二分图上的公平性感知极大双团枚举

Ziqi Yin†、Qi Zhang†、Wentao Zhang‡、Rong-Hua Li†、Guoren Wang†

†北京理工大学，北京，中国； ‡北京大学，北京，中国
ZIQI003@e.ntu.edu.sg； {qizhangcs,rhli,wanggr}@bit.edu.cn；wentao.zhang@pku.edu.cn

arXiv:2303.03705v1 [cs.DS] 7 Mar 2023

摘要——极大双团枚举是二分图数据分析中的一个基础问题。现有的双团枚举方法主要针对无属性二部图，同时忽略了图属性的公平性。本文首次将公平性引入双团模型，并研究公平性感知双团枚举问题。具体而言，我们提出了两种公平性感知双团模型，分别称为单侧公平双团和双侧公平双团。为了高效枚举所有单侧公平二分团，我们首先提出两种非平凡的剪枝技术，分别称为公平α-β 核剪枝和彩色公平 α-β 核剪枝，在不损失准确性的情况下减小图规模。随后，我们设计了一种称为 FairBCEM 的分支定界算法，用于在缩减后的二分图上枚举所有单侧公平二分团。为进一步提高效率，我们提出了一种结合精心设计的组合枚举技术的高效分支定界算法。值得注意的是，我们的所有技术也可以扩展用于枚举所有双侧公平双团。我们还通过约束每种属性的顶点数与顶点总数之比，对这两种公平性感知双团模型进行了扩展，并给出了相应的枚举算法。在五个大型真实世界数据集上的大量实验结果表明，我们的方法具有良好的效率、有效性和可扩展性。

I. 引言

二分图 $G(U、V、E)$ 包含两个不相交的顶点集 U 和 V 以及一个边集 E，其中每条边都连接 U 中的一个节点和 V中的一个节点。许多现实世界网络，例如在线用户-物品网络 [1]–[5] 和基因共表达网络 [6]–[9]，都可以建模为二分图。近年来，由于极大双团枚举 [6], [10]–[12], 、蝴蝶计数 [13]–[16], 和最大二部团搜索 [17]–[20] 等众多现实应用，二分图分析问题受到了广泛关注。

近年来，公平性这一概念也在数据分析相关领域得到广泛研究 [21]–[24]。许多现有研究表明，有偏差的机器学习模型可能会对受歧视群体造成歧视，例如性别偏见和种族偏见 [25]–[28]。为解决这一问题，人们提出了多种方法（例如群体公平性和个体公平性 [21], [29]–[30], 等）。尽管这些方法在数据分析申请中有效，但图数据分析中的公平性 [31] 仍未得到充分研究。一个值得注意的例子是，Pan 等人提出了两种公平感知极大团模型，用于在属性图中发现公平社群 [31]。然而，这些模型主要针对传统属性图设计，无法直接推广到本文研究的其他类型图，例如二分图。

本文主要关注属性二部图，这是因为许多真实图（例如在线客户-产品网络）都可以建模为属性二部图。我们将公平性概念引入经典双团模型，并研究在属性二部图中挖掘公平性感知二部团的问题。这里，二分团是二分图的一个子图，其中属于不同侧的每一对节点之间都有一条边。需要注意的是，属性二部图上侧和下侧的节点通常具有不同类型的属性。公平性可以定义在节点的一侧，也可以定义在节点的两侧。因此，我们提出了两种用于刻画二部图中二分团公平性的新模型，分别称为单侧公平双团和双侧公平双团。单侧公平双团是指一侧节点满足公平性属性的二分团，并且它还是满足该属性的极大子图。也就是说，每种属性对应的顶点数量不少于阈值 β，且各种属性对应的顶点数量之间的最大差异不超过阈值 δ。类似地，双侧公平双团是指在两侧都保证公平性的二分团，并且它还是满足该属性的极大子图。在双侧公平双团中，上侧和下侧每种属性对应的顶点数量分别不少于阈值 α 和 β，且各种属性对应的顶点数量之间的最大差异不超过阈值 δ。值得注意的是，通过引入公平性比例 θ，单侧公平双团和双侧公平双团都可以扩展为比例公平双团模型。具体而言，阈值 θ 要求在公平侧，每种属性对应的顶点数量占顶点总数的比例不低于 θ。

在二分图中挖掘公平二部团有多种应用。例如，在科学合作网络（如 DBLP）中，我们可能希望找到一支初级和高级专家人数相近、研究领域也各不相同的团队。通过在作者-出版物网络中挖掘双侧公平双团，可以识别出这样的团队，因为双侧公平双团能够确保团队中初级和高级研究人员人数相近，且研究领域各不相同。在职位推荐系统（如 Jobs）中，可能存在国籍偏见。也就是说，即使外国人拥有更高的学历和更丰富的工作经验，他们也可能被推荐较不热门的职位。电影推荐系统（如 Movies）中也存在同样的问题，

其中存在曝光偏差。其直观表现是，即使受欢迎程度相同，已经热门的电影通常也比相对较新的电影获得更多推荐机会。为消除这些偏差，我们可以通过在职位侧和电影侧定义公平性，挖掘单侧公平二部团，以确保推荐结果不受国籍或时间因素影响。

尽管我们的公平二分团模型具有实际意义，目前尚无现有方法能够从二分图中挖掘所有单侧公平二分团或双侧公平双团。此外，我们证明，在二分图上枚举所有单侧公平二分团或双侧公平双团是 NP-hard 问题。为解决这些问题，我们首先提出一种名为 FairBCEM 的分支定界算法，并设计了两种精心设计的剪枝技术，用于枚举所有单侧公平二分团。为进一步提高效率，我们提出一种新颖的 FairBCEM++ 算法：该算法首先枚举所有极大二部团，然后使用一种精心设计的组合枚举技术，在所有极大二部团的集合中枚举所有结果，而非在原始二分图中进行枚举。我们证明，所有这些技术也可以扩展用于解决双侧公平二部团枚举问题。总而言之，我们的贡献如下。

新模型。我们提出单侧公平双团模型和双侧公平双团模型，用于刻画紧密二部子图的公平性。此外，我们还提出比例单侧公平双团模型和比例双侧公平双团模型，该模型考虑每种属性的顶点数占顶点总数的比例。据我们所知，我们是首个将公平性概念引入二分图双团挖掘任务的研究者。

新颖算法。为枚举所有单侧公平二分团，我们首先提出一种公平 α-β 核剪枝技术，以剪去原始二分图中不具潜力的节点。随后，我们通过在公平侧顶点上构建二跳图，并应用多彩核剪枝技术来缩减公平侧顶点，据此提出一种名为多彩 α-β 核剪枝的剪枝技术。我们提出一种名为 FairBCEM 的分支定界算法，用于枚举所有单侧公平双团。为进一步提升性能，我们开发了一种名为 FairBCEM++ 的新算法，该算法以极大二部团作为候选集，再通过精心设计的组合枚举技术，在这些候选集中枚举所有单侧公平双团。此外，我们还将所提出的剪枝技术和枚举算法扩展到双侧公平双团枚举问题，由此得到基础枚举算法 BFairBCEM 和改进算法 BFairBCEM++。此外，我们还提出名为 FairBCEMPro++ 和 BFairBCEMPro++ 的算法，用于枚举所有比例单侧公平双团和比例双侧公平双团。

大量实验。我们使用五个真实世界网络开展了大量实验，以评估所提出算法的效率和有效性。结果表明：(1) 用于单侧公平双团枚举和双侧公平双团枚举的剪枝技术能够显著剪除无望的顶点；(2) 对于单侧公平双团枚举，FairBCEM++ 的速度至少比FairBCEM快两个数量级；(3) 对于双侧公平双团枚举，BFairBCEM++ 的速度约为 BFairBCEM的3至100倍；(4) 我们改进后的两种算法都能处理包含7,577,304个节点和12,282,059条边的大型二分图。此外，我们在DBLP、Jobs和Movies上开展了三项案例研究，以评估所提方案的有效性。结果表明，单侧公平双团和双侧公平双团都能在DBLP中发现有意义且有趣的公平社群，并在Jobs和Movies中生成公平推荐结果。为便于复现，本文源代码已发布于 https://github.com/Heisenberg-Yin/fairnesss-biclique。

二、预备知识

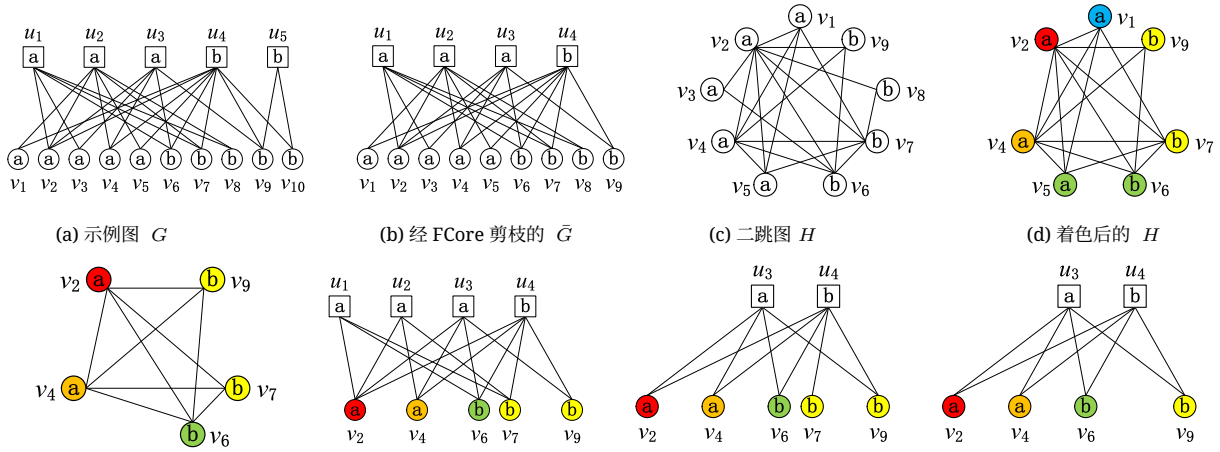
设 $G=(U,V,E,A)$ 为一个无向、无权的属性二部图，其中 $U(G)$ 和 $V(G)$ 是两个不相交的顶点集，$E(G) \subseteq U(G) \times V(G)$ 表示 G 的边集。通常，我们分别将顶点集 $U(G)$ 和 $V(G)$ 称为 G 的上侧和下侧。$A(G)=\{A_U, A_V\}$ 是 G 的属性集，其中 A_U 是 $U(G)$ 中顶点的属性，而 A_V 是 $V(G)$ 中顶点的属性。对于任意顶点 u，我们使用 $u.val$ 表示其属性值。设 $A(U)$ 为 A_U 的所有属性值构成的集合，即 $A(U)=\{u.val|u \in U(G)\}$。类似地，我们记 $A(V)=\{u.val|u \in V(G)\}$。$A(U)$ 和 $A(V)$ 的基数分别为 A_n^U 和 A_n^V。我们主要关注 G 的每一侧均具有二维属性的情形，即 $A_n^U=A_n^V=2$。不失一般性地，我们记 $A(U)=\{a_i^U|0 \le i < A_n^U\}$，并记 $A(V)=\{a_i^V|0 \le i < A_n^V\}$。图 G 中顶点 u 的邻居集合记为 $N(u,G)=\{v|(u,v) \in E(G)\}$，而 u 在 G 中的度表示为 $D(u,G)=|N(u,G)|$。给定一个顶点集 S，我们使用 $N(S)=\{v|v \in N(u)\ \forall u \in S\}$ 表示 S 的邻居集合。集合 S 中属性值为 a_i^* 的顶点数量为 $S_{a_i^*}=\{v|v.val=a_i^*\}$，其中符号“$*$”可以是 U 或 V。当上下文明确时，在上述记号中省略符号 G。

定义 1：（二分团）给定一个二分图 $G(U,V,E)$，子图 C 是二分团，当且仅当：(1) $E(C)=U(C) \times V(C)$；(2) $U(C) \subseteq U(G)$；(3) $V(C) \subseteq V(G)$。

定义 2：（极大双团）给定一个二分图$G(U,V,E)$和一个子图 C，C 是极大双团，当且仅当：(1) C 是二分团；(2) 不存在其他满足条件 (1) 的二分团 $C' \supset C$。

下面，我们介绍两种新颖的公平感知双团模型，即单侧公平双团（SSFBC）和双侧公平双团（BSFBC）。不失一般性，在 SSFBC 模型中，我们将 V 视为公平侧；在 BSFBC 中，则将 U 和 V 都视为公平侧。

定义 3：（单侧公平双团）给定一个属性二部图 $G(U,V,E,A)$ 和三个整数 α,β,δ，若 G 的一个二分团 $C(U,V,E,A)$ 满足：(1) $|C(U)| \ge \alpha$；(2) $\forall a_i^V \in A(V)\ |C(V)_{a_i^V}| \ge \beta$ 且



(a) 示例图 G　(b) 经 FCore 剪枝的 $\bar{G}$　(c) 二跳图 H　(d) 着色后的 H

(e) 自我彩色2-核 $\bar{H}$　(f) 经 $\bar{G}$ 剪枝的 $\bar{H}$ (g) 经 FCore 剪枝的 $\bar{G}$　(h) 经 CFCore 剪枝后的 $\bar{G}$

