



# Paw 图 – 边删除问题的线性顶点核心化算法

盛子默, 肖鸣宇\*

电子科技大学计算机科学与工程学院, 成都 611731

\* 通信作者. E-mail: myxiao@uestc.edu.cn

收稿日期: 2023-12-31; 修回日期: 2024-02-24; 接受日期: 2024-03-28; 网络出版日期: 2024-07-09

国家自然科学基金 (批准号: 62372095, 61972070) 资助项目

**摘要** 图边删除问题中一类重要问题是研究是否可以删除图中不超过  $k$  条边之后使得剩余的图不存在某个子图结构  $H$ , 而子图  $H$  为顶点个数不超过 4 的连通图的情况被研究得最为广泛. 本文主要考虑  $H$  为 Paw 图 (三角形其中一个顶点再邻接一条边) 的情况, 称为 Paw 图 – 边删除问题, 并为该问题设计了一个  $32k$  个顶点的问题核. 这是该问题的第 1 个线性顶点大小的问题核. 文中主要的技术是结合两个新的皇冠分解的变体来分析图的结构从而对图进行简化.

**关键词** 图算法, 核心化算法,  $H$ -边删除问题, Paw 图 – 边删除问题, 皇冠分解技术

## 1 引言

给定一个简单无向图  $G$ , 一个最基础的问题是能否通过删掉  $k$  条边, 使得图具有某种性质  $\mathcal{F}$ . 这一问题通常被称为图边删除问题 (edge deletion problem). 具体而言, 如果性质  $\mathcal{F}$  为图  $G$  不包含与  $H$  同构的子图, 该问题被称为  $H$ -边删除问题 ( $H$ -edge covering problem); 如果性质  $\mathcal{F}$  为图  $G$  不包含与  $H$  同构的导出子图 (induced subgraph), 该问题被称为导出  $H$ -边删除问题 ( $H$ -free edge deletion problem).

在 (导出)  $H$ -边删除问题中, 当  $H$  包含最多 2 个顶点的时候, 问题很平凡; 在  $H$  包含 3 个或者 4 个顶点的时候, 出现一些结构使得问题变得很难了; 而在  $H$  包括 5 个或者更多顶点的时候,  $H$  图的情况太多并且许多  $H$  的结构过于特殊, 因此现有研究通常关注  $|H| \leq 4$  且  $H$  为连通图 (connected graph) 的情况. 图 1 中列举了  $|H| \leq 4$  的各种连通图, 其中 Diamond 是从  $K_4$  中删去任意一条边形成的子图; Paw 是三角形其中一个顶点再邻接一条边形成的子图; Claw 是一个点邻接三条边形成的子图, 即  $K_{1,3}$ . 事实上, 导出  $H$ -边删除问题是 NP-难的当且仅当图  $H$  中至少存在两条边<sup>[1]</sup>. 但是非导出版本的  $H$ -边删除问题却不同, 比如说 Claw 包含三条边, 但是 Claw-边删除问题就是多项式可

**引用格式:** 盛子默, 肖鸣宇. Paw 图 – 边删除问题的线性顶点核心化算法. 中国科学: 信息科学, 2024, 54: 1604–1619, doi: 10.1360/SSI-2023-0418  
Sheng Z M, Xiao M Y. A linear vertex kernel for the Paw edge covering problem (in Chinese). Sci Sin Inform, 2024, 54: 1604–1619, doi: 10.1360/SSI-2023-0418

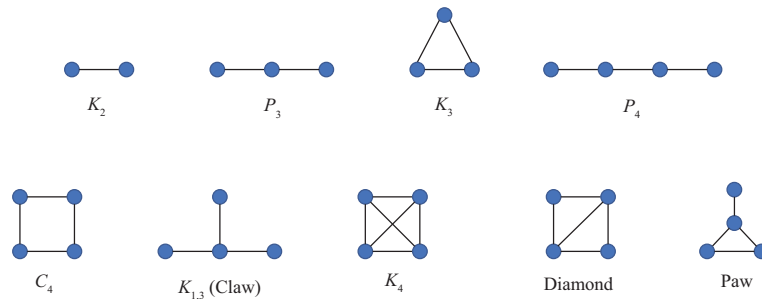
图 1 (网络版彩图)  $|H| \leq 4$  的各种连通图

Figure 1 (Color online) Connected graphs on at most 4 vertices

解的<sup>[2]</sup>. 由于(导出) $H$ -边删除问题的困难性, 这两个问题在精确算法(exact algorithm)、参数算法(parameterized algorithm)、核心化算法(kernelization algorithm)、近似算法(approximation algorithm)等领域得到深入研究<sup>[3~6]</sup>. 本文考虑 $H$ 为Paw图时候的 $H$ -边删除问题的核心化算法. 核心化算法是参数算法领域中的热门研究分支, 可以看做一种高效的预处理算法, 在算法理论研究非常火热.

导出 $H$ -边删除问题的核心化算法研究较多. 不久前, Crespelle等<sup>[7]</sup>在综述中阐述了导出 $H$ -边删除问题的核心化算法研究现状. 当 $|H| = 3$ 时, 导出 $P_3$ -图边删除问题存在 $2k$ 个点的线性核<sup>[8]</sup>. 当 $|H| = 4$ 时, 导出 $C_4$ -图边删除问题没有多项式核<sup>[9]</sup>; 而当 $H$ 为 $P_4$ ,  $K_4$ , Diamond, 或者Paw的时候, 导出 $H$ -边删除问题均存在多项式核<sup>[10~13]</sup>; 导出Claw-图边删除问题是否存在多项式核依然是一个公开问题. 当 $|H| \geq 5$ 时, Marx和Sandeep<sup>[14]</sup>猜想若 $H$ 不为完全图或空图, 在 $\text{coNP} \not\subseteq \text{NP/poly}$ 假设下任意导出 $H$ -边删除问题都不具有多项式核. 他们给出了9种特殊的 $H$ , 如果这9种 $H$ 的导出 $H$ -边删除问题不存在多项式核, 则该猜想是正确的.

对于非导出版本的 $H$ -边删除问题的核心化算法, 研究相对滞后一些. 表1中列举了一些 $H$ -边删除问题的现有核心化算法研究, 其中“P”意味着对应问题是多项式可解的, 故不讨论其核心化算法. 当 $|H| = 3$ 时,  $K_3$ -边删除问题即三角形-边删除问题存在 $3k$ 个点的线性核<sup>[15]</sup>. 当 $|H| = 4$ 时, Claw-边删除问题是多项式可解的<sup>[2]</sup>;  $K_4$ -边删除问题存在 $O(k^3)$ 个点的多项式核<sup>[11]</sup>;  $C_4$ -边删除问题存在 $6k^2$ 个点的多项式核<sup>[16]</sup>.  $d$ -命中集问题( $d$ -hitting set problem)是一个经典的NP-难问题, 其问题描述为给定一些大小不超过 $d$ 的集合, 判定是否能够用 $k$ 个元素“命中”所有集合, 使得每一个集合中至少存在一个被选中的元素.  $d$ -命中集问题存在一个大小为 $O(k^{d-1})$ 个元素的多项式核<sup>[17]</sup>. Paw图-边删除问题、Diamond图-边删除问题和 $P_4$ -边删除问题都是NP-难的, 虽然目前没有专门针对它们设计的核心化算法, 但是利用 $d$ -命中集问题 $O(k^{d-1})$ 的核心化算法<sup>[17]</sup>, 通过把 $H$ -边图删除问题中的边映射为 $d$ -命中集问题的元素;  $H$ -边图删除问题中每个构成 $H$ 的边集映射为 $d$ -命中集问题的一个集合的方式可以推导出以上3个问题大小为 $O(k^{|E(H)|-1})$ 的多项式核.

令 $\mathcal{H}$ 为一个图的集合, 我们类似的定义导出的 $\mathcal{H}$ 图集-边删除问题( $\mathcal{H}$ -free edge deletion problem)为判定能否删掉 $k$ 条边使得图 $G$ 不包含与图集 $\mathcal{H}$ 中任意子图同构的导出子图. 导出的 $\mathcal{H}$ 图集-边删除问题与 $H$ -边删除问题存在天然的联系: 给定一个图 $H$ , 令图集 $\mathcal{H}$ 为一个包含了点集为 $V(H)$ , 边集为 $E(H)$ 的超集的所有子图组成的集合; 则导出的 $\mathcal{H}$ 图集-边删除问题等价于 $H$ -边删除问题. 例如Diamond图-边删除问题等价于导出的 $\{\text{Diamond}, K_4\}$ 图集-边删除问题; Paw图-边删除问题等价于导出的 $\{\text{Paw}, \text{Diamond}, K_4\}$ 图集-边删除问题. 特别地, 如果 $H$ 是完全图, 导出 $H$ -边删除问题与 $H$ -边删除问题等价.

