段精确算法，结合了图约简、新颖的下界估计，以及采用惰性约束分离的迭代分支割过程，从而高效求解 EICP。

• 在 DIMACS基准图、大型真实世界稀疏网络和随机图上的大量实验表明，RLCM 的表现始终优于现有方法。具体而言，在 600秒时间限制内，对于每个测试的 k 值，RLCM 求解的困难 DIMACS实例均多于其他算法。在大型真实世界稀疏网络上，RLCM 在 695 个实例中的 664 个实例上，运行时间短于其他求解器。消融研究进一步证实了所提各组件的有效性。

2. 符号

考虑一个简单无向图 $G = (V, E)$，其中 V 表示顶点集，E 表示边集。顶点 $v \in V$ 的邻域记为 $N_G(v)$，即与 v 相邻的顶点集合。为简化符号，除非另有说明，我们使用简写形式 $N(v)$ 代替 $N_G(v)$。对于顶点的任意子集 $S \subseteq V$，记号 $G[S]$ 表示由 G 中的 S 诱导的子图，记号 $E(S)$ 表示 $G[S]$ 的边集。

给定图 $G = (V, E)$，若顶点集 $C \subseteq V$ 中任意一对不同顶点 $u, v \in C$ 之间的边 $\{u, v\}$ 属于 E，则称 $C \subseteq V$ 为 G 的团。G 的团数记为 $\omega(G)$，即 G 中最大团的基数。如前所述，$\omega(G)$ 的计算已得到深入研究。我们用 MaxClique(G) 表示一个原语，该原语从图 G 中返回一个最大团。在我们的实现中，MaxClique(G) 基于两种性能优异的开源算法构成的组合策略。具体而言，当 $|V| \leq 500$ 时，我们采用 MoMC 求解器 [16], ; 对于 $|V| > 500$，则使用 MC-BRB 求解器 [3]。这种组合策略旨在使算法在不同规模的图上都能保持稳健的性能。

如果一个团不是任何更大团的真子集，则称其为极大 团。全文中，在上下文明确时，我们稍微滥用记号，用 C 同时表示一个团及其诱导的完全子图，即，$G[C]$。给定图 $G = (V, E)$ 和非负预算值 $k \in \mathbb{N}$，$\eta(G, k)$ 表示通过从 G 中至多移除 k 条边后所得子图的最小团数，即

$$\eta(G, k) = \min_{\substack{F \subseteq E \\ |F| \leq k}} \omega(G' = (V, E \setminus F))$$

计算 $\eta(G,\ k)$，并找到达到该最小值的相应边集 F^*，这一问题称为边干预团问题 (EICP)。

3. EICP 的混合整数线性规划公式

3.1. EICP 的一种现有集合覆盖形式

文献 [12], 提出了一种针对 EICP 的集合覆盖型混合整数线性规划公式。该公式假设事先已知 $\eta(G,k)$ 的有效下界和上界，分别记为 lb 和 ub。对于每条边 $e \in E$，引入一个二元变量 x_e：若边 e 被干预，则 $x_e = 1$；否则 $x_e = 0$。此外，对于每个 $i \in \{lb, lb+1, ..., ub\}$，定义二元变量 y_i，使得当 $\eta(G,k) = i$ 时，$y_i = 1$；否则 $y_i = 0$。根据构造，恰有一个变量 y_i 必须取值为 1，其余变量均为 0。利用这些变量，EICP 的集合覆盖型混合整数线性规划公式（记为 EICP-MILP）如下所示。

$$
\begin{aligned}
\textbf{(EICP-MILP)}\quad \min\quad & \sum_{i=lb}^{ub} i y_i \\
& \text{s.t.} \sum_{e\in E(C)} x_e \geq \sum_{i=lb}^{ub} \gamma_{clq}(|C|,i) y_i, \quad \forall C \in \mathcal{C},\ |C| \geq 2, && (1)\\
& \sum_{e\in E} x_e \leq k, && (2)\\
& \sum_{i=lb}^{ub} y_i = 1 && (3)\\
& y_i \in \{0,1\}, \quad \forall i \in \{lb, lb+1, \ldots, ub\} \\
& x_e \in \{0,1\}, \quad \forall e \in E
\end{aligned}
$$

在公式化中，$\mathcal{C}$ 表示 G中所有团的集合。函数$\gamma_{clq}(n,\ i)$，其中 $n,\ i \in \mathbb{N}$和 $i \leq n$，表示从一个有 n 个顶点的完全图中必须移除的最少边数，使所得图的团数至多为 i。根据图兰定理 [27],

$\gamma_{clq}(n,\ i)$可由以下闭式表达式给出

$$\gamma_{clq}(n,i)=\begin{cases} n_{\alpha-1}\binom{\alpha-1}{2}+n_{\alpha}\binom{\alpha}{2}, & \text{if } i<n, \\ 0, & \text{if } i\geq n, \end{cases}$$

其中 $\alpha=\left\lceil \frac{n}{i} \right\rceil$，$n_{\alpha-1}=i\alpha-n$，$n_{\alpha}=\underline{^{n-n_{\alpha-1}\ (\alpha-1)\ \alpha}}$。

因此，不等式（1）规定，对于 G中的每个团 C ，只要 $\eta(G,\ k)\ =\ i$，就必须在 $E(C)$中阻断至少$\gamma_{clq}(|C|,\ i)$条边。不等式（2）和（3）将被阻断边的总数限制为 k ，并确保恰有一个变量 y_i 取值为1。对于上下界 lb 和ub，傅等人建议，对于非平凡图，可采用简单的下界 $lb\ =\ 2$，并引入一种重新公式化的混合整数线性规划，以获得有效的上界ub [12]。

3.2. EICP 的参数化公式

边干预团问题混合整数线性规划通过变量 y_i 同时对所得团数进行编码，并通过变量 x_e对被阻断的边进行编码，因此搜索空间非常庞大。本节提出一种参数化公式，以顺序方式确定 $\eta(G,\ k)$，从而将问题分解。其核心思想是将 EICP 化简为前文提到的边阻断团问题（EBCP） [19], ，该问题的形式化定义如下。

定义 1（边阻断团问题）。给定一个无向图$G\ =\ （V,E）$和一个参数 $p\in\mathbb{N}$，边阻断团问题要求找出一个最小基数边子集 $F\subseteq E$ ，使其移除后，剩余图 $G'=（V,E\setminus F）$的团数至多为 p，即$\omega（G'）\ \leq p$。我们将最优边集的最小基数记为 $\gamma（G,p）$ 。

特别地，当 G 是一个有 n 个顶点的完全图时，我们有 $\gamma(G,\ p)\ =\gamma_{clq}(n,\ p)$。因此，对于完全图，EBCP 存在闭式解，并且可以在常数时间内求解。

基于边干预团问题混合整数线性规划，可以轻松推导出一种混合整数线性规划公式——

用于 EBCP 的公式如下。

$$\begin{aligned}\textbf{(EBCP-MILP)}\gamma(G,p) = \min \quad & \sum_{e\in E} x_e \\ \text{s.t.} \quad & \sum_{e\in E(C)} x_e \geq \gamma_{clq}(|C|,p), \quad \forall C \in \mathcal{C},\ |C| \geq 2, \\ & x_e \in \{0,1\}, \quad \forall e \in E\end{aligned} \tag{4}$$

在此公式化中，二元变量 x_e 表示边 $e \in E$是否被阻断，而 $\mathcal{C}$ 仍表示图 G 中所有团的集合。（我们使用术语 阻断 ，以区别于 EICP 的语境中使用的术语 阻截 。）不等式 (4)规定，在每个团 C 中至少阻断 $\gamma_{clq}(|C|, p)$ 条边，即足以将团数限制为 p。