图 1. FCore 和 CFCore 在示例图 G上的剪枝过程。

$\forall a_i^V, a_j^V \in A(V)$, $||C(V)_{a_i^V}| - |C(V)_{a_j^V}|| \leq \delta$; (3) 不存在满足 (1) 和 (2) 的二分团 $C' \supset C$。

定义 4: （双侧公平双团）给定一个带属性的二分图 $G(U, V, E, A)$ 和三个整数 α, β, δ，若 G 的一个二分团 $C(U, V, E, A)$ 满足: （1）$\forall a_i^U \in A(U)$ $|C(U)_{a_i^U}| \geq \alpha$ 且 $\forall a_i^U, a_j^U \in A(U)$, $||C(U)_{a_i^U}| - |C(U)_{a_j^U}|| \leq \delta$; （2）$\forall a_i^V \in A(V)$ $|C(V)_{a_i^V}| \geq \beta$ 且 $\forall a_i^V, a_j^V \in A(V)$, $||C(V)_{a_i^V}| - |C(V)_{a_j^V}|| \leq \delta$; （3）不存在满足（1）和（2）的二分团 $C' \supset C$，则称其为双侧公平双团。

示例 1: 考虑图 1(a) 中的属性二部图 $G = (U, V, E, A)$。对于上侧 $U(G)$，属性 A_U 的值分别用方框中的 a 和 b 表示。对于下侧 $V(G)$，属性 A_V 的属性值分别用圆圈中的 a 和 b 表示。假设 $\alpha = 1, \beta = 2$和 $\delta = 1$。根据定义 3，由顶点集 $\{u_3, u_4, v_2, v_4, v_6, v_9\}$ 诱导的子图 C_S 是 G的单侧公平双团，而由 $\{u_3, u_4, v_2, v_4, v_6, v_9\}$ 诱导的子图 C_B是一个 BSFBC。显然， C_B 是 C_S的子图，这意味着 BSFBC 必须包含在单侧公平双团中。□

此外，公平性不仅考虑每种属性对应的顶点数量，还考虑公平侧上每种属性对应的顶点数量与顶点总数之比。下面，我们提出 SSFBC 和 BSFBC 的两种扩展模型，即比例单侧公平双团（PSSFBC）和比例双侧公平双团（PBSFBC），通过引入公平比例阈值 θ进一步保证公平性。

定义 5: （比例单侧公平双团）给定一个属性二部图 $G(U, V, E, A)$、三个整数 α, β, δ 和一个浮点数 θ，若 G 的二分团 $C(U, V, E, A)$ 满足以下条件，则称其为比例单侧公平双团: (1) $|C(U)| \geq \alpha$; (2) $\forall a_i^V \in A V$ $|C V_{a_i^V}| \geq \beta$ $\forall a_i^V a_j^V \in A V$ $||C V_{a_i^V}| - ( ) ( )$ and $, ( ), |C V_{a_j^V}|| \leq \delta$ 3 $\forall a_i^V \in A V$ $|C V_{a_i^V}| / |C V| \geq \theta$ 4 $( ) ( ) ;( ) ( ) ( ) ;( )$; 且不存在满足条件 (1) 、 (2) 和 (3) 的二分团 $C' \supset C$。

定义 6: （比例双侧公平双团）给定一个属性二部图 $G(U, V, E, A)$、三个整数 α, β, δ 和一个浮点数 θ，若 G 的二分团 $C(U, V, E, A)$ 满足以下条件，则称其为比例双侧公平双团: （1）$\forall a_i^U \in A(U)$ $|C(U)_{a_i^U}| \geq \alpha$ 且 $\forall a_i^U, a_j^U \in A(U)$, $||C(U)_{a_i^U}| - |C(U)_{a_j^U}|| \leq \delta$; （2） $\forall a_i^V \in A V$ $|C V_{a_i^V}| \geq \beta$ $\forall a_i^V a_j^V \in A V$ $||C V_{a_i^V}| - ( ) ( )$ 且 $, ( ), ( ) ( ) ; |C V_{a_j^V}|| \leq \delta$ 3 $\forall a_i^V \in A V$ $|C V_{a_i^V}| / |C V| \geq \theta$ $( ) ( ) ( ) ( ), \forall a_i^U \in A U$ $|C U_{a_i^U}| / |C U| \geq \theta$ 4 $( ) ( ) ( )$; ( ) 不存在满足条件（1）、（2）和（3）的二分团 $C' \supset C$。

问题定义。给定一个属性二部图$G(U, V, E, A)$、三个整数 α, β, δ 和一个浮点数 θ，我们的目标是在 G 中找出所有单侧公平双团、比例单侧公平双团、双侧公平双团和 PBSFBC。

计算困难性。我们首先讨论单侧公平双团枚举问题的计算困难性。考虑一个特殊情况： $\alpha = 0$、 $\beta = 0$、 $\delta = n$，其中 n 是图规模。显然，在这些参数下，单侧公平双团枚举问题退化为传统极大二分团枚举问题，而该问题是 NP-hard。因此，找出所有单侧公平双团也是一个 NP-hard 问题。双侧公平双团枚举问题比枚举所有单侧公平双团更具挑战性，因为双侧公平双团的数量通常远多于单侧公平双团。根据定义可知，双侧公平双团始终包含于某个单侧公平双团中。反之，单侧公平双团不一定是双侧公平双团。

与传统双团枚举问题相比，公平性感知双团枚举问题更加困难。首先，单侧公平双团和双侧公平双团模型都不满足遗传性质。也就是说，由于属性约束，SSFBC 或 BSFBC 的子图并不总是公平子图。因此，检验单侧公平双团和双侧公平双团是否为极大双团都更加困难。其次，公平性感知双团的数量通常多于传统极大双团，导致枚举所有公平性感知双团需要更长时间。例如，在 IMDB 上，参数为 $\alpha = 8$、 $\beta = 10$、 $\delta = 2$ 时，极大双团和单侧公平双团的数量分别为 12,614 和 3,502,746。在 $\alpha = 4$、 $\beta = 6$、 $\delta = 2$ 的情况下，我们可以找到 42,023 个极大双团和 11,091,721 个双侧公平双团。

下面，我们分析查找所有单侧公平双团和双侧公平双团的时间复杂度下界。我们首先介绍一个

重要定理，该定理已在 [32] 中得到证明。

定理 2.1：每个含有 n 个顶点的二分图至多包含 $2^{n/2}$ 个二部团 [32]。

在最坏情况下，所有二部团都可能满足定义 3 中的 α 和 β约束，因此我们只考虑参数 $delta$。给定一个二分团 $C(U,V,E,A)$，不失一般性，我们假设 $|C(V)_{a_1^V}| = |C(V)_{a_2^V}| + n_1$和$|C(U)_{a_1^U}| = |C(U)_{a_2^U}| + n_2$成立，其中$n_1 > \delta$和 $n_2 > \delta$。那么，SSFBC 的数量为 $\binom{|C(V)_{a_2^V}|}{|C(V)_{a_1^V}|}$，其最大值为 $\binom{|C(V)_{a_1^V}|}{\lfloor |C(V)_{a_1^V}|/2 \rfloor}$。类似地，BSFBC 的最大数量等于 $\binom{|C(V)_{a_1^V}|}{\lfloor |C(V)_{a_1^V}|/2 \rfloor}\binom{|C(U)_{a_1^U}|}{\lfloor |C(U)_{a_1^U}|/2 \rfloor}$。由于共有 $2^{n/2}$ 个二部团（定理 2.1），且 $C(V)$ $C(u) \leq n$成立，因此，找出所有 SSFBC 和 BSFBC 分别至少需要 $O(C_n^{\lfloor n/2 \rfloor} * 2^{n/2})$ 和 $O(C_n^{\lfloor n/2 \rfloor})^2 * 2^{n/2})$ 时间，因为算法需要输出这些公平二部团。

对于所有 PSSFBC 和 PBSFBC 的枚举，其时间复杂度下界可以通过类似于寻找 SSFBC 和 BSFBC 的方法轻松推导。由于篇幅限制，我们省略相关分析。

III. 单侧公平双团枚举

本节首先介绍两种非平凡的剪枝技术，分别称为公平 α-β 核剪枝和彩色公平 α-β 核剪枝，用于缩小图的规模。随后，我们提出两种分支限界枚举算法 FairBCEM 和 FairBCEM++，用于枚举所有单侧公平二分团。最后，我们开发了 FairBCEMPro++ 算法，以解决 PSSFBC枚举问题。

A. 公平 α-β 核剪枝

下面首先给出属性度的定义，该定义对于推导公平 α-β 核剪枝技术至关重要。

定义 7：（属性度）给定一个属性二部图 $G = (U, V, E, A)$ 和一个属性值 $a_i \in A(U) \cup A(V)$。顶点 u 的属性度记为 $D_{a_i}(u, G)$，是 u 的邻居中属性值为 a_i 的顶点数量，即 $D_{a_i}(u, G) = |\{v | v \in N(u)\ v.val = a_i\}|$。

定义 8：（公平 α-β 核）给定一个属性二部图 $G = (U, V, E, A)$，若子图 $H = (L, R, E, A)$ 满足以下条件，则称其为公平 α-β 核：(1) $D_{a_i}(u, H) \geq \beta, u \in L, a_i \in A(V)$；(2) $D(v, H) \geq \alpha, v \in R$；(3) 在 G 中不存在满足条件 (1) 和 (2) 的子图 $H' \supset H$。

根据定义 8，我们得到以下引理。由于篇幅所限，本文省略所有证明。

引理 1：给定一个属性二部图 $G = (U, V, E, A)$ 和两个整数 α, β，任何单侧公平双团都必须包含在一个公平 α-β 核中。

根据引理 1，我们提出了一种公平的 α-β 核计算算法，称为 FCore，用于剪除那些必定不属于任何单侧公平双团、因而不具备潜力的顶点。FCore 的伪代码概述于算法 1 中，它是经典核分解算法 [33]、[34]的一种变体。具体而言，算法使用优先队列 Q 来维护

算法 1：FCore

```
输入：G = (U, V, E, A)，两个整数 α, β
输出：公平的 α-β 核 G^
1  令 Q 为一个优先队列；Q ← ∅;
2  对于 u ∈ U 执行
3      对于 v ∈ N(u) 执行 D_{v.val}(u)++;
4      D_min(u) ←最小值{D_{a_i^V}(u) | a_i^V ∈ A(V)};
5  对于 u ∈ U 执行
6      如果 D_min(u) < β 则 Q.push(u); 从 G中移除 u;
7  对于 v ∈ V 执行
8      对于 u ∈ N(v) 执行 D(v)++;
9  对于 v ∈ V 执行
10     如果 D(v) < α 则 Q.push(v); 从 G中移除 v;
11 当 Q ≠ ∅ 时执行
12     u ← Q.pop();
13     对于 v ∈ N(u), 执行
14         如果 v 未被移除，则
15             如果 v ∈ U，则
16                 D_{u.val}(v)--;
17                 D_min(v) ←最小值{D_{a_i^V}(v) | a_i^V ∈ A(V)};
18                 如果 D_min(v) < β 则
19                     Q.push(v); 从 G 中移除 v;
20             如果 v ∈ V 则
21                 D(v)--;
22                 如果 D(v) < α，则
23                     Q.push(v); 从 G中移除 v;
24                 Q.push(v); 从 G中移除 v;
25 G ← G的剩余图；
26 返回 G;
```

将在剥离过程中被移除的顶点（第 1 行）。FCore 首先分别计算上侧和下侧顶点的属性度和度，以初始化 Q （第 2–10 行）。根据定义 8，对于顶点$u \in U$ （即上侧顶点），一旦其最小属性度 $D_{min}(u)$ 小于 β，FCore 就会将其从 G 中移除；对于$v \in V$（即下侧顶点），一旦其度 $D(v)$ 小于 α，就会将 v 从 G 中移除。随后，算法根据顶点的度和属性度，通过迭代地从剩余图中剥离顶点，计算公平的 α-β 核（第 11–24 行）。最后，FCore 返回剩余图 G 作为公平的 α-β 核。容易证明，FCore 使用 $O(U \times A_n^V + V)$ 空间，耗时$O(E + V)$。

B. 彩色公平 α-β 核剪枝

公平 α-β 核剪枝可能并不十分有效，因为它只采用属性度约束，而忽略了团的性质。为此，我们通过建立我们的问题与 [31]提出的弱公平团模型之间有趣的联系，提出了一种更强大的剪枝技术，称为彩色公平 α-β 核（CFCore）剪枝。

回顾定义 3，在单侧公平双团 C 中，$C(V)$ 中的任意两个顶点至少有 α 个公共邻居。因此，我们可以在 G 的公平侧构造一个二跳图 $H(V, E, A)$，具体如下。我们保留 H 中的顶点，即 G 的下侧中的顶点，也就是 $H(V) = G(V)$ 和 $A = A_V$。给定两个顶点 $v_i, v_j \in V(H)$，如果 v_i 和 v_j 在 G 中的公共邻居数不少于 α，我们就在 H 中连接 v_i 和 v_j

因为 v_i 和 v_j 可能出现在同一个单侧公平双团中。借助二跳图 H，我们有如下观察。

观察 1：给定属性二部图 G 及其二跳图 H。对于任意单侧公平双团 C，C（V）中的顶点在 H 中构成一个团 $\hat{C}$，其中属性值等于 a_i^V 的顶点数量不少于 β。