表 1  $|H|$  为不超过 4 个点的连通图时  $H$ -边删除问题现有的最优问题核大小  
 Table 1 Kernels of  $H$  edge covering problem for connected graphs with  $|H| \leq 4$

$H$	Kernels of $H$ -edge covering problems
$P_2$	P
$P_3$	P
$K_3$	$3k$ -vertex <sup>[15]</sup>
$P_4$	$O(k^2)$
$C_4$	$6k^2$ -vertex <sup>[16]</sup>
$K_4$	$O(k^3)$ -vertex <sup>[11]</sup>
Diamond	$O(k^4)$
Paw	$O(k^3)$
Claw	P

本文研究  $H$  为 Paw 图时的  $H$ -边删除问题, 即 Paw 图 – 边删除问题 (Paw edge covering problem). 该问题等价于导出  $\{Paw, Diamond, K_4\}$  图集 – 边删除问题 ( $\{Paw, Diamond, K_4\}$ -free edge deletion problem). 正如前面所说, Paw 图 – 边删除问题存在一个由  $d$ -命中集问题的核心化算法推导而来的大小为  $O(k^3)$  的问题核. 但是目前没有针对这个问题专门设计的核心化算法. 本文深入研究这个问题, 通过分析图中的各种子图结构和引入了新的归约规则 (reduction rule) 设计更为高效的核心化算法. 文中最重要的技术贡献是介绍了两种新的皇冠分解 (crown decomposition) 的变体, 在这两个新技术的帮助下, 最终我们的算法可以将问题压缩到大小不超过  $32k$  个点和  $128k^2 + 20k$  条边, 从而得到了该问题第 1 个线性顶点个数的问题核.

## 2 基本性质

对于简单无向图  $G = (V, E)$ , 本文给出如下符号和定义.  $V(G)$  和  $E(G)$  分别表示  $G$  的顶点集合和边集合. 如果两个顶点之间存在一条边, 则其中一个顶点是另一个顶点的邻居. 顶点  $v$  的邻居集合用  $N(v)$  表示, 对于顶点子集  $V' \subseteq V$ , 我们定义  $N(V') = \cup_{v \in V'} N(v) \setminus V'$  和  $N[V'] = N(V') \cup V'$ . 我们用  $G[V']$  表示点集  $V'$  构成的导出子图. 对图  $G$  中的两个顶点  $u$  和  $v$ , 我们称  $u$  和  $v$  之间的最短路径的长度为  $u$  和  $v$  的距离. 图  $G$  中一个顶点和一个点集的距离为该顶点到点集中所有顶点的最短路径长度的最小值; 两个点集的距离为两个点集之间所有顶点的最短路径的最小值.

一个由 3 个顶点构成的完全图称为三角形. 我们用  $vuw$  表示由顶点  $v, u$  和  $w$  组成的三角形. 如果图  $G$  中存在三角形  $vuw$ , 我们称顶点  $v$  支撑边  $uw$ . 一个 Paw 是由 4 个顶点和 4 条边组成的一个图形, 其中 3 条边组成一个三角形, 另一条边连接三角形中的第 1 个顶点和第 4 个顶点. 一个 Paw 可能会用 4 个顶点  $uvw x$  来表示, 我们总会把三角形中的 3 个点放在前面, 把三角形外的一个点放在最后, 且最后两个点在  $G$  中相邻. 我们称最后两个点所组成的边为 Paw 的尾巴, 我们也称该边是点  $v$  (第 3 个点) 的尾巴.

如果边集  $S$  覆盖了图  $G$  中的所有 Paw, 即图  $G-S$  中任意一个子图均不为 Paw, 我们称  $S$  是  $G$  的一个 Paw 图 – 边删除集. Paw 图 – 边删除问题问题定义如下:

### Paw 图 – 边删除问题:

输入: 一个简单无向图  $G = (V, E)$  和一个正整数  $k$ .

**问题:** 判定图  $G$  中是否存在一个大小小于等于  $k$  的 Paw 图 - 边删除集?

在为 Paw 图 - 边删除问题设计核心化算法时, 我们需要用到填装 (packing) 的概念. 图  $G$  的一个大小为  $k$  的边不相交 Paw (或 Diamond) 填装  $P$  是由  $k$  个两两之间边不相交的 Paw (或 Diamond) 组成的 Paw (或 Diamond) 的集合, 我们用  $V(P)$  表示 Paw (或 Diamond) 填装中出现的点,  $E(P)$  表示 Paw (或 Diamond) 填装中出现的边. 本文所提到的所有填装均仅要求边不相交, 即填装中的两个 Paw (或 Diamond) 允许共用某些顶点, 但不能共用任意一条边.

核心化算法是参数算法领域中的一种高效预处理算法, 其定义如下: 给定一个问题的实例  $(I, k)$ , 一个核心化算法在多项式时间内输出相同问题的等价实例  $(I', k')$ .  $(I, k)$  是一个“是”实例当且仅当  $(I', k')$  是一个“是”实例, 其中  $k' \leq k$  且  $|I'| \leq g(k)$ ,  $g$  是某个仅依赖于  $k$  的可计算函数. 新实例  $(I', k')$  被称为问题核,  $g(k)$  是问题核的大小. 如果  $g(\cdot)$  是一个多项式函数或线性函数, 我们分别称该问题具有多项式或线性核.

### 3 两种皇冠分解变体

皇冠分解是构造著名顶点覆盖问题核心化算法的关键技术之一. 该技术已扩展用于解决若干顶点覆盖的相关问题<sup>[18~20]</sup>. 本文将设计两种皇冠分解的变体, Paw 皇冠分解和 Diamond 皇冠分解, 来解决结构性更强的 Paw 图 - 边删除问题.

二分图的  $q$ -扩展 ( $q$ -expansion) 和拓展引理是证明皇冠分解存在性的关键. 本节同样需要用到这些概念来证明 Paw 皇冠分解和 Diamond 皇冠分解的存在性. 考虑一个二分图  $G$ , 其顶点被分为两个部分  $(A, B)$ . 边集合  $M \subseteq E(G)$  是从  $A$  到  $B$  的  $q$ -扩展 ( $q \geq 1$ ), 则  $M$  同时满足以下条件:

- 每个  $A$  中的顶点正好与  $M$  中的  $q$  条边相邻;
- $M$  与  $B$  中  $q|A|$  个不同的顶点相邻.

事实上 1-拓展完全等价于匹配, 图 2 是一个 2-拓展的例子. 对任意大于 1 的整数  $q$  已知以下两个引理<sup>[6, 21]</sup>.

**引理1** (文献 [21] 中引理 7) 存在一个多项式时间算法, 对于给定的二分图  $G = ((A, B), E)$  和整数  $q$ , 该算法可以判断 (并在肯定情况下输出) 是否存在集合  $X \subseteq A, Y \subseteq B$ , 使得  $X$  到  $Y$  有一个  $q$ -扩展, 且  $Y$  中的顶点没有在  $X$  之外的邻居, 即  $N(Y) \subseteq X$ .

**引理2** (文献 [6] 中引理 5.2) 设  $q$  为正整数,  $G = ((A, B), E)$  是一个二分图, 满足条件  $|B| \geq q|A|$  且  $B$  中没有孤立点. 那么, 存在非空顶点集  $X \subseteq A$  和  $Y \subseteq B$ , 使得以下条件同时满足:

- 存在一个  $X$  到  $Y$  的非空  $q$ -扩展;
- $Y$  中的顶点没有在  $X$  之外的邻居, 即  $N(Y) \subseteq X$ .

#### 3.1 Paw 皇冠分解

**定义1** (Paw 皇冠分解) 对于图  $G = (V, E)$ , Paw 皇冠分解是一个三元组  $(C_1, C_2, H)$ , 其中  $C_1 \subseteq V$  和  $C_2 \subseteq V$  是两个点集,  $H \subseteq E$  是一个边集, 同时满足以下条件:

- $C_1, C_2$  和  $V(H)$  3 个集合两两之间互不相交;
- $C_2$  中的点不能支撑任何边;
- $C_1$  中的点只能支撑  $H$  中的边;
- 存在一个边不相交的 Paw 填装  $P$ , 大小为  $|P| = |H|$ , 使得  $P$  中的每个 Paw 都包含  $C_1, C_2$  中的各一个顶点和  $H$  中的一条边的两个端点, 填装  $P$  也称为 Paw 皇冠分解的见证填装.

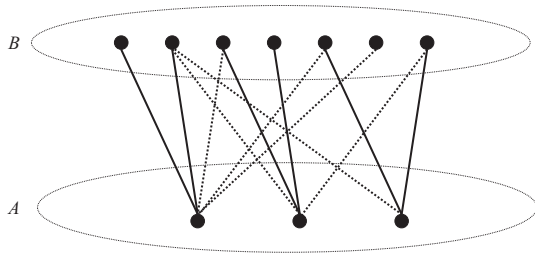


图 2 一个 2 扩展的例子  
Figure 2 Illustration of 2-expansion

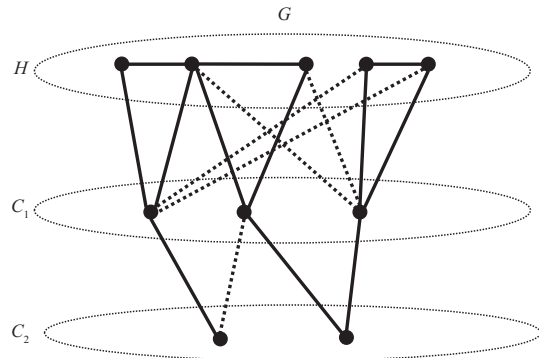


图 3 一个 Paw 皇冠分解的例子, 黑色的边所构成的 3 个边不相交的 Paw 为见证填充  $P$   
Figure 3 Illustration of a Paw crown decomposition, where the three edge-disjoint Paws formed by the black edges are witness Paw packing

图 3 是 Paw 皇冠分解的一个例子. 针对 Paw 皇冠分解, 我们给出下面两条引理, 第 1 条引理用于判定 Paw 皇冠分解在一个给定的图结构中是否存在, 第 2 条引理则证明了我们可以从一个非空的 Paw 皇冠分解中删去一些特定边而不改变 Paw 图 - 边删除问题的性质. 在设计核心化算法时, 我们会用到这两条引理.