我们进一步注意到，对于固定的图 G，$\gamma(G, p)$ 关于 p 单调不增。因此，如果能够高效计算 $\gamma(G, p)$，那么也可以通过对 p 进行二分搜索来求解 EICP，这需要进行 $O(\log(ub - lb))$ 次 $\gamma(G, p)$ 计算。具体而言，如果 $\gamma(G, p) \leq k$，则 $\eta(G, k) \leq p$；否则，$\eta(G, k) > p$。

在 EBCP-MILP 中，不等式 (4) 的数量等于图中团的总数。由于团的任意子集本身也是团（这一性质通常称为 遗传性），大小为 n 的团包含 $2^n - n - 1$ 个非平凡子团。若 $\mathcal{C}_{\max}$ 是图 G 中极大团的集合，则 EBCP-MILP 可能需要的不等式总数为 $O(|\mathcal{C}_{\max}|2^{\omega(G)})$。

3.3. 使用极大团重新表述 *EBCP-MILP*

我们对 EBCP-MILP 进行重新表述，使其只需考虑极大团。设 $C \in \mathcal{C}_{\max}$ 是 G 的一个极大团。给定 C 的顶点排列，记为 $\overrightarrow{C}$，我们根据该排列对诱导子图 $G[C]$ 中的每条边进行定向，从而得到一个竞赛图。具体而言，对于每条边 $e \in E(C)$，将其从 $\overrightarrow{C}$ 中排名较低的顶点指向排名较高的顶点。对于顶点 $u \in C$，令 $N_{\overrightarrow{C}}(u)$ 表示其在所得有向图中的出邻居集合。基于 $\overrightarrow{C}$，我们引入二元变量$z_{u,\ \overrightarrow{C}}$ 以及不等式（5）–（8）。直观而言，$z_{u,\ \overrightarrow{C}} = 1$ 表示 u 的所有出边均未被阻断。这些约束确保数量

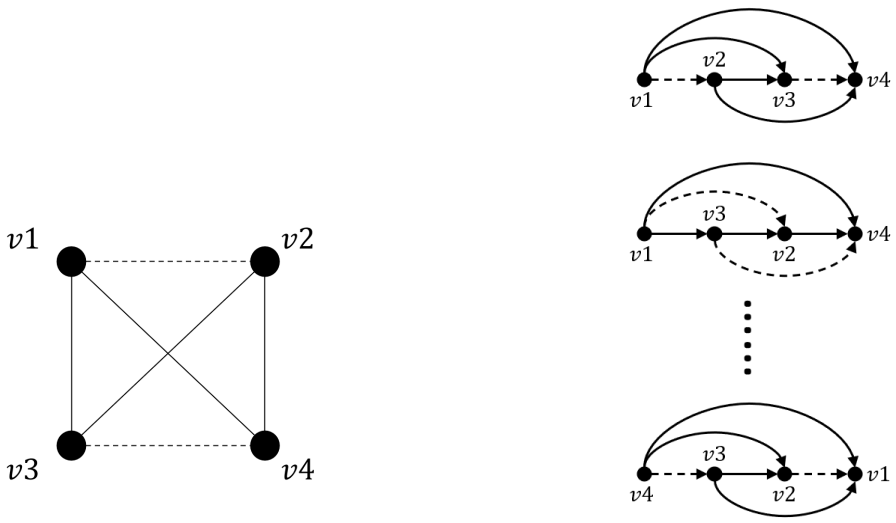
其出边仍未被阻断的顶点数受 p 限制，从而对边阻断后的团数施加上界。引理 1 表明，这些不等式对于 EBCP 是有效的。

$$z_{u,\overrightarrow{C}} + \sum_{v \in N_{\overrightarrow{C}}(u)} x_{u,v} \geq 1, \qquad \forall\, u \in C, \tag{5}$$

$$z_{u,\overrightarrow{C}} + x_{u,v} \leq 1, \qquad \forall\, u \in C,\ \forall\, v \in N_{\overrightarrow{C}}(u), \tag{6}$$

$$\sum_{u \in C} z_{u,\overrightarrow{C}} \leq p, \tag{7}$$

$$z_{u,\overrightarrow{C}} \in \{0,1\}, \qquad \forall\ u \in C \tag{8}$$



(a) 来自原图的一个团 C 。

(b) 有向图 $\overrightarrow{C}$ 得到来自 C的不同排列。

图2：团 C 和有向图的一个示例。

引理 1。 设 C 为一个团，并令 $x_e = 1$ 表示边 $e \in E(C)$ 被阻断（移除），否则为 $x_e = 0$ 。给定一个整数 p，假设对于 C 的每个排列 $\overrightarrow{C}$，都存在一个变量赋值 z，使得不等式 *(5)-(7)* 得到满足。则不存在满足 $|C'| \geq p$ 的子团 $C' \subseteq C$。

证明。 我们用反证法证明这个引理。假设阻断边 e where $x_e = 1$ 后，$C' \subseteq C$ 中存在一个团，使得 $|C'| = p + 1$。令

$C' = \{v_1, ..., v_{p+1}\}$。考虑 C 的一个排列 $\overrightarrow{C}$，其中 C' 的顶点按 v_1，……，v_{p+1} 的顺序连续出现。由于 C' 是剩余图中的一个团，因此 C' 中顶点之间的边均未被阻断。因此，$x_{v_i,\ v_j} = 0,\ \forall 1 \leq i < j \leq p+1$。对于此排列 $\overrightarrow{C'}$，由不等式（5）和（6）可得 $z_{u,\ \overrightarrow{C'}} = 1,\ \forall u \in C'$。因此，$\sum_{u \in C} z_{u,\ \overrightarrow{C'}} \geqslant |C'| = p+1$，这与不等式（7）矛盾。这个矛盾完成了证明。

□

示例。考虑图 2(a) 所示的团 C。假设$x_{v_1,\ v_2} = 1$, $x_{v_3,\ v_4} = 1$ 和 $x_e = 0$ 是 $E(C)$中的其他边；也就是说，在当前整数解下，边$\{v_1, v_2\}$ 和 $\{v_3, v_4\}$ 被阻断。令$p = 2$。图 2(b) 展示了 C的顶点排列中的一些排列。对于 $\overrightarrow{C}$的每种排列，如果 v 的出边均未被阻断，则令 $z_{v,\ \overrightarrow{C}} = 1$；否则令 $z_{v,\ \overrightarrow{C}} = 0$。可以直接验证，对于每种排列 $\overrightarrow{C}$，不等式 $\sum_{v \in C} z_{v,\ \overrightarrow{C}} \leq 3$ 均成立。

现在，我们准备给出一种仅基于极大团的 EBCP 精细混合整数线性规划公式。

$$\textbf{(EBCP-MILP-2)}\gamma(G,p) = \min \quad \sum_{e \in E(G)} x_e$$

$$\text{s.t.} \sum_{e \in E(C)} x_e \geq \gamma_{clq}(|C|, p), \quad \forall C \in \mathcal{C}_{\max},\ |C| \geq 2 \tag{9}$$

$$\text{Inequality (5)-(8)}, \quad \forall \text{permutation } \overrightarrow{C}, \forall C \in \mathcal{C}_{\max},\ |C| \geq 2 \tag{10}$$

$$x_e \in \{0,1\}, \quad \forall e \in E$$

根据引理 1，该公式能够正确约束边阻断后所得图的团数不超过 p。因此，**EBCP-MILP-2** 为 EBCP 提供了一个有效公式。

3.4. 分支定界算法中两种公式化方法的结合

尽管 EBCP-MILP-2 只需要极大团，但不等式总数仍有上界 $O(|\mathcal{C}_{\max}|\omega(G)!)$，其渐近增长速度高于原始 EBCP-MILP。为了解决这一问题，我们在分支割框架中结合这两种公式化。总体思路如下：首先提取一部分极大

