软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(4):1796−1810 [doi: 10.13328/j.cnki.jos.007270] [CSTR: 32375.14.jos.007270]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向二部图的极大缺陷二团高效枚举算法*

代强强[1], 于瀚文[1], 李荣华[1,2], 李振军[2], 王国仁[1]

[1](北京理工大学 计算机学院, 北京 100081)

[2](深圳城市职业技术学院 信息与通信学院, 广东 深圳 518038)

通信作者: 李荣华, E-mail: lironghuabit@126.com

摘　要: 极大二团枚举问题是二部图分析中的一个基本研究问题. 然而, 在实际应用中, 传统二团模型要求子图必须为完全二部图的约束往往过于严格, 因此需要一些更为宽松的二团模型作为代替. 为此, 提出一种新的称之为 k-缺陷二团的松弛二团模型. 该模型允许二部图子图与完全子图二团最多相差 k 条边. 由于极大 k-缺陷二团枚举问题属于 NP-难问题, 设计高效的枚举算法是一项极具挑战性的任务. 为解决此问题, 提出一种基于对称集合枚举的算法. 该算法的思想是通过 k-缺陷二团中缺失边的数量约束来控制子分支的数量. 为进一步提高计算效率, 还提出一系列优化技术, 包括基于排序的子图划分方法、基于上界的剪枝方法、基于线性时间的更新技术以及分支的优化方法. 此外, 提出的优化算法的时间复杂度与 $O(\gamma_k^n)$ 有关, 其中 $\gamma_k < 2$, 突破了传统 $O(2^n)$ 的时间复杂度. 最后, 大量的实验结果表明, 在大部分参数条件下所提方法的效率相较于传统分支定界方法提高了 100 倍以上.

关键词: 二部图; 稠密子图挖掘; k-缺陷二团

中图法分类号: TP311

中文引用格式: 代强强, 于瀚文, 李荣华, 李振军, 王国仁. 面向二部图的极大缺陷二团高效枚举算法. 软件学报, 2025, 36(4): 1796–1810. http://www.jos.org.cn/1000-9825/7270.htm

英文引用格式: Dai QQ, Yu HW, Li RH, Li ZJ, Wang GR. Efficient Algorithms for Maximal Defective Biclique Enumeration on Bipartite Graphs. Ruan Jian Xue Bao/Journal of Software, 2025, 36(4): 1796–1810 (in Chinese). http://www.jos.org.cn/1000-9825/7270.htm

Efficient Algorithms for Maximal Defective Biclique Enumeration on Bipartite Graphs

DAI Qiang-Qiang[1], YU Han-Wen[1], LI Rong-Hua[1,2], LI Zhen-Jun[2], WANG Guo-Ren[1]

[1](School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

[2](School of Information and Communication, Shenzhen City Polytechnic, Shenzhen 518038, China)

Abstract: Maximal biclique enumeration is a fundamental research problem in the bipartite graph analysis field. However, the traditional biclique model, which requires the subgraph to be a complete bipartite graph, is often overly constrained in practical applications, and thus some looser biclique models are needed to substitute. In this study, a new relaxation biclique model called k-defective biclique is proposed. This model allows a bipartite subgraph to differ from a complete bipartite subgraph biclique by up to k edges. Since enumerating maximal k-defective bicliques is NP-hard, designing efficient enumeration algorithms is a challenging task. To solve this problem, an algorithm based on symmetric set enumeration is proposed. The idea of this algorithm is to control the number of sub-branches through a constraint on the number of missing edges in the k-defective bicliques. To further improve the computational efficiency, a series of optimization techniques are also proposed, including ordering-based subgraph partitioning method, upper-bound based pruning method, linear time-based updating technique, and optimization method for branching. In addition, the time complexity of the proposed optimization algorithms is related to, where breaks through the traditional limitation. Finally, a large number of experimental results show that the efficiency of the proposed method is over a hundred times higher than that of the traditional branch-and-bound approach for most parameter settings.

* 基金项目: 新一代人工智能国家科技重大专项 (2020AAA0108503); 国家自然科学基金 (U2241211, 62072034); 中国博士后创新人才支持计划 (BX20240467); 中国博士后科学基金 (2023M740245); 广东省哲学社会科学规划项目 (GD21CYj21); 深圳市教育科学“十四五”规划: 2023 年度项目 (rgzn23021)
收稿时间: 2024-05-10; 修改时间: 2024-06-27; 采用时间: 2024-08-02; jos 在线出版时间: 2025-01-08
CNKI 网络首发时间: 2025-01-15

Key words: bipartite graph; cohesive subgraph mining; k-defective biclique

随着信息技术的广泛应用, 各领域产生了大量关系复杂的图数据, 如社交网络数据、生物信息网络数据以及金融交易数据等. 这些网络数据中通常包含了许多的稠密子图结构, 如何从图数据中挖掘稠密子图结构是图分析中一个重要的研究问题. 因为, 稠密子图挖掘问题在社交网络分析[1]、蛋白质络合物分析[2]以及金融统计分析[3]等领域具有重要的应用价值. 例如在社交网络数据中, 稠密子图挖掘可以用于了解社区的内在结构特征, 识别用户群体之间的重要联系, 以及发现潜在的社交圈子或社交事件, 对社交媒体营销、舆情分析以及社交网络动态的理解具有重要意义.

然而, 某些特定领域中的数据一般具有不同的实体, 而且实体之间的关系仅在不同类型的实体之间产生, 利用传统图结构对数据进行建模已经无法满足各领域的应用需求. 例如电子商务中用户与商品的关系数据[4]、合作网络中作者与出版物的关系数据[5]、社交网络中用户与网页间的关系数据[6]以及生物信息网络中基因与蛋白质的关系数据[7]等均存在不同的实体. 为此, 一种常见的图结构, 即二部图, 也广泛用于建模现实应用中包含不同实体类型的图数据, 其表示顶点可以分为两个相互独立的集合, 并且图上每条边上的顶点分别属于不同的集合中.

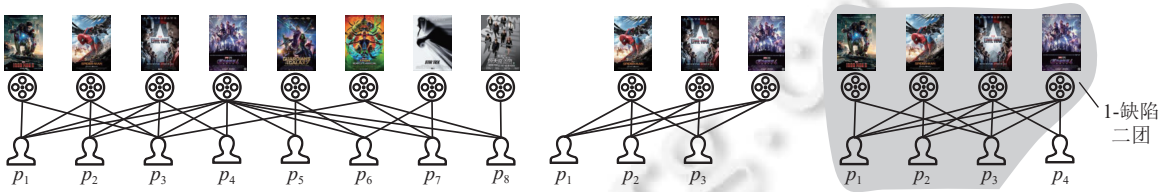
二团作为二部图中一种基本的稠密子图模型, 其表示为完全二部图子图, 即相互独立的集合 X 和 Y 之间的任何一对顶点都存在边连接. 由于该模型在社区检测[8−11]、欺诈检测[12]和生物网络分析[13]等领域的重要应用价值, 研究者针对极大二团枚举问题进行了广泛研究, 并提出了一系列的高效算法[14−20]. 然而, 要求二部图中两部分顶点集中任意一对顶点都存在边连接可能过于严格, 因为许多真实的二部图数据是基于传感器或者实验数据而收集生成的, 其中可能存在噪声或者错误. 此外, 在二部图数据中, 对于没有直接连接关系的顶点仍然可能具有较强的关联关系. 例如在电子商务的刷单行为检测中, 欺诈用户与被刷单的产品形成了一种稠密子图结构, 但每个欺诈用户可能不会对同一组商品进行频繁刷单. 因此, 利用二团模型进行欺诈检测极有可能导致挖掘结果不准确. 为了提高实际应用的效果, 一种可行的解决方法是从二部图数据中挖掘松弛的二团子图以替代二团模型的挖掘任务.

近年来, 研究者们还提出了许多其他的松弛的二团模型, 例如 (α, β)-核[21]、k-bitruss[22]以及 k-biplex[23]等. 然而, 上述松弛的二团模型在特定应用场景中仍然存在各自的缺陷性. 具体地, 基于最小度定义的 (α, β)-核模型和基于蝴蝶 (2×2 的二团) 数量定义的 k-bitruss 模型对子图的稠密度的限制过于松弛. 此类模型在处理真实图数据时, 可能导致检索出的子图社区规模过大 (顶点数量可能多达上千甚至上万个). 这样的结果可能无法直接使用, 仍然需要耗费较高的人力资源进行进一步的筛选. 此外, 虽然基于非邻居数量定义的 k-biplex 模型可以搜索到规模较小的社区结构, 但该模型仍然难以细粒度地检索二部图中社区的不同结构关系. 例如, 若二部图中稠密子图两边顶点数不大于 6 时, 该模型只能将 k 设置为 1 或者 2, 难以发现具有不同结构的社区.

