

基于贪婪策略的紧密 k 核子图查询

赵丹枫¹, 姚贤标¹, 包晓光¹, 黄冬梅², 郭伟其³

(1. 上海海洋大学 信息学院, 上海 201306; 2. 上海电力大学 电子与信息工程学院, 上海 200090;

3. 国家海洋局 东海海洋环境调查勘察中心, 上海 200137)

摘要: k 核查询是一种社团查询, 由于其可以在线性时间内被有效计算, 因此在社团检测中具有较广泛的应用。图中边的权值在很多场景下具有较强的语义关系, 但现有研究较少考虑图中边的权值。为提升 k 核查询的效率, 在 k 核的基础上定义加权图中的紧密 k 核子图查询(CRKSQ)问题, 并使用归约方法证明该问题是NP-难的。基于贪婪策略设计启发式算法CRK-G, 通过迭代删除节点为CRKSQ问题找到一个近似解。在此基础上, 从降低图规模和减少迭代次数两方面研究CRK-G算法的优化策略, 分别提出使用图压缩策略的算法CRK-C及使用单次多节点删除策略的算法CRK-F。在Bio-GRID、Email-Enron、DBLP 3个数据集上的实验结果表明, 相对于CRK-G算法, CRK-C、CRK-F算法在查询速度上有较大的提升, 且平均误差均在8%以内。

关键词: 社团检测; k 核; 加权图; 紧密子图; 贪婪策略

开放科学(资源服务)标志码(OSID):



中文引用格式: 赵丹枫, 姚贤标, 包晓光, 等. 基于贪婪策略的紧密 k 核子图查询[J]. 计算机工程, 2022, 48(10): 55-66.

英文引用格式: ZHAO D F, YAO X B, BAO X G, et al. Closely related k -core subgraph query based on greedy strategy[J]. Computer Engineering, 2022, 48(10): 55-66.

Closely Related k -Core Subgraph Query Based on Greedy Strategy

ZHAO Danfeng¹, YAO Xianbiao¹, BAO Xiaoguang¹, HUANG Dongmei², GUO Weiqi³

(1. College of Information Technology, Shanghai Ocean University, Shanghai 201306, China;

2. College of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China;

3. East China Sea Marine Environment Survey and Survey Center, State Oceanic Administration, Shanghai 200137, China)

[Abstract] k -core query is a type of community query because it can be calculated effectively in linear time and has a wide range of applications in community detection. In some scenarios, the weights of edges in graphs exhibit strong semantics, but the weights of edges in graphs were rarely considered in previous studies. First, the Closely Related k -core Subgraph Query (CRKSQ) problem in weighted graphs is defined based on the k -core, and the problem is confirmed to be Non-deterministic Polynomial-time hard (NP-hard) using the reduction method. Second, a heuristic algorithm CRK-G is designed based on the greedy strategy to obtain an approximate solution to the CRKSQ problem by iteratively deleting nodes. Finally, the optimization strategy of the CRK-G algorithm is evaluated based on two aspects: reducing the graph size and the number of iterations. This study proposes a CRK-C algorithm using the graph compression strategy and a CRK-F algorithm using the single-time multinode deletion strategy. The experimental results for three datasets (Bio-GRID, Email-Enron, and DBLP) show that compared with the CRK-G algorithm, the CRK-C and CRK-F algorithms exhibited significantly improved query speed, and the average error is within 8%.

[Key words] community detection; k -core; weighted graph; closely related subgraph; greedy strategy

DOI: 10.19678/j.issn.1000-3428.0062639

0 概述

图查询是图数据分析与处理的基础, 其中社团检测作为一种子图查询, 其目标是从给定的图中寻找所

有紧密连接的子图, 并且各个图内部的节点是连通的^[1]。紧密子图查询作为社团检测的核心, 在很多领域都有着重要的应用, 例如: 在社交网络中, 查询联系紧密的社团有助于用户的特征分析^[2]; 在作者的合作网络中,