团 G，并构建公式化 EBCP-MILP。当获得整数解 $\{x_e\}$ ——对应于边集 F ——时，我们构造剩余图 $G' = (V,\ E \setminus F)$，并从中G'计算一个最大团 C。如果该团违反不等式 (4)，我们会抽取少量排列 $\overrightarrow{C}$（在我们的实现中为两个），并针对每个排列，将不等式 (4) 和额外不等式 (5)-(8) 加入混合整数线性规划模型。总体而言，该过程以割平面方式引入 $O(|C|)$ 个额外变量和 $O(|C|^2)$ 个额外不等式。在下一节中，我们将说明这一策略如何提升算法的效率，尤其是在稠密图上。

4. 两阶段算法 - RLCM

本节介绍用于求解 EICP 的 RLCM 算法。该算法遵循两阶段框架，每个阶段分别处理该问题的一个不同方面。

- 预处理。给定输入图 G和预算值 k，第一阶段计算 $\eta(G,\ k)$ 的下界 lb，并基于该下界应用一组约简规则。这些约简能够保证最优性，因此不会改变原问题的最优值。

- 迭代分支割。在第二阶段，RLCM 估计 $\eta(G,\ k)$ 的上界 ub，然后迭代确定满足 $\gamma(G,\ p) \leq k$ 的最小整数 $p \in \{lb, \ldots, ub\}$。具体而言，该算法初始化 $p = ub - 1$，并使用分支割过程求解相应的 EBCP 实例，以计算 $\gamma(G,\ p)$。如果 $\gamma(G,\ p) \leq k$，则将 p 减一并继续该过程。否则，算法终止，并返回 $p + 1$ 作为 $\eta(G,\ k)$ 的最优值。

RLCM 的两个阶段将在以下小节中详细介绍，我们将进一步阐述其组成部分及整合方式。

4.1. 下界估计

我们提出了一种组合过程，用于计算 $\eta(G,\ k)$ 的有效下界，具体总结于算法 1 中。该方法基于以下观察

对于 G 的任意子图 G'， $\eta(G', k)$ 的值为 $\eta(G, k)$ 提供了一个下界；这一思想此前已应用于顶点阻断问题 [11]。

该算法构造了一个子图 G'，由一组顶点不相交团 $\mathcal{C}' \subseteq \mathcal{C}$组成。对于此类图，可以利用完全图的 $\gamma_{clq}(\cdot, \cdot)$ 闭式表达式，高效地计算受限 EICP。特别地，将 G' 的团数降低至不超过 p 所需的最少边数为

$$\gamma(G', p) = \sum_{C \in \mathcal{C}'} \gamma_{clq}(|C|, p),$$

由此可以通过二分搜索确定 p 的最小可行值。该过程的总体时间复杂度为 $O(|V|^2 + |\mathcal{C}'| \log(\omega(G))) = O(|V|^2)$，因为 $|\mathcal{C}'| < |V|$。因此，时间主要由不相交团的构造所占用。

算法 1： $\eta(G, k)$ 的下界。

输入： 图 $G = (V, E)$ 和阻断预算 k

输出： $\eta(G, k)$ 的下界值 lb

估计下界(G, k)

开始

  初始化一个空集合 $\mathcal{C}' \leftarrow \{\emptyset\}$；

  对于所有 $v \in V$ **执行**

    如果存在一个 $C \in \mathcal{C}'$，使得 $C \cup \{v\}$ 是一个团 **则**

      通过用 $C \cup \{v\}$ 替换 C 来更新 $\mathcal{C}'$

    else

      将 $\{v\}$ 添加到 $\mathcal{C}'$ 中

  令 $\Delta = \max\{|C| : C \in \mathcal{C}'\}$；

  使用二分搜索计算最小的 $p \in \{1, 2, ..., \Delta\}$ 使得 $\sum_{C \in \mathcal{C}'} \gamma_{clq}(|C|, p)$ 不超过 k；

  返回 p

4.2. 图约简

在获得关于 $\eta(G, k)$ 的有效下界 lb 后，我们应用图约简规则来简化输入图 G。其基本原理如下：如果 G 的某个顶点或某条边不属于任何大于 lb 的团，则可以安全地将其从 G 中移除，而不改变

$\eta(G,\ k)$。为便于记号表示，令 $\mathcal{F}(G,\ k)$ 表示最优阻断集的族，即，$\mathcal{F}(G,\ k) = \{F^* \subseteq E|\omega((V,\ E \setminus F^*)) = \eta(G,\ k)\}$。下面给出一条基于顶点的约简规则。

引理 2（顶点团约简）。设 $G = (V, E)$ 是一个图，且k 和 lb 是两个整数，使得 lb 是 $\eta(G, k)$ 的一个有效下界。若存在一个满足 $\omega(G[N(u)]) \leq lb - 2$的顶点 $u \in V$，则 $\eta(G, k) = \eta(G[V \setminus \{u\}], k)$。

证明。设 $G' = G[V \setminus \{u\}]$和 $F \in \mathcal{F}(G', k)$。由于 G'是G的子图，因此 $\eta(G, k) \geq \eta(G', k) \geq \eta(G, k) - 1$。此外，因为$F \in \mathcal{F}(G', k)$，所以有 $\omega((V \setminus \{u\}, E \setminus F)) = \eta(G', k)$。而且，由于删除边不会增加团的大小，$\omega((N(u)\ E \setminus F)) \leq \omega(G[N(u)]) \leq lb - 2$。由于 $lb \leq \eta(G, k)$ 且 $\eta(G', k) \geq lb$，我们得到 $\omega((N(u)\ E \setminus F)) \leq \eta(G', k) - 1$。因此，顶点 u不能属于图 $(V, E \setminus F)$ 中大小为 $\eta(G', k)$ 的任何团，这意味着 $\omega((V, E \setminus F)) = \omega((V \setminus \{u\}, E \setminus F)) = \eta(G', k)$。由此，$\eta(G, k) \leq \eta(G', k)$。由于已证明 $\eta(G, k) \geq \eta(G', k)$，我们得出结论：$\eta(G, k) = \eta(G', k)$。 □

引理 *3*（边团约简）。设 $G = (V, E)$ 是一个图，且 k 和 lb 是两个整数，使得 lb 是 $\eta(G, k)$ 的一个有效下界。若存在一条边 $\{u, v\} \in E$ 满足 $\omega(G[N(u) \cap N(v)]) \leq lb - 3$，则 $\eta(G, k) = \eta((V, E \setminus \{\{u, v\}\}), k)$。

证明。设 $G' = (V, E \setminus \{\{u, v\}\})$ 和 $F \in \mathcal{F}(G', k)$。由于 G' 是 G 的一个子图，因此 $\eta(G, k) \geq \eta(G', k) \geq \eta(G, k) - 1$。此外，由于 $F \in \mathcal{F}(G', k)$，我们有 $\omega((V, E \setminus \{\{u, v\}\} \setminus F)) = \eta(G', k)$。我们还有 $\omega((V, E \setminus F) \geq \eta(G, k) \geq lb$。由于 $\omega((N(u) \cap N(v)\ E \setminus F)) + 2 \leq \omega(G[N(u) \cap N(v)]) + 2 \leq lb$，我们有 $\omega((V, E \setminus F)) = \omega((V, E \setminus \{\{u, v\}\} \setminus F]) = \eta(G', k)$ 因此，$\eta(G, k) \leq \omega((V, E \setminus F)) = \eta(G', k)$。由于 $\eta(G, k) \geq \eta(G', k)$ 已经得到证明，我们得出结论：$\eta(G, k) = \eta(G', k)$。 □