为了解决上述问题, 本文将传统图数据中一种广泛研究的 k-缺陷团模型[24]引入至二部图数据的稠密子图建模中, 并提出了一种称之为 k-缺陷二团的稠密子图模型. 该模型表示与完全子图二团相差最多 k 条边的二部图子图. 显然, 当 $k = 0$ 时, k-缺陷二团等价于二团, 为此提出的模型是二团模型的一种泛化, 可为稠密子图挖掘任务提供更高的灵活性 (基于不同的 k 值搜索不同的稠密子图). 此外, 相较于 k-biplex, k-缺陷二团模型是基于缺失的边数 k 约束稠密子图的, 可以基于不同的阈值 k 以更细粒度地检测社区的结构变化情况. 图 1 进一步说明了提出的 k-缺陷二团模型相较于二团模型在真实社区挖掘中的优势. 具体地, 图 1(a) 展示了互联网电影资料库 (https://www.imdb.com/) 中包含 8 部电影和 8 个演员的二部图网络, 可以看出基于二团模型仅可以挖掘出 3×3 的二团社区关系 (图 1(b) 所示). 然而, 基于提出的 k-缺陷二团模型, 当设置 k 大于 0 时, 可以得到具有更多关系的稠密子图社区. 如当 k=3 时, 所提模型可以挖掘出 4×4 的社区关系 (图 1(c)). 此外, 所得社区仍然具有较高的稠密度, 表示所提模型在社区挖掘方面具有实际意义. 基于 Yannakakis 定理[25], 从二部图中枚举极大二团是一个 NP-难问题. 为此, 如何设计高效的算法从二部图中枚举极大 k-缺陷二团是一个极具挑战的问题. 针对该问题, 本文提出了一种新的对称集合枚举技术. 其主要思想是, 设 D 为集合 S 的非共同邻居集合, 当枚举包含 S 的极大 k-缺陷二团时, 当前任务可以划分为包含 S 和 D 中前 i 个顶点但不包含第 $i+1$ 个顶点的子任务, 其中 $0 \leqslant i \leqslant k$. 由于 k-缺陷二团中最多允许缺失 k 条边, 则子任务的个数最多被限制在 $k+1$ 个以内.

为了进一步提高计算效率, 本文还研究了 k-缺陷二团的相关性质, 提出了一系列的优化技术, 包括基于子图划分的预处理技术、基于上界的剪枝技术、线性时间的更新技术以及分支优化技术. 值得注意的是, 本文证明提出的优化枚举算法的时间复杂度与 $O(\gamma_k^n)$ 有关, 其中 $\gamma_k < 2$, 突破了传统枚举方法 $O(2^n)$ 的上界. 实验结果表明, 所提出的优化枚举算法, 相较于基准算法具有极大的性能提升, 即在大部分参数情况下, 优化枚举算法的时间性能要比传统枚举算法高 100 倍以上. 此外, 具体的案例分析证明了所研究模型的有效性.

本文第 1 节介绍面向二部图的稠密子图挖掘的相关工作. 第 2 节介绍本文的符号与问题定义. 第 3 节介绍本文提出的基于对称集合枚举技术的搜索算法. 第 4 节介绍所提算法的优化技术, 并证明优化算法的时间复杂度. 第 5 节通过大量的实验验证所提方法的高效性与有效性. 最后总结本文的研究工作.



(a) 部分电影-演员网络 (b) 极大二团社区 (c) 3-缺陷二团社区

图 1 互联网电影资料库 (IMDB) 电影-演员网络中的社区

1 相关工作

二团表示一个完全二部图, 是二部图数据中一种基本的稠密子图结构. 近年来, 研究者针对极大二团的枚举问题进行了广泛地研究, 并提出了一系列的高效算法[14−20]. 其中, Zhang 等人[16]提出了一种基于集合枚举的算法备受关注, 该算法的主要思想是从给定二部图 $G=(L,R,E)$ 中较小的顶点集合 L 或者 R 中递归枚举所有可能的子集. 因为对于任何子集 $A \subseteq L$, 若 B 是 A 的共同邻居, 则 (A,B) 组成的子图一定是一个二团. 同时, 作者证明该算法具有多项式延迟的时间复杂度. 为进一步提高计算效率, Das 等人[17]提出了一种基于排序的子图划分技术以将原始任务划分 $|R|$ (或者 $|L|$) 个子任务, Abidi 等人[18]提出了一种基于支配集的支撑点剪枝枚举技术, 以及 Chen 等人[19]基于一种新的排序技术和批量支撑点技术, 提出了一种具有多项式延迟的枚举算法. 最近, Dai 等人[20]基于新定义的支撑点技术开发了一种新的枚举框架, 并证明该框架同时具有接近最优的时间复杂度和多项式延迟的时间复杂度.

由于二团模型在某些特定情况下对稠密子图的限定过于严格, 为此研究者们还引入了一些典型的松弛二团模型[21−23]用于挖掘二部图数据中的稠密子图社区. 例如, Cerinšek 等人[21]基于最小度定义了一种 (α,β)-核的概念; Liu 等人[26]提出了一种在最优时间内查询 (α,β)-核的索引算法; 张毅豪等人[27]提出泛化距离的 (α,β)-核分解算法; Zou 等人[22]基于蝴蝶 (2×2 的二团) 的数量定义了 k-bitruss 模型; Wang 等人[28]同时提出了一种 k-bitruss 分解的算法以及 k-bitruss 的社区索引算法; Sim 等人[23]提出了基于非邻居的数量定义的 k-biplex 模型; Yu 等人[29]以及 Dai 等人[20]针对极大 k-biplex 枚举问题提出了一系列的改进算法; Ignatov 等人[30]提出了一种基于边密度定义的伪团模型. 基于之前的分析, 上述松弛的二团模型在特定应用场景下仍然存在一定的缺陷性. 为此, 本文定义了一种新的称之为 k-缺陷二团的稠密子图模型. 该模型表示与完全子图二团相差最多 k 条边的二部图子图. 据了解, 目前还没有针对二部图数据的 k-缺陷二团搜索算法. 虽然利用传统的分支定界枚举方法可用于解决该问题, 但是该方法存在大量的冗余计算导致其效率往往不高. 为此, 需要设计一种新的算法以提高其计算效率.

2 问题定义

令 $G=(L,R,E)$ 为一个无向无权的二部图, 其中 L 和 R 分别表示二部图中两个相互独立的顶点集, E 表示 G 的边集, 则对于任何边 $(u,v) \in E$ 都满足 $u \in L$ 且 $v \in R$ (或者 $v \in L$ 且 $u \in R$). 本文令 n 和 m 分别为 G 中的顶点数和边数, 即 $n=|L|+|R|$, $m=|E|$. 给定 L 中的一个顶点 v, 本文用 $N_v(G)$ 表示 v 在 G 中的邻居集合, 即 $N_v(G)=\{u \in R|(u,v) \in E\}$, 则 v 在 G 中的度表示为 $d_v(G)=|N_v(G)|$. 给定图 G 中的顶点集合 (A,B), 本文用 $G(A,B)=(A,B,E_{(A,B)})$ 表示 G 由

(A,B) 组成的诱导子图, 其中 $E_{(A,B)}=\{(u,v)\in E|u\in A,v\in B\}$. 为了方便描述, 本文所述的子图在未特别说明情况下均为顶点集合组成的诱导子图. 此外, 本文用 $d_v(A)$ 或者 $d_v(B)$ 表示顶点 v 在集合 A 或者 B 中的邻居个数, 即 $d_v(A)=|\{u\in A|(u,v)\in E\}|$ ($d_v(B)=|\{u\in B|(u,v)\in E\}|$). 下文正式定义了 k-缺陷二团的概念.

定义 1 (k-缺陷二团). 给定一个二部图 G 以及一个正整数 k, 若子图 $G(A,B)$ 中至少存在 $|A|\times|B|-k$ 条边, 则称 $G(A,B)$ 为 G 中的一个 k-缺陷二团.

基于定义 1, 若 G 中不存在其他满足 $A\subseteq A'$ 和 $B\subseteq B'$ 的子图 $G(A',B')$ 也是一个 k-缺陷二团, 则称 $G(A,B)$ 为 G 中的极大 k-缺陷二团. 可以看出, 当 $k=0$ 时, k-缺陷二团等价于二团. 因此二团可以看作是 k-缺陷二团的一种特殊情况. 下面将介绍两个关于 k-缺陷二团的重要性质, 其为设计具体的算法提供了方便. 为了叙述方便, 下文将直接使用 (A,B) 表示 k-缺陷二团 $G(A,B)$.

性质 1. k-缺陷二团满足继承性, 即对于给定的任意 k-缺陷二团 (A,B), 其每个子图 H 仍然是一个 k-缺陷二团.

证明: 假设 H 不为 k-缺陷二团, 则 H 中至少存在 $k+1$ 条缺失的边. 当将 $G(A,B)$ 中 H 之外的边加入 H 时, 子图 H 中缺失的边数不变, 表明 $G(A,B)$ 不是一个 k-缺陷二团. 因此, 造成矛盾.

在二部图 G 中, 设 $dis(G,u,v)$ 为顶点 u 与 v 在 G 中的最短距离. 本文用顶点间最长的最短距离表示该图的直径, 即 $\max_{\forall u,v\in G}(dis(G,u,v))$, 则对 k-缺陷二团, 有如下性质.

性质 2. 给定任意的 k-缺陷二团 (A,B), 若 $|A|\geqslant k+1$ 且 $|B|\geqslant k+1$, 则 $G(A,B)$ 的直径最大为 3.

证明: 基于 k-缺陷二团 (A,B) 的定义, 对于集合 A 中任意两个顶点 u_1 与 u_2, 满足 $d_{u_1}(B)+d_{u_2}(B)\geqslant 2|B|-k$, 其中 $d_{u_1}(B)$ 表示 u_1 在 B 中的邻居个数. 此外, 若 $G(A,B)$ 的直径为 3, u_1 与 u_2 一定存在共同邻居, 则有 $2|B|-k\geqslant|B|+1$. 因此, 可以推出 $|B|\geqslant k+1$. 类似地, 基于集合 B 中任意两个顶点 v_1 与 v_2 之间的邻居关系也可以推出 $|A|\geqslant k+1$. 因此, 得证.

基于性质 1, 可以容易判断给定的 k-缺陷二团 (A,B) 在 G 中是否是极大的, 即当 G 中存在其他顶点 v 加入 (A,B) 组成更大的 k-缺陷团, 则 (A,B) 是非极大的; 否则, (A,B) 是非极大的. 基于性质 2 可以看出, 对于任何 $|A|\geqslant k+1$ 且 $|B|\geqslant k+1$ 的极大 k-缺陷二团, 其一定是紧密连接的. 表明较大的 k-缺陷二团可以作为二部图中的稠密子图. 因此本文将重点研究枚举顶点数满足一定阈值的极大 k-缺陷二团, 其正式的问题定义如下.