**引理 3** 给定图  $G = (V, E)$ , 点集  $V' \subseteq V$  和  $V'' \subseteq V$ , 以及边集  $E' \subseteq E$ , 同时满足以下条件:

- $V', V''$ , 和  $V(E')$  3 个集合两两之间互不相交;
- $V''$  中的顶点不会支撑任何边;
- $V'$  中的顶点只支撑  $E'$  中的边;
- $V'$  中的每一个点至少与  $V''$  中的一个点相邻.

则存在一个多项式时间算法, 判断是否存在 Paw 皇冠分解  $(C_1, C_2, H)$  满足  $C_1 \subseteq V', C_2 \subseteq V''$  且  $\emptyset \neq H \subseteq E'$ , 并在存在时输出一个 Paw 皇冠分解, 以及一个大小为  $|P| = |H| > 0$  的边不相交见证填充  $P$ .

特别地, 若  $|V'| > |E'|$ , 且  $V'$  中的每个顶点至少支撑了  $E'$  中的一条边时, 符合要求的 Paw 皇冠分解一定存在.

**证明** 为了方便证明, 我们构造一个辅助二分图  $G^* = ((A, B), E^*)$ ,  $A$  中的每个顶点对应  $G$  中的边集  $E'$  中的一条边,  $B$  中的每个顶点对应  $G$  中点集  $V'$  中的一个顶点. 二分图  $G^*$  的  $A$  部分中的一个顶点  $e'$  与  $B$  部分中的一个顶点  $v'$  之间存在一条边当且仅当图  $G$  中对应  $e'$  的边  $e$  被对应  $v'$  的顶点  $v$  支撑. 图 4 是辅助二分图构造的一个例子.

我们断言图  $G$  中存在一个 Paw 皇冠分解  $(C_1, C_2, H)$  满足  $C_1 \subseteq V', C_2 \subseteq V''$  且  $\emptyset \neq H \subseteq E'$  当且仅当图  $G^*$  中存在点集  $C'_1 \subseteq B$  和  $H' \subseteq A$ , 使得导出子图  $G^*[C'_1 \cup H']$  中存在大小为  $|H'| > 0$  的匹配且满足  $N_{G^*}(C'_1) \subseteq H'$ , 其中  $H'$  表示图  $G^*$  中对应了边集  $H$  的点集,  $C'_1$  表示图  $G^*$  中对应了点集  $C_1$  的点集. 该断言的充分性由辅助二分图  $G^*$  的构造方式和 Paw 皇冠分解的定义直接得到.

我们现在证明该断言的必要性. 已知导出子图  $G^*[C'_1 \cup H']$  中存在大小为  $|H'| > 0$  的匹配  $P$  且满足  $N_{G^*}(C'_1) \subseteq H'$ . 我们令  $C_1$  为  $G$  中对应了  $C'_1$  的点集,  $C_2 = N(C_1) \cap V'', H$  为图  $G$  中对应了  $G^*$  中点集  $H'$  的边集. 我们证明三元组  $(C_1, C_2, H)$  是图  $G$  的一个 Paw 皇冠分解. 首先, 由  $C_2 \subseteq V''$  且

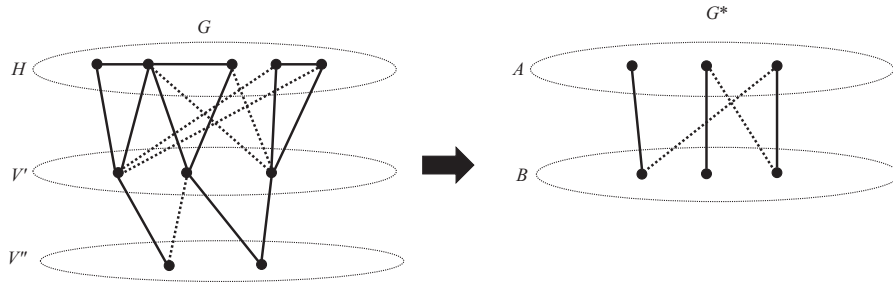


图 4 引理 3 证明中的辅助二分图  $G^*$  的构造实例,  $G$  中黑色的边构成的 Paw 填充对应了  $G^*$  中黑色的边构成的匹配

Figure 4 Construction example of the auxiliary bipartite graph  $G^*$  in the proof of Lemma 3, the black edges in  $G$  form a Paw packing which corresponds to the matching formed by the black edges in  $G^*$

$V''$  中的顶点不会支撑任何边可知  $C_2$  中的顶点不会支撑任何边. 又因为  $N_{G^*}(C'_1) \subseteq H'$ , 我们知道二分图中  $C'_1$  中的顶点只会与  $H'$  相邻, 而在构造二分图  $G^*$  时, 两个顶点  $v' \in A$  和  $e' \in B$  相邻当且仅当点  $v$  在图  $G$  中支撑了边  $e$ . 因此,  $C_1$  中的顶点只会支撑  $H$  中的边. 我们现在只需要构造一个边不相交的 Paw 填充使得其中的每个 Paw 都包含  $C_1, C_2$  中的各一个顶点和  $H$  中的一条边的两个端点即可完成证明.

对于匹配  $P$  中的每条边  $v'e'$  满足  $e' \in H'$  和  $v' \in C'_1$ , 我们知道  $v$  在  $G$  中支撑了边  $e$ , 即  $v$  和  $e$  的两个端点构成了一个三角形. 因此, 匹配  $P$  在图  $G$  中对应了一个三角形填充  $P'$ , 且由于匹配  $P$  中的所有边两两之间互不相交, 三角形填充  $P'$  是一个边不相交的三角形填充. 又因为  $V'$  中的每一个点至少与  $V''$  中的一个点相邻, 对于  $V' \cap C_1$  中的每一个顶点  $v$ , 令  $u = N(v) \cap V'' \subseteq C_2 = N(C_1) \cap V''$ , 我们将边  $uv$  作为尾巴与三角形填充中点  $v$  所在的三角形拼接就得到了一个 Paw. 按照这种方式我们即得到 Paw 填充  $P''$ , 又因为  $V', V''$ , 和  $V(E')$  3 个集合两两之间互不相交, 这样得到的 Paw 填充一定是边不相交的. 至此, 我们证明了断言的必要性.

由引理 1 可知, 对于给定的二分图  $G = ((A, B), E)$ , 存在多项式时间算法判断是否存在集合  $X \subseteq A, Y \subseteq B$ , 使得  $X$  到  $Y$  有一个 1-扩展 (即匹配), 且  $N(Y) \subseteq X$  (并在肯定情况下输出). 我们结合断言可知该算法也可以在多项式时间内判定图  $G$  中是否存在一个 Paw 皇冠分解  $(C_1, C_2, H)$  满足  $C_1 \subseteq V', C_2 \subseteq V''$  且  $\emptyset \neq H \subseteq E'$  (并在肯定情况下输出).

特别地, 若  $|V'| > |E'|$ , 且  $V'$  中的每个顶点至少支撑了  $E'$  中的一条边时, 对应的二分图  $G^*$  中满足  $|B| > |A|$ , 且  $B$  中没有孤立点. 那么根据引理 2, 满足条件的匹配一定存在, 由断言可知满足条件的 Paw 皇冠分解也一定存在.

**引理4** 给定图  $G = (V, E)$ , 和  $G$  中的一个 Paw 皇冠分解  $(C_1, C_2, H)$ . 令图  $G'$  为从图  $G$  中删除边集  $H$  得到的图. 则图  $G$  中存在大小为  $k$  的 Paw 图 - 边删除集当且仅当图  $G'$  中存在大小为  $k - |H|$  的 Paw 图 - 边删除集.