基于引理 2 和引理 3 中确立的约简规则，我们采用算法 2 中概述的预处理过程。该算法首先计算一个有效下界 lb，然后迭代地移除局部邻域无法容纳大于 lb 的团的顶点和边。该过程利用基于度数和基于着色的上界来执行这些约简，而无需显式计算团数。此过程广泛用于计算图的团数上界

[17, 11, 3, 5] 对于剩余的边，在 $G[N(u) \cap N(v)]$ 上执行精确的团计算，以应用边约简规则。每当此类计算得到一个团时，就将其存入团池 $\mathcal{C}_{pool}$，以供稍后在 EBCP-MILP 表述中使用。

算法 2： 预处理算法

输入： 初始图 $G = (V, E)$，阻断预算 k

输出： 约简后的图 G' 和团集 $\mathcal{C}_{pool}$

```
预处理(G, k)
开始
    lb ← 估计下界(G, k);
    递归删除所有满足 u with |N(u)| ≤ lb − 2 的顶点，以及所有
      满足 {u, v} with |N(u) ∩ N(v)| ≤ lb − 3 from G 的边;
    for u ∈ V 执行
        如果 χ(G[N(u)]) ≤ lb − 2 则
            从 G 中删除 u;
    对于 {u, v} ∈ E，执行
        如果 χ(G[N(u) ∩ N(v)]) ≤ lb − 3 则
            从 G 中移除 {u, v} ;
    初始化一个空的团池 C_pool;
    对于 {u, v} ∈ E，执行
        |C| ← 最大团(G[N(u) ∩ N(v)]);
        如果 |C| ≤ lb − 3 那么
            E ← E \ {{u, v}};
        else
            C_pool ← C_pool ∪ {C ∪ {u, v}};
    返回 G, C_pool
```

4.3. 上界启发式

现在，我们将介绍一种启发式过程，用于计算 $\eta(G, k)$ 的上界。所提出启发式方法的伪代码见算法 3。该方法的基本观察是，对于任意满足 $|F| \leq |E|$ 的边集 $F \subseteq E$，移除 F 后图的团数可作为 $\eta(G, k)$ 的上界。逐条移除边并在每次删除后重新计算最大团，会导致最大团计算次数过多，难以承受；而直接移除 k

来自 G 最大团的边可能很容易错过高质量解。为平衡解的质量与计算效率，我们采用一种自适应策略，确定每次迭代移除多少条边。具体而言，在初始迭代 $i = 0$中，我们计算输入图的最大团C_0 =MaxClique(G)，并将待移除的边数设为 $r = 1$。在迭代 $i \geq 1$中，我们首先计算当前图的最大团C_{i+1}。根据以下规则更新 r 的值。如果与前一次迭代相比团数没有下降，即 $|C_{i+1}| = |C_i|$，则将待移除的边数增加，设为 $r = \min(2r,\ |C_i|,\ k)$。否则，如果 $|C_{i+1}| < |C_i|$，则将 r 重置为 1。

随后，从当前图中均匀随机地选取大小为 r 的边集 $F \subseteq E(C_i)$并将其移除。剩余阻断预算更新为 $k \leftarrow k - r$，并继续执行该过程，直至预算耗尽。

算法 3： 用于估计上界的算法

输入： 图 $G = (V,\ E)$，整数 $k \geq 0$

输出： $\eta(G,\ k)$ 的一个上界。

EstimateUB(G, k)

开始

  $i \leftarrow 0$, $C_i \leftarrow$ 最大团(G) $r \leftarrow 1$;

  当 $k > 0$ **时**

    $E_{\text{del}} \leftarrow$ 随机选择 r 条边，从 $E(C_i)$;

    从图 G 中移除边集 E_{del} ;

    $C_{i+1} \leftarrow$ 最大团$(G = (V, E \setminus E_{\text{del}}))$;

    if $|C_{i+1}| = |C_i|$ **则**

      $r \leftarrow \min(2r, |C_{i+1}|, k)$;

    else

      $r \leftarrow 1$;

    $k \leftarrow k - r,\ \ i \leftarrow i + 1$;

  返回 $|C_{i-1}|$

4.4. 完整的 *RLCM* 算法

现在，我们将此前介绍的各个组件整合到 RLCM 中。RLCM 是一种用于求解 EICP 的精确算法，其总体结构总结于算法 4。该算法首先进入预处理阶段，对输入图进行缩减，并初始化一个团池 {v1}。

利用该团池，通过仅针对 $\mathcal{C}_{pool}$ 中的团加入不等式 (4)，并结合额外的基于排列的不等式来强化松弛，从而构建一个部分 EBCP-MILP（ P-EBCP-MILP）。

从初始上界 ub 出发，RLCM 搜索满足 $\gamma(G,p) \leq k$ 的最小参数 p。对于给定的 p，P-EBCP-MILP 的最优值为 $\gamma(G,p)$ 提供了一个有效下界。如果该下界超过 k，则可以安全地舍弃当前的 p 值。否则，采用结合惰性约束生成的分支定界方案。每当解在剩余图中诱导出一个大小至少为 p 的团时，就分离相应的阻断不等式和基于排列的不等式，并将其加入模型。当确定了最小可行的 p 值时，该过程终止。

备注 在 RLCM 中，参数 p 从 $ub-1$ 到 lb 依次考察，而不是采用二分搜索策略。原因如下。假设对于给定的 $p=p$，根据当前团池 $\mathcal{C}_{pool}$ 构建的 P-EBCP-MILP 已证明 $\eta(G,k) < p$。随后考察更小的值 $p < p$ 时，所有团 $C \in \mathcal{C}_{pool}$ 必然满足 $|C| > p$，因此可以直接复用同一团池来构建 P-EBCP-MILP。相比之下，二分搜索策略可能需要考察值 $p > p$。此时，现有团池 $\mathcal{C}_{pool}$ 可能包含大小小于 p 的团，这些团无法用于在 P-EBCP-MILP 中构造有效约束，因此需要额外进行团分离。实践中，这使得依次考察的策略比二分搜索策略更高效。

算法 4： 两阶段算法，RLCM

输入： 图 $G=(V,E)$，一个整数 k

输出： $\eta(G,k)$

RLCM(G, k)

开始

  $G, \mathcal{C}_{pool} \leftarrow$ 预处理(G,k);

  通过将 $\mathcal{C}$ 松弛为 $\mathcal{C}_{pool}$来构建 EBCP-MILP，并将该模型记为 P-EBCP-MILP;

  对于每个 $C \in \mathcal{C}_{pool}$，**执行**

    随机生成一个排序 $\overrightarrow{C}$ 及其逆序 $\overrightarrow{C'}$。然后，添加额外不等式（5）–（8），这些不等式由.....构造 $\overrightarrow{C}$ 和 $\overrightarrow{C'}$ 到 P-EBCP-MILP 模型;

  $ub \leftarrow$ EstimateUB$(G,\ k)$;

  $p \leftarrow ub-1$;