问题定义. 给定二部图 G 以及两个正整数 k 和 $q\geqslant k+1$, 本文的目标是从图 G 中枚举所有顶点数量满足 $|A|\geqslant q$ 和 $|B|\geqslant q$ 的极大 k-缺陷二团 (A,B).

基于文献 [25], 从二部图中搜索满足继承性的最大子图是一个 NP-难问题, 则枚举所有极大 k-缺陷二团的问题也是一个 NP-难问题. 接下来, 本文将介绍一种高效的算法以解决该问题.

3 极大缺陷二团枚举算法

本节将介绍一种新的极大 k-缺陷二团枚举算法, 其主要思想是基于一种对称集合技术. 该技术与传统的集合枚举形成对称关系, 并且具有更强的冗余分支剪枝能力, 从而具有更高的计算性能.

3.1 对称集合枚举技术

在介绍具体的枚举技术之前, 首先介绍几种频繁使用的概念.

定义 2 (当前解). 集合 (S_L,S_R), 满足 k-缺陷二团的定义, 但不一定是极大的.

定义 3 (候选集). 集合 (C_L,C_R), 其中任何顶点均可加入当前解 (S_L,S_R) 组成更大的解.

定义 4 (排除集). 集合 (X_L,X_R), 候选集中已经用于扩展过当前解 (S_L,S_R) 的顶点集合.

基于上述定义, 一种最基本的极大 k-缺陷二团的方法是: 首先设置当前解为空集, 并设置候选集为 (L,R), 然后递归地从候选集中选择顶点来扩展当前解以枚举候选集中的所有子集. 可以看出每次递归一定存在与候选集大小相等的子分支数量, 所以具有非常低的性能. 为此, 下面将介绍一种对称集合枚举的方法来减少子分支的数量.

给定当前解 (S_L,S_R) 与候选集 (C_L,C_R), 可以发现集合 $S_L\cup C_L$ 中任何包含 S_L 的子集一定满足如下情况之一, 即包含 C_L 的前 $i-1$ 个顶点但不包含第 i 个顶点, 其中 $i\in[1,n+1]$ 的整数. 基于该思想, 可以设计一种递归方法以枚举给定二部图 G 中所有极大的 k-缺陷二团. 令 $Br=\langle S_L,S_R,C_L,C_R,X_L,X_R\rangle$ 为某个递归分支. 设 $C_L=\{v_1,\ldots,v_n\}$, 则

基于 C_L 中的顶点可以将分支 Br 划分为如下 $|C_L|+1$ 个子分支:

$$Br_i=\langle S_L\cup D_{i-1},S_R,C_L\backslash D_i,C_L\backslash\{v_i\},C_R\rangle,\text{ 其中 }D_i=\{v_1,\ldots,v_i\}\text{ 且 }i\in[1,n+1].$$

上述针对 Br 的分支过程称之为对称集合枚举. 值得注意的是, 同样可以利用候选集 C_R 扩展当前解. 为了进一步阐述对称集合枚举的思想, 图 2 展示了使用候选集 $C=\{v_1,v_2,\ldots,v_n\}$ 来扩展当前解的枚举树.

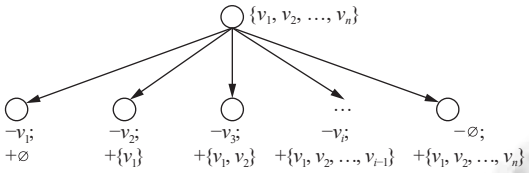


图 2 对称集合枚举树

然而直接利用上述方法, 递归分支 Br 的子分支数量仍然与 C_L 的大小相关. 考虑到 k-缺陷二团最多允许缺陷 k 条边的限制, 则候选集 $C_L\backslash N(S_R)$ 中最多允许 k 个顶点同时扩展当前解, 其中 $N(S_R)$ 表示顶点 S_R 的共同邻居, 即 $N(S_R)=\{u\in L|S_R\subseteq N_u(G)\}$. 为此可以得到如下针对 k-缺陷二团枚举的扩展方式.

定理 1. 给定递归分支 $Br=\langle S_L,S_R,C_L,C_R,X_L,X_R\rangle$, 设 $D=C_L\backslash N(S_R)=\{v_1,v_2,\ldots,v_d\}$, 其中 $d=|C_L\backslash N(S_R)|$. 若 $|D|>k$, 则如下 k+1 个子分支可用于枚举所有包含 (S_L,S_R) 的极大 k-缺陷二团:

- 第 1 个子分支为 $Br_1=\langle S_L,S_R,C_L\backslash\{v_1\},C_L\backslash\{v_1\},C_R\rangle$;
- 第 i 个子分支为 $Br_i=\langle S_L\cup\{v_1,\ldots,v_{i-1}\},S_R,C_L\backslash\{v_1,\ldots,v_i\},C_L\backslash\{v_i\},C_R\rangle$, 其中 i 为 2 到 k 的整数;
- 第 k+1 个分支为 $Br_{k+1}=\langle S_L\cup\{v_1,\ldots,v_k\},S_R,C_L\backslash\{v_1,\ldots,v_d\},C_L,C_R\rangle$.

由定理 1 可知, 若候选集中存在至少 k+1 个当前解的非共同邻居时, 递归分支 Br 的子分支个数仅与 k 的大小有关与候选集的大小无关. 在实际应用中, k 通常取值较小, 为此该分支方法相较于基本的对称集合枚举具有更高的效率. 下面的例子进一步阐明了定理 1 相较于基本的对称集合枚举的优势.

为了阐述定理 1 从图 3 的二部图 G 中枚举包含 (S_L,S_R) 的极大 k-缺陷二团的分支过程, 其中 $k=2$, $S_L=\{u_1\}$ 以及 $S_R=\{v_1\}$, 图 4 展示了一个具体的实例, 其中黑色部分的子分支由定理 1 得到, 子分支数量为 3; 灰色部分的子分支为冗余分支. 在递归分支之前, 需要计算 S_L 在候选集 C_R 中的非共同邻居, 即 $C_R\backslash N(S_L)=\{v_3,v_4,v_5,v_6\}$. 由于 $C_R\backslash N(S_L)$ 的大小大于 k, 为此可以选择 $C_R\backslash N(S_L)$ 中前 k 个顶点执行定理 1 的分支过程. 具体地, 第 1 个子分支为枚举不包含 v_3 的极大 k-缺陷二团; 第 2 个子分支为枚举包含 v_3 但是不包含 v_4 的极大 k-缺陷二团; 第 3 个子分支为枚举包含 $\{v_3,v_4\}$ 的极大 k-缺陷二团. 值得注意的是, 基于 k-缺陷二团的定义, $C_R\backslash N(S_L)$ 中最多允许 k 个顶点加入当前 (S_L,S_R) 中, 为此在枚举包含 $\{v_3,v_4\}$ 的子分支中, 候选集一定不包含 $\{v_5,v_6\}$. 此外, $C_R\backslash N(S_L)$ 中不存在大于 k 个的顶点加入当前解中, 因此该分支方法的子分支数量仅为 $k+1=3$ 个, 与候选集 C_L (或者 C_R) 的大小无关. 表明该方法可以极大地减少冗余分支的数量.

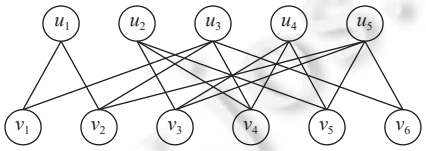


图 3 二部图例图

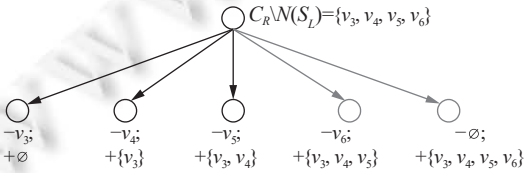


图 4 当 $k=2$ 时枚举包含 $(S_L=\{u_1\},S_R=\{v_1\})$ 的对称集合枚举树

对于某个递归分支 Br, $C_L\backslash N(S_R)$ 的大小还可能小于 k. 在此情况下, 定理 1 将不再适用. 为了确保递归能够继续进行, 本文针对此情况还设计了一种简单的分支技术, 其思想是从 C_L 中任意选择一个顶点 v, 然后将递归分支划分为两个子分支 Br_1 与 Br_2 以分别枚举包含 v 与不包含 v 的极大 k-缺陷二团.

值得注意的时, 文献 [12] 中也使用了一种称之为对称集合枚举的技术以解决极大 k-biplex (定义如文献 [12] 所示) 的枚举问题, 其基本思想与图 2 一致. 然而, 针对极大 k-缺陷二团枚举问题, 本文提出的枚举策略与文献 [12] 中的方法具有较大差异. 具体而言, 文献 [12] 的方法首先从当前递归的搜索空间 (由候选集和当前解组成的诱导子图) 中找出一个具有最小度的顶点 v, 然后基于顶点 v 是否存在于当前解中, 决定是否执行对称集合枚举技术. 而本文提出的方法 (如定理 1 所示), 主要基于当前解中的顶点在候选集中非共同邻居的个数来执行对称集合枚举技术. 此外, 极大 k-biplex 的枚举问题可以基于最小度的大小提前终止当前递归, 但极大 k-缺陷二团枚举问题没有类似的提前终止优化. 因此, 设计高效的极大 k-缺陷二团枚举算法相较于极大 k-biplex 的枚举问题更具挑战性.

3.2 算法实现

基于第 3.1 节中提出的分支方法, 本节将设计一种算法以枚举给定图中的极大 k-缺陷二团. 其具体的伪代码如算法 1 所示.

算法 1. 基本枚举算法.