**证明** 已知对于 Paw 皇冠分解  $(C_1, C_2, H)$ , 存在一个大小为  $|P| = |H|$  的见证填充  $P$ .

我们先考虑充分条件, 令  $M$  为图  $G$  中大小为  $k$  的一个 Paw 图 - 边删除集. 我们先用如下的迭代方式得到边集  $M'$ ,  $M'$  仍然是  $G$  的一个大小为  $k$  的 Paw 图 - 边删除集, 且满足  $H \subseteq M'$ . 对于见证填充中的每一个 Paw  $T \in P$ , 如果  $M$  在  $T$  中所选的边没有一条在  $H$  中即  $(M \cap E(T)) \cap H = \emptyset$ , 则我们将  $M \cap E(T)$  中的其中任意一条边  $e$  替换为  $E(T) \cap H$  中的边  $e'$ . 我们证明对见证填充  $P$  中的每一个 Paw 进行替换后得到的边集  $M'$  仍然是  $G$  的一个 Paw 图 - 边删除集, 注意根据  $M'$  的构造方式

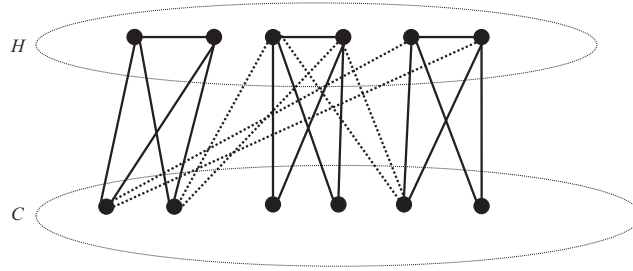


图 5 一个 Diamond 皇冠分解的例子, 黑色的边为见证 Diamond 填装  
Figure 5 Illustration of diamond crown decomposition

可知  $H \subseteq M'$ .

假设以上结论不成立,  $M'$  不是  $G$  的一个 Paw 图 – 边删除集, 即存在 Paw  $T' \in G$  满足  $E(T') \cap M' = \emptyset$ . 由于我们只从  $M$  中删除了一些属于边集  $E(P) - H$  中的边, 因此  $T'$  一定包括了一条被替换的边  $e \in E(P) - H$ , 我们令  $M'$  中的边  $e' \in H$  是替换了  $e$  的边. 由 Paw 皇冠分解的结构可知,  $e$  可能为  $C_1$  与  $C_2$  之间的边或者  $C_1$  与  $H$  之间的边. 如果  $e$  为  $C_1$  与  $C_2$  之间的边, 由于  $C_2$  中的点不会支撑任何边, 且  $C_1$  中的点只能支撑  $H$  中的边, 那么  $e$  所在的所有 Paw 只能由  $e$  作为尾巴, 且包含一个  $C_1$  中的点支撑的边构成三角形, 由于  $C_1$  中的点只会支撑  $H$  中的边, 所以这些三角形都会被  $H$  覆盖, 因此这些 Paw 也都会被  $H \subseteq M'$  覆盖, 矛盾. 如果  $e$  为  $C_1$  与  $H$  之间的边, 那么  $e$  可能作为尾巴或者三角形中的一条边存在于 Paw  $T'$  中. 由于  $C_1$  只能支撑  $H$  中的边, 所以  $e$  作为三角形中的一条边存在于 Paw 中的情况也会被  $H \subseteq M'$  覆盖, 矛盾. 如果  $e$  作为尾巴出现在  $T'$  中, 且  $T'$  未被  $M'$  覆盖, 那么  $T'$  一定不包含  $H$  中的边, 即  $T'$  拥有一个位于  $H$  外, 且存在一个顶点与  $e$  相连的三角形. 那么  $T' - \{e\} \cup \{e'\}$  是一个未被  $M$  覆盖的 Paw, 与  $M$  是  $G$  的 Paw 图 – 边删除集矛盾. 由于我们总能推出矛盾, 因此  $M'$  仍然是  $G$  的一个 Paw 图 – 边删除集且大小为  $k$ ,  $M' - H$  即为大小为  $k - |H|$  的  $G'$  中的 Paw 图 – 边删除集.

我们再考虑必要条件, 令  $M'$  为图  $G'$  中大小为  $k - |H|$  的一个 Paw 图 – 边删除集, 我们证明  $M' \cup H$  是图  $G$  中合法的 Paw 图 – 边删除集. 假设以上结论不成立, 即存在图  $G$  中存在 Paw  $T$  未被  $M' \cup H$  覆盖. 由于  $G'$  中的所有 Paw 都被  $M'$  覆盖, 因此 Paw  $T$  一定包含  $H$  中的至少一条边. 那么 Paw  $T$  一定会被  $M' \cup H$  覆盖, 矛盾, 必要性证毕.

### 3.2 Diamond 皇冠分解

**定义2** (Diamond 皇冠分解) 对于图  $G = (V, E)$ , Diamond 皇冠分解是一个二元组  $(C, H)$ , 其中  $C \subseteq V$  是一个点集,  $H \subseteq E$  是一个边集, 同时满足以下条件:

- $C \cap V(H) = \emptyset$ ;
- $C$  中的点只能支撑  $H$  中的边;
- 存在一个边不相交的 Diamond 填装  $P$ , 大小为  $|P| = |H|$ , 使得  $P$  中的每个 Diamond 都包含  $C$  中的两个顶点和  $H$  中的一条边的两个端点. 填装  $P$  也称为 Diamond 皇冠分解的见证填装.

图 5 是 Diamond 皇冠分解的一个例子. 针对 Diamond 皇冠分解, 我们给出两条引理, 第 1 条引理用于判定 Diamond 皇冠分解在一个给定的图结构中是否存在, 第 2 条证明我们可以从一个非空的 Diamond 皇冠分解中删去一些特定的图结构而不改变 Paw 图 – 边删除问题的性质.

**引理5** 给定图  $G = (V, E)$ , 点集  $V' \subseteq V$  和边集  $E' \subseteq E$ , 同时满足以下条件:

- $V' \cap V(E') = \emptyset$ ;
- $V'$  中的顶点只支撑  $E'$  中的边.

则存在一个多项式时间算法, 判断是否存在 Diamond 皇冠分解  $(C, H)$  满足  $C \subseteq V'$  且  $\emptyset \neq H \subseteq E'$ , 并在存在时输出一个 Diamond 皇冠分解, 以及一个大小为  $|P| = |H| > 0$  的见证填充  $P$ .

特别地, 若  $|V'| > 2|E'|$ , 且  $V'$  中的每个顶点至少支撑了  $E'$  中的一条边时, 符合要求的 Diamond 皇冠分解一定存在.

**证明** 我们构造一个辅助二分图  $G^* = ((A, B), E^*)$ ,  $A$  中的每个顶点对应  $G$  中的边集  $E'$  中的一条边,  $B$  中的每个顶点对应  $G$  中点集  $V'$  中的一个顶点. 二分图  $G^*$  的  $A$  部分中的一个顶点  $e'$  与  $B$  部分中的一个顶点  $v'$  之间存在一条边当且仅当图  $G$  中对应  $e'$  的边  $e$  被对应  $v'$  的顶点  $v$  支撑.

我们断言图  $G$  中存在一个 Diamond 皇冠分解  $(C, H)$  满足  $C \subseteq V'$  且  $\emptyset \neq H \subseteq E'$  当且仅当图  $G^*$  中存在点集  $C' \subseteq B$  和  $H' \subseteq A$ , 使得导出子图  $G^*[C' \cup H']$  中存在大小为  $|H'| > 0$  的  $H'$  到  $C'$  2-扩展且满足  $N_{G^*}(C') = H'$ , 其中  $H'$  表示图  $G^*$  中对应了边集  $H$  的点集,  $C'$  表示图  $G^*$  中对应了点集  $C$  的点集. 该断言的充分性由辅助二分图  $G^*$  的构造方式和 Diamond 皇冠分解的定义直接得到.

我们现在证明该断言的必要性. 已知导出子图  $G^*[C' \cup H']$  中存在大小为  $|H'| > 0$  的  $H'$  到  $C'$  的 2-扩展  $P$  且满足  $N_{G^*}(C') = H'$ . 我们令  $C$  为  $G$  中对应了  $C'$  的点集,  $H$  为图  $G$  中对应了  $H'$  的边集. 我们证明二元组  $(C, H)$  是图  $G$  的一个 Diamond 皇冠分解. 首先, 因为  $N_{G^*}(C') \subseteq H'$ , 我们知道二分图中  $C'$  中的顶点只会与  $H'$  相邻, 而在构造二分图  $G^*$  时, 两个顶点  $v' \in A$  和  $e' \in B$  相邻当且仅当点  $v$  在图  $G$  中支撑了边  $e$ . 因此,  $C$  中的顶点只会支撑  $H$  中的边. 我们现在只需要构造一个边不相交的 Diamond 填充使得其中的每个 Diamond 都包含  $C$  中的两个顶点和  $H$  中的一条边的两个端点即可完成证明.

对于 2-扩展  $P$  中的每个  $P_3 = v'_1 e v'_2$ , 满足  $e' \in H'$  和  $v'_1, v'_2 \in C'$ , 我们知道  $v_1$  和  $v_2$  在  $G$  中支撑了边  $e$ , 即  $v_1, v_2$  和  $e$  的两个端点构成了一个 Diamond. 因此, 2-扩展  $P$  在图  $G$  中对应了一个 Diamond 填充  $P'$ , 且由于 2-扩展  $P$  中的所有  $P_3$  两两之间点不相交, Diamond 填充  $P'$  在  $G$  中一定点不相交. 至此, 我们证明了断言的必要性.