基金项目: 国家自然科学基金青年科学基金项目(42106190); 上海市科委地方能力建设项目(20050501900)。

作者简介: 赵丹枫(1982—), 女, 讲师、博士, 主研方向为图计算、大数据技术; 姚贤标, 硕士研究生; 包晓光, 讲师、博士; 黄冬梅(通信作者), 郭伟其, 教授。

收稿日期: 2021-09-08

修回日期: 2021-11-01

E-mail: dfzhao@shou.edu.cn

联系紧密的作者之间可能具有相似的研究领域^[3];在商品销售数据中,联系紧密的商品更有可能被相似的消费者所关注^[4];在蛋白质网络中,关系紧密的蛋白质通常更有可能形成特定的官能团^[5]。

在现有的研究中,研究人员提出许多社团检测方法,如基于模块度优化的算法^[6]、谱方法^[7]、非负矩阵分解^[8]等。对于社团的定义也有多种,常见的有 k 架(k -truss)、 k 团(k -clique)、 k 核(k -core)等,其中 k 核的定义^[9]要求图中每个节点至少有 k 个邻接节点,即每个节点的度至少为 k 。 k 核可以在线性时间复杂度内计算得到,具有较好的应用基础^[10-12]。近年来的研究多结合部分场景的应用需求,讨论各类属性图上的 k 核子图查询^[13-15]。然而,由于考虑边权值的研究较少,但在很多场景下,图中边的权值往往具有很强的语义关系。例如在社交网络中,两个用户之间的交流越频繁意味两者之间的联系越紧密。在作者的合作网络中,若两个作者之间合作的次数越多,则说明他们的联系越紧密,对应图中边的权值越大。因此,为了使社团检测更加符合实际的语义场景,提高查询结果的合理性,则需要将图中边的权值考虑在内。

此外,考虑到查询最紧密社团往往会耗费较多的时间,并且在一些应用场景下不需要查询社团权重的最值,例如在蛋白质网络中,有时仅需要找到满足一定紧密关系的蛋白质集合即可。为此,不局限于社团检测中最值的查询,给定核值 k 、节点平均权值 w_0 ,在加权图中查询联系紧密的连通 k 核子图问题,记为紧密 k 核子图查询(Closely Related k -Core Subgraph Query, CRKSQ)问题。该问题要求查询得到的连通 k 核子图的节点平均权值大于等于 w_0 ,其进一步限制了子图的紧密程度,且由于 w_0 可以根据实际需求给出,具有更好的灵活性,更加切合实际应用场景。

为解决CRKSQ问题,本文首先证明该问题是一个NP-难问题,即无法为其找到一个在多项式时间复杂度内的最优算法,并基于贪婪策略设计启发式算法CRK-G。在此基础上,研究CRK-G算法的优化策略,提出CRK-C和CRK-F算法,分别从降低图规模和减少迭代次数两个方面提高查询效率。最后通过实验对比3种算法在不同场景下的效率。

1 相关工作

目前针对社团的查询已经有了广泛的研究,除 k 核外,主要还包括基于 k -truss和 k -clique的查询问题。 k -truss和 k -clique相对于 k 核有着更加严格的定义,但都与 k 核类似,其本身并未考虑图中的其他属性,仅仅只是从图中寻找稠密且内聚的结构。近年来的很多研究也开始结合图的语义关系,探索更加高质量的 k -truss和 k -clique社团^[16-18]。

与上述两种社团的定义相比, k 核可以被有效地在线性时间内计算,其概念最初由SEIDMAN^[9]在1983年提出。为了高效地进行 k 核的查询,

BATAGELJ等^[19]于2003年提出了 k 核分解算法,其能在 $O(m)$ 时间复杂度下得到每个节点的最大核值,进而能够在线性时间内得到 k 核查询的结果。文献^[20]为应对大图的查询,又提出了效率更高且能在个人计算机上运行的3种 k 核分解算法,其查询速度有了很大的提升。为提升 k 核查询的效率,文献^[21]提出了面向 k 核查询的图压缩算法SC,进一步将压缩图转化为树的算法TC,其算法效率比可以比直接在原图上查询的效率高出1~2个数量级。