$\hat{}$ 根据观察 1，团 C 满足 [31]中弱公平团模型的公平性约束。由于弱$\hat{}$公平团是极大的，C 必须包含在一个弱公平团中。因此，我们可以应用 [31] 中提出的多彩核剪枝技术，对 H 中无法构成弱公平团的无希望顶点进行剪枝。然而，[31] 中的多彩核剪枝并未考虑顶点自身的属性值。下面我们通过纳入顶点属性，给出多彩度和多彩核的变体，分别称为自我多彩度和自我多彩核。

定义 9：（自我多彩度）给定属性图$G=$（V、E、A）和属性值 $a_i \in A$。顶点 u的自我多彩度记为 ED_{a_i}（u、G），其定义为 u 以及属性值为 a_i的 u的邻居所包含的颜色数量，即ED_{a_i}（u、G）$=|\{color\ (v)\ |v \in N\ (u)\ \cup\{u\},\ v.val=a_i\}|$。

在定义 9 中，每个节点的颜色可以通过经典的贪心图着色算法 [35], 获得，该算法确保两个相邻节点的颜色不同。令$ED_{\min}(u,G)$表示顶点 u 的最小自我多彩度，即 $ED_{\min}(u,G)=\min\{ED_{a_i}(u,G)|a_i \in A\}$。当上下文清楚时，我们省略 $ED_{a_i}(u,G)$ 和 $ED_{\min}(u,G)$ 中的符号 G。

定义 10：（自我多彩 k-核）给定一个属性图 $G=(V,E,A)$ 和一个整数 k，若其子图 $H=(V_H,E_H,A)$ 满足以下条件，则称其为自我多彩 k-核：(1) 对每个顶点 $u \in V_H$，$ED_{\min}(u,H) \geq k$；(2) 不存在满足条件 (1) 和 $H' \supset H$ 的子图H'。

根据定义 10，我们得到以下引理。

引理 2：给定一个属性二部图 G、其二跳图 H 以及参数 α、β、δ。对于任意单侧公平二分团 C，$C(V)$ 中的顶点必须包含在 H 的自我多彩 β-核中。

根据引理 2，我们可以基于公平侧 V构建二跳图 H，并通过计算 H 的自我彩色β-核，剪除 $G(V)$ 中无法构成单侧公平双团的顶点。显然，自我彩色 β-核的规模小于 H。这意味着下侧的一些顶点可以从 G 中移除，因此我们可以进一步应用 FCore，剪除 G 上侧和下侧的顶点。基于这一思路，我们提出了一种彩色公平 α-β 核剪枝算法，即 CFCore，如算法 2 所示。CFCore 算法的工作过程如下。首先，根据引理 1 执行 FCore（算法 1），计算公平 α-β 核 G （第 1 行）。随后，CFCore 在公平（下）侧 $G(V)$ 上构建二跳图 H （算法 3），并删除度小于 $A_n^V \times \beta - 1$ 的顶点，因为这些顶点显然无法构成单侧公平双团（第 3-5 行）。之后，CFCore 对 H 使用贪心着色，根据度 [33], [34], 的顺序为顶点着色，并计算

算法 2：CFCore

输入：$G=(U,V,E,A)$，两个整数 α、β
输出：剪枝后的图 $\hat{G}$

```
1  G(U,V,E,A) ← FCore(G,α,β);
2  令 Q 为一个优先队列；Q ← ∅;
3  H(V,E,A_V) ← Construct2HopGraph(G,α,G(V));
4  对 u ∈ H(V) 执行
5      若 D(u,H) < A_n^V × β − 1 则从 H中移除 u；
6  通过调用基于度的贪心着色算法，对 H 中的所有顶点进行着色;
7  对 u ∈ H(V) 执行
8      对 v ∈ N(u) ∪ {u} 执行
9          若 M_u(v.val, color(v)) = 0 则 ED_{v.val}(u)++;
10         M_u(v.val, color(v))++;
11     ED_min(u) ←最小值{ED_{a_i^V}(u)|a_i^V ∈ A(V)};
12 对于 u ∈ H(V) 执行
13     若 ED_min(u) < β 则 Q.push(u); 从 H中移除 u；
14 当 Q ≠ ∅ 时
15     u ← Q.pop();
16     对于 v ∈ N(u,H)
17         如果 v 未被移除，则
18             M_v(u.val, color(u))--;
19             如果 M_v(u.val, color(u)) ≤ 0，则
20                 ED_{u.val}(v) ← ED_{u.val}(v) − 1;
21                 ED_min(v) ← min{ED_{a_i^V}(v)|a_i^V ∈ A(V)};
22                 if ED_min(v) < β 则
23                     Q.push(v); 从 H 中移除 v；
24 剩余图 H 的自我彩色 β-核 H̄ ← ；
25 对 u ∈ Ḡ(V) − H̄(V) 执行
26     从 Ḡ(V) 中移除 u；
27 Ĝ ← FCore(Ḡ=(U,V,E,A) α,β);
28 返回 Ĝ;
```

算法3：Construct2HopGraph

输入：$G=(U,V,E,A)$，一个整数 α，公平侧 V
输出：基于公平侧 V 的二跳图 H

```
1  令 H=(V=G(V) E=∅,A=A_V) 为一个属性图;
2  对于 v ∈ G(V) 执行
3      初始化一个数组 C，其值为 C[i]=0, 1 ≤ i ≤ |G(V)|;
4      对于 u ∈ N(v,G) 执行
5          对于 w ∈ N(u,G) 执行
6              如果 w ≠ v 则 C[w] ← C[w] + 1;
7      对于 u ∈ G(V) 执行
8          如果 C[u] ≥ α 且 u < v 则 E(H) ← E(H) ∪ (u,v);
9  返回 H;
```

通过根据顶点的自我多彩度从剩余图中迭代剥离顶点，得到自我多彩 β-核 H （第 6–24 行）。根据引理 2，CFCore 会安全地移除不包含在自我多彩 β-核 H 中的顶点。它还会再次执行 FCore（算法 1），以减少 G 的上侧和下侧顶点（第 25–27 行）。最后，CFCore 返回剪枝后的图 $\hat{G}$，其中包含所有单侧公平二分团。算法 2 的时间复杂度为 $O(E+V+\sum_{u\in U} d(u,G)^2+\sum_{v\in V} d(v,G)^2)$，空间复杂度为 $O(V \times A_n^V \times color)$。

示例 2：考虑图 1(a) 中的二分图 $G=(U,V,E,A)$。假设我们设置了 $\alpha=2,\beta=2$。CFCore 首先执行 FCore，计算公平的 α-β 核，记为 G，如图 1(b) 所示。然后，它为 G 的公平侧 V构建二跳图 H （即圆圈中的顶点），如图 1(c) 所示。图 1(c) 中有两个邻居的顶点 v_3

无法构成单侧公平双团，因此我们将其从H中移除。这是因为单侧公平双团 C 在下侧 V至少包含 $A_n^V \times \beta$ 个顶点，这要求 $V(C)$ 中的顶点在二跳图 H中至少有 $A_n^V \times \beta - 1 = 2 \times 2 - 1 = 3$ 个邻居。类似地，图 1(c) 中的顶点 v_8不属于单侧公平双团，因此我们也将 v_8 从 H中移除。度剪枝后，我们使用贪心着色算法 [35] 为 H着色，如图 1(d) 所示，然后计算自我彩色2-核 H。以 v_1 为例，我们推导出 v_1的自我彩色度，即 $ED_a(v_1, H) = 4$和 $ED_b(v_1, H) = 1$。此外，我们有 $ED_{\min}(v_1, H) = 1 \leq \beta = 2$。因此，可以安全地移除 v_1，因为根据引理 2，v_1 不属于自我彩色2-核，也不属于单侧公平双团。图 1(e) 展示了自我彩色2-核 H。我们使用 H 对二分图 G进行剪枝。剩余图如图 1(f) 所示。显然，在下侧，剪枝后的 G 只有 5 个顶点，而图 1(b) 中先前的 G 有 9 个顶点。接着，CFCore 再次执行 FCore，移除 G 中的顶点，如图 1(g) 和图 1(h) 所示。CFCore 剪枝后的最终图如图 1(h) 所示，与图 1(a) 中的原始图相比规模显著减小。□

C. FairBCEM算法

在介绍FairBCEM算法之前，我们首先给出两个重要定义，即公平集和极大公平子集。

定义11：（公平集）给定一个属性集 S，其属性值属于 A，以及两个整数 k、δ，如果 S满足以下条件，则称其为公平集：（1）$\forall a_i \in A$、$|S_{a_i}| \geq k$；（2）$\forall a_i$、$a_j \in A$、$||S_{a_i}| - |S_{a_j}|| \leq \delta$。

定义12：（极大公平子集）给定一个属性集$\hat{S}$，其属性值属于 A，以及两个整数 k、δ，如果 $S \subseteq S$ 满足以下条件，则称其为极大公平子集：（1）S 是基于 k、δ的公平集；（2）不存在满足 $S \subset S$的公平集 $S \subset S$。

如算法4所示，我们提出了一种高效算法，用于识别集合$\hat{S}$ 是否是集合 S 的极大公平子集。显然，当满足不存在能够添加到$\hat{S}$中的 $S - \{\hat{S}\}$ 的子集，且不会损害其公平性时，S 是极大公平子集。

借助 CFCore剪枝技术，我们提出了 FairBCEM算法，该算法基于分支定界搜索方法枚举所有单侧公平双团。在 FairBCEM 中，有四个重要集合：L、R、P、Q，它们控制搜索树的生成。具体而言，我们用R 表示下侧 V 中当前已找到且可扩展为单侧公平双团的顶点。L 是上侧 U 中的顶点集，其中每个顶点都是 R中所有顶点的邻居。P 是 V中的候选集，可用于在搜索树中扩展 R。Q 是这样一个顶点集，其中每个顶点都可用于扩展 R，但已在先前的搜索路径中访问过。下面，我们给出一些观察结果来解释 FairBCEM算法。

观察2：如果 $\forall a_i^V \in A\ (V)$，我们可以发现至少存在一个满足 $\forall u \in L$的顶点 $v \in Q$，其 $v.val = a_i^V$为 $\forall u \in L$，$(u,\ v) \in E$，R 不是极大的，因此我们可以结束当前搜索及所有更深层的搜索。

观察3：给定一个公平集 R，如果不存在一个与 L 完全连接且可以

算法4：MFSCheck

输入：集合 S，$\hat{S}$，属性值集合 A，两个整数 k, δ
输出：true：S 是极大公平子集；false：$\hat{S}$ 不是极大公平子集

```
1 如果 ∃a_i ∈ A, S_{a_i} < k 则返回 false;
2 C ← S − Ŝ;
3 如果 ∀a_i ∈ A, |C_{a_i}| > 0 则返回 false;
4 对于 a_i ∈ A，执行
5 如果|C_{a_i}| > 0，则
6     如果∃u ∈ C_{a_i}，Ŝ ∪ {u} 是一个公平集，则返回 false;
7 返回 true;
```

加入 R 而不破坏公平性的顶点集 L，那么 (L, R) 就是一个单侧公平双团。

观察4：如果 P 中的所有节点都与 R 完全连接，且 $R \cup P$ 是公平集，那么我们可以将 P 中的所有顶点加入 R，而不会丢失解。

观察5：$|L| < \alpha$ 或 $\exists a_i^V \in A\ V\ |R_{a_i^V}| + |P_{a_i^V}| < \beta$：如果或()，我们可以终止当前搜索分支。

基于上述观察，单侧公平二分团枚举的 FairBCEM 算法如算法 5 所示。该算法首先使用 CFCore 剪枝移除不可能属于单侧公平二分团的顶点，并初始化四个集合 L、R、P 和 Q，然后调用 BackTrackFBCEM过程，通过分支限界技术查找所有单侧公平二分团。在 BackTrackFBCEM 中，P 中的每个顶点 x 都用于扩展当前找到的 R。加入 x 后，必须更新 L，以排除那些与 x 不相邻的顶点，因为 L 中的每个顶点都与 R 中的所有顶点相邻（第 7-8 行）。变量 $flag$ 初始化为 true，用于指示当前分支中是否存在单侧公平二分团。我们分别将 Q 和 P 中与 L 完全连接的顶点记为 Q^{FC} 和 P^{FC}，并用它们检查 R 的极大性。显然，如果 $|L'| < \alpha$，则无法找到单侧公平二分团，因为这违反了定义 3 对上侧顶点数量的限制，因此我们将 $flag$ 设为 false（第 9 行）。接着，BackTrackFBCEM过程依据观察2，使用集合 Q 判断 R 是否为极大的，并维护 $flag$ 的值和集合 Q'（第 10-15 行）。一旦 $flag$ 等于 false，当前分支中就不存在单侧公平二分团；我们将 x 从 P 移至 Q，以表示 x 已经搜索过（第 29-30 行）。否则，BackTrackFBCEM 使用候选集 P 计算集合 P' 和 P^{FC}（第 17-20 行）。如果 $P' = P^{FC}$，则 P 中的所有顶点都与 R' 完全连接，因此我们可以根据观察4直接检查 $(L', R' \cup P^{FC})$ 是否为单侧公平二分团。若是，BackTrackFBCEM 会将二分团 $(L', R' \cup P^{FC})$ 加入结果集 Res，并将 P' 和 P^{FC} 更新为空集（第 21-23 行）。之后，该过程通过算法4判断 R' 是否为 $R' \cup P^{FC} \cup Q^{FC}$ 的极大公平集，并根据观察 3 将 (L', R') 加入 Res（第 24-26 行）。随后，如果 $P' \neq \emptyset$ 且 $\forall a_i^V \in A(V)\ |R'_{a_i^V}| + |P'_{a_i^V}| \geq \beta$，BackTrackFBCEM 会使用新的 L'、R'、P'、Q' 执行下一步回溯（第 27-28 行）。最终集合 Res 保存了 G 中的所有单侧公平二分团（第 4 行）。