由引理 1 可知, 对于给定的二分图  $G = ((A, B), E)$ , 存在多项式时间算法判断是否存在集合  $X \subseteq A, Y \subseteq B$ , 使得  $X$  到  $Y$  有一个 2-扩展, 且  $N(Y) \subseteq X$  (并在肯定情况下输出). 我们结合断言可知该算法也可以在多项式时间内判定图  $G$  中是否存在一个 Diamond 皇冠分解  $(C, H)$  满足  $C \subseteq V'$  且  $\emptyset \neq H \subseteq E'$  (并在肯定情况下输出).

特别地, 若  $|V'| > 2|E'|$ , 且  $V'$  中的每个顶点至少支撑了  $E'$  中的一条边时, 对应的二分图  $G^*$  中满足  $|B| > 2|A|$ , 且  $B$  中没有孤立点. 那么根据引理 2, 满足条件的 2-扩展一定存在, 由断言可知满足条件的 Diamond 皇冠分解也一定存在.

**引理6** 给定图  $G = (V, E)$ , 和  $G$  中的一个 Diamond 皇冠分解  $(C, H)$ . 令图  $G'$  为从图  $G$  中删除边集  $H$  得到的图. 则图  $G$  中存在大小为  $k$  的 Paw 图 - 边删除集当且仅当图  $G'$  中存在大小为  $k - |H|$  的 Paw 图 - 边删除集.

**证明** 已知对于 Diamond 皇冠分解  $(C, H)$ , 存在一个大小为  $|P| = |H|$  的见证填充  $P$ .

我们先考虑充分条件, 令  $M$  为图  $G$  中大小为  $k$  的一个 Paw 图 - 边删除集. 我们先用如下的迭代方式得到边集  $M'$ ,  $M'$  仍然是  $G$  的一个大小为  $k$  的 Paw 图 - 边删除集, 且满足  $H \subseteq M'$ . 对于见证填充中的每一个 Diamond  $T \in P$ , 如果  $M$  在  $T$  中所选的边没有一条在  $H$  中即  $(M \cap E(T)) \cap H = \emptyset$ , 则我们将  $M \cap E(T)$  中的其中任意一条边  $e$  替换为  $E(T) \cap H$  中的边  $e'$ . 我们证明对见证填充  $P$  中的



每一个 Diamond 进行替换后得到的边集  $M'$  仍然是  $G$  的一个 Paw 图 – 边删除集, 注意根据  $M'$  的构造方式可知  $H \subseteq M'$ .

假设以上结论不成立,  $M'$  不是  $G$  的一个 Paw 图 – 边删除集, 即存在  $\text{Paw } T' \in G$  满足  $E(T') \cap M' = \emptyset$ . 由于我们只从  $M$  中删除了一些属于边集  $E(P) - H$  中的边, 因此  $T'$  一定包括了一条被替换的边  $e \in E(P) - H$ , 我们令  $M'$  中的边  $e' \in H$  是替换了  $e$  的边. 由 Diamond 皇冠分解的结构可知,  $e$  为  $C$  与  $H$  之间的边, 且  $e$  可能作为尾巴或者三角形中的一条边存在于  $\text{Paw } T'$  中. 由于  $C$  只能支撑  $H$  中的边, 所以  $e$  作为三角形中的一条边存在于  $\text{Paw}$  中的情况也会被  $H \subseteq M'$  覆盖, 矛盾. 如果  $e$  作为尾巴出现在  $T'$  中, 且  $T'$  未被  $M'$  覆盖, 那么  $T'$  一定不包含  $H$  中的边, 即  $T'$  拥有一个位于  $H$  外, 且存在一个顶点与  $e$  相连的三角形. 那么  $T' - \{e\} \cup \{e'\}$  是一个未被  $M$  覆盖的 Paw, 与  $M$  是  $G$  的 Paw 图 – 边删除集矛盾. 由于我们总能推出矛盾, 因此  $M'$  仍然是  $G$  的一个 Paw 图 – 边删除集且大小为  $k$ ,  $M' - H$  即为大小为  $k - |H|$  的  $G'$  中的 Paw 图 – 边删除集.

我们再考虑必要条件, 令  $M'$  为图  $G'$  中大小为  $k - |H|$  的一个 Paw 图 – 边删除集, 我们证明  $M' \cup H$  是图  $G$  中合法的 Paw 图 – 边删除集. 假设以上结论不成立, 即存在图  $G$  中存在 Paw  $T$  未被  $M' \cup H$  覆盖. 由于  $G'$  中的所有 Paw 都被  $M'$  覆盖, 因此 Paw  $T$  一定包含  $H$  中的至少一条边. 那么 Paw  $T$  一定会被  $M' \cup H$  覆盖, 矛盾, 必要性证毕.

#### 4 Paw 图 – 边删除问题的核心化算法

本节介绍 Paw 图 – 边删除问题的核心化算法. 该算法由一系列归约规则组成, 我们依次判定每一条规则是否可以应用于当前实例. 若不行, 则继续判定下一条规则; 若可以, 则应用该规则后再回到第一条规则进行判定, 直到所有的归约规则均不能应用于当前实例. 基于我们的算法运行方式, 每条归约规则在判定是否可以被应用时, 此条归约规则之前的所有归约规则均不能应用于当前实例.

若原始实例  $(G, k)$  是一个“是”实例当且仅当应用归约规则后得到的实例  $(G', k')$  是一个“是”实例, 则我们称该归约规则为正确的. 为了证明核心化算法的正确性, 我们需要证明所有的归约规则都是正确的. 以下两条规则显然是正确的.

**归约规则1** 如果  $k \geq 0$  且图为空, 则返回“是”; 如果  $k < 0$ , 则返回“否”.

**归约规则2** 如果图中存在一个顶点或边没有出现在任何 Paw 中, 则从图中删除该顶点或边.

**归约规则3** 如果图中存在一个顶点  $v \in V$ , 使得与  $v$  相邻的所有边可以被划分为两个部分  $E_1$  和  $E_2$ , 并且图  $G$  中的任何一个 Paw 不会同时包含  $E_1$  中的一条边和  $E_2$  中的一条边, 那么将  $v$  拆分为两个顶点  $v'$  和  $v''$ , 使得  $E_1$  中的所有边与  $v'$  相邻, 而  $E_2$  中的所有边与  $v''$  相邻.

这个归约规则会增加图中的顶点数, 但是它将简化图的结构并便于我们的分析. 图 6 为该归约规则的一个应用实例.

**引理7** 归约规则 3 是正确的, 并且可以在多项式时间内运行.

**证明** 首先, 我们考虑正确性. 设  $G' = (V', E')$  是在顶点  $v$  上应用归约规则 3 后得到的图,  $v'$  和  $v''$  是点  $v$  分裂得到的两个顶点. 根据此规则的应用方式, 边集  $E$  和  $E'$  的大小是一样的. 我们将顶点  $v'$  和  $v''$  同时视为  $v$  在边集  $E$  和边集  $E'$  之间建立一一对应关系. 由于图  $G$  中没有包含  $E_1$  中的一条边和  $E_2$  中的一条边的 Paw, 因此  $G$  中的 Paw 与  $G'$  中的 Paw 也一一对应, 即图  $G$  中存在一个 Paw 图 – 边删除集当且仅当图  $G'$  中存在一个 Paw 图 – 边删除集, 正确性得证.

我们给出一个简单的贪心算法来找到给定顶点  $v$  的边集  $E_1$  和  $E_2$ . 初始时令  $E_1$  包含与  $v$  相邻

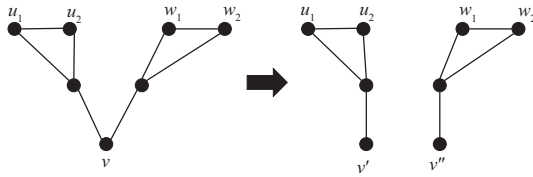


图 6 归约规则 3 的一个应用实例: 点  $v$  的两条邻边不出现在同一个 Paw 中, 因此将点  $v$  分裂成点  $v'$  和  $v''$  分别邻接这两条边.

Figure 6 Illustration of Reduction Rule 3

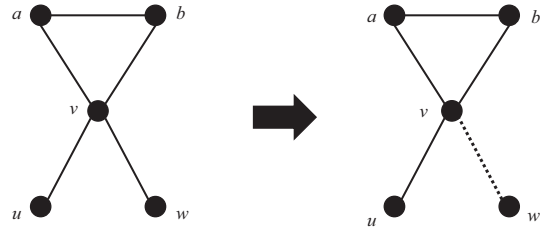


图 7 归约规则 4 的一个应用实例: 边  $vu$  和  $vw$  只能作为点  $v$  的尾巴构成爪, 因此删去边  $vw$

Figure 7 Illustration of Reduction Rule 4

的一条任意边  $e$ . 我们反复执行以下步骤, 直到不再更新为止: 如果存在一个 Paw, 其中包含  $E_1$  中的一条边和  $v$  的一条邻边  $e'$ , 但  $e'$  不在  $E_1$  中, 则将边  $e'$  添加到  $E_1$  中. 如果  $E_1 \neq E$ , 那么我们可以将  $E$  划分为两个部分  $E_1$  和  $E_2 = E \setminus E_1$ . 否则, 与  $v$  相邻的边不能被划分.

**归约规则 4** 如果图中存在一个顶点  $v$ , 使得图中有两条与点  $v$  相邻的边, 它们均只能作为  $v$  的尾巴构成一个 Paw, 则删去其中一条.

图 7 是归约规则 4 的一个应用实例.