输入: 二部图 G, 正整数 k 与 $q \geqslant k+1$;
输出: 所有不小于 q 的极大 k-缺陷二团.

```
1. Branch(∅, ∅, L, R, ∅, ∅);
2. Function Branch(S_L, S_R, C_L, C_R, X_L, X_R)
3.   if C_L ∪ C_R = ∅ then
4.     if X_L ∪ X_R = ∅ ∧ |S_L| ⩾ q ∧ |S_R| ⩾ q then
5.       输出 (S_L, S_R) 为极大解;
6.     end if
7.     return;
8.   end if
9.   D_L = C_L\N(S_R) = {v_1, ..., v_{d_L}}; D_R = C_R\N(S_L) = {u_1, ..., u_{d_R}}; s ← k − d̄(S_L, S_R);
10.  if |D_L| ⩾ s then
11.    Branch(S_L, S_R, C_L, C_R, X_L\{v_1}, X_R);
12.    for i = 2 to s then
13.      S'_L ← S_L ∪ {v_1, ..., v_{i−1}};
14.      C'_L ← C_L\{v_1, ..., v_i}; X'_L ← X_L\{v_i};
15.      更新 C'_L, C_R, X'_L, X_R 使其中任意顶点均可用于扩展当前解 (S'_L, S_R);
16.      Branch(S'_L, S_R, C'_L, C_R, X'_L, X_R);
17.    end for
18.    S'_L ← S_L ∪ {v_1, ..., v_s}; C'_L ← C_L\D_L;
19.    更新 C'_L, C_R, X'_L, X_R 使其中任意顶点均可用于扩展当前解 (S'_L, S_R);
20.    Branch(S'_L, S_R, C'_L, C_R, X_L, X_R);
21.  else if |D_R| > s then
22.    用 D_R 中的顶点执行 10–20 行;
23.    else
24.      C_L ∪ C_R 中选择一个顶点 v, 假设 v ∈ C_L; S'_L ← S_L ∪ {v};
```

25. 更新 C'_L, C_R, X'_L, X_R 使其中任意顶点均可用于扩展当前解 (S'_L,S_R);
26. *Branch*(S'_L, S_R, $C_L\backslash\{v\}$, C_R, X_L, X_R);
27. *Branch*(S_L, S_R, C_L, C_R, $X_L\backslash\{v\}$, X_R);
28. **end if**

算法 1 主要调用 *Branch* 函数来给定二部图 G 中所有的极大 k-缺陷二团, 该函数需要调用 6 个参数, 分别为: S_L、S_R、C_L、C_R、X_L 以及 X_R. 其中 (S_L,S_R) 表示当前解, (C_L,C_R) 表示候选集以及 (X_L,X_R) 表示排除集. 初始阶段, 参数 S_L、S_R、X_L 和 X_R 均设置为空集, 而 C_L 和 C_R 分别设置为 L 与 R (第 1 行). 函数 *Branch* 在递归调用之前, 首先需要判断 (C_L,C_R) 和 (X_L,X_R) 是否为空集, 若 (C_L,C_R) 为空集则表示没有顶点可用于扩展当前解, 该递归调用终止迭代 (第 3、7 行). 与此同时, 若 (X_L,X_R) 也为空集, 则当前解 (S_L,S_R) 在 G 中一定是一个极大解, 可作为结果输出 (第 4、5 行), 因为二部图 G 中在 (C_L,C_R) 与 (X_L,X_R) 之外的顶点一定无法扩展 (C_L,C_R).

然后, 函数 *Branch* 计算 (S_L,S_R) 在候选集 (C_L,C_R) 中的非共同邻居集合 D_L 与 D_R (第 9 行). 若 D_L 与 D_R 中存在一个集合满足定理 1 中用于扩展当前解的条件, 则算法利用其中之一来执行对称集合枚举技术 (第 10 或 21 行). 值得注意的是, 当前解 (S_L,S_R) 中可能已经存在一些边缺失, D_L 或 D_R 中最多有 $k-\bar{d}(S_L,S_R)$ 个顶点加入 (S_L,S_R) 中, 其中 $\bar{d}(S_L,S_R)$ 表示子图 (S_L,S_R) 中缺失的边数. 令 s 为 $k-\bar{d}(S_L,S_R)$ (第 9 行), 基于定理 1, 最多可产生 $s+1$ 个子分支. 假设函数是从 D_L 中选择顶点执行子递归调用, 设 $D_L=\{v_1,\ldots,v_{d_L}\}$, 函数 *Branch* 首先调用子递归枚举不包含 v_1 的极大 k-缺陷二团 (第 11 行), 其次枚举包含 $\{v_1,\ldots,v_{i-1}\}$ 但不包含 v_i 的极大 k-缺陷二团 (第 12–17 行), 最后枚举包含 $\{v_1,\ldots,v_s\}$ 的极大 k-缺二团 (第 18–20 行). 此外, 当 D_L 与 D_R 的大小都不满足条件时 (大于等于 s), 函数将从候选集 (C_L,C_R) 中随机选择一个顶点分别枚举包含 v 与不包含 v 的极大 k-缺陷团以保证递归的继续执行 (第 23–28 行).

需要注意的是, 当某个顶点 v 或者集合 $\{v_1,\ldots,v_i\}$ 加入 (S_L,S_R) 之后, 候选集中可能存在某些顶点不再能够用于扩展新的当前解 (S'_L,S_R) 了. 为了删除这些顶点, 最直接的方法是, 算法依次遍历候选集 (C_L,C_R) (以及排除集 (X_L,X_R)) 中每个顶点, 然后尝试加入 (S'_L,S_R) 中, 并判断是否能够组成更大的 k-缺陷二团. 如果不能, 则该顶点从候选集 (或者排除集) 中删除, 剩余的每个顶点均可用于扩展新的当前解.

定理 2. 算法 1 正确并唯一地输出了二部图 G 中所有满足给定阈值条件的极大 k-缺陷二团.

证明: 设 (A,B) 为 G 中任意的极大 k-缺陷二团, 其中 $A=\{v_1,\ldots,v_A\}$ 以及 $B=\{u_1,\ldots,u_B\}$, 则只需证明 (A,B) 能够被算法 1 唯一枚举. 定理 1 证明了算法 1 一定会输出极大解 (A,B), 则只需证明当 (A,B) 被枚举后, 包含 (A,B) 的非极大解不会被输出. 由于算法是基于深度优先的方式扩展当前解 (S_L,S_R), 因此算法一旦输出 (A,B) 则顶点 v_A 或者 u_B 将被放入到排除集中. 那么对于其他满足 $S_L\subseteq A$ 且 $S_R\subseteq B$ 的子递归分支, v_A 或者 u_B 一定存在于当前的排除集合中, 则任何包含 (A,B) 的非极大解都不会输出. 因此, 得证.

4 优化技术

为了进一步提高所提出算法的性能, 本节将介绍一系列的优化方法. 主要包括基于排序的子图划分技术, 基于上界的剪枝技术, 线性时间的更新技术以及分支优化技术.

4.1 基于排序的子图划分技术

给定 L 中的一个顶点 v, 令 $N_v^2(G)$ 表示 G 中与顶点 v 距离为 2 的顶点集合, 即 $N_v^2(G)=\{u\in L\backslash\{v\}|N_v(G)\cap N_u(G)\neq\varnothing\}$. 本文用 G_v 表示 G 中由顶点集合 $\Gamma_v^2(G)=N_v^2(G)\cup\{v\}$ 与 $\cup_{u\in\Gamma_v^2(G)}N_u(G)$ 组成的诱导子图. 对于 R 中的顶点 u 同样有如上定义, 则可以得到如下结论.

定理 3. 给定 G 中任意极大 k-缺陷二团 (A,B), 若 $|A|\geqslant k+1$ 且 $|B|\geqslant k+1$, 则 (A,B) 一定包含在子图 G_v 中, 其中 v 为 (A,B) 中的任意顶点.

证明: 基于性质 2, 若 $|A|\geqslant k+1$ 且 $|B|\geqslant k+1$, 则 $G(A,B)$ 的直径一定不大于 3. 又因为 G_v 包含了与 v 距离小于

等于 3 的所有顶点, 则 (A,B) 一定存在于 G_v 中.

由于同一个极大 k-缺陷二团 (A,B) 同时包含在多个子图 G_v 中, 从而造成了重复枚举. 为了避免该情况, 可以利用排序进行子图划分. 设 $O=\{v_1,v_2,\ldots,v_n\}$ 为二部图 G 中所有顶点 (包括 L 和 R) 按照某种规则由低到高的排序. 令 $N_{v_i}^{>2}(G)$ 表示 G 中与顶点 v_i 距离为 2 且基于排序 O 比 v_i 排名高的顶点集合, 即 $N_{v_i}^{>2}(G)=\{v_j\in O\wedge L|i<j, N_{v_i}(G)\cap N_{v_j}(G)\neq\varnothing\}$. 设 $G_{v_i}^{>}$ 表示 G 中由顶点集合 $\Gamma_{v_i}^2(G)=N_{v_i}^2(G)\cup\{v_i\}$ 与 $\cup_{u\in\Gamma_{v_i}^2(G)}N_u(G)$ 组成的诱导子图, 则有如下结论.

定理 4. 给定 G 中任意极大 k-缺陷二团 (A,B), 若 $|A|\geqslant k+1$ 且 $|B|\geqslant k+1$, 则 (A,B) 一定包含在子图 $G_v^{>}$ 中, 其中 v 为 A 中基于排序 O 排名最低的顶点.

证明: 因为 v 为 A 中基于 O 排名最低顶点, 则 $\Gamma_v^2(G)$ 包含了 A 中所有的顶点. 此外, B 中任意顶点都与 A 中至少某一个顶点存在邻居关系, 则子图 $G_v^{>}$ 包含 (A,B).