正确性分析。显然，我们会枚举所有可能的 R

算法 5：FairBCEM

输入：二分图 $G=(U,V,E,A)$，三个整数 α,β,δ
输出：所有单侧公平二分团的集合 Res

1 $G=(U,V,E,A) \leftarrow$ CFCore(G,α,β);
2 $L \leftarrow \hat{U}$; $R \leftarrow \emptyset$; $P \leftarrow \hat{V}$; $Q \leftarrow \emptyset$;
3 BackTrackFBCEM(L,R,P,Q);
4 返回 Res;
5 过程 BackTrackFBCEM(L,R,P,Q)
6 当 $P \neq \emptyset$ 时
7 $x \leftarrow$ 是 P 中的一个顶点; $flag \leftarrow true$;
8 $R' \leftarrow R \cup \{x\}$; $L' \leftarrow \{u \in L | (u, x) \in \hat{E}\}$;
9 如果 $|L'| < \alpha$，则 $flag \leftarrow false$;
10 对于 $u \in Q$，执行
11 $N(u) = \{v \in L' | (u, v) \in \hat{E}\}$;
12 如果 $|N(u)| = |L'|$，则 $Q^{FC} \leftarrow Q^{FC} \cup \{u\}$;
13 如果 $|N(u)| \geq \alpha$，则 $Q' \leftarrow Q' \cup \{u\}$;
14 如果 $\forall a_i^V \in A(V)$ $Q^{FC}_{a_i^V} > 0$ 则
15 $flag \leftarrow false$;
16 如果 flag 则
17 对于 $v \in P$、$v \neq x$ 执行
18 $N(v) = \{u \in L' | (u, v) \in \hat{E}\}$;
19 如果 $|N(v)| = |L'|$，则 $P^{FC} \leftarrow P^{FC} \cup \{v\}$;
20 如果 $|N(v)| \geq \alpha$，则 $P' \leftarrow P' \cup \{v\}$;
21 如果 $P^{FC} = P'$，则
22 如果 $(L', R' \cup P^{FC})$ 是公平单侧二部团，则
23 $R' \leftarrow R' \cup P^{FC}$; $P^{FC} \leftarrow \emptyset$; $P' \leftarrow \emptyset$;
24 如果 R' 是公平集，则
25 如果 R' 是 $R' \cup P^{FC} \cup Q^{FC}$ 的极大公平子集，则
26 $Res \leftarrow Res \cup (L', R')$;
27 如果 $P' \neq \emptyset$ 且 $\forall a_i^V \in A(V)$ $|R'_{a_i^V}| + |P'_{a_i^V}| \geq \beta$ 则
28 BackTrackFBCEM(L', R', P', Q');
29 $P \leftarrow P - \{x\}$;
30 $Q \leftarrow Q \cup \{x\}$;

算法 6：FairBCEM++

输入：二分图 $G=(U,V,E,A)$，三个整数 α,β,δ
输出：所有单侧公平二分团的集合 Res

1 $G=(U,V,E,A) \leftarrow$ CFCore(G,α,β);
2 $L \leftarrow \hat{U}$; $R \leftarrow \emptyset$; $P \leftarrow \hat{V}$; $Q \leftarrow \emptyset$;
3 BackTrackFBCEM++(L,R,P,Q);
4 返回 Res;
5 过程 BackTrackFBCEM++(L,R,P,Q)
6 当 $P \neq \emptyset$ 时执行
7 $x \leftarrow$ 是 P 中的一个顶点; $flag \leftarrow true$;
8 $R' \leftarrow R \cup \{x\}$; $L' \leftarrow \{u \in L | (u, x) \in \hat{E}\}$;
9 如果 $|L'| < \alpha$ 则 $flag \leftarrow false$;
10 对于 $u \in Q$ 执行
11 $N(u) = \{v \in L' | (u, v) \in \hat{E}\}$;
12 若 $|N(u)| = |L'|$，则 $flag \leftarrow false$; 中断;
13 若 $|N(u)| > 0$，则 $Q' \leftarrow Q' \cup \{u\}$;
14 $C \leftarrow C \cup \{u\}$;
15 若标志为真，则
16 对于 $v \in P, v \neq x$，执行
17 $N(v) = \{u \in L' | (u, v) \in \hat{E}\}$;
18 如果 $|N(v)| = |L'|$，则
19 $R' \leftarrow R' \cup \{v\}$;
20 $N^{lap}(v) = \{u | u \in L/L', (u, v) \in \hat{E}\}$;
21 如果 $|N^{lap}(v)| = 0$，则 $C \leftarrow C \cup \{v\}$;
22 如果 $|N(v)| \geq \alpha$，则 $P' \leftarrow P' \cup \{v\}$;
23 如果 (L', R') 是单侧公平双团，则
24 $Res \leftarrow Res \cup (L', R')$;
25 **else**
26 $\mathcal{R}' \leftarrow$ 组合($R', A(V)$ β, δ);
27 对于 $r' \in \mathcal{R}'$ 执行
28 如果 $N(r') = L$，则 $Res \leftarrow Res \cup (L', r')$;
29 如果 $P' \neq \emptyset$ 且 $\forall a_i^V \in A(V)$ $|R'_{a_i^V}| + |P'_{a_i^V}| \geq \beta$ 然后
30 BackTrackFBCEM++(L', R', P', Q');
31 $P = P - C$;
32 $Q = Q \cup C$;

基于集合 P、Q，所有单侧公平二分团都包含在枚举树中，因此我们的算法满足完备性。二分团的公平性和极大性分别由算法 5 第 22 行和第 25 行保证。此外，集合 Q 可确保每个单侧公平二分团只被枚举一次，因此我们的算法也满足无冗余性。总之，我们的 FairBCEM算法能够正确输出所有单侧公平二分团。

D. FairBCEM++算法

由于单侧公平二分团数量庞大，FairBCEM算法可能面临巨大的搜索空间。为进一步提高效率，我们提出了一种新算法，称为 FairBCEM++。该算法首先枚举所有极大二部团，然后使用组合枚举技术，在所有极大二部团的集合中找出所有单侧公平二分团。我们的算法基于以下关键观察：任何单侧公平二分团都必须包含在某个二分团中。

更具体地说，FairBCEM++ 首先找出所有满足 $|L| \geq \alpha$ 和 $R_{a_i^V} \geq \beta, \forall a_i^V \in A(V)$的极大团，然后枚举其中所有单侧公平二分团。FairBCEM++ 的伪代码如算法 6 所示。与 FairBCEM 类似，FairBCEM++ 使用 CFCore 剪枝移除不具潜力的顶点，然后执行 BackTrackFBCEM++ 回溯过程，以找出所有单侧公平二分团（第 1–3 行）。在 BackTrackFBCEM++的每次迭代中，

我们基于 MBEA++ 算法 [6] 的思路找出所有极大二部团；该算法每次将一组顶点（即集合 C）加入 R。具体而言，首先通过添加 x 扩展 R，并得到集合 L'，其中的顶点均与 x 相连（第 7–8 行）。随后，通过尝试将 u 中的每个顶点加入当前二部团，判断 (L', R') 是否为极大二部团。显然，如果不是，我们就可以终止当前搜索，因为任何单侧公平二分团都必须是某个二部团的子集（第 10–13 行）。否则，我们一次性将与 L' 中所有顶点相连的顶点从 P 移至 R'，并更新集合 C 和 P'（第 16–22 行）。对于 (L', R')，我们考虑两种情况：(1) 若 R' 是公平集，则 (L', R') 是单侧公平二分团（第 23–24 行）；(2) 若 R' 不是公平集，则我们计算 R' 的所有极大公平子集，以便进一步枚举单侧公平二分团（第 25–28 行）。如算法 7 所示，可以通过组合枚举方法获得这些极大公平子集。设 $r' \in \mathcal{R}'$ 是 R'的一个极大公平子集。若 $N(r')$等于 L，则得到一个单侧公平二分团，BackTrackFBCEM++ 回溯过程会将 (L', r') 加入结果集 Res（第 28 行）。与 FairBCEM 类似，当 $P' \neq \emptyset$和 $\forall a_i^V \in A(V)$ $|R'_{a_i^V}| + |P'_{a_i^V}| \geq \beta$成立时，BackTrackFBCEM++ 会调用下一次回溯过程（第 29–30 行）。最后，集合 Res保存 G 中的所有单侧公平二分团（第 4 行）。

正确性分析。包含 $|L| \geq \alpha$、$|R_{a_i}| \geq \beta$ 的二分团

算法 7：组合

输入：集合 S、属性值集合 A、两个整数 k、δ
输出：所有组合的集合 $\mathcal{CanSet}$
1 如果 $\exists a_i \in A,\ S_{a_i} < k$ 则
2 返回 $\emptyset$;
3 $msize = $ 最小值$_{a_i \in A} S_{a_i}$;
4 对于 $a_i \in A$ 执行
5 $csize = $ 最小值$(S_{a_i}, msize + \delta)$;
6 $\mathcal{Res}(a_i) \leftarrow S_a$的所有子集 $_i$ 其大小等于 $csize$;
7 $\mathcal{CanSet} \leftarrow \mathcal{Res}(a_0)$;
8 对于 $a_i \in A, i \neq 0$ 执行
9 $\mathcal{CanSet} = \mathcal{CanSet} \times \mathcal{Res}(a_i)$;
10 返回 $\mathcal{CanSet}$;

算法 8：BiConstruct2HopGraph

输入：$G = (U, V, E, A)$，一个整数 α，公平侧 V
输出：基于公平侧 V 的二跳图 H
1 令 $H = (V = G(V)\ E = \emptyset, A = A_V)$ 为一个属性图;
2 对于 $v \in G(V)$ 执行
3 C 是一个包含 $C[i][j] = 0, 1 \le i \le |G(V)|, 1 \le j \le |A(U)|$ 的数组;
4 对于 $u \in N(v, G)$ 执行
5 对于 $w \in N(u, G)$ 执行
6 如果 $w \neq v$，则 $C[w][w.val] \leftarrow C[w][w.val] + 1$;
7 对于 $u \in G(V)$ 执行
8 如果 $\forall a_i^U \in A(U)\ C[u][a_i^U] \ge \alpha$ 且 $u < v$ 则
9 $E(H) \leftarrow E(H) \cup (u, v)$;
10 返回 H;

由于 MBEA++ [6] 的正确性，这些二分团都会被枚举。对于任意极大双团 B（L，R'），算法都会枚举 B 中所有单侧公平二分团。由于每个单侧公平二分团都包含于某个极大双团中，FairBCEM++ 满足完备性。在第 26 行，我们通过 Combination 算法找出 R' 的所有极大公平子集，并判断它们是否与 L 构成二分团。因此，公平性约束得以满足。由于 L 在搜索过程中不断缩小，根据第 28 行也满足极大性。同时，每个单侧公平二分团 B'（L，R'）的 L 都是某个极大双团 B（L，R'）的 L，且每个极大双团都有不同的 L，因此，每个单侧公平二分团只会在一个极大双团中被枚举，从而避免重复枚举。

扩展到寻找所有比例单侧公平双团。我们提出了一种名为 FairBCEMPro++ 的算法，通过对 FairBCEM++ （算法 6）稍作修改来枚举所有比例单侧公平双团。具体而言，在算法 6 的第 23 行，FairBCEMPro++ 将对单侧公平双团的检查替换为对比例单侧公平双团的检查，后者可以轻松实现。此外，在算法 6 的第 26 行，我们使用另一种名为 CombinationPro 的算法，而非 Combination，来枚举比例单侧公平双团。CombinationPro 的工作流程与 Combination 类似，不同之处在于，CombinationPro 通过 $min(S_{a_i}, msize + \delta, \overline{msize * (1-\theta)\theta})$ 计算 $csize$（算法 7 第 5 行）。第三项来自比例约束，可由不等式 $\frac{msize}{msize+csize} \ge \theta$ 轻松推导。受篇幅限制，我们省略 FairBCEMPro++ 和 CombinationPro 的伪代码。

IV. 双侧公平双团枚举

本节首先修改用于求解单侧公平双团枚举问题的剪枝技术，使其适用于我们的双侧公平双团枚举问题。然后，我们通过扩展 FairBCEM，提出一种名为 BFairBCEM 的算法，用于枚举所有双侧公平双团。同样地，我们还通过扩展 FairBCEM++ 算法，提出一种名为 BFairBCEM++ 的算法。最后，我们通过调整 BFairBCEM++ 算法，提出 BFairBCEMPro++ 算法，以求解 PBSFBC 枚举问题。

A. 剪枝技术

在单侧公平双团枚举中，我们通过考虑公平侧（即下侧 V）顶点的属性度，推导出两种剪枝技术。在双侧公平双团模型中，属性约束扩展到了上侧和下侧，因此一个自然的想法是利用 U 和 V 中顶点的属性度来设计剪枝方法。下面，我们介绍两种剪枝技术，即 BFCore 和 BCFCore，它们分别是 FCore 和 CFCore 的变体。

双侧公平 α-β 核剪枝（BFCore）。与 FCore 类似，我们将双侧公平 α-β 核的概念作为定义 13 引入，并推导出引理 3，以剪除 U 和 V 中肯定不属于任何双侧公平双团的顶点。