  当$true$ **时执行**

    使用混合整数线性规划求解器求解 P-EBCP-MILP，令最优值为 $\hat{}\gamma(G,\ p)$，最优解为 $\hat{}x$;

    若 $\hat{}\gamma(G,\ p) \leq k$ **则**

      令 F 为满足 $\hat{}x_e = 1$ 的边集 e ;

      $C \leftarrow$ 最大团$(G=(V,\ E \setminus F))$;

      若 $|C| < p$ **则**

        $p \leftarrow |C|-1$;

        重建 P-EBCP-MILP，其中 $\mathcal{C}$ 是 $\mathcal{C}_{pool}$

      将不等式 $\sum_{e \in E(C)} x_e \geq \gamma_{clq}(|C|, p)$ 添加到 P-EBCP-MILP;

      随机生成一个排序 $\overrightarrow{C}$ 及其逆序排序 $\overrightarrow{C'}$;

      添加由以下内容构造的额外不等式（5）–（8）：$\overrightarrow{C}$ 和 $\overrightarrow{C'}$ 至 P-EBCP-MILP模型;

    else

      返回 $p+1$

5. 实验

5.1. 实验设置

本节评估所提出的算法 RLCM 的性能。所有实验均在运行 Ubuntu 22.04 的服务器上进行，该服务器配备 Intel Xeon® Platinum 8360Y CPU（2.40GHz）和 1TB RAM。该算法使用 C语言++ 编写，并采用 g++ 10.5.0 编译，启用了 -O3 优化标志。所有混合整数线性规划均使用 IBM CPLEX 22.1.1 求解，采用单线程和默认设置。

5.2. 基准算法与数据集

我们将所提出的算法 RLCM 与两种用于求解 EICP 的现有方法进行比较：

- BILEVEL，一种先进的通用混合整数双层优化求解器 [9, 10]。

- EDGE-INTER，专为 EICP [12] 设计的最新精确算法。

此外，为确保公平比较，我们在调用 BILEVEL 之前对输入图应用预处理过程。因此，BILEVEL 和 RLCM 均在相同的缩减实例上运行。我们在三个结构特性各异、来源不同的基准数据集上评估这些算法。

- **DIMACS2：** 我们选取了第二届 DIMACS 最大团问题实现挑战中的 16 个图，每个图均包含$|V| = 200$。这些实例已广泛用于最大团算法的相关文献 [3, 28, 31] ，并且也在 EDGE-INTER 中进行了测试。

- **大型稀疏图：**我们使用了来自网络资源库的 95 个大型稀疏图 [22], ，每个图最多包含 1 000000 条边。这些图涵盖了种类繁多的真实世界网络，包括合著、引文、街道、聚类、几何和数值模拟网络。

- **随机图：**我们生成了具有 $|V| \in\{25, 50, 75, 100\}$ 且边密度为 $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,0.9, 0.95\}$ 的 Erdős–Rényi 图。对于每种 $(|V|, \rho)$ 组合，我们使用不同的随机种子生成 5 个图，共得到 $4 \times 10 \times 5 = 200$ 个合成图。

对于每个 DIMACS2 实例，我们测试集合 $\{10, 15, 20, 25, 30\}$ 中的五个阻断预算 k 。对于大型真实世界图，我们同时考虑五个固定值 $k \in \{10, 15, 20, 25, 30\}$，以及五个与边数成比例的值，即 $c \in \{0.0001, 0.0005, 0.0010, 0.0015, 0.0020\}$ 对应的 $k = \lceil c \cdot |E| \rceil$ 。对于每个随机图，我们评估相对于 $|E|$的十二个 k 值，具体为 $c \in \{0.01, \ldots, 0.10, 0.15, 0.20\}$ 对应的 $k = \lceil c \cdot |E| \rceil$ 。每项实验的时间限制均为 600 秒。计算平均运行时间时，超时的实例按达到时间限制计，即 600 秒。最后，我们指出，DIMACS2 和随机图实例的实验设置与 EDGE-INTER [12] 中使用的设置一致。

5.3. DIMACS2 的结果

表1：EDGE-INTER 和 RLCM 在 DIMACS2 上的 CPU 时间（秒）比较。

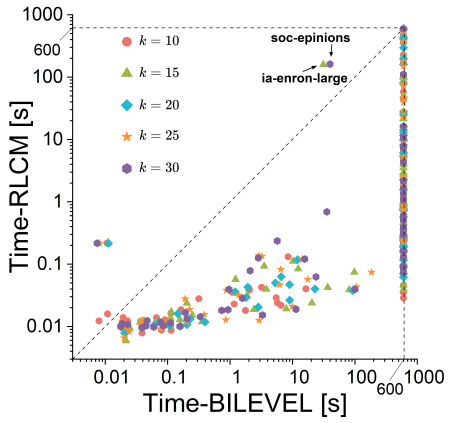
		$k=10$			$k=15$			$k=20$			$k=25$			$k=30$	
实例	$\omega(G)$	Ed-In.	RLCM	η	Ed-In.	RLCM	η	Ed-In.	RLCM	η	Ed-In.	RLCM	η	Ed-In.	RLCM	η
brock200_1	21	33.1	**13.5**	19	**21.3**	27.1	19	t.l	**19.9**	19	-	**481.8**	19	-	t.l	-
brock200_2	12	0.6	0.6	10	0.8	**0.7**	10	2.7	**0.7**	10	-	**8.7**	9	-	**8.5**	9
brock200_3	15	**1.4**	2.3	13	6.2	**1.9**	13	9.8	**2.0**	13	-	**2.1**	13	-	**396.9**	12
brock200_4	17	**4.0**	4.1	15	9.1	**4.4**	15	**19.0**	21.4	14	-	**21.0**	14	-	**21.2**	14
c-fat200-1	12	0.1	0.1	11	0.1	0.1	10	0.1	0.1	10	-	**0.1**	10	-	**0.1**	10
c-fat200-2	24	**0.2**	0.3	22	**0.2**	0.4	21	**0.2**	0.6	20	-	**0.4**	20	-	**2.2**	19
c-fat200-5	58	t.l	**5.9**	55	t.l	**9.4**	53	t.l	**8.4**	52	-	**16.3**	50	-	**20.8**	49
gen200_p0.9_44	44	t.l	t.l	-	t.l	t.l	-	**21.9**	t.l	16	-	t.l	-	-	t.l	-
gen200_p0.9_55	55	t.l	**26.7**	45	t.l	t.l	-	*t.l*	t.l	-	-	t.l	-	-	t.l	-
san200_0.7_1	30	**5.5**	7.7	20	119.4	**65.2**	17	*t.l*	**25.9**	17	-	**26.6**	17	-	**30.5**	17
san200_0.7_2	18	**0.7**	2.5	15	2.7	**2.6**	15	*t.l*	**2.6**	15	-	t.l	-	-	**306.6**	14
san200_0.9_1	70	**66.2**	121.7	60	t.l	t.l	-	*t.l*	t.l	-	-	t.l	-	-	t.l	-
san200_0.9_2	60	**66.5**	70.1	50	t.l	**39.2**	45	**55.6**	t.l	$15^{?}$	-	t.l	-	-	t.l	-
san200_0.9_3	44	t.l	**181.5**	36	t.l	t.l	-	*t.l*	t.l	-	-	t.l	-	-	t.l	-
sanr200_0.7	18	15.4	**6.0**	17	**20.0**	34.7	16	*t.l*	**25.5**	16	-	**26.2**	16	-	**31.3**	16
sanr200_0.9	42	t.l	t.l	-	t.l	t.l	-	*t.l*	t.l	-	-	t.l	-	-	t.l	-
#opt		11	**14**		9	**11**		7	**10**		-	**9**		-	**9**	