定理 5. 给定 G 中任意极大 k-缺陷二团 (A,B), 若 $|A|\geqslant k+1$ 且 $|B|\geqslant k+1$, 若 v 不是 A 中基于排序 O 排名最低的顶点, 则 (A,B) 一定不包含于子图 $G_v^{>}$ 中.

证明: 因为 v 不是 A 中基于 O 排名最低的顶点, 则 A 一定不包含于 $\Gamma_v^2(G)$ 中, 因此 A 不包含 (A,B).

基于上述结论, 从原始二部图 G 中枚举所有极大 k-缺陷二团等价于从各个子图 $G_v^{>}$ 中枚举所有极大 k-缺陷二团, 其中 v 属于 $O\cap L$ (或者 $O\cap R$) 中. 本文利用传统图中广泛使用的退化排序 (degeneracy ordering)[20]对 G 进行排序, 其定义如下.

定义 5 (退化排序). 给定二部图 G, 退化排序为 G 中所有顶点 (包括 L 和 R) 的一种排列 $\{v_1,v_2,\ldots,v_n\}$ 使得任意 v_i 在由 $\{v_i,v_{i+1},\ldots,v_n\}$ 组成的诱导子图中度最小, 其中 i 为 $[1,n]$ 中的整数.

基于一种传统的迭代删除方法[31,32], 退化排序可在线性时间内完成计算. 具体地, 给定二部图 G, 依次删除在剩余顶点组成的子图中度数最小的顶点, 顶点的删除顺序组成了一种退化排序.

4.2 基于上界的剪枝技术

当从 $G_v^{>}$ 中枚举大小 (顶点数) 约束的极大 k-缺陷二团时, 其中可能存在某些顶点一定不包含于任何满足条件的极大 k-缺陷二团中. 为此, 还需要对 $G_v^{>}$ 进行进一步的削减以减少不必要的计算. 下面首先介绍一种基于 (α,β)-核[21]的剪枝方法.

定义 6 ((α,β)-核). 给定二部图 G, 若子图 $G(A,B)$ 为 G 中的一个 (α,β)-核, 则对于 $\forall v\in A$ (以及 $\forall u\in B$) 均满足 $d_v(G(A,B))\geqslant\alpha$ ($d_u(G(A,B))\geqslant\beta$).

基于 k-缺陷二团的定义, 显然有如下定理.

定理 6. 给定任意 k-缺陷二团 (A,B), $G(A,B)$ 一定为一个 ($|B|-k$, $|A|-k$)-核.

为此从 $G_v^{>}$ 中枚举包含 v 的极大 k-缺陷二团之前, 可以基于定理 6 使子图成为一个 ($q-k$, $q-k$)-核. 为进一步减少冗余顶点, 本文发现, 在枚举包含 v 且大小约束的极大解时, 任何其他顶点与 v 至少存在 $q-k$ 个共同邻居. 则可以得到如下结论.

定理 7. 给定任意 k-缺陷二团 (A,B), 对于任何两个顶点 $v,u\in A$, 一定有 $|N_v(G)\cap N_u(G)|\geqslant|B|-q$; 若 $v,u\in B$, 同样有 $|N_v(G)\cap N_u(G)|\geqslant|A|-q$.

证明: 因为顶点 $v,u\in A$ 包含在同一个 k-缺陷二团中, 表明 (A,B) 中最多允许有 k 个顶点在 v 与 u 的非共同邻居中. 为此 v 与 u 的共同邻居的数量一定不小于 $|B|-q$. 此外, 若 $v,u\in B$, 也有类似结论.

基于定理 7, 本文设计了一种新的剪枝技术. 即给定子图 $G_v^{>}=(L_v^{>},R_v,E_v)$, 若 $v\in L_v^{>}$, 首先从 $L_v^{>}$ 中删除与 v 共同邻居数量小于 $q-k$ 的顶点, 然后从 R_v 删除度数 $q-k$ 的顶点, 之后迭代重复上述操作直到所有顶点不能删除为止. 基于现有的剥离算法[31,32], 该剪枝技术的时间与子图的顶点数和边数成线性关系.

4.3 线性时间的更新技术

在算法 1 中, 当利用某个顶点 v 扩展当前解 (S_L,S_R) 时, 需要更新候选集和排除集以保持剩余顶点仍然可以扩展新的当前解. 然而传统方法的时间复杂度与 (S_L,S_R) 的大小成多项式关系. 为提高更新效率, 本节介绍一种线性时间的更新方法. 设候选集 (C_L,C_R) 中每个顶点是一个对结构, 即任何顶点 $u\in C_L$ (或者 $u\in C_R$) 包含了元素 id 与 nnd, 其

中 id 为顶点 u 的编号, 表示为 $u.id$; nnd 为顶点 u 在 (S_L,S_R) 中的非邻居数量, 表示为 $u.nnd$. 基于该结构, 当 v 从候选集中加入 (S_L,S_R) 时, 只需利用 v 与候选集中其他顶点 u 是否是邻居关系以及 $v.nnd$ 与 $u.nnd$ 的大小即可确定顶点 u 是否可以扩展新的当前解. 基于该思想, 本文设计了一种新的更新算法, 其伪代码如算法 2 所示. 容易看出其时间复杂度与候选集的大小线性相关, 因为判断 $(v.id,u.id)$ 是否是图 G 中的边可以基于索引技术在线性时间内完成.

算法 2. $Update(S_L,v,S_R,C_L,C_R)$.

1. $\bar{d}(S_L,S_R) \leftarrow G(S_L,S_R)$ 中缺失的边数;
2. $C'_L \leftarrow \varnothing$; $C'_R \leftarrow \varnothing$;
3. **foreach** $u \in C_L$ **then**
4. **if** $\bar{d}(S_L,S_R)+v.nnb+u.nnb \leqslant k$ **then**
5. $C'_L.push(u)$;
6. **end if**
7. **end for**
8. **foreach** $u \in C_R$ **then**
9. **if** $(v.id,u.id) \notin E$ **then** $u.nnb \leftarrow u.nnb+1$; **end if**
10. **if** $\bar{d}(S_L,S_R)+v.nnb+u.nnb \leqslant k$ **then**
11. $C'_R.push(u)$;
12. **end if**
13. **end for**
14. **return** (C'_L,C'_R);

值得注意的是算法 2 仅展示了顶点 v 加入 S_L 时更新候选集 (C_L,C_R) 的方法, 而排除集 (X_L,X_R) 的更新方法与此类似. 同时当 v 加入 S_R 时, 候选集与排除集的更新方法与 v 加入 S_L 时也是类似的. 为了避免重复, 本文省略了针对上述情况的更新方法.

4.4 分支优化技术

在当前递归分支无法利用定理 1 来产生子递归分支时, 算法 1 将使用传统的分支定界技术以保证递归的继续. 然而该方法的效率很低, 可能产生大量的冗余分支. 本文发现, 在执行分支定界技术过程中, 若用于扩展候选集 (S_L,S_R) 的顶点 v 能够使得新的当前解在候选集中具有更多的非邻居, 则可以提升枚举算法的性能. 因为当前解 (S_L,S_R) 在候选集中的非邻居数量越多, 不仅能够减少分支定界枚举的使用, 而且还有利于提高对称集合枚举的效率. 为此, 当递归分支不满足定理 1 的要求时, 本文将从候选集 (C_L,C_R) 中选择在 $G(S_L \cup C_L,S_R \cup C_R)$ 中度数最小的顶点执行分支定界枚举.

除了上述提到的, 在算法 1 的 *Branch* 函数中, 本文还发现可以利用排除集中的顶点来判断当前递归是否可以提前终止, 以进一步减少冗余计算. 其主要思想是, 如果能够确定当前搜索空间中任何包含当前解 (S_L,S_R) 的 k-缺陷二团在原始二部图中一定不是极大的, 则当前递归可以直接终止计算. 为此, 可以得到如下结论.

定理 8. 在某个递归分支中, 若排除集 (X_L,X_R) 中存在顶点 $v \in X_L$ (或者 $u \in X_R$) 满足 $(S_R \cup C_R) \subseteq N_v(G)$ (或者 $(S_L \cup C_L) \subseteq N_u(G)$), 则当前搜索空间中不存在极大解.

证明: 设 (A,B) 是当前递归搜索到的极大 k-缺陷二团, 基于条件 $(S_R \cup C_R) \subseteq N_v(G)$ (或者 $(S_L \cup C_L) \subseteq N_u(G)$), 可以容易看出 $(A \cup \{v\},B)$ (或者 $(A,B \cup \{u\})$) 一定是一个更大的 k-缺陷二团. 因此, 该递归分支中的任何解都不是极大的.

综合上述所有的优化技术, 可以得到一种改进的极大 k-缺陷二团枚举算法, 其具体的伪代码如算法 3 所示. 具体的, 算法 3 首先基于排序优化将原始二部图划分成一系列的子图 $G^{>}_{v_i} = (L^{>}_{v_i},R_{v_i},E_{v_i})$ (第 1–3 行), 然后利用上界剪枝技术以减少子图的规模 (第 4 行), 最后在各个子图中调用 *Branch* 函数 (第 5 行) 枚举极大 k-缺陷二团. 值得注意的是, 递归调用 *Branch* 加入了上界剪枝技术、线性更新技术以及分支优化技术.

算法 3. 优化枚举算法.

输入: 二部图 G 以及正整数 k 与 $q \geqslant k+1$;
输出: 所有不小于 q 的极大 k-缺陷二团.