定义 13：（双侧公平 α-β 核）给定一个带属性的二分图 $G = (U, V, E, A)$，若子图 $H = (L, R, E, A)$ 满足以下条件，则称其为双侧公平 α-β 核：(1) $D_{a_i}(u, H) \ge \beta, u \in L, a_i \in A(V)$；(2) $D_{a_i}(v, H) \ge \alpha, v \in R, a_i \in A(U)$；(3) 在 G 中不存在满足条件 (1) 和 (2) 的子图 $H' \supset H$。

引理 3：给定一个属性二部图 $G = (U, V, E, A)$ 和两个整数 α, β，任何双侧公平双团都必须包含在一个双公平 α-β 核中。

根据引理 3，一个问题是如何计算二分图 G 的双公平 α-β 核。我们通过稍微修改 FCore（算法 1），设计了一种称为 BFCore 的剥离算法，因为定义 13 也是经典 k- 核 [33],[34] 的一种变体。具体而言，对于 V 中的每个顶点 v，BFCore 计算属性度 $D_{a_i^U}(v)$，而非度 $D(v)$（第 2-6 行）。移除顶点 u 时，算法会更新其邻居的属性度，并维护优先队列 Q。如果邻居 v 位于下侧 V，BFCore 会像处理上侧 U 中的顶点一样，计算新的属性度 $D_{a_i^U}(v)$（第 16-19 行）。BFCore 的其他步骤与 FCore 类似，因此我们省略 BFCore 的伪代码。

双色公平 α-β 核剪枝（BCFCore）。在 CFCore 中，我们通过为公平侧 V 上至少有 α 个公共邻居的两个顶点添加边，来构建二跳图（即定义 3 中的条件 (1)）。而双侧公平二分团模型考虑了 U 和V两侧的公平性。因此，构建 V上的二跳图时，只有当两个顶点对于每个属性值 $a_i^U \in A(U)$ 都至少有 α 个公共邻居时，我们才为它们添加边（即定义 4 中的条件 (1)）。这里，我们修订了二跳图算法，使其适用于双侧公平二部团枚举问题，算法概要见算法 8。在由

算法 9：BFairBCEM

```
输入：一个二分图 G = (U, V, E, A)，三个整数 α、β、δ
输出：所有双侧公平双团的集合 Res^
1 G =( U, V, E, A) ← BCFCore(G, α, β);
2 L ← Û;  R ← ∅;  P ← V̂;  Q ← ∅;
3 通过 FairBCEM( ^G, α, β, δ) 枚举所有单侧公平二分团;
4 对于每个单侧公平二分团 B(L′, R′)，执行
5     L′ ← 组合(L′, A(U) α, δ);
6     对于 l′ ∈ L′，执行
7         如果 R′ 是 N(l′) 的极大公平子集，则
8             Res ← Res ∪ (l′, R′);
9 返回 Res;
```

BiConstruct2HopGraph 构建的图中，我们仍可计算自我双色β-核，以剪除 V中不具潜力的顶点。

此外，双侧公平双团模型还要求上侧 U 满足公平性，因此我们可以采用处理下侧 V 的方式，对 U 中的顶点进行剪枝。基于这一想法，我们提出了 BCFCore 算法，该算法与 FCore 类似，仅进行了以下细微修改。具体而言，对于下侧 V，BCFCore 通过 BiConstruct2HopGraph 而不是 Construct2HopGraph 构建二跳图（算法 2 中的第 3 行），并计算自我彩色 β-核，以剪枝 V 中的顶点。对于上侧 U，BCFCore 再次通过 BiConstruct2HopGraph 构建二跳图，参数为 (G, β, U)，并计算自我彩色 α-核，以剪枝 U 中无望的顶点。由于篇幅限制，我们省略 BCFCore 的伪代码。

B. BFairBCEM算法

在介绍 BFairBCEM算法之前，我们先给出以下观察。

观察 6：双侧公平双团必然包含于单侧公平双团中。

根据观察 6，我们给出算法 9 所示的 BFairBCEM算法。我们首先搜索所有单侧公平双团，然后通过组合上侧来枚举所有双侧公平双团。具体而言，BFairBCEM 调用 FairBCEM 搜索所有单侧公平双团（第 3 行）。给定一个单侧公平双团 $B(L', R')$，它满足下侧的公平性约束；随后，我们通过 Combination 算法枚举上侧 L' 的所有极大公平子集，以确保公平性（第 5 行）。对于 $\mathcal{L}'$ 中的一个极大公平子集 l'，BFairBCEM算法会判断 R' 是否为 $N(l')$ 的极大子集（第 7 行）。显然，若是，则 (l', R') 是一个双侧公平双团，我们将其加入 Res。根据观察 6，所有双侧公平双团都包含在所有单侧公平双团中。因此，BFairBCEM算法能够正确返回所有双侧公平双团。

正确性分析。FairBCEM 能正确枚举所有单侧公平二分团，而任何双侧公平双团都必须包含在某个单侧公平双团中，因此满足完备性。算法 9 的第 7 行保证了极大性，因为 l' 是 $N(R)$ 的极大公平子集，且 R' 是 $N(l')$ 的极大公平子集，这也验证了公平性约束。对于无冗余性，显然，枚举得到的任何双侧公平双团都属于一个单侧公平双团，且具有相同的 R'；不存在两个不同的单侧公平双团具有相同的 R，因此任何双侧公平双团都只会被枚举一次。

C. BFairBCEM++算法

基于观察 6，我们还可以调用 FairBCEM++ 算法来搜索所有单侧公平二部团，然后通过组合枚举方法枚举所有双侧公平双团。因此，我们提出了 BFairBCEM++ 算法，该算法只需对算法 9 进行少量修改即可轻松设计。也就是说，我们使用 FairBCEM++ 替代第 3 行中的 FairBCEM，以查找所有单侧公平二部团。由于篇幅限制，我们省略 BFairBCEM++的伪代码。

扩展到查找所有 PBSFBC。我们可以对 BFairBCEM++ 算法进行少量调整，以解决 PBSFBC枚举问题，该算法称为 BFairBCEMPro++。也就是说，我们将 Combination替换为 CombinationPro（算法 9 第 5 行），并使用针对 PBSFBC 的检查替代针对 BSFBC 的检查（算法 9 第 3—4 行）。值得注意的是，在检查极大公平子集时，我们还需要检查比例约束是否得到满足（算法 9 第 7 行）。由于篇幅限制，我们省略 BFairBCEMPro++ 的详细内容。

V. 实验

A. 实验设置

对于单侧公平二部团枚举问题，我们实现了配备 FCore（算法 1）和 CFCore（算法 2）剪枝技术的 FairBCEM（算法 5）和 FairBCEM++ （算法 6）。为了枚举所有双侧公平双团，我们实现了采用 BFCore 和 BCFCore 剪枝技术的 BFairBCEM（算法 9）和 BFairBCEM++ 。作为对比，我们实现了两种朴素搜索算法，即 NSF 和 BNSF，用于寻找所有 SSFBC 和 BSFBC；这些算法保留了算法 1 和算法 2 等剪枝技术，并在搜索过程中去除了观察 2、观察 4 和观察 5 等所有剪枝技术。我们还采用两种不同的顶点选择顺序实现了上述枚举算法，即 DegOrd 和 IDOrd；这两种顺序分别通过按照顶点度和 ID 的非递增顺序对顶点进行排序获得。所有算法均使用 C++ 实现。我们在配备 2.10GHz 英特尔至强处理器和 256GB 内存的 PC 上进行所有实验。我们将所有算法的时间限制设为 24 小时，并使用符号“INF”表示算法无法在 24 小时内终止。

数据集。我们在五个现实世界图上评估所提算法的效率。具体而言，Wiki-catis 是一个特征网络。YouTube 和 IMDB 是隶属关系网络，Twitter 是交互网络，DBLP 是作者关系网络。所有数据集均可从 http://konect.cc/ 下载。请注意，这些数据集都是无属性二部图，因此我们随机为每个顶点分配一个属性，以生成属性图，用于评估所有算法的效率。

参数。我们的算法中有四个参数：α、β、δ 和 θ。α 和 β 用于限制公平二部团的规模。

表 I 数据集和参数

数据集	$\|U\|$	$\|V\|$	$\|E\|$	密度	α^{*s}	β^{*s}	α^{*b}	β^{*b}	δ^*	θ^*
YouTube	94 , 238	30, 087	293, 360	1.0×10^{-4}	8	8	5	5	2	0.4
Twitter	175, 214	530, 418	1, 890, 661	2.0×10^{-5}	8	8	6	7	2	0.4
IMDB	303 , 617	896 , 302	3, 782, 463	1.4×10^{-5}	10	10	6	6	2	0.4
Wiki-cat	1 853 493	182 947	3795	1.1×10^{-5}	7	7	6	6	2	0.4
DBLP	1 ,953, 085	5 , 624, 219	12 , 282, 059	1.1×10^{-6}	7	7	4	4	2	0.4

注： α^{*s}, β^{*s} 和 α^{*b}, β^{*b} 分别是 SSFBC (PSSFBC) 和 BSFBC (PBSFBC) 模型中 α, β 的默认值， δ^*, θ^* 是 δ 和 θ 的默认值。

如果 α 和 β 过小，就会得到过多没有意义的小型二分团。如果 α 和 β 过大，剪枝过程中大多数顶点都会被剪除，剩余图将丢失大量结构信息，导致输出的二分团很少。我们根据现实数据集中二分团的数量，仔细调整这些参数，以提取有意义的公平二部团。δ 表示各属性对应的顶点数之间的最大差值。随着 δ 增大，顶点集内不同属性之间的公平性会降低。因此， δ 不应设置得过大，否则问题将退化为极大二部团枚举问题。参数θ 是公平性比例阈值，我们可以轻易推出 θ 不大于0.5。因此， θ 也不应设置得过大。由于不同数据集的规模各异，参数 α 和 β 设置为不同的整数。对于 SSFBC（PSSFBC）和 BSFBC（PBSFBC）枚举问题，我们也将参数设置为不同的整数。详细的参数设置见网站

h

ttps://github.co

m/Heisenberg-Yin/fairnesss-biclique。

B. 效率测试

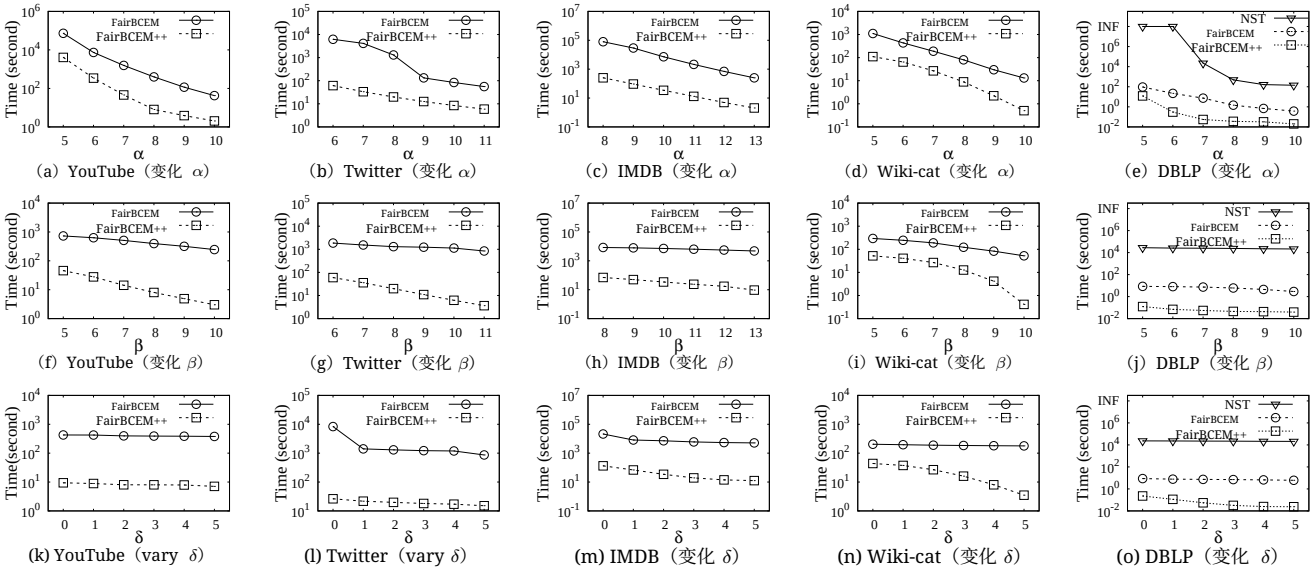
实验1：剪枝技术评估。对于单侧公平双团枚举问题，FairBCEM 和 FairBCEM++ 算法都可以使用 FCore 和 CFCore 剪除无望节点。对于双侧公平二部团枚举问题，剪枝技术 BFCore 和 BCFCore 可以在 BFairBCEM 和 BFairBCEM++ 中缩小图规模。在本实验中，我们通过比较剪枝后剩余顶点的数量及其耗时，评估这些剪枝技术在不同 α 和 β下的表现。图3和图4分别展示了在 IMDB 上进行单侧公平双团枚举和双侧公平双团枚举的结果。其他数据集上的结果也一致。图3 (a)-(b)显示，与原始图相比，FCore 和 CFCore 都能显著减少顶点数量，这符合预期。此外，随着 α 或 β增大，剩余顶点数量会减少。总体而言，就剪枝性能而言，CFCore 优于 FCore，尤其是在 α或 β 值相对较小时。如图3 (c)-(d)所示，随着 α 或 β 增大，FCore 和 CFCore 的运行时间会减少，而 CFCore 剪除无望顶点所需的时间比 FCore 更长。这是因为 CFCore 会先执行 FCore，然后在二跳图中通过 ego fair α-β core 剪枝进一步缩减图规模（算法 2）。例如，在图3(a)中，当 $\alpha = 8$时，FCore 将顶点数量从 9,266,649 减少到 12,507；CFCore 则进一步将顶点数量减少到 1,318。当 β 等于 8 时，如图3(b)所示，经过 FCore 和 CFCore 剪枝后，剩余顶点数量分别为 13,757 和 1,490。因此，CFCore 剪枝只需略微增加耗时，就能获得优于 FCore 的剪枝效果。此外，图4中双侧公平双团枚举也呈现出类似结果。总之，上述实验结果验证了 FCore、CFCore、BFCore 和 BCFCore 剪枝技术的有效性和效率。