**引理 8** 归约规则 4 是正确的.

**证明** 令图  $G = (V, E)$  中存在点  $v$ ,  $v$  有 2 条只能作为  $v$  的尾巴构成 Paw 的相邻边  $vw$  和  $vu$ . 设被删掉的边为  $vw$ , 保留在  $G$  中的边为  $vu$ , 删掉  $vw$  后的图为  $G'$ , 我们证明  $G$  中存在大小为  $k$  的 Paw 图 - 边删除集当且仅当  $G'$  中存在大小为  $k$  的 Paw 图 - 边删除集.

由于  $G'$  是  $G$  的子图, 故充分性可直接由 Paw 图 - 边删除问题的遗传性得到, 我们证明必要性. 令  $M'$  为  $G'$  中大小为  $k$  的 Paw 图 - 边删除集. 如果  $G$  中所有 Paw 都被  $M'$  覆盖, 则  $M'$  也是  $G$  的大小为  $k$  的 Paw 图 - 边删除集, 必要性得证.

假设图  $G$  中存在两个 Paw 没有被  $M'$  覆盖, 那么这两个 Paw 一定都包含边  $vw$ , 否则会与  $M'$  是  $G'$  的 Paw 图 - 边删除集矛盾. 由于  $vw$  只能作为  $v$  的尾巴构成 Paw, 因此这两个 Paw 一定分别包含一个以点  $v$  作为其中一个端点的三角形. 此时我们总可以在这两个 Paw 组成的边导出子图上找到一个不包含  $vw$  的 Paw, 这与  $M'$  是图  $G'$  的 Paw 图 - 边删除集矛盾.

假设图  $G$  中最多存在一个 Paw 没有被  $M'$  覆盖, 令该 Paw 为  $abvw$ . 由于三角形  $abv$  未被  $M'$  覆盖, 除  $va, vb$  和  $vw$  外的点  $v$  的任何一个邻边均在  $M'$  中, 否则我们一定能在图  $G'$  中找到一个未被  $M'$  覆盖的 Paw. 因此  $M' \setminus \{vu\} \cup \{va\}$  为图  $G$  中大小为  $k$  的 Paw 图 - 边删除集, 定理得证.

在对图  $G$  应用前面所有归约规则之后, 我们用一个简单的贪心算法在图  $G$  中找到一个极大的边不相交 Paw 填装  $S$ , 即我们迭代的把图  $G$  中与  $S$  边不相交的一个 Paw 加入  $S$ , 直到  $G$  中不存在这样的 Paw. 接下来的每一条归约规则会与这个极大边不相交 Paw 填装  $S$  相关. 首先, 如果  $|S| > k$ , 那么我们至少需要  $k + 1$  条边才能覆盖  $S$  中的 Paw, 即  $(G, k)$  是一个“否”实例.

**归约规则 5** 如果  $|S| > k$ , 则返回“否”.

在本文的剩余部分, 我们用  $F = V \setminus V(S)$  表示  $G$  里不出现在  $S$  中的顶点集合.

在介绍接下来的归约规则之前, 我们先给出一些执行规则 5 之后的图  $G$  的性质和 Paw 填装  $S$  的性质. 注意由于算法的迭代执行, 我们可能会多次找到一些互不相同的 Paw 填装  $S$ , 但是这些性质总

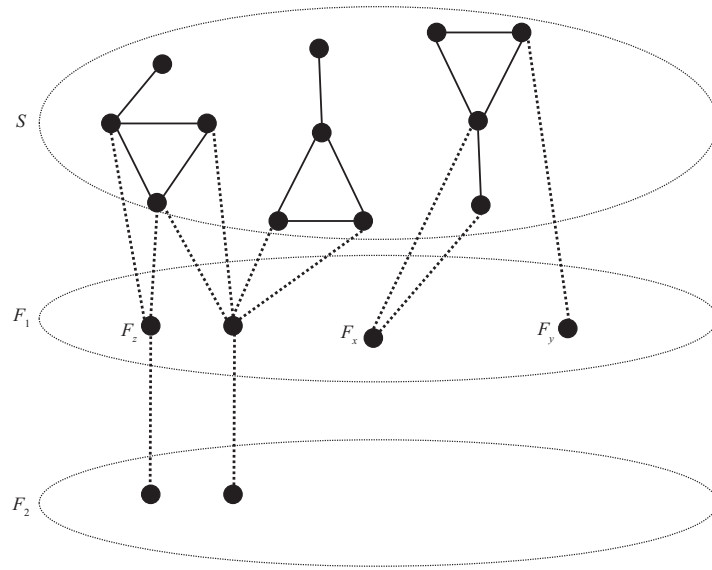


图 8 图  $G$  中的不同点集的实例  
 Figure 8 Illustration of vertex sets in  $G$

是可以保证的. 由归约规则 2 和  $S$  的极大性, 我们得到如下引理.

**引理9** 执行归约规则 5 之后, 图  $G$  中的任意一条边或一个顶点一定存在于一个 Paw 中; 图  $G$  中的任意一个 Paw 一定包含了  $E(S)$  中的一条边.

由上述引理可知执行归约规则 5 之后,  $F$  中任意一个点与  $V(S)$  的距离最多为 2. 我们用  $F_1$  表示  $F$  中与  $V(S)$  距离为 1 的顶点;  $F_2$  表示  $F$  中与  $V(S)$  距离为 2 的顶点;  $F_z$  表示  $F_1$  中与  $F_2$  某些点相邻的点, 即  $F_z = \{v \in F_1 | N(v) \cap F_2 \neq \emptyset\}$ .  $F_x$  是  $F_1 - F_z$  中支撑了  $E(S)$  中至少一条边的顶点的集合;  $F_y$  是  $F_1$  中的剩余点集, 即  $F_y = F_1 - F_z - F_x$ . 图 8 给出了这些点集的一个例子.

- 引理10**
- (1)  $F_z$  中的每一个顶点只能支撑  $E(S)$  中的边且至少支撑一条.
  - (2)  $F_2$  中的顶点不能支撑任何边且只与  $F_z$  中的顶点相邻.
  - (3)  $F_x$  中的顶点只支撑  $E(S)$  中的边.

**证明** 我们考虑顶点  $v_2 \in F_2$ , 由  $F_2$  的定义可知  $v_2$  与  $V(S)$  的距离为 2, 因此  $v_2$  不与  $V(S)$  中的顶点相邻, 又由  $F_z$  的定义可知  $v_2$  只与  $F_z$  中的顶点相邻, 令  $v_1 = F_z \cap N(v_2)$ . 假设  $v_2$  支撑了一条不在  $E(S)$  中的边  $ab$ , 那么 Paw  $abv_2v_1$  是一个不含有  $E(S)$  中的边的 Paw, 这与引理 9 所描述的图  $G$  中的任意一个 Paw 一定包含了  $E(S)$  中的边矛盾, 因此  $v_2$  不会支撑任何边, (2) 成立.

我们考虑顶点  $v'_1 \in F_z$ , 由  $F_z$  的定义可知  $N(v'_1) \cap F_2 \neq \emptyset$ , 令  $v'_2 \in N(v'_1) \cap F_2$ . 如果顶点  $v'_1$  支撑了一条不在  $E(S)$  中的边  $uw$ , 那么有 Paw  $uwv'_1v'_2$  不包含  $E(S)$  中的任何边, 这与引理 9 中的图  $G$  中的任意一个 Paw 一定包含了  $E(S)$  中的一条边矛盾, 因此  $F_z$  中的每一个顶点只能支撑  $E(S)$  中的边. 如果  $v'_1$  不支撑任何边, 由 (2) 可知  $v'_2 \in F_2$  不会支撑任何边, 那么边  $v'_1v'_2$  不会存在于图中的任何一个 Paw 中, 这与引理 9 所描述的图  $G$  中任意一个顶点至少存在在一个 Paw 中矛盾, 因此  $v'_1$  至少会支撑一条边. 综上可知 (1) 成立.

考虑顶点  $v_x \in F_x$ , 由  $F_x$  的定义可知  $v_x$  支撑了一条  $E(S)$  中的边, 令该边为  $u_xw_x$ . 假设  $v_x$  支撑了一条不在  $E(S)$  中的边  $a_xb_x$ , 显然  $a_xb_x \neq u_xw_x$ . 由于点  $u_x$  和  $w_x$  至少有一个不等于  $a_x$  或者  $b_x$ , 四

元组  $a_x b_x v_x u_x$  和  $a_x b_x v_x w_x$  中至少有一个是合法的不包含  $E(S)$  中的边的 Paw. 这与引理 9 中要求的图中所有 Paw 均包含一条  $E(S)$  中的边矛盾, 因此  $F_x$  中的顶点只会支撑  $E(S)$  中的边, (3) 成立.

由引理 10 可知  $F_z, F_2$  和  $E(S)$  是符合引理 3 中的  $V', V''$  和  $E'$  的要求的集合, 因此, 我们设计如下的基于 Paw 皇冠分解的归约规则, 其正确性由引理 4 保证.