1. $O \leftarrow \{v_1, v_2, \ldots, v_n\}$ 为 G 中顶点的退化排序;
2. **foreach** $v_i \in O \wedge v_i \in L$ **then**
3. 构造子图 $G_{v_i}^{>} = (L_{v_i}^{>}, R_{v_i}, E_{v_i})$ 并利用上界剪枝技术删除 $G_v^{>}$ 中的冗余顶点;
4. $X_L \leftarrow \{v_j \in R_{v_i} | j < i, d_{v_j}(G_{v_i}^{>}) \geqslant q-k\}$;
5. *Branch*(v_i, $\varnothing$, $L_{v_i}^{>}$, R_{v_i}, X_L, $\varnothing$); /*该递归加入了上界剪枝技术、线性更新技术以及分支优化技术*/
6. **end for**

定理 9. 算法 3 的时间复杂度为 $O(nm'\gamma_k^{n'})$, 其中 γ_k 为线性方程 $x^{2k+5} - 2x^{2k+4} + x^3 - 2x + 2 = 0$ 的最大实根, n' 与 m' 分别为最大子图 $G_v^{>}$ 的顶点数和边数, 如当 $k = 0$, 1 和 2 时, γ_k 分别等于 1.544, 1.891 和 1.975.

证明: 容易看出算法 3 的时间复杂度为处理最大子图 $G_v^{>}$ 的时间乘以 n. 此外, 在处理子图 $G_v^{>}$ 时, 算法 3 的时间复杂度主要与 *Branch* 的递归数量以及每层递归消耗的时间有关. 为此, 设 $T(n)$ 为 *Branch* 处理子图 $G_v^{>}$ 的总递归次数, 则算法的时间复杂度为 $O(nm'T(n))$. 因为 *Update* 的时间复杂度为线性的所以每层递归的时间复杂度为 $O(m')$. 下面将重点分析 $T(n)$ 的大小上界.

当 $|C_L \backslash N(S_R)| > k$ 时, 算法将利用对称集合枚举技术搜索极大 k-缺陷二团. 由于该技术最多生成 $k+1$ 个子分支, 则容易得到 $T(n) = \sum_{i=1}^{k+1} T(n-i)$ 的递归关系.

当 $|C_L \backslash N(S_R)| \leqslant k$ 时, 算法直接使用分支定界算法枚举极大 k-缺陷二团, 容易得到 $T(n) = T(n-1) + T(n-1)$ 的递归关系. 然而, 由于被选择分支的顶点 v 在当前搜索空间中都具有最小的度. 为此, 在枚举包含 v 的子分支 $T(n-1)$ 中, 在最坏情况下, 该节点与候选集中的顶点最少存在一个非邻居. 而且, 在该子分支中, 候选集中另外一个具有最小度的顶点 u 将用于扩展新的当前解, 从而使得当前解在候选集中的非共同邻居数量进一步增大. 在最坏情况下, 当 $k+1$ 个顶点加入当前解 (S_L, S_R) 时, 该子递归分支 $T(n-k-1)$ 中满足条件 $|C_L \backslash N(S_R)| > k$. 因此, 可以得到 $T(n-k-1) = \sum_{i=1}^{k+1} T(n-i-k-1)$. 综上, 可以得到一种更紧的最终递归关系 $T(n) = \sum_{i=1}^{2k+2} T(n-i)$. 此外, 由于最深的递归 $T(n-2k-2)$ 等价于二团枚举, 可以容易验证, 在该情况下的递归关系为 $T(n-2k-2) = 2T(n-2k-4)$. 综上, 算法 3 的最终的递归关系为 $T(n) = \sum_{i=1}^{2k+1} T(n-i) + 2T(n-2k-4)$, 基于文献 [33] 中的定理 2.1, 可以证明 $T(n)$ 的大小与 γ_k^n 有关, 其中 γ_k^n 为方程 $x^{2k+5} - 2x^{2k+4} + x^3 - 2x + 2 = 0$ 的最大实根. 因此, 得证.

5 实验分析

5.1 实验设置

(1) 算法: 本文实现了 3 个算法, 用于枚举给定二部图中的极大 k-缺陷二团, 分别为 iMDCE、MDCE 和 Basic. 其中, iMDCE 是提出的算法 3, 包含了本文提出的所有优化技术; MDCE 是在 iMDCE 的基础上去除了第 4.4 节分支优化部分的极大 k-缺陷二团枚举算法; Basic 是在 MDCE 的基础上去除对称集合枚举技术的极大 k-缺陷二团枚举算法. 值得注意的是, iMDCE 与 MDCE 和 Basic 的唯一区别仅在于分支策略上, 3 个算法均包含了第 4.1–4.3 节的优化技术. 所有算法均由 C++实现, 并在一台配置为 2.2 GHz CPU 和 64 GB 内存的服务器中进行测试.

(2) 数据集: 本实验使用了 6 个真实的二部图数据以测试所提算法的性能, 数据主要包括了社交网络数据、合作网络数据以及用户标签网络数据等. 具体统计如表 1 所示, 其中 $d_{1\max}$ 与 $d_{2\max}$ 分别表示 L 与 R 中顶点的最大度. 所有的数据均可从 http://konect.cc/networks 下载.

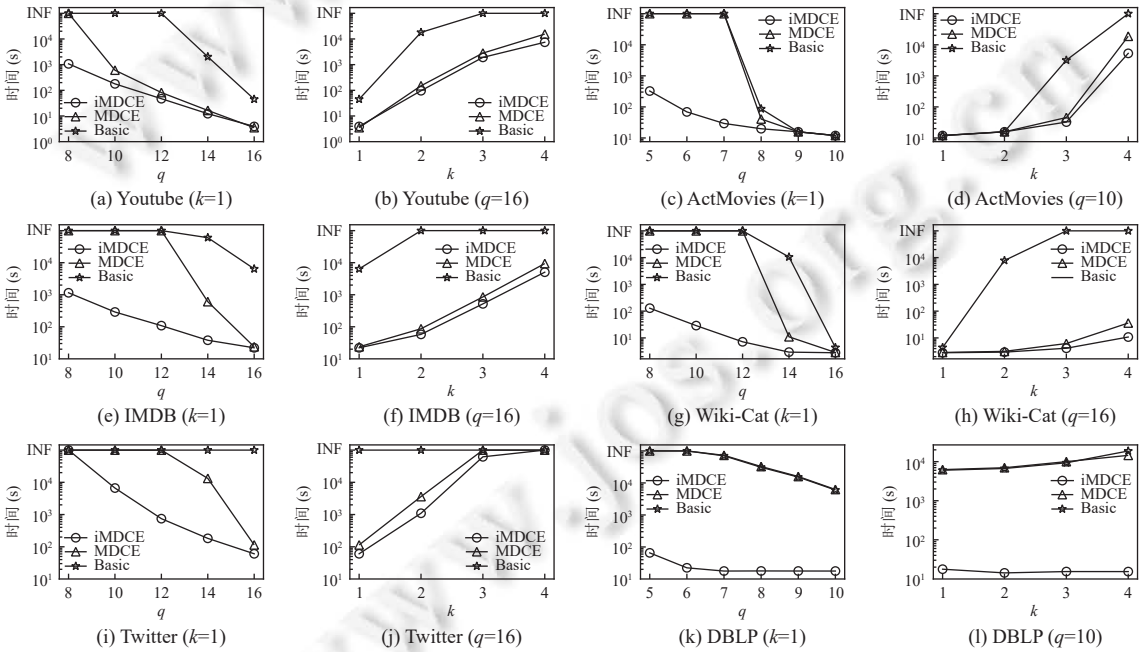
表 1 真实二部图数据

数据集	$\|L\|$	$\|R\|$	m	$d_{1\max}$	$d_{2\max}$
Youtube	94 238	30 087	293 360	1 035	7 591
ActMovies	127 823	383 640	1 470 404	294	646
IMDB	303 617	896 302	3 782 463	1 334	1 590
Wiki-Cat	1 853 493	182 947	3 795 796	54	11 593
Twitter	175 214	530 418	4 664 605	968	19 805
DBLP	1 953 085	5 624 219	12 282 059	1 386	287

(3) 参数设置: 所有的算法均包含了两个参数 k 和 q, 其 q 表示极大 k-缺陷二团的顶点数量的阈值, 即任何所输出的 k-缺陷二团 (A,B) 均满足 $|A|\geqslant q$ 且 $|B|\geqslant q$. 本文选择 k 为 1–4 的整数, 其默认值为 1; q 为 8–16 的整数. 由于 ActMovies 和 DBLP 中的极大 k-缺陷二团比较小, 本文针对该数据设置 q 的取值范围为 5–10 的整数.

5.2 实验结果

实验 1: 算法的性能分析. 本文首先测试了各个算法枚举大小约束的极大 k-缺陷二团时的性能. 图 5 展示了各个算法在分别变化参数 k 和 q 时处理不同真实二部图的运行时间. 值得注意的是, 本实验限定各个算法的运行时间为 24 h, 当超过约束时间时, 其运行时间将设置为 INF. 从图 5 的实验结果可以看出, 基准算法 Basic 在绝大部分参数情况下的运行时间均超过了 24 h 的时间限制, 而且除了少数参数所有算法均能高效计算外, 所提出的优化分支算法 iMDCE 的运行时间均远低于所提出的 MDCE 算法. 例如, 在 IMDB 二部图中, 当 $k=1$ 以及 $q=14$ 时, iMDCE 的运行时间为 37.99 s, 但是 MDCE 和 Basic 在相同条件下的运行时间为 606.83 s 和 60 279.2 s. 表明了所提出的分支技术相较于传统分枝定界方法在减少冗余计算方面具有极大的性能优势. 此外, 当 k 较大或者 q 较小时, 所提算法 iMDCE 相较于 MDCE 具有更高的加速比. 例如, 在二部图 IMDB 中, 当 $k=1$, $q=14$ 时, iMDCE 相较于 MDCE 的加速比为 15.9, 但是当 $q=12$ 时, iMDCE 相较于 MDCE 的加速比超过了 300 倍. 该实验进一步表明提出的优化分支技术在减少冗余分支方面的优越性.