表1报告了 RLCM 和 EDGE-INTER 的计算时间。表中未列出求解器 BILEVEL，因为它未能在 600秒时间限制内求解任何 EICP实例。

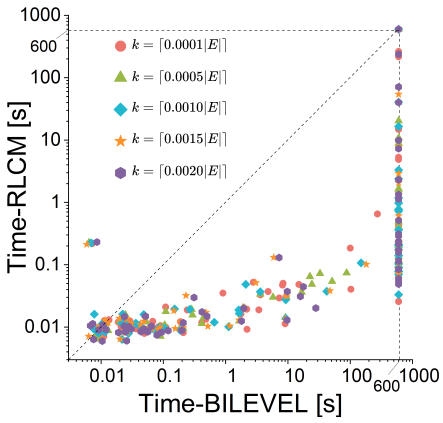
EDGE-INTER 的源代码使用 C语言 实现，但无法在我们的平台上成功编译。尽管与作者进行了沟通，我们仍未能获得完整且可用的代码版本。与作者沟通后，我们仍未能获得完整的源代码。因此，EDGE-INTER 的实验结果直接取自 [12]。 [12] 中的实验设置与我们的设置一致。此外，他们的实验是在一台配备 3.40 GHz、8 核 Intel i7-3770 处理器的机器上进行的，其时钟频率约比我们的平台高 1.4 倍。因此，我们认为 EDGE-INTER 报告的结果可以与我们的结果进行公平、直接的比较。

在表1中，η 表示最优值 $\eta(G,\ k)$。若超过时间限制，则显示“t.l”；“-”表示由于源代码不可用而无法评估 EDGE-INTER，或没有任何算法能在时间限制内计算出 $\eta(G,\ k)$ 的实例。对于 $k = 20$处的若干 EDGE-INTER 结果，我们用“?”标记原始论文中看似不一致的数值。例如，对于实例 san200_0.9_2，EDGE-INTER 报告了 $\eta(G,\ 15) = 45$ 和 $\eta(G,\ 20) = 15$，这是不可能的，因为再移除五条边，不可能将一个大小为45的团缩减到大小为15。表格最后一行报告了每种算法解决的 EICP实例总数。

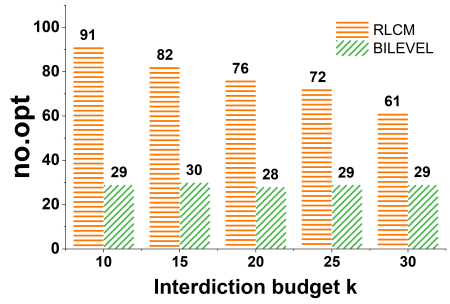
从表1可以看出，对于每个测试的 k值，RLCM解决的实例都比竞争算法更多。在许多情况下，EDGE-INTER未能在时间限制内返回解，而RLCM则能在600秒内成功解决相应实例。相反，据报道，只有两个实例 gen200_p0.9_44 和 san200_0.9_2 with $k = 20$由 EDGE-INTER 成功解决，但 RLCM 未能解决。值得注意的是，后一种情况下 EDGE-INTER 报告的解似乎值得怀疑。总体而言，即使预算较大（如 $k = 25$ 或30），RLCM在这些困难的团实例上仍具有良好的可扩展性。进一步检查求解过程可知，预处理阶段无法缩减该数据集输入图的规模。因此，BILEVEL无法有效处理这些实例也就不足为奇了。



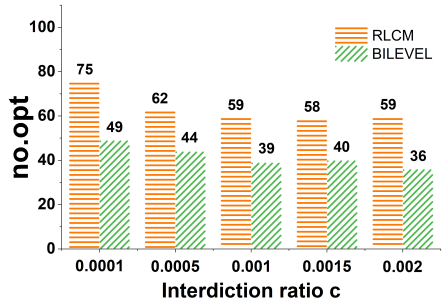
(a) 每个实例在不同固定 k 值下的运行时间。



(b) 每个实例在不同 $k = \lceil c \cdot |E| \rceil$ 值下的运行时间。



(c) 每个实例在不同固定 k 值下的最优解数量。



(d) 每个实例在不同 $k = \lceil c \cdot |E| \rceil$ 值下的最优解数量。

图3：两种算法在不同 k 值下的性能。

5.4. 大型稀疏图的结果

图3比较了在不同 k值下，RLCM和BILEVEL在大型稀疏图上的性能。如前所述，由于无法获得完整的源代码及相应的实验数据，此处未将EDGE-INTER纳入比较。图3(a)和(b)中的每个点都对应一个EICP实例（G, k）。点的坐标表示两种算法在同一个实例上的运行时间。位于对角线下方的点表示RLCM比BILEVEL更快的实例，而位于对角线上方的点则表示相反的情况。图3(c)和(d)总结了对于不同 k值，每种算法解决的实例数量。

对于所有测试的 k取值，RLCM 和 BILEVEL 均在时间限制内求解了 695 个实例。在这些实例中，就计算时间而言，RLCM 在 664 个实例上优于 BILEVEL。BILEVEL 的优势主要体现在较简单的实例上，这些实例约可在 0.1 秒内求解。此外，即使采用相同的预处理过程，BILEVEL 仍因内存消耗过高而无法求解 180 个实例。相比之下，RLCM 能将其中 113 个实例求解至最优。

5.5. 评估约简、上界估计与增强不等式

图约简、上界估计以及采用增强不等式的分支割是 RLCM 框架的关键组成部分。在本小节中，我们通过消融研究评估这些组成部分各自的影响。我们考虑 RLCM 的三个消融变体：

- NRed，禁用所有图归约规则；
- NUB，其移除了上界估计过程；以及
- NCon，其省略了第3.3节在割生成过程中引入的额外不等式（5）–（8）。

图4报告了在大型稀疏图集合上进行的这项研究结果，因为图约简主要适用于大型稀疏图。该图展示了每种算法变体的运行时间箱线图，其中每个EICP实例均以一个点表示。此外，图的顶部还报告了每种变体在时间限制内求解的实例数量。

结果清楚地表明，与其他变体相比，不进行约简的变体 NRed 的 CPU 时间显著更长，且始终能以最优性求解的问题实例更少。这一观察凸显了图约简在 RLCM 框架中求解大型稀疏图时的关键作用。

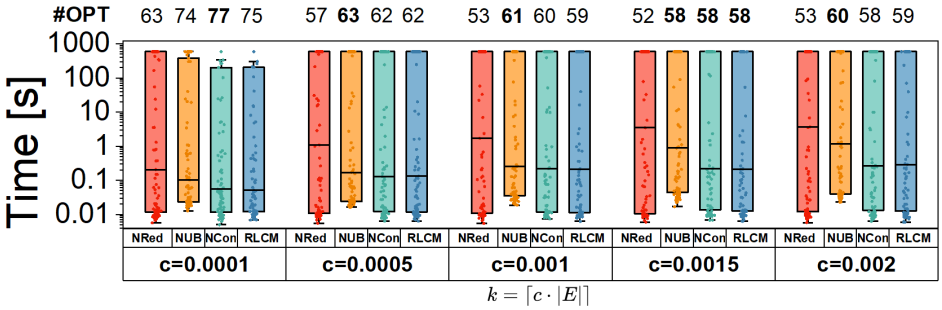
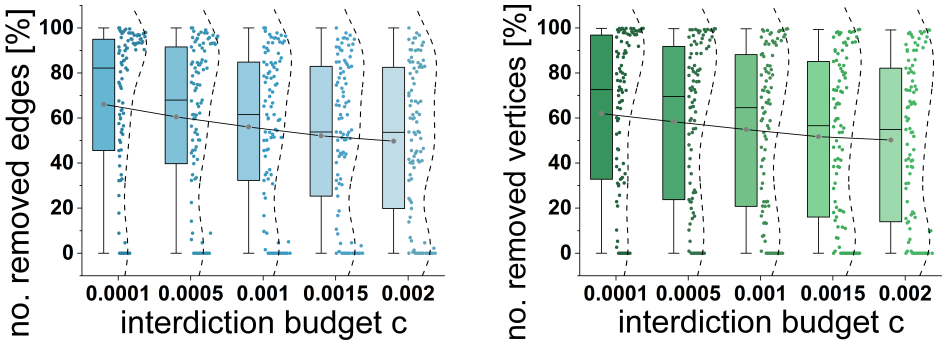