实验2：SSFBC 枚举算法的评估。这里，我们通过改变 α、β 和 δ，评估采用降序 DegOrd 的 FairBCEM 和 FairBCEM++ 算法。结果如图2所示。正如预期，在所有数据集上，随着 α、β、δ增大，FairBCEM 和 FairBCEM++的运行时间都会减少。这是因为当 α较大时，FCore 和 CFCore 剪枝技术可以剪去许多顶点，并且在分支定界过程中，搜索空间也相应缩小。当 δ较大时，由于极大性约束，单侧公平二分团的数量会随着 δ 增大而减少，因此运行时间呈下降趋势。此外，我们还可以看到，在所有数据集的所有参数设置下，FairBCEM++ 的运行时间至少比 FairBCEM 低两个数量级。例如，当 $\alpha = 10$ 采用默认 β 和δ时，FairBCEM 在 IMDB 上耗时 29,192 秒，才能找到所有单侧公平二分团；而 FairBCEM++ 仅用 91 秒就输出了结果，速度几乎比 FairBCEM算法快三个数量级。这些结果验证了所提出的 FairBCEM 和 FairBCEM++ 算法的效率。

在 FairBCEM 和 FairBCEM++ 算法中，会从候选集中选择一个顶点加入当前二部团，以执行回溯搜索过程。由于采用不同排序方式时的搜索空间存在显著差异，我们还使用 IDOrd 和 DegOrd排序对这两种算法进行了评估。表II展示了在默认 α、β、δ情况下，FairBCEM 和配备 IDOrd 与 DegOrd排序的 FairBCEM++ 在所有数据集上的运行时间。如表II所示，采用 DegOrd排序的 FairBCEM 明显快于采用 IDOrd排序的 FairBCEM。例如，在 IMDB 中，采用 IDOrd 和 DegOrd排序的 FairBCEM算法分别耗时 4,378 秒和 2,098 秒来输出所有单侧公平二分团。显然，后者的速度几乎是前者的 2 倍。对于采用 IDOrd 和 DegOrd排序的 FairBCEM++ 算法，也可以观察到类似结果。同样，FairBCEM++ 算法在所有数据集上都优于 FairBCEM，这与我们之前的发现一致。结果表明，在搜索过程中，DegOrd 排序比 IDOrd排序更高效。

此外，我们在所有数据集上将 NSF 与所提出的 FairBCEM 和 FairBCEM++ 进行比较。由于在大多数参数设置下，NSF 在其他数据集上都会超时，因此我们仅展示图2中的 DBLP 结果。如图所示，FairBCEM 至少比 NSF 快两个数量级。这些结果证实，我们提出的算法显著优于 NSF算法。

实验3：BSFBC 枚举算法评估。我们通过改变++ 和 DegOrd，评估 BFairBCEM 和 BFairBCEM++ 的运行时间。

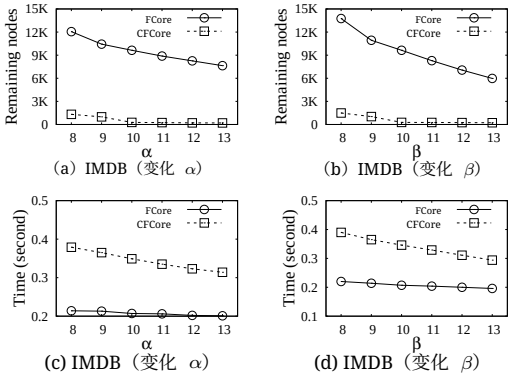


（a）YouTube（变化 α） （b）Twitter（变化 α） （c）IMDB（变化 α） （d）Wiki-cat（变化 α） （e）DBLP（变化 α）

（f）YouTube（变化 β） （g）Twitter（变化 β） （h）IMDB（变化 β） （i）Wiki-cat（变化 β） （j）DBLP（变化 β）

(k) YouTube（vary δ） (l) Twitter（vary δ） (m) IMDB（变化 δ） (n) Wiki-cat（变化 δ） (o) DBLP（变化 δ）

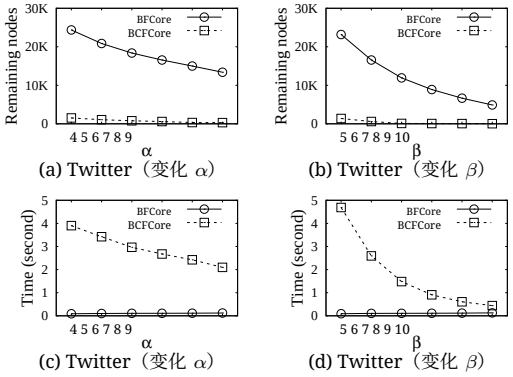
图2. NSF、FairBCEM 和 FairBCEM++ 算法在不同数据集上的运行时间。



（a）IMDB（变化 α） （b）IMDB（变化 β）

(c) IMDB（变化 α） (d) IMDB（变化 β）

图 3. FCore 和 CFCore 的剪枝时间及剩余节点。



(a) Twitter（变化 α） (b) Twitter（变化 β）

(c) Twitter（变化 α） (d) Twitter（变化 β）

图4. BFCore 和 BCFCore 的剪枝时间及剩余节点。表 II 不同算法的运行时间IDOrd 和DegOrd。

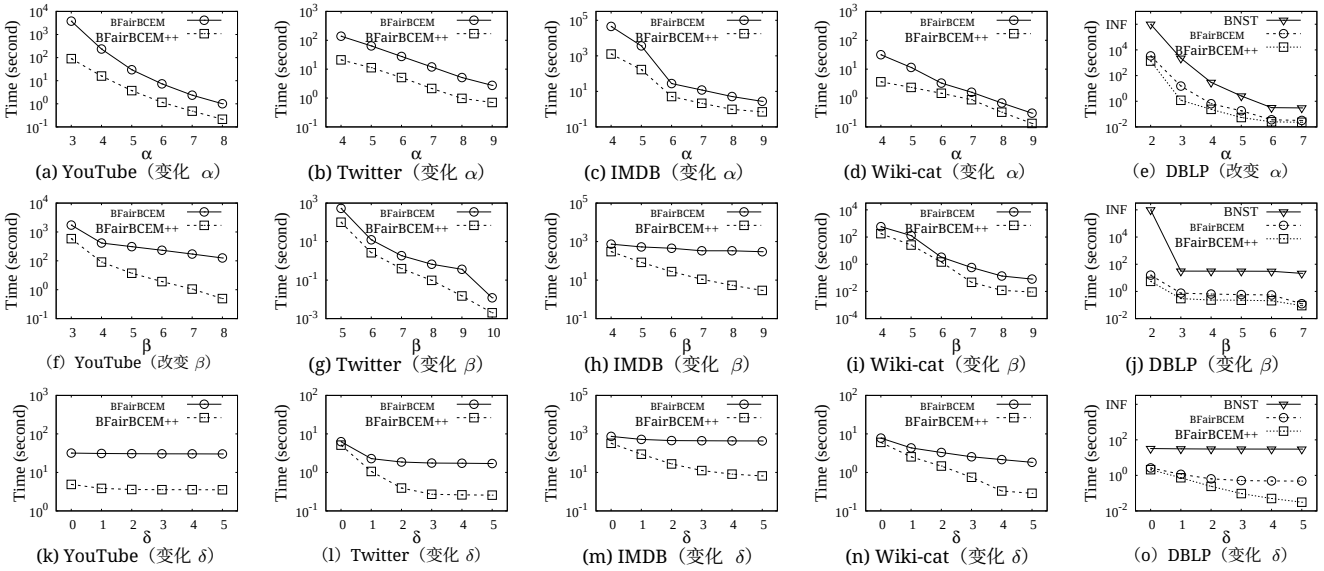
算法（秒）	排序	IMDB	YouTube	Twitter	Wiki-cat	DBLP
FairBCEM	IDOrd	7,022.7	157.1	854.2	90.6	6.3
	DegOrd	1,612.9	43.6	611.8	45.9	2.6
FairBCEM++	IDOrd	78.6	16.1	72.5	13.2	0.6
	DegOrd	61.9	8.3	65.1	12.4	0.5
BFairBCEM	IDOrd	174.2	2.3	76.8	0.9	1.5
	DegOrd	68.1	1.4	69.1	0.4	1.1
BFairBCEM++	IDOrd	19.8	7.4	63.8	0.3	0.7
	DegOrd	17.2	1.7	59.7	0.2	0.6

具体改变 α、β和δ。结果如图5所示。正如预期，随着 α、β和δ增大，BFairBCEM 和 BFairBCEM++ 的运行时间会减少，这与单侧公平双团枚举算法的情况相似。此外，我们还观察到，在所有数据集的所有参数设置下，BFairBCEM++ 算法的速度几乎比 BFairBCEM 算法快 3 至 100 倍。例如，在 $\beta = 7$ 、采用默认 α 和 δ时，BFairBCEM 和 BFairBCEM++ 分别用 17 秒和 1 秒在 YouTube 上输出所有双侧公平双团。显然，前者明显快于后者。这些结果验证了所提出的 BFairBCEM 和 BFairBCEM++ 算法的效率。

此外，我们比较了在默认 α、β、δ下，采用IDOrd和DegOrd的BFairBCEM与BFairBCEM++ 算法的运行时间。如表II所示，采用DegOrd的BFairBCEM明显优于采用IDOrd的版本。例如，在IMDB中，采用IDOrd的BFairBCEM算法需要253秒才能找到所有双侧公平双团，而采用DegOrd的算法只需169秒。采用IDOrd和DegOrd的BFairBCEM++ 算法也呈现出类似结果。此外，在所有数据集上，BFairBCEM++ 算法的速度都快于BFairBCEM。这些结果还证明了DegOrd排序的效率，与我们先前的发现一致。

此外，我们还评估了所有数据集上BNSF与BFairBCEM++ 和BFairBCEM++的运行时间。我们在图5中展示了DBLP上的结果，因为在其他数据集上，NSF在给定的时间限制内无法完成计算。可以看到，BFairBCEM的速度至少比BNSF快两个数量级。这些结果证实，我们的算法明显快于BNSF算法。

实验4：SSFBC 和 BSFBC 的数量。图6报告了 Wiki-cat 上随着 α、β、δ 变化，单侧公平双团和双侧公平双团的数量。请注意，我们找到了满足 $|L| \geq \alpha$ 和 $|R| \geq 2 \times \beta$ 的极大双团 $B(L, R)$

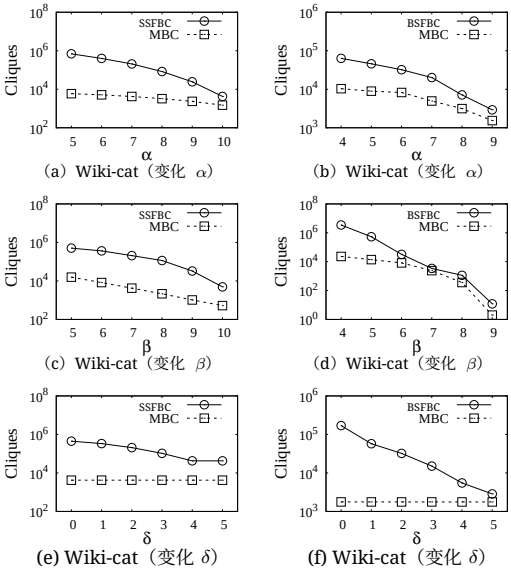


(a) YouTube（变化 α） (b) Twitter（变化 α） (c) IMDB（变化 α） (d) Wiki-cat（变化 α） (e) DBLP（改变 α）

(f) YouTube（改变 β） (g) Twitter（变化 β） (h) IMDB（变化 β） (i) Wiki-cat（变化 β） (j) DBLP（变化 β）

(k) YouTube（变化 δ） (l) Twitter（变化 δ） (m) IMDB（变化 δ） (n) Wiki-cat（变化 δ） (o) DBLP（变化 δ）