(a) Youtube (k=1) (b) Youtube (q=16) (c) ActMovies (k=1) (d) ActMovies (q=10)

(e) IMDB (k=1) (f) IMDB (q=16) (g) Wiki-Cat (k=1) (h) Wiki-Cat (q=16)

(i) Twitter (k=1) (j) Twitter (q=16) (k) DBLP (k=1) (l) DBLP (q=10)

图 5 不同算法在真实二部图上的运行时间

实验 2: 优化技术的效果. 为了测试所提优化技术对所提算法性能的影响, 本实验还测试了所提算法在排除各个优化技术时的运行时间. 表 2 和表 3 展示了算法 iMDCE 在不包含不同优化技术条件下变化参数 k 和 q 时枚举

极大 k-缺陷二团的运行时间, 其中"-O"表示无排序优化, "-U"表示无上界优化, "-D"表示线性更新优化, "—"表示算法在 24 h 的限定时间内无法完成计算. 值得注意的是, ActMovies 与 DBLP 中的极大 k-缺陷二团较小, 本实验对这两数据集中的大小约束 q 设置为 6–10 的整数. 从表 2 或者表 3 中可以看出, 当不包含第 4 节中提出的任何一个优化技术, 所提算法 iMDCE 的性能均会有所下降. 首先, 上界优化对所提算法的影响最大, 当不包含此项优化时, 所提算法在所有参数情况下均无法在 24 h 内完成计算. 其次, 排序优化在参数 q 相对较小或者 k 较大时具有明显的性能优化, 例如, 在 ActMovies 中, 当 $k=3$, $q=10$ 时, iMDCE 相较于不包含排序优化的 iMDCE 的运行时间加速比超过 30 倍. 最后, 线性更新优化对枚举算法的影响最小, 但是在许多参数条件下仍然具有明显的加速. 该实验结果表明所提出的优化技术对提高所提算法具有重要作用.

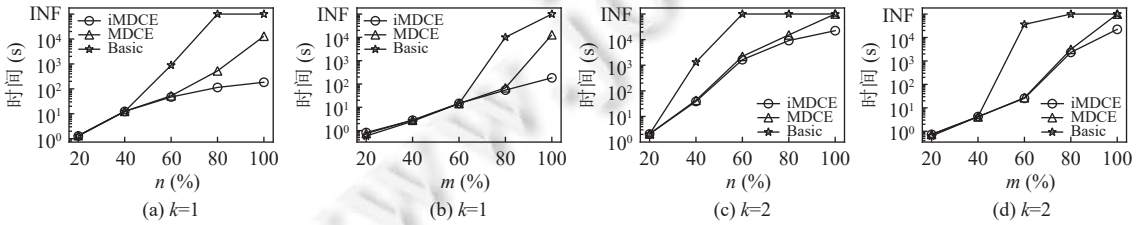
表 2 iMDCE 在 k=1 时使用不同优化技术的运行时间 (s)

数据集	q=8 (6)			q=10 (7)			q=12 (8)			q=14 (9)		
	-O	-U	-D	-O	-U	-D	-O	-U	-D	-O	-U	-D
Youtube	2 663.49	—	1 236.39	817.43	—	254.75	214.14	—	66.60	97.95	—	16.92
ActMovies	465.98	—	107.80	154.14	—	35.33	22.83	—	19.90	16.03	—	15.80
IMDB	5 897.60	—	1 352.31	980.88	—	310.12	291.47	—	109.82	57.08	—	39.54
Wiki-Cat	277.29	—	114.16	51.68	—	26.11	8.90	—	6.51	3.01	—	2.91
Twitter	—	—	—	—	—	13 501.4	10 903.3	—	1 311.30	1 300.95	—	241.04
DBLP	18.64	—	17.86	14.42	—	14.21	14.48	—	14.29	14.08	—	14.03

表 3 iMDCE 在 q=16 (ActMovies 和 DBLP 为 10) 时使用不同优化技术的运行时间 (s)

数据集	k=1			k=2			k=3			k=4		
	-O	-U	-D	-O	-U	-D	-O	-U	-D	-O	-U	-D
Youtube	28.95	—	4.91	1 391.11	—	167.98	—	—	3 566.09	—	—	—
ActMovies	12.03	—	11.95	17.53	—	15.80	1 068.90	—	41.30	—	—	16 815.2
IMDB	26.63	—	22.61	165.05	—	70.93	1 395.09	—	747.81	15 902.8	—	7 975.75
Wiki-Cat	2.78	—	2.76	3.16	—	2.94	5.68	—	4.82	32.27	—	19.70
Twitter	77.82	—	64.90	19 050.6	—	2 176.75	—	—	—	—	—	—
DBLP	14.36	—	13.80	14.08	—	13.84	14.21	—	13.90	15.58	—	14.85

实验 3: 可扩展性分析. 为研究所提算法的可扩展性, 本实验通过对二部图 Twitter 随机抽样 20%–80% 的顶点和边以生成 8 个子图, 然后测试各个算法在这些图上的时间性能.图 6 展示了其实验结果. 可以看出, 除了所有算法均能够快速计算的子图外, 提出的算法 iMDCE 和 MDCE 始终优于基准算法 Basic. 此外, 随着二部图数据的规模增大, iMDCE 的运行时间平稳增长, 然而其他算法 (包括 MDCE 和 Basic) 的运行时间急剧增加. 例如, 在 $k=1$, iMDCE、MDCE 以及 Basic 在顶点抽样为 60% 的子图中的运行时间分别为 47.03 s, 52.32 s 以及 885.12 s. 然而当顶点抽样增加到 80% 时, MDCE 和 Basic 的运行时间分别是 iMDCE 的 4.6 倍和超过 800 倍. 表明所提出的优化算法具有较高的可扩展性, 可用于真实应用中的较大的二部图数据.



(a) k=1 (b) k=1 (c) k=2 (d) k=2

图 6 可扩展性分析

实验 4: 算法内存消耗. 本实验还测试了各个算法在处理真实二部图数据时的内存消耗. 图 7 展示了各个算法在变化参数 k 和 q 时最大的内存消耗. 可以看出所有算法 (包括 iMDCE、MDCE 以及 Basic) 的内存消耗几乎相

同, 这主要是因为所提算法的内存消耗与分支数量无关, 仅与二部图数据的大小成线性关系. 此外所提算法的内存消耗与二部图的存储大小仅相差数倍, 这是因为算法使用了额外的索引以判断顶点之间的邻居关系. 该实验表明所提算法具有很高的空间效率.

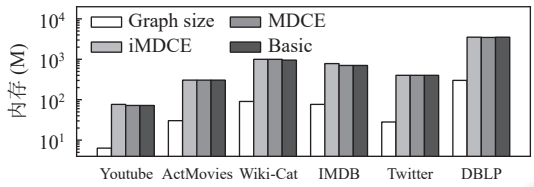


图 7 内存消耗

实验 5: 案例研究. 此外, 本文还进一步展开了一项案例研究以分析所提出的极大 k-缺陷二团在社区挖掘方面的效果. 本实验利用 DBLP 上的数据信息 (https://dblp.uni-trier.de/xml) 构建二部图数据, 其中顶点分别表示作者和文章, 作者与文章之间的关系则表示为边. 本实验基于不同二部图稠密子图社区模型检测了包含“研究者 2”的社区 (为保护个人隐私, 本文仅以不同编号代替具体作者姓名), 其中参数 k 和 q 分别设置为 2 和 5. 值得注意的是, 极大 k-biplex 模型在此参数条件下难以在限定时间内找出所需结果, (α, β)-核模型所检测的社区规模过于庞大, 顶点数超过 100 个. 为此本实验忽略了这两个模型的搜索结果. 图 8 展示了由二团模型和 k-缺陷二团模型的搜索结果. 可以看出, 相较于二团模型, 基于极大 k-缺陷二团模型可以从该数据中搜索到更多具有价值的信息. 具体地, 二团模型的结果不包含研究者“研究者 1”和“研究者 7”, 然而 k-缺陷二团模型能够成功找出这两名研究者. 通过查找 DBLP 原始数据可知, 这两名研究者确实与其他研究者也存在非常密切的合作关系, 表明所提出的极大 k-缺陷二团在真实二部图中检测社区具有一定的意义.

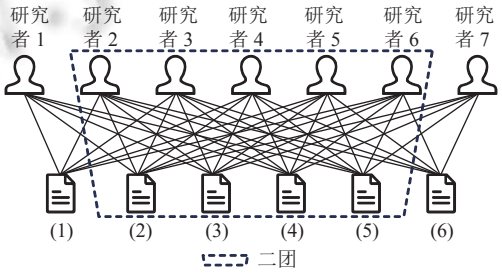


图 8 利用 k-缺陷二团搜索包含“研究者 2”的社区结果 ($k=2$, $q=5$)

6 总 结

本文研究了面向二部图数据的稠密子图挖掘问题. 由于传统二团模型的限制过于严格, 为此本文提出了一种称之为 k-缺陷二团的松弛二团模型. 为解决极大 k-缺陷二团枚举问题, 本文首先提出了一种基于对称集合枚举的搜索算法. 为进一步提高算法的性能, 本文还提出了一系列的优化方法, 主要包括基于排序的子图划分方法、基于上界的剪枝方法、线性时间的更新方法以及分支优化方法. 本文证明所提出的优化方法具有非平凡的理论时间复杂度, 主要与 $O(\gamma_k)$ 有关, 其中 $\gamma_k < 2$. 最后, 大量的实验表明所提出的极大 k-缺陷二团枚举算法在大部分参数情况下相较于传统枚举方法速度提高了 100 倍以上.