**归约规则6** 给定应用了归约规则 5 之后的图  $G = (V, E)$ , 调用引理 3 中的算法检查图中是否存在一个 Paw 皇冠分解  $(C_1, C_2, H)$ , 使得  $C_1 \subseteq F_z, C_2 \subseteq F_2$  且  $\emptyset \neq H \subseteq E(S)$ , 如果存在, 则删除边集  $H$ , 令  $k = k - |H|$ .

同样由引理 10 可知,  $F_x$  和  $E(S)$  是符合引理 5 中的  $V'$  和  $E'$  要求的集合, 因此, 我们设计如下的基于 Diamond 皇冠分解的归约规则, 其正确性由引理 6 保证.

**归约规则7** 给定应用了归约规则 6 之后的图  $G = (V, E)$ , 调用引理 5 中的算法检查图中是否存在一个 Diamond 皇冠分解  $(C, H)$ , 使得  $C \subseteq F_x$ , 且  $\emptyset \neq H \subseteq E(S)$ , 如果存在, 则删除边集  $H$ , 令  $k = k - |H|$ .

以上 7 条归约规则组成了我们的核心化算法. 如果一个实例不能应用以上任何一条归约规则, 我们称该实例是简化的, 相应的图也被称为简化图.

**引理11** 对于任何实例, 以上核心化算法都可以在多项式时间内运行, 并输出简化实例.

**证明** 对于任意给定实例  $(G = (V, E), k)$ , 所有归约规则的一次判定都可以在多项式时间内完成, 其中归约规则 6 和 7 的判定时间由引理 3 和 5 确定. 我们证明所有归约规则的执行次数均为多项式的.

归约规则 1 最多执行一次就会使算法终止. 归约规则 2, 4, 6 和 7 的每次执行都会使得图  $G$  中的顶点数或者边数减少 1, 因此这 4 条归约规则一共最多执行  $|V| + |E|$  次. 归约规则 5 以及贪心算法的执行不会改变图  $G$  的结构, 同时其执行次数最多为其他所有规则的执行次数之和. 归约规则 3 的每次执行不会增加图中边的条数, 同时此归约规则最多连续执行  $2|E|$  次, 因为执行此规则最多只能使得图中顶点个数为  $2|E|$ . 那么归约规则 3 的总执行次数最多为其他改变图结构的归约规则执行次数的总和乘  $2|E|$  次. 因此, 整个核心化算法一定可以在多项式时间内终止, 并返回一个所有归约规则都无法再应用的简化实例.

## 5 Paw 图 – 边删除问题的问题核大小分析

本节对核心化算法的输出 – 简化图进行分析. 首先, 由归约规则 5 可知简化图中存在一个极大的边不相交 Paw 填充  $S$  满足  $|S| \leq k$ , 因此  $|V(S)| \leq 4k$  且  $|E(S)| \leq 4k$ . 我们用  $E(V(S))$  表示导出子图  $G[V(S)]$  中的边组成的集合, 有  $|E(V(S))| \leq \frac{4k \times (4k-1)}{2} \leq 8k^2$ . 我们接下来分别考虑  $F$  中的点集  $F_z, F_2, F_x$  和  $F_y$  中点的个数以及这些点集内部和彼此之间边的条数. 我们最终证明了简化图  $G$  中的顶点个数最多为  $32k$ , 边的条数最多为  $128k^2 + 20k$ .

**引理12**  $|F_z| \leq |E(S)| \leq 4k$ .

**证明** 假设简化图  $G$  中满足  $|F_z| > |E(S)|$ , 我们证明归约规则 6 可以应用于  $G$ . 这与简化图不能再应用任何归约规则矛盾, 并推出  $|F_z| \leq |E(S)|$ , 从而证明本引理.

首先由定义可知  $F_z, F_2, V(S)$  3 个点集两两之间互不相交, 且  $F_z$  中的每一个顶点至少与  $F_2$  中的一个顶点相邻. 又由引理 10 可知,  $F_z$  中的每一个顶点只能支撑  $E(S)$  中的边且至少支撑一条;  $F_2$  中的顶点不能支撑任何边且只与  $F_z$  中的顶点相邻. 再结合假设  $|F_z| > |E(S)|$ , 引理 3 的算法一定可以

输出一个满足  $C_1 \subseteq F_z, C_2 \subseteq F_2$  且  $\emptyset \neq H \subseteq E(S)$  的 Paw 皇冠分解. 该皇冠分解可以被归约规则 6 所应用.

**引理13**  $|F_2| \leq |F_z| \leq 4k$ , 且  $F_2$  与  $F_z$  之间的边数最多为  $|F_z| \leq 4k$  条.

**证明** 由引理 10 可知  $F_2$  中的顶点不能支撑任何边且只能与  $F_z$  中的顶点相邻. 因此  $F_z$  与  $F_2$  之间的边只能作为  $F_z$  中的顶点的尾巴来参与组成 Paw. 我们考虑  $F_z$  中的一个顶点  $v$ , 如果  $v$  在  $F_2$  中的邻居大于等于两个, 即存在两条  $v$  的邻边只能作为  $v$  的尾巴构成 Paw, 这就会与归约规则 4 不能再应用矛盾. 因此每个  $F_z$  中的顶点在  $F_2$  中的邻居最多为一个, 即  $F_2$  中的顶点个数和  $F_2$  与  $F_1$  之间边的条数均不会超过  $|F_z|$ .

**引理14**  $|F_x| \leq 2|E(S)| \leq 8k$ .

**证明** 假设简化图  $G$  中满足  $|F_x| > 2|E(S)|$ , 我们证明归约规则 7 可以应用于  $G$ . 这与简化图不能再应用任何归约规则矛盾, 并推出  $|F_x| \leq 2|E(S)|$ , 从而证明本引理.

首先由定义可知  $F_x$  与  $V(S)$  互不相交, 且  $F_x$  中的每一个顶点至少支撑  $E(S)$  中的一条边. 又由引理 10 可知,  $F_x$  中的顶点只支撑  $E(S)$  中的边. 再结合假设  $|F_x| > 2|E(S)|$ , 引理 5 的算法一定可以输出一个满足  $C \subseteq F_x$ , 且  $\emptyset \neq H \subseteq E(S)$  的 Diamond 皇冠分解. 该皇冠分解可以被归约规则 7 所应用.

**引理15**  $|F_y| \leq 3|V(S)| \leq 12k$ ,  $F_y$  中的点只与  $F_y \cup V(S)$  中的点相邻且边最多为  $4|V(S)| \leq 16k$  条.

**证明** 由  $F_y$  的定义可知,  $F_y$  中的顶点与  $S$  的距离为 1 且不会支撑任何  $E(S)$  中的边. 我们把  $F_y$  中的点分为两类. 第 1 类  $F'_y$  中的顶点支撑了一条不属于  $E(S)$  中的边, 第 2 类  $F''_y$  中的顶点不支撑图  $G$  中的任何边.

考虑  $v' \in F'_y$ , 支撑了一条不属于  $E(S)$  中的边  $uw$ . 如果  $u, w \notin V(S)$  则包含三角形  $v'uw$  的 Paw 一定不包含  $E(S)$  中的边, 这与引理 9 所描述的任意 Paw 一定包含至少一条  $E(S)$  中的边矛盾. 因此  $u, w$  中至少有一个顶点属于  $V(S)$ . 不失一般性的, 我们令  $u \in V(S)$ , 由于三角形  $v'uw$  不包含  $E(S)$  中的任意一条边, 点  $u, v'$  和  $w$  不能邻接其他任何不在  $E(S)$  中的边, 否则我们一定可以找到一个不包含  $E(S)$  中的边的 Paw, 与引理 9 矛盾. 我们称三角形  $v'uw$  为点  $u$  的私有三角形. 另一个方面, 对于  $V(S)$  中的任意一点来说, 其最多拥有一个私有三角形, 否则, 我们可以在这两个私有三角形的边导出子图中找到一个 Paw 不包含  $E(S)$  中的边, 从而推出矛盾. 因此  $F'_y$  中的点最多有  $2|V(S)| \leq 8k$  个,  $F'_y$  内部的边最多  $|V(S)| \leq 4k$  条且  $F'_y$  与  $V(S)$  中的边最多有  $2|V(S)| \leq 8k$  条.

考虑  $v'' \in F''_y$ , 不支撑图中的任何边. 我们证明  $v''$  所相邻的边只能作为尾巴通过连向  $V(S)$  中的某个点组成 Paw. 由  $F''_y$  定义可知,  $v''$  与  $V(S)$  的距离为 1,  $v''$  不存在  $F_2$  中的邻居, 且  $v''$  不会支撑图  $G$  中的任何边. 由于  $v''$  与  $V(S)$  的距离为 1, 令  $u'' \in N(v'') \cap V(S)$ . 我们证明边  $v''u''$  只能作为  $u''$  的尾巴存在于 Paw 中. 首先由于  $v''$  不支撑图中任何边, 边  $v''u''$  只能作为尾巴存在于 Paw 中. 假设  $v''u''$  作为  $v''$  的尾巴存在于 Paw 中, 同样由于  $v''$  不会支撑任何边, 该 Paw 一定不会包含  $E(S)$  中的边, 这与引理 9 矛盾. 因此边  $v''u''$  只会作为  $u''$  的尾巴存在于 Paw 中. 我们再证明  $N(v'') \setminus V(S) = \emptyset$ . 假设存在点  $w'' \in N(v'') \setminus V(S)$ , 由于点  $v''$  不会支撑任何边, 边  $v''u''$  和  $v''w''$  不可能同时存在于一个 Paw 中, 因此这种情况会被归约规则 3 所应用, 矛盾. 综上可知  $v''$  的所有邻居都在  $V(S)$  中, 且  $v''$  的相邻边只能作为  $V(S)$  中的点的尾巴构成 Paw. 另一个方面, 对于  $V(S)$  中的每一个顶点来说, 其最多拥有一个在  $F''_y$  中的邻居, 否则归约规则 4 就可以应用了. 因此  $|F''_y| \leq |V(S)| \leq 4k$  且  $F''_y$  与  $S$  之间的边最多有  $|V(S)| \leq 4k$  条.