图5。BNSF、BFairBCEM 和 BFairBCEM++ 算法在不同数据集上的运行时间。

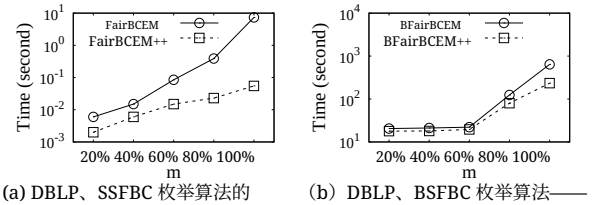


（a）Wiki-cat（变化 α） （b）Wiki-cat（变化 α）

（c）Wiki-cat（变化 β） （d）Wiki-cat（变化 β）

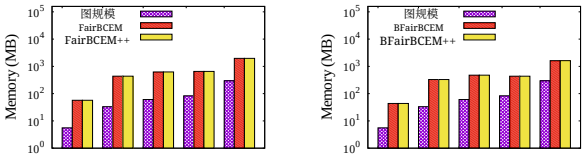
(e) Wiki-cat（变化 δ） (f) Wiki-cat（变化 δ）

图6. 极大双团、单侧公平双团和双侧公平双团的数量。



(a) DBLP、SSFBC 枚举算法的变化 m) （b）DBLP、BSFBC 枚举算法——法（vary m）

图 7。 所提算法的可扩展性。



(a) SSFBC 枚举算法 (b) BSFBC 枚举算法

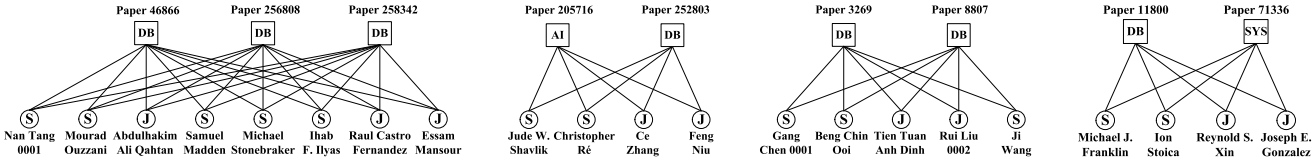
图 8. 内存开销。

以便与单侧公平双团进行比较。为了与双侧公平双团进行比较，我们搜索满足 $|L| \geq 2\times\alpha$ 和 $|R| \geq 2\times\beta$ 的极大双团 B（L，R）。显然，Wiki-cat 上存在大量单侧公平双团和双侧公平双团。例如，对于单侧公平二部团枚举问题中的 $\alpha = 6$、$\beta = 6$、$\delta = 2$，共有 9,548 个极大双团和 346,411 个单侧公平双团。对于双侧公平二部团枚举问题中的 $\alpha = 3$、$\beta = 6$、$\delta = 2$，共有 546,411 个双侧公平双团和 9,548 个极大双团。总体而言，单侧公平双团和双侧公平双团的数量多于极大双团。这一发现与我们在第 II 节中的分析一致，因为任何单侧公平双团或双侧公平双团都必须包含在一个极大双团中。此外，我们可以看到，随着 α、β、δ 增大，极大双团、单侧公平双团和双侧公平双团的数量都会减少。这是因为 $\alpha/\beta/\delta$ 越大，公平性约束和大小约束就越严格，因而

单侧公平双团/单侧公平双团模型和极大二分团模型，分别。

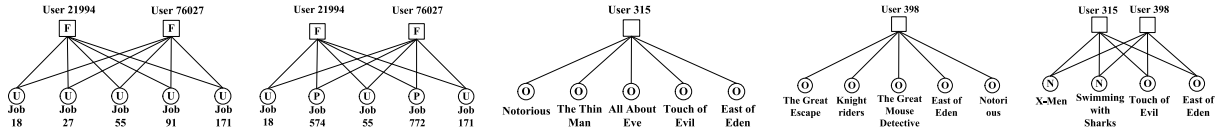
实验 5：可扩展性测试。这里我们评估所提出算法的可扩展性。为此，我们从每个数据集中随机选取 20%-80% 的边，生成四个子图，并评估单侧公平双团枚举和双侧公平双团枚举算法的运行时间。图 7 展示了 DBLP 上的结果，其他数据集上的结果也类似。对于 SSFBC 枚举算法，如图 7(a) 所示，随着图规模增大，FairBCEM 的运行时间平稳增长，而 FairBCEM++ 的运行时间在不同 m 值下相对稳定。同样，在所有参数设置下，FairBCEM++ 的速度至少比 FairBCEM 快 10 倍，这与我们此前的发现一致。对于 SSFBC 枚举算法，如图 7(b) 所示，与 BFairBCEM 相比，BFairBCEM++ 的运行时间随图规模增大而增长得更加平缓。这些结果证明了所提出算法具有很强的可扩展性。

实验 6：内存开销。图 8 展示了各数据集上枚举算法的内存开销。请注意



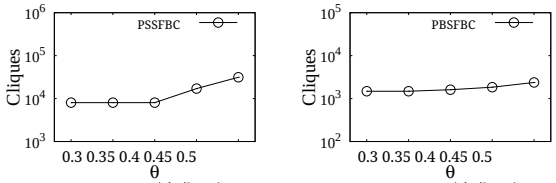
(a) DBDA、SSFBC、($\alpha = 3, \beta = 3, \delta = 2$)　(b) DBDA、BSFBC, ($\alpha = 1, \beta = 2, \delta = 2$)　(c) DBDS、SSFBC、($\alpha = 2, \beta = 2, \delta = 2$)　(d) DBDS、BSFBC, ($\alpha = 1, \beta = 2, \delta = 2$)

图 9. DBDA 和 DBDS 的案例研究。



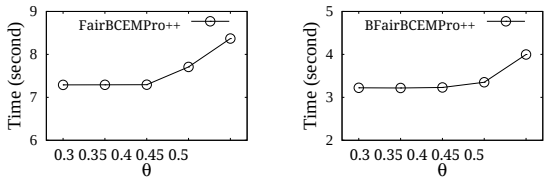
(a) Jobs，CF算法　(b) Jobs，SSFBC，($\alpha = 2, \beta = 2, \delta = 1$)　(c) Movies，CF算法　(d) Movies，CF算法　(e) Movies，SSFBC，($\alpha = 2, \beta = 2, \delta = 1$)

图10. Jobs和Movies上的案例研究。



(a) YouTube（变化 θ）　(b) YouTube（变化 θ）

图11. 比例单侧公平二分团和 PBSFBC 的数量。



(a) YouTube，FairBCEMPro++ al-算法（变化 θ）　(b) YouTube, BFairBCEMPro++al-gorithm（vary θ）

图12. FairBCEMPro++ 和 BFairBCEMPro++的运行时间。

不同算法的内存开销不包括图的规模。从图 8 可以看出，FairBCEM 和 FairBCEM++ 的内存用量几乎相同，且始终大于原始图的规模。这是因为它们都采用 CFCore 剪枝技术，并以深度优先方式枚举单侧公平二分团，因此空间开销主要取决于 CFCore 中的数据结构。这些结果与我们在第III-C节中的分析一致。对于 BFairBCEM 和 BFairBCEM++ 算法，也可以观察到类似的结果。

实验7：PSSFBC和BSFBC枚举算法的评估。这里，我们通过改变附加参数 θ，评估FairBCEMPro++ 和 BFairBCEMPro++ 算法。图11和图12展示了FairBCEMPro++和BFairBCEMPro++ 在YouTube上的PSSFBC和PBSFBC数量及运行时间。其他数据集上的结果类似。可以看出，比例公平二部团的数量和运行时间会随着 θ的增大而增加。当 $\theta = 0.5$时，PSSFBC枚举问题退化为 $\delta = 0$的SSFBC枚举问题。因此，求解PSSFBC枚举问题所需的时间与求解SSFBC枚举问题相近。PBSFBC枚举问题的情况也类似。当 θ 接近0.5时，更多二分团符合比例公平二部团的定义，因此其数量

PSSFBC和PBSFBC的数量增加，算法的运行时间也随之增加。

C. 案例研究

DBLP 案例研究。我们对 DBLP 合作网络进行案例研究，以展示我们算法的有效性。DBLP 数据集从 dblp.uni-trier.de/xml/ 下载。我们通过定义两类节点，在 DBLP 上构建一个二分图：论文位于上侧，学者位于下侧。当某位学者是某篇论文的作者时，两者之间就有一条边。基于 DBLP，我们进一步构建了两个属性二部子图：DBDA 和 DBDS，具体如下。对于 DBDA，我们保留至少在数据库（DB）和人工智能（AI）相关会议上发表过一篇论文的学者。每位学者都有一个属性 A_V，其取值为 $A(V) = \{S, J\}$，其中S 表示资深学者，J 表示青年学者。我们通过判断某位学者是否发表论文超过 10 年，来为其分配属性值 v。如果是，则将 $v.val$ 设为 S；否则，$v.val$ 为 J。每篇论文都关联一个属性 A_U，其取值为 $A(U) = \{DB, AI\}$，用于表示该论文是否发表于 DB 和 AI 相关会议。对于 DBDS，我们仅保留至少在数据库（DB）和系统（SYS）相关会议上发表过一篇论文的学者。每位学者也具有属性 A_V，其取值为 $A(V) = \{S, J\}$；我们采用 DBDA 中的方法为学者分配属性值。每篇论文都有一个属性 A_U，其取值为 $A(U) = \{DB, SYS\}$，用于表示该论文是否发表于 DB 和 SYS 相关会议。最后，DBDA 包含 260,605 篇论文和 240,420 位学者，共有 781,378 条边，即 $|U| = 240,420$ 和 $|V| = 260,605$。DBDS 则包含 163,545 篇论文和 139,703 位学者，共有 433,928 条边，即 $|U| = 163,545$ 和 $|V| = 139,703$。我们运行 FairBCEM++ 和 BFairBCEM++ 算法，以找出所有单侧公平二分团和双侧公平双团。

例如，图 9 (a)-(b) 和图 9 (c)-(d) 分别展示了 DBDA 和 DBDS 上的一个单侧公平二分团和一个双侧公平双团。由于论文标题过长，我们不展示标题。在图 9(a) 中，可以看到有五位资深学者和三位青年学者，这显然是 DBDA 上一个包含 $\alpha = 3$、$\beta = 3$、$\delta = 2$的单侧公平二分团。从他们的个人主页可以看出，所有学者

在图 9(a) 中的研究方向都是数据库相关领域，这与他们所关联论文的属性一致。Michael Stonebraker 和 Samuel Madden 等资深作者确实是数据库领域的知名学者。这一结果表明，我们的 FairBCEM++ 能够在现实应用中找到单侧公平二分团，从而保证一侧的公平性。图 9(b) 中的二分图包含下侧的两位资深学者和两位青年学者，以及上侧的一篇 AI 论文 [36]和一篇 DB 论文 [37] ，构成一个双侧公平双团。此外，Christopher R´e 和 Jude W. Shavlik 两位教授是数据库和人工智能领域的科学家，而 Ce Zhang 相较于前两位学者更年轻，前两位学者曾是 Christopher R´e 的学生。这一结果证实，所提出的 BFairBCEM++ 确实能够在现实世界图中找到双侧公平双团，从而确保两侧的公平性。在 DBDS 上也能发现类似结果。图 9(c) 展示了一个包含五位资深学者和三位青年学者的单侧公平双团。在图 9(d) 中，两位资深学者和两位青年学者共同撰写了一篇发表于 SIGMOD 的 DB 论文 [38]和一篇发表于 OSDI 的 SYS 论文 [39] 。在所有学者中，Michael Frankin 和 Ion Stoica 两位教授也是数据科学和分布式系统领域的知名学者。这些结果证明了单侧公平二分团和双侧公平双团模型以及我们所提出算法的有效性。

Jobs案例研究。我们使用职位推荐数据集 Jobs 开展案例研究，该数据集可从 https://www.kaggle.com/competitions/job-recommendation 下载。该数据集包含 7 个窗口。为简单起见，我们只考虑窗口 1，因为每个窗口彼此独立。我们通过定义两类节点来构建二分图 G ，即上侧的用户和下侧的职位。职位的属性是受欢迎程度，根据该职位的申请数量设定。为避免冷启动问题，我们仅保留申请数量最多的前 1000 个职位，并将其中受欢迎程度最高的前 500 个职位设为更受欢迎的职位（属性为 P），其余职位设为较不受欢迎的职位（属性为 U）。我们还为每位用户分配属性值 A 或 F ，以表示其是美国人还是外国人。因此，二分图 G 包含 63,412 位用户和 1,000 个职位，拥有 $A_V = \{P, U\}$ 和 $A_U = \{A, F\}$。我们使用协同过滤（CF）算法计算推荐结果，如图 10(a) 所示。在图 10(a) 中，如果某职位位于 CF 算法推荐的前 5 个职位之列，则用户与该职位之间存在一条边。从 Jobs 的信息中可以得知，用户 21,994 来自印度，拥有硕士学位和 9 年工作经验；用户 76,027 是加拿大人，拥有硕士学位和 23 年工作经验。显然，这两位外国人的学历和工作经验相似，但推荐给他们的职位全都不太受欢迎。为消除偏差，我们构建二分图 G' ，其中每条边表示该职位的 CF 计算推荐分数位列前 10，即 G 包含 63,412 位用户、1,000 个职位和 63,4120 条边。随后，我们将职位设为公平侧，运行 FairBCEM++ 以查找单侧公平双团。图 10(b) 展示了一个包含用户 21,994 和用户 76,027 的单侧公平双团。正如预期，更受欢迎和较不受欢迎的职位都推荐给了这两位外国人。这些结果证明了我们提出的公平二分团模型和算法的有效性。