References:

[1] Lancichinetti A, Fortunato S. Community detection algorithms: A comparative analysis. Physical Review E, 2009, 80(5): 056117. [doi: 10.1103/PhysRevE.80.056117]

[2] Harley E, Bonner A, Goodman N. Uniform integration of genome mapping data using intersection graphs. Bioinformatics, 2001, 17(6): 487–494. [doi: 10.1093/bioinformatics/17.6.487]

[3] Boginski V, Butenko S, Pardalos PM. Mining market data: A network approach. Computers & Operations Research, 2006, 33(11): 3171–3184. [doi: 10.1016/j.cor.2005.01.027]

[4] Wang J, De Vries AP, Reinders MJT. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proc. of the 29th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Seattle: ACM, 2006. 501–508. [doi: 10.1145/1148170.1148257]

[5] Ley M. The DBLP computer science bibliography: Evolution, research issues, perspectives. In: Proc. of the 9th Int'l Symp. on String Processing and Information Retrieval. Lisbon: Springer, 2002. 1–10. [doi: 10.1007/3-540-45735-6_1]

[6] Beutel A, Xu WH, Guruswami V, Palow C, Faloutsos C. CopyCatch: Stopping group attacks by spotting lockstep behavior in social networks. In: Proc. of the 22nd Int'l Conf. on World Wide Web. Rio de Janeiro: ACM, 2013. 119–130. [doi: 10.1145/2488388.2488400]

[7] Li HQ, Li JY, Wong L. Discovering motif pairs at interaction sites from protein sequences on a proteome-wide scale. Bioinformatics, 2006, 22(8): 989–996. [doi: 10.1093/bioinformatics/btl020]

[8] Lehmann S, Schwartz M, Hansen LK. Biclique communities. Physical Review E, 2008, 78(1): 016108. [doi: 10.1103/PhysRevE.78.016108]

[9] Lyu BQ, Qin L, Lin XM, Zhang Y, Qian ZP, Zhou JR. Maximum biclique search at billion scale. Proc. of the VLDB Endowment, 2020, 13(9): 1359–1372. [doi: 10.14778/3397230.3397234]

[10] Zhao YW. Community search algorithms of dense subgraphs in bipartite graph [MS. Thesis]. Shanghai: East China Normal University, 2023 (in Chinese with English abstract). [doi: 10.27149/d.cnki.ghdsu.2023.002346]

[11] Zhao XW, Xue JF. Community discovery algorithm for attributed networks based on bipartite graph representation. Computer Science, 2023, 50(11): 107–113 (in Chinese with English abstract). [doi: 10.11896/jsjkx.221000226]

[12] Yu KQ, Long C. Maximum k-Biplex search on bipartite graphs: A symmetric-BK branching approach. Proc. of the ACM on Management of Data, 2023, 1(1): 49. [doi: 10.1145/3588729]

[13] Voggenreiter O, Bleuler S, Gruissem W. Exact biclustering algorithm for the analysis of large gene expression data sets. BMC Bioinformatics, 2012, 13(S18): A10. [doi: 10.1186/1471-2105-13-S18-A10]

[14] Liu GM, Sim K, Li JY. Efficient mining of large maximal bicliques. In: Proc. of the 8th Int'l Conf. on Data Warehousing and Knowledge Discovery. Krakow: Springer, 2006. 437–448. [doi: 10.1007/11823728_42]

[15] Li JY, Liu GM, Li HQ, Wong L. Maximal biclique subgraphs and closed pattern pairs of the adjacency matrix: A one-to-one correspondence and mining algorithms. IEEE Trans. on Knowledge and Data Engineering, 2007, 19(12): 1625–1637. [doi: 10.1109/TKDE.2007.190660]

[16] Zhang Y, Phillips CA, Rogers GL, Baker EJ, Chesler EJ, Langston MA. On finding bicliques in bipartite graphs: A novel algorithm and its application to the integration of diverse biological data types. BMC Bioinformatics, 2014, 15(1): 110. [doi: 10.1186/1471-2105-15-110]

[17] Das A, Tirthapura S. Shared-memory parallel maximal biclique enumeration. In: Proc. of the 26th IEEE Int'l Conf. on High Performance Computing, Data, and Analytics (HiPC). Hyderabad: IEEE, 2019. 34–43. [doi: 10.1109/HiPC.2019.00016]

[18] Abidi A, Zhou R, Chen L, Liu CF. Pivot-based maximal biclique enumeration. In: Proc. of the 29th Int'l Conf. on Int'l Joint Conf. on Artificial Intelligence. Yokohama: ACM, 2020. 3558–3564.

[19] Chen L, Liu CF, Zhou R, Xu JJ, Li JX. Efficient maximal biclique enumeration for large sparse bipartite graphs. Proc. of the VLDB Endowment, 2022, 15(8): 1559–1571. [doi: 10.14778/3529337.3529341]

[20] Dai QQ, Li RH, Ye XW, Liao MH, Zhang WP, Wang GR. Hereditary cohesive subgraphs enumeration on bipartite graphs: The power of pivot-based approaches. Proc. of the ACM on Management of Data, 2023, 1(2): 138. [doi: 10.1145/3589283]

[21] Cerinšek M, Batagelj V. Generalized two-mode cores. Social Networks, 2015, 42: 80–87. [doi: 10.1016/j.socnet.2015.04.001]

[22] Zou ZN. Bitruss decomposition of bipartite graphs. In: Proc. of the 21st Int'l Conf. on Database Systems for Advanced Applications. Dallas: Springer, 2016. 218–233. [doi: 10.1007/978-3-319-32049-6_14]

[23] Sim K, Li JY, Gopalkrishnan V, Liu GM. Mining maximal quasi-bicliques: Novel algorithm and applications in the stock market and protein networks. Statistical Analysis and Data Mining: The ASA Data Science Journal, 2009, 2(4): 255–273. [doi: 10.1002/sam.10051]

[24] Yu HY, Paccanaro A, Trifonov V, Gerstein M. Predicting interactions in protein networks by completing defective cliques. Bioinformatics, 2006, 22(7): 823–829. [doi: 10.1093/bioinformatics/btl014]

[25] Yannakakis M. Node-deletion problems on bipartite graphs. SIAM Journal on Computing, 1981, 10(2): 310–327. [doi: 10.1137/0210022]

[26] Liu BG, Yuan L, Lin XM, Qin L, Zhang WJ, Zhou JR. Efficient (α, β)-core computation in bipartite graphs. The VLDB Journal, 2020, 29(5): 1075–1099. [doi: 10.1007/s00778-020-00606-9]

[27] Zhang YH, Hua ZY, Yuan L, Zhang F, Wang K, Chen Z. Distance-generalized (α, β)-core decomposition on bipartite graphs. Computer

Science, 2024, 51(11): 95–102 (in Chinese with English abstract) [doi: 10.11896/jsjkx.231000130]
[28] Wang K, Lin XM, Qin L, Zhang WJ, Zhang Y. Efficient bitruss decomposition for large-scale bipartite graphs. In: Proc. of the 36th IEEE Int'l Conf. on Data Engineering (ICDE). Dallas: IEEE, 2020. 661–672. [doi: 10.1109/ICDE48307.2020.00063]
[29] Yu KQ, Long C, Liu SX, Yan D. Efficient algorithms for maximal *k*-Biplex enumeration. In: Proc. of the 2022 Int'l Conf. on Management of Data. Philadelphia: ACM, 2022. 860–873. [doi: 10.1145/3514221.3517847]
[30] Ignatov DI, Ivanova P, Zamaletdinova A, Prokopyev O. Preliminary results on mixed integer programming for searching maximum quasi-bicliques and large dense biclusters. In: Proc. of the 2019 ICFCA Conf. and Workshops (Supplements). 2019. 28–32.
[31] Batagelj V, Zaversnik M. An O(*m*) algorithm for cores decomposition of networks. arXiv:cs/0310049, 2003.
[32] Li RH, Song QS, Xiao XK, Qin L, Wang GR, Yu JX, Mao R. I/O-efficient algorithms for degeneracy computation on massive networks. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(7): 3335–3348. [doi: 10.1109/TKDE.2020.3021484]
[33] Fomin FV, Kaski P. Exact exponential algorithms. Communications of the ACM, 2013, 56(3): 80–88. [doi: 10.1145/2428556.2428575]

附中文参考文献:

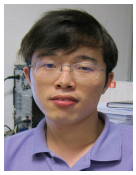
[10] 赵奕威. 基于二分图中稠密子图的社区搜索算法研究 [硕士学位论文]. 上海: 华东师范大学, 2023. [doi: 10.27149/d.cnki.ghdsu.2023.002346]
[11] 赵兴旺, 薛晋芳. 基于二部图表示的属性网络社区发现算法. 计算机科学, 2023, 50(11): 107–113. [doi: 10.11896/jsjkx.221000226]
[27] 张毅豪, 华征宇, 袁龙, 张帆, 王凯, 陈紫. 基于距离泛化的二分图 (*α*, *β*)-core 高效分解算法. 计算机科学, 2024, 51(11): 95–102. [doi: 10.11896/jsjkx.231000130]

代强强(1992－), 男, 博士, CCF 学生会员, 主要研究领域为图数据挖掘与管理, 社交网络数据分析, 并行算法设计.

李振军(1979－), 男, 博士, 正高级工程师, 主要研究领域为数据分析和挖掘, 隐私计算, 区块链.

于瀚文(2001－), 男, 硕士生, 主要研究领域为图数据挖掘, 图计算.

王国仁(1966－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为不确定数据管理, 数据密集型计算, 可视媒体数据管理与分析, 非结构化数据管理, 分布式查询处理与优化技术, 生物信息学.



李荣华(1985－), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为图数据管理与挖掘, 图计算系统, 图神经网络, 图表示学习, 知识图谱, 图论算法的设计与分析, 谱图理论.