图4：RLCM及其变体在大型稀疏图上的统计CPU时间。

为进一步说明预处理阶段的有效性，图5报告了在图约简过程中从大型稀疏图中移除的顶点和边的比例。在该数据集中，约简实现了大幅度的精简，从而显著简化了问题实例。不过，我们观察到，随着阻断预算 k 增加，约简的有效性会降低。我们推测，这种现象主要是因为对于更大的 k值， η 上的下界（G， k）会逐渐变得不再那么紧。



(a) 不同 $k = \lceil c \cdot |E| \rceil$ 下每个实例移除的边的百分比。

(b) 不同 $k = \lceil c \cdot |E| \rceil$ 下每个实例移除的顶点的百分比。

图5：大型稀疏图上的约简效果。请注意，每个点代表一个图。

关于上界估计，不含这一组成部分的变体 NUB 在测试的所有 k 值中，除 $k = \lceil 0.0001|E| \rceil$ 外，解决的实例都更多。然而，仔细考察各实例的运行时间后，会发现情况有所不同。尽管 NUB 总体上解决的实例略多，但就单个实例而言，它通常效率较低。具体来说，在两种算法都能解决的实例中，NUB 所需的

表2：DIMACS2 数据集上不同 k 的 RLCM 和 NCon 计算时间比较。

		$k=10$			$k=15$			$k=20$			$k=25$			$k=30$		
实例	$\omega(G)$	NCon	RLCM	η	NCon	RLCM	η	NCon	RLCM	η	NCon	RLCM	η	NCon	RLCM	η
brock200_1	21	13.7	**13.5**	19	**20.1**	27.1	19	**18.2**	19.9	19	t.l	**481.8**	19	t.l	t.l	-
brock200_2	12	0.6	0.6	10	0.7	0.7	10	0.7	0.7	10	**7.6**	8.7	9	**5.0**	8.5	9
brock200_3	15	2.7	**2.3**	13	1.9	1.9	13	2.0	2.0	13	2.1	2.1	13	t.l	**396.9**	12
brock200_4	17	4.2	**4.1**	15	4.5	**4.4**	15	23.4	**21.4**	14	22.2	**21.0**	14	30.4	**21.2**	14
c-fat200-1	12	0.1	0.1	11	0.1	0.1	10	0.1	0.1	10	0.1	0.1	10	0.1	0.1	10
c-fat200-2	24	0.3	0.3	22	**0.3**	0.4	21	**0.4**	0.6	20	0.6	**0.4**	20	**0.8**	2.2	19
c-fat200-5	58	6.1	**5.9**	55	**6.4**	9.4	53	**7.8**	8.4	52	**16.1**	16.3	50	153.9	**20.8**	49
gen200_p0.9_44	44	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-
gen200_p0.9_55	55	54.2	**26.7**	45	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-
san200_0.7_1	30	17.6	**7.7**	20	81.2	**65.2**	17	28.3	**25.9**	17	34.5	**26.6**	17	33.5	**30.5**	17
san200_0.7_2	18	3.0	**2.5**	15	**2.5**	2.6	15	2.6	2.6	15	t.l	t.l	-	**191.4**	306.6	14
san200_0.9_1	70	**16.3**	121.7	60	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-
san200_0.9_2	60	367.8	**70.1**	50	t.l	**39.2**	45	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-
san200_0.9_3	44	465.5	**181.5**	36	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-
sanr200_0.7	18	**5.9**	6.0	17	**29.8**	34.7	16	33.8	**25.5**	16	**20.6**	26.2	16	**27.9**	31.3	16
sanr200_0.9	42	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-	t.l	t.l	-

时间比 RLCM 多的实例有 290 个，而 RLCM 速度较慢的实例仅有 28 个。

接下来，我们将 RLCM 与省略附加不等式的变体 NCon 进行比较。在大型稀疏图数据集上，这两种算法在求解实例数量和计算时间中位数方面表现相当。不过，这一观察结果并不意味着附加不等式不重要，因为它们在稠密且具有挑战性的实例上影响更为明显。为突出这一效果，表2报告了 RLCM 和 NCon 在稠密 DIMACS2 图上的运行时间。在这些实例上，RLCM 明显占优，比 NCon 多求解了三个未能在时间限制内完成的 EICP 实例，分别为 san200_0.9_2for $k=15$、brock200_1 for $k=25$ 和 brock200_3 for $k=30$。

5.6. 随机图上的结果

表3总结了 Erdős–Rényi 随机图上的实验结果，并按不同的 k 值分组。每组包含 200 个实例，顶点数量和边密度各不相同。对于每组，我们报告了获得最优解的实例平均数量（#opt），以及计算最优解所需的平均运行时间。总体而言，在该数据集上，无论是已求解实例的数量

还是运行时间，RLCM 始终优于另外两种算法。当 $k \geq \lceil 0.10 \cdot |E| \rceil$ 时是个例外：EDGE-INTER 在 #opt 方面略占优势。

表3：按不同的 k 分组的算法性能比较，其中 $k = \lceil c \cdot |E| \rceil$。

$k = \lceil c \cdot \lvert E \rvert \rceil$	BILEVEL		EDGE-INTER		RLCM	
c	最优值数量	时间（秒）	#opt	时间（秒）	最优值数量	时间（秒）
0.01	18	340.8	26	210.5	**32.0**	**131.1**
0.02	15.2	386.2	25	234.8	**27.0**	**197.1**
0.03	13.4	409.3	22	282.2	**24.4**	**241.7**
0.04	11.6	435.0	22	287.6	**24.8**	**236.7**
0.05	11.4	437.2	22	300.2	**24.0**	**242.3**
0.06	9.6	462.5	18	348.7	**21.6**	**280.8**
0.07	8.8	474.0	17	362.8	**20.0**	**302.3**
0.08	8.6	476.7	18	357.0	**18.8**	**320.1**
0.09	9.2	468.7	17	364.7	**18.8**	**325.2**
0.10	9	474.1	**19**	345.4	18.4	**333.4**
0.15	8	485.1	**17**	397.1	16	**367.0**
0.20	9	475.2	**17**	383.1	16.6	**362.3**

6. 结论与未来工作

本文研究了边干预团问题（EICP）的混合整数线性规划公式及精确求解方法。该问题旨在从图中删除至多 k 条边，以最小化其团数。该问题在计算上具有挑战性，且属于超出 NP 的复杂性类。

我们将 EICP 重新表述为一系列参数化边阻断团问题（EBCP），并通过引入利用不同团排序的新有效不等式，构建了强化的混合整数线性规划公式。在这些公式的基础上，我们提出了 RLCM，这是一种两阶段精确算法，整合了图约简技术、上界估计和迭代分支割框架。这些组件共同构成了一种在实践中可行的 EICP 求解方法。