Movies案例研究。我们还使用电影推荐数据集 Movies 开展案例研究，该数据集可从 https://www.kaggle.com/code/rounakbanik/movie-recommender-system 下载。我们构建了一个二分图，其中上侧为用户，下侧为电影。对于每部电影，我们将其属性设置为 O ，表示该电影是在1990年之前上映的老电影；否则，将其属性设置为N ，表示新电影。该二分图包含9,000部电影和700个用户，即 $|U| = 700$ 和 $|V| = 9,000$。传统协同过滤算法得到的推荐结果如图10(c)和图10(d)所示，其中一条边表示某部电影位于某用户的前5个推荐结果中。可以看出，对于两个具有相似兴趣的用户，图10(c)和图10(d)中的五部电影全部都是老电影。CF算法存在爆炸偏差，即已经流行的电影将获得更多被推荐的机会，而相对较新的电影即使质量相当，也获得较少的推荐机会，这通常被称为冷启动问题。为了解决这一问题，我们根据CF计算出的个性化推荐分数，将每个用户与排名前10的电影相连，并调用 FairBCEM++ 寻找单侧公平双团。图10(e)展示了一个包含用户310和用户512的单侧公平双团。通过将公平性引入电影推荐任务，新推荐的电影“X战警”相比老电影更受欢迎，也更为知名。该结果表明，公平二分团模型能够缓解爆炸偏差问题。

VI. 相关工作

凝聚二分子图挖掘。我们的工作与二分图中的凝聚子图挖掘相关，这一方向近年来备受关注。例如，Zhang等人 [6] 提出了一种分支定界算法，即MBEA，用于搜索所有极大二部团。为提高搜索效率，Abidi等人 [10] 进一步提出了一种名为PMBE的枢轴枚举算法，该算法基于包含有向无环图（CDAG）。Yang等人[40] 研究了(p, q)-团计数问题，并提出了BCList和BCList++ 算法；该算法采用基于层的探索策略和成本模型来加速搜索过程。Lyu等人 [41] 提出了一种搜索最大二分团的新算法，可用于处理十亿级规模的二分图。Wang等人 [42]开发了一种新颖的索引结构，以帮助查找(α, β)-社区；该社区是最小边权的(α, β)-核。Wang等人 [43] 提出了基于顶点优先级的范式 BFC-VP，大幅加速了蝴蝶计数。上述所有算法均未考虑凝聚子图的公平性，

且主要针对非属性二分图。据我们所知，公平感知双团的定义是首次提出的，我们的工作也是首次研究在二分图中查找公平感知双团的问题。

公平感知数据挖掘。我们的工作受到机器学习领域广泛研究的一种概念——公平性——的启发。Verma 等人 [21] 提出了许多概念，以更好地衡量公平性。Zehlike 等人 [25]提出了一种生成排序的方法，可保证群体公平性，确保排序中受保护元素的比例不低于给定阈值。Serbos 等人 [26] 研究了套餐到群体推荐中的公平性问题，并提出了一种用于求近似解的贪心算法。Beutel 等人 [27] 也研究了推荐系统中的公平性，并提出了一组用于评估算法公平性的度量指标。另一条公平性研究路线聚焦于分类任务。一些值得关注的工作包括人口统计均等 [23] 和机会均等 [22]。例如，Hardt 等人 [22] 提出了一个框架，可以对任意已学习的预测器进行最优调整，以减少偏差。我们对公平性的定义要求群体内不同属性值相等，这与机器学习文献中的上述研究不同。在数据挖掘领域，Pan 等人 [31] 将公平性引入团模型，并提出了多种寻找公平团的算法。与他们的工作不同，我们着重研究二分图上的公平感知二分团枚举问题，所采用的技术也与他们的方法有显著差异。

VII. 结论

本文研究了二分图中公平感知二分团的枚举问题。我们提出了单侧公平双团模型和双侧公平双团模型，将公平性引入二分图。为枚举所有单侧公平二分团，我们首先提出 FCore 和 CFCore 剪枝技术，以剪除不具潜力的顶点；随后开发了分支定界算法 FairBCEM，用于枚举剪枝后图中的所有单侧公平二分团。为提高效率，我们提出了 FairBCEM++ 算法，通过将极大团作为候选项来缩小搜索空间，从而搜索所有单侧公平二分团。针对双侧公平二分团枚举问题，我们还提出了 BFCore 和 BCFCore 剪枝技术，并开发了采用分支定界技术的 BFairBCEM算法。此外，我们还提出了改进算法 BFairBCEM++，用于找出所有双侧公平双团。我们还考虑了每种属性对应的顶点数占顶点总数的比例，并提出了比例单侧公平双团和比例双侧公平双团模型及其枚举算法。我们使用五个大型真实图开展了大量实验，结果证明了所提方法的效率、有效性和可扩展性。

致谢

本研究部分得到了以下项目的资助：(i) 国家重点研发计划 2021YFB3301300，(ii) 国家自然科学基金委员会资助项目 U2241211、62072034、U1809206，以及 (iii) CCF-Huawei Pop- ulus Grove Fund。Rong-Hua Li 是本文的通讯作者。

参考文献

[1] H. Wang、C. Zhou、J. Wu、W. Dang、X. Zhu 和 J. Wang，《用于欺诈检测的深度结构学习》，载于 ICDM，2018。[2] J. Wang、A. P. de Vries 和 M. J. T. Reinders，《通过相似度融合统一基于用户和基于物品的协同过滤方法》，载于 SIGIR，2006。[3] X. Zhu、H. Tao、Z. Wu、J. Cao、K. Kalish 和 J. Kayne，《在线数字广告中的欺诈防范》，施普林格计算机科学简明丛书，2017。[4] F. Colace、M. D. Santo、L. Greco、V. Moscato 和 A. Picariello，《在线社交网络中以用户为中心的协同物品推荐框架》，《计算机与人类行为》，第 51 卷，第 694–704 页，2015。[5] S. Wu、W. Zhang、F. Sun 和 B. Cui，《推荐系统中的图神经网络：综述》，CoRR，第 abs/2011.02260 卷，2020。

[ ] 2011 02260 在线。可通过以下网址

[6] 访问：https://arxiv.org/abs/ 。Y. Zhang、C. A. Phillips、G. L. Rogers、E. J. Baker、E. J. Chesler 和 M. A. Langston，《在二部图中寻找二分团：一种新颖算法及其在整合多种生物数据中的应用》，BMC生物信息学，第 15 卷，第 110 页，2014。[7] E. Corel、R. M´eheust、A. K. Watson、J. O. McInerney、P. Lopez 和 E. Bapteste，《微生物世界中基因共享的二部网络分析》，《分子生物学与进化》，第 35 卷，第 899–913 页，2018。[8] C. Chi、Y. Ye、B. Chen 和 H. Huang，《基于二分图的方法：通过基因表达—药物反应关联对细胞系进行聚类》，生物信息学，第 37 卷，第 17 期，第 2617–2626 页，2021。[9] X. Xing、F. Yang、H. Li、J. Zhang、Y. Zhao、M. Gao、J. Huang 和 J. Yao，《基于共表达基因模块、用于疾病诊断和预后的多层注意力图神经网络》，生物信息学，第 38 卷，第 8 期，第 2178–2186 页，2022。[10] A. Abidi、R. Zhou、L. Chen 和 C. Liu，《基于支点的极大双团枚举》，载于 IJCAI，2020。[11] Z. Ma、Y. Liu、Y. Hu、J. Yang、C. Liu 和 H. Dai，《二部图流中极大双团的高效维护》，万维网，第 25 卷，第 2 期，第 857–877 页，2022。[12] L. Chen、C. Liu、R. Zhou、J. Xu 和 J. Li，《大型稀疏二部图中的高效极大双团枚举》，Proc. VLDB Endow.，第 15 卷，第 8 期，第 1559–1571 页，2022。[13] J. Wang、A. W. Fu 和 J. Cheng，《大型二部图中的矩形计数》，载于 IEEE国际大数据大会，2014。[14] K. Wang、X. Lin、L. Qin、W. Zhang 和 Y. Zhang，《基于顶点优先级的大规模二部网络蝴蝶计数》，Proc. VLDB Endow.，第 12 卷，第 10 期，第 1139–1152 页，2019。[15] S. Sanei-Mehri、A. E. Sariy¨uce 和 S. Tirthapura，《二部网络中的蝴蝶计数》，载于 KDD，2018。[16] A. Zhou、Y. Wang 和 L. Chen，《不确定二部网络中的蝴蝶计数》，Proc. VLDB Endow.，第 15 卷，223，2021 2，211第。页– 。[17] L. Chen、C. Liu、J. Xu 和 J. Li，《二分图中最大平衡二部团搜索的高效精确算法》，载于 SIGMOD会议，2021。[18] Y. Wang、S. Cai 和 M. Yin，《最大平衡二部团问题的新启发式方法》，《信息科学》，第 432 卷，第 362–375 页，2018。[19] P. Manurangsi，《基于小集扩张假设的最大二部团问题、最小k割和最稠密的至少k顶点子图的不可近似性》，《算法》，第 11 卷，第 1 期，第 10 页，2018。[20] J. Pardalos 和 M. Resende，《关于超大图中的最大团问题》，DIMACS丛书，第 50 卷，第 119–130 页，1999。[21] S. Verma 和 J. Rubin，《公平性定义详解》，载于 FairWare，2018。[22] M. Hardt、E. Price 和 N. Srebro，《监督学习中的机会均等》，载于 NIPS，2016。[23] C. Dwork、M. Hardt、T. Pitassi、O. Reingold 和 R. Zemel，《通过认知实现公平》，载于 ITCS，2012。[24] Y. Dong、J. Ma、C. Chen 和 J. Li，《图挖掘中的公平性：综述》，CoRR，第 abs/2204.09888 卷，2022。[25] M. Zehlike、F. Bonchi、C. Castillo、S. Hajian、M. Megahed 和 R. Baeza-Yates，《Fa* ir：一种公平的top-k排序算法》，载于 CIKM，2017。

[26] D. Serbos、S. Qi、N. Mamoulis、E. Pitoura 和 P. Tsaparas，《套餐到群组推荐中的公平性》，载于 WWW，2017。[27] A. Beutel、J. Chen、T. Doshi 等，《通过成对比较实现推荐排序的公平性》，载于 SIGKDD，2019。[28] H. Ma、S. Guan、C. Toomey 和 Y. Wu，《具有群体公平性的多样化子图查询生成》，载于 WSDM。ACM，2022，第 686–694 页。[29] A. Beutel、J. Chen、T. Doshi、H. Qian、L. Wei、Y. Wu、L. Heldt、Z. Zhao、L. Hong、E. H. Chi 和 C. Goodrow，《通过成对比较实现推荐排序的公平性》，载于 KDD，2019。[30] G. S. Sankar、A. Louis、M. Nasre 和 P. Nimbhorkar，《具有群体公平性约束的匹配：在线与离线算法》，CoRR，第 abs/2105.09522 卷，2021。[31] M. Pan、R. Li、Q. Zhang、Y. Dai、Q. Tian 和 G. Wang，《公平性感知的最大团枚举》，ICDE，2021。[32] E. Prisner，《图中的二分团 I：数量界》，Combina-torica，第 20 卷，第 1 期，第 109–117 页，2000。[33] V. Batagelj 和 M. Zaversnik，《一种用于网络核心分解的 o(m) 算法》，CoRR，第 cs.DS/0310049 卷，2003。[34] D. W. Matula 和 L. L. Beck，《最小度优先排序、聚类与图着色算法》，J. ACM，第 30 卷，第 3 期，第 417–427 页，1983。[35] W. Hasenplaugh、T. Kaler、T. B. Schardl 和 C. E. Leiserson，《并行图着色的排序启发式方法》，载于 SPAA，2014。[36] C. Zhang、F. Niu、C. R´e 和 J. W. Shavlik，《大数据与群体：在所有正确的地方寻找关系》，载于 ACL，2012 年。[37] F. Niu、C. Zhang、C. R´e 和 J. W. Shavlik，《通过对偶分解扩展马尔可夫逻辑推理》，载于 ICDM，M. J. Zaki、A. Siebes、J. X. Yu、B. Goethals、G. I. Webb 和 X. Wu 编，2012 年。[38] R. S. Xin、J. E. Gonzalez、M. J. Franklin 和 I. Stoica，《GraphX：基于 Spark 的弹性分布式图系统》，载于 GRADES。CWI/ACM，2013 年，第 2 页。[39] J. E. Gonzalez、R. S. Xin、A. Dave、D. Crankshaw、M. J. Franklin 和 I. Stoica，《GraphX：分布式数据流框架中的图处理》，载于 OSDI。USENIX 协会，2014 年，第 599–613 页。[40] J. Yang、Y. Peng 和 W. Zhang，《大型稀疏二部图的 (p, q)-二部团计数与枚举》，Proc. VLDB Endow.，第 15 卷，第 2 期，第 141–153 页，2021 年。[41] B. Lyu、L. Qin、X. Lin、Y. Zhang、Z. Qian 和 J. Zhou，《十亿级规模的最大二部团搜索》，Proc. VLDB Endow.，第 13 卷，第 9 期，第 1359–1372 页，2020 年。[42] K. Wang、W. Zhang、X. Lin、Y. Zhang、L. Qin 和 Y. Zhang，《大规模二部图上的高效且有效的社区搜索》，载于 ICDE，2021 年。[43] K. Wang、X. Lin、L. Qin、W. Zhang 和 Y. Zhang，《面向大规模二部网络的基于顶点优先级的蝴蝶计数》，Proc. VLDB Endow.，第 12 卷，第 10 期，第 1139–1152 页，2019 年。