表 2 简化图  $G$  中各点集之间的最大边数Table 2 Upper bound of edges between vertex sets in a reduced graph  $G$ 

	$V(S)$	$F_z$	$F_2$	$F_x$	$F_y$
$V(S)$	$ V(S) ^2/2$	$ E(S)  \cdot  V(S) $	0	$2 E(S)  \cdot  V(S) $	$3 V(S) $
$F_z$	-	$ E(S) ^2/2$	$E(S)$	$2 E(S)  \cdot  V(S) $	0
$F_2$	-	-	0	0	0
$F_x$	-	-	-	$2 E(S) ^2$	0
$F_y$	-	-	-	-	$ V(S) $

由于  $F_y = F'_y \cup F''_y$ , 综合以上两种情况, 我们证明了引理 15.

由引理 12~15 可得如下引理和定理.

**引理16** 给定一个执行全部归约规则后的简化图  $G = (V, E)$ ,  $V(G) \leq 4|V(S)| + 4|E(S)| \leq 32k$ ,  $E(G) \leq 5|E(S)| \cdot |V(S)| + 2.5|E(S)|^2 + 0.5|V(S)|^2 + E(S) + 4|V(S)| \leq 128k^2 + 20k$ .

**证明** 给定简化图  $G = (V, E)$ , 根据我们对于点集  $V(G)$  的划分有  $|V(G)| = |V(S)| + |V(F)| = |V(S)| + |F_z| + |F_2| + |F_x| + |F_y| \leq 4|V(S)| + 4|E(S)| \leq 32k$ . 而在分析简化图的边集  $E(G)$  时, 我们则要分别分析每个点集内部的边以及两个点集之间的边. 我们用表 2 表示两个点集之间的边数, 如果两个点集相同则表示其内部边数. 由于对称性, 我们只填写表格的上半部分. 其中需要特别说明的是, 由于  $F_2$  的定义, 其只能与  $F_z$  中的点相邻, 且由引理 12 可知  $F_2$  与  $F_z$  中的边数最多有  $|E(S)|$  条. 由引理 15 可知,  $F_y$  中的点只与  $V(S) \cup F_y$  中的点相邻且  $|E(F_y)| \leq |V(S)|$ ,  $F_y$  与  $V(S)$  之间的边最多有  $3|V(S)|$  条. 因此, 简化图  $G$  中最多有  $5|E(S)| \cdot |V(S)| + 2.5|E(S)|^2 + 0.5|V(S)|^2 + E(S) + 4|V(S)| \leq 128k^2 + 20k$  条边.

**定理1** Paw 图 - 边删除问题存在一个  $32k$  个点和  $128k^2 + 20k$  条边的问题核.

## 6 结论

本文研究了图删除问题中  $H$ -边删除问题的一个重要子类, 特别关注了  $H$  为 Paw 时的情况, 即 Paw 图 - 边删除问题. 在这一问题中, 对于给定的简单无向图  $G$ , 我们致力于确定通过删除  $k$  条边是否能使图中不存在任何子图为 Paw. 为了研究 Paw 图 - 边删除问题的核心化算法, 我们对图中的各种子图结构进行了深入分析, 引入了多条归约规则, 并设计了两种皇冠分解技术的变体 —— 这是我们能够得到线性顶点核最关键的技术. 对于  $|H| = 4$  的  $H$ -边删除问题, Paw 图 - 边删除问题是第 1 个做到线性顶点的核心化算法, 其他几个图形结构是否存在线性顶点的核心算法依然未知. 此外, 对于  $P_4$  图 - 边删除问题和 Diamond 图 - 边删除问题, 是否存在比直接从  $d$ -命中集问题的核心化算法推导出的算法给出的更小的问题核, 也值得进一步研究.

## 参考文献

- 1 Aravind N R, Sandeep R B, Sivadasan N. Dichotomy results on the hardness of  $H$ -free edge modification problems. SIAM J Discrete Math, 2017, 31: 542–561
- 2 Anstee R P. A polynomial algorithm for b-matchings: an alternative approach. Inf Process Lett, 1987, 24: 153–157
- 3 Yannakakis M. Edge-deletion problems. SIAM J Comput, 1981, 10: 297–309
- 4 Cai L. Fixed-parameter tractability of graph modification problems for hereditary properties. Inf Process Lett, 1996, 58: 171–176

- 5 Belova T, Bliznets I. Hardness of approximation for  $H$ -free edge modification problems: towards a dichotomy. In: Proceedings of the 33rd International Symposium on Algorithms and Computation, 2022
- 6 Bodlaender H L, Fomin F V, Lokshtanov D, et al. Kernelization. J ACM, 2016, 63: 44
- 7 Crespelle C, Drange P G, Fomin F V, et al. A survey of parameterized algorithms and the complexity of edge modification. Comput Sci Rev, 2023, 48: 100556
- 8 Cao Y X, Ke Y P. Improved kernels for edge modification problems. In: Proceedings of the 16th International Symposium on Parameterized and Exact Computation, 2021
- 9 Cai L, Cai Y. Incompressibility of  $H$ -free edge modification problems. Algorithmica, 2015, 71: 731–757
- 10 Guillemot S, Havet F, Paul C, et al. On the (non-)existence of polynomial kernels for  $P_1$ -free edge modification problems. Algorithmica, 2013, 65: 900–926
- 11 Tsur D. Kernel for  $K$ -free edge deletion. Inf Process Lett, 2021, 167: 106082
- 12 Cao Y, Rai A, Sandeep R B, et al. A polynomial kernel for diamond-free editing. Algorithmica, 2022, 84: 197–215
- 13 Eiben E, Lochet W, Saurabh S. A polynomial kernel for paw-free editing. In: Proceedings of the 15th International Symposium on Parameterized and Exact Computation, 2020
- 14 Marx D, Sandeep R B. Incompressibility of  $H$ -free edge modification problems: towards a dichotomy. J Comput Syst Sci, 2022, 125: 25–58
- 15 Sheng Z M, Xiao M Y. A discharging method: improved kernels for edge triangle packing and covering. In: Proceedings of the 29th International Conference on Computing and Combinatorics, 2023. 171–183
- 16 Xia G, Zhang Y. Kernelization for cycle transversal problems. Discrete Appl Math, 2012, 160: 1224–1231
- 17 Abu-Khazam F N. A kernelization algorithm for  $d$ -hitting set. J Comput Syst Sci, 2010, 76: 524–531
- 18 Dehne F, Fellows M, Rosamond F, et al. Greedy localization, iterative compression, modeled crown reductions: new FPT techniques, an improved algorithm for set splitting, and a novel  $2k$  kernelization for vertex cover. In: Proceedings of the 1st International Workshop on Parameterized and Exact Computation, 2004. 271–280
- 19 Xiao M, Kou S. A  $5k$ -vertex kernel for 3-path vertex cover. Theor Comput Sci, 2023, 959: 113872
- 20 Červený R, Choudhary P, Suchý O. On kernels for  $d$ -path vertex cover. In: Proceedings of the 47th International Symposium on Mathematical Foundations of Computer Science, 2022
- 21 Kumar M, Lokshtanov D. A  $2lk$  kernel for  $l$ -component order connectivity. In: Proceedings of the 11th International Symposium on Parameterized and Exact Computation, 2016

## A linear vertex kernel for the Paw edge covering problem

Zimo SHENG & Mingyu XIAO\*

*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

\* Corresponding author. E-mail: myxiao@uestc.edu.cn

**Abstract** One important class of problems in graph theory involves investigating whether it is possible to delete at most  $k$  edges from a graph in such a way that the remaining graph does not contain a certain subgraph structure  $H$ . The case where the subgraph  $H$  is a connected graph with at most 4 vertices has been widely studied. In this paper, we specifically consider the case where  $H$  is a Paw graph (a triangle with one vertex incident to an additional edge). The corresponding problem is called the Paw edge covering problem (also known as the  $\{\text{Paw}, \text{Diamond}, K_4\}$ -free edge deletion problem). In this paper, by using two new variants of crown decompositions, we show that the Paw edge covering problem has a kernel of  $32k$  vertices, which is the first linear-size vertex kernel for this problem.

**Keywords** graph algorithms, kernelization,  $H$ -edge covering, Paw-edge covering, crown decomposition