大量计算实验表明，所提出的方法在具有不同结构特性的图上兼具高效性与稳健性。在 DIMACS2 基准实例上，RLCM 在 600秒时间限制内求解的 EICP实例多于现有方法，并成功解决了若干此前未能解决的案例。对于大型稀疏真实世界网络，该算法展现出很强的可扩展性，而通用双层求解器往往因内存消耗过大而无法求解。针对随机图的补充实验进一步证实了 RLCM 的适应能力

能够适应不同的图密度和问题规模。最后，消融研究突显了图约简、上界估计以及所提出的不等式各自对算法整体性能的贡献。

未来研究有几个值得探索的方向。首先，针对其他稠密子图概念的阻断问题值得进一步研究。团在模拟真实系统时往往限制过多，而 s-plexes [30]和 s-bundles [32] 等松弛结构则能更真实地刻画稠密社群。[29] 近期的研究通过考察准团——边密度高于给定阈值的子图——开启了这一研究方向；在同一框架下探讨更广泛的松弛结构也顺理成章。另一个方向是将本文提出的归约和建模思路应用于时序网络，因为其中的稠密结构会随时间演变。此外，阻断操作还可扩展至边删除或添加之外；基于顶点的修改（包括加权顶点添加或删除） [11], 的影响仍基本未经探索。

参考文献

[1] Anand, R., Sarmah, D. T., 和 Chatterjee, S.（2018）。利用时间连通网络提取参与疾病进展的蛋白质。*BMC 系统生物学*，12(1):78。[2] Becker, T.（2017）。双层团阻断及相关问题。博士论文，莱斯大学。[3]Chang, L.（2019）。大型稀疏图上的高效最大团计算。载于第*25*届 *ACM SIGKDD* 知识发现与数据挖掘国际会议论文集，KDD ’19，第 529–538 页，美国纽约州纽约市。计算机协会。[4] Chang, L.（2023）。通过改进时间复杂度高效计算最大 k 缺陷团。*ACM 数据管理会议录*，1(3):1–26。[5] Chen, X., Zhou, Y., Hao, J.-K., 和 Xiao, M.（2021）。计算超大规模图中的最大 k 缺陷团。《计算机与运筹学》，127:105131。

[6] Dai, Q., Li, R., Cui, D., 和 Wang, G.（2024）。理论与实践上均高效的最大缺陷团搜索。*ACM* 数据管理会议录，2(4):1–27。[7] Dempe, S.（2020）。双层优化：理论、算法、应用与文献目录。双层优化：进展与未来挑战，第 581–672 页。[8] Downey, R. G. 和 Fellows, M. R.（2012）。参数化复杂性。施普林格科学与商业媒体。[9]Fischetti, M., Ljubić, I., Monaci, M., 和 Sinnl, M.（2017）。一种用于混合整数双层线性规划的通用新算法。《运筹学》，65(6):1615–1637。[10] Fischetti, M., Monaci, M., 和 Sinnl, M.（2018）。广义阻断问题的动态重构启发式算法。《欧洲运筹学杂志》，267(1):40–51。[11] Furini, F., Ljubić, I., Martin, S., 和 San Segundo, P.（2019）。最大团阻断问题。《欧洲运筹学杂志》，277(1):112–127。[12] Furini, F., Ljubić, I., San Segundo, P., 和 Zhao, Y.（2021）。边干预团问题的分支割平面算法。《欧洲运筹学杂志》，294(1):54–69。[13] Gao, J., Xu, Z., Li, R., 和 Yin, M.（2022）。一种针对超大规模稀疏图中最大 k-缺陷团问题、具有新上界的精确算法。载于 *AAAI* 人工智能会议论文集，第 36 卷，第 10174–10183 页。[14] Gschwind, T., Irnich, S., 和 Podlinski, I.（2018）。重新探讨最大权松弛团与俄罗斯套娃搜索。《离散应用数学》，234:131–138。[15] Hastad, J.（1996）。团问题难以在 n/sup 1-/spl epsiv 内近似。载于第*37*届计算机科学基础会议论文集，第 627–636 页。IEEE。

[16] Li, C.-M., Jiang, H. 和 Manyà, F.（2017）。关于最大团问题分支定界算法中分支数量的最小化。《计算机与运筹学》，84：1–15。[17] Li, C. M. 和 Quan, Z.（2010）。一种基于 MaxSAT 的高效最大团问题分支定界算法。载于*AAAI*，第 10 卷，第 128–133 页。[18] Luo, C., Zhou, Y., Wang, Z. 和 Xiao, M.（2024）。一种更快的最大 k-缺陷团问题分支算法。载于*ECAI 2024*，第 4132–4139 页。IOS Press。[19] Mahdavi Pajouh, F.（2020）。最小成本边阻断团问题。《运筹学年鉴》，294(1)：345–376。[20] Mattia, S.（2024）。通过图映射重新表述团禁阻问题及其复杂性。《离散应用数学》，354：48–57。第 18 届科隆—特温特图与组合优化研讨会（CTW 2020）。[21] Pattillo, J., Youssef, N. 和 Butenko, S.（2013）。网络分析中的团松弛模型。《欧洲运筹学杂志》，226(1)：9–18。[22]Rossi, R. A. 和 Ahmed, N. K.（2015）。包含交互式图分析与可视化功能的网络数据仓库。https:// networkrepository.com/。访问日期：2025年1月12日。[23] Samudrala, R. 和 Moult, J.（1998）。一种用于蛋白质结构比较建模的图论算法[11]（由 F. Cohen 编辑）。《分子生物学杂志》，279(1)：287–302。[24] San Segundo, P. 和 Artieda, J.（2015）。一种用于视觉特征匹配问题的新型团公式。《应用智能》，43(2)：325–342。[25] San Segundo, P., Furini, F., Álvarez, D. 和 Pardalos, P. M.（2023）。Clisat：一种用于困难最大团问题的新型精确算法。《欧洲运筹学杂志》，307(3)：1008–1025。[26] Stentiford, F.（2014）。通过检测点匹配团进行人脸识别。载于《图像处理：机器视觉应用*VII*》，第 9024 卷，第 148–158 页。SPIE。

[27] 图兰, P.（1941）。关于图论中的一个外部问题。*Mat. Fiz.Lapok*，48：436–452。[28] Wu, Q. 和 Hao, J.-K.（2015）。最大团问题算法综述。《欧洲运筹学杂志》，242(3)：693–709。[29] Zhong, H., Mahdavi Pajouh, F., Butenko, S. 和 Prokopyev, O. A.（2024）。网络中稠密簇的阻断问题。*INFORMS*《计算杂志》。[30] Zhou, Y. 和 Hao, J.-K.（2017）。用于最大 s-plex 问题的频率驱动禁忌搜索。《计算机与运筹学》，86：65–78。[31] Zhou, Y., Hu, S., Xiao, M. 和 Fu, Z.-H.（2021）。通过二阶约简和图着色界定改进最大 k-plex 求解器。载于*AAAI* 人工智能会议论文集，第 35 卷，第 12453–12460 页。[32] Zhou, Y., Lin, W., Hao, J.-K., Xiao, M. 和 Jin, Y.（2022）。一种求解最大 s-bundle 问题的有效分支定界算法。《欧洲运筹学杂志》，297(1)：27–39。[33] Zhu, C., Zhou, Y. 和 Jiang, H.（2025）。一种用于团禁阻问题的基于约简的算法。