为了优化 k 核社团查询,文献^[22]提出一种能够支持多个节点查询的社团搜索问题,解决了以往单节点查询带来的通用性的不足。文献^[23]考虑到现有的社团查询方法使用的都是全局搜索,算法代价较高,故提出一种局部搜索方法。上述两种方法虽然在查询条件上做了优化,但却都未结合图上的其他属性,仅面向简单图的查询。

近年来,研究人员开始致力于寻找更加紧密的 k 核子图,文献^[13]结合图中节点的影响力,提出了 k -influential社团模型,用于在大图中查询最具影响的社团。文献^[14]考虑边的有向性,研究了有向图上的社团搜索问题,其利用最小内度和外度的度量方法寻找紧密连接的子图。文献^[15]研究了轮廓图的社团搜索(PCS)问题,其能够识别具有语义共性的节点,从而发现更多高质量的社团。上述研究结合了图上的其他属性,在相应的场景下能得到更加合理有效的社团,而对于边权值属性的社团查询方面,目前的研究还不充分。虽然文献^[24]提出了查询加权图的亲密核心组问题,但由于其模型属于社团搜索,仅针对给定节点集的查询,并且是查询总权值最小的子图,与部分应用场景不符,具有一定的局限性。

本文考虑了节点的语义关系,不局限于社团检测中最值的查询,提出了在加权图中查询联系紧密的连通 k 核子图问题,以实现更好的灵活性,且更贴合实际应用场景。

2 定义与证明

2.1 紧密 k 核子图的定义

本文研究的是无向加权图上的子图查询,该研究是在 k 核的基础上进行的,故先给出 k 核的定义。

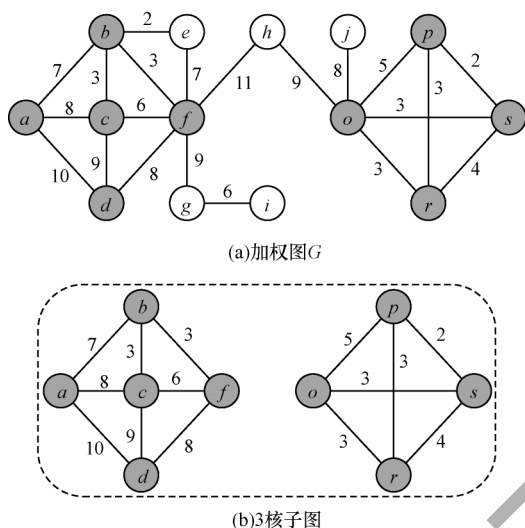
定义1(k 核)^[9] 给定图 $G=(V,E)$, k 核是图 G 的极大子图 H ,使得 H 中的每个节点至少有 k 个邻节点。

定义2(核值) 给定图 $G=(V,E)$,对于 G 中的节点 $u, u \in V$, u 的核值 $\text{core}(u)$ 为 u 所在全部 k 核中的最大值,即:

$$\text{core}(u) = \max_{V(H) \subseteq V} \{k | u \in V(H) \text{ 且 } H \text{ 为 } k \text{ 核} \} \quad (1)$$

例如,在图1中,节点集 $\{a, b, c, d, f\}$ 和 $\{o, p, r, s\}$ 的导出子图中节点核值都为3。但需要注意的是,节点的核值并不等于节点的度,如图1中的节点 c ,

其节点的度为4,但核值却为3。此外, k 核并不一定是一个连通的图,如图1(b)表示的是图1(a)中加权图 G 的3核子图,该图并不是连通的。

图1 k 核查询实例Fig.1 Example of k -core query

定义3(连通 k 核) 给定图 $G=(V,E)$,连通 k 核是图 G 的极大连通子图 H ,使得 H 中的每个节点至少有 k 个邻节点。

连通 k 核是在 k 核的基础上定义的,其表示的子图一定是一个连通的图。本文研究的是连通 k 核,若不加特别说明,下文中提到的 k 核均表示的是连通 k 核。

定义4(节点权值 ω) 给定一个加权图 $G=(V,E,W)$,对于 G 中的节点 $u, u \in V$,节点 u 的权值为与其相连边的权值之和,即节点权值:

$$\omega(u) = \sum_{e=(u,v), v \in V} w(e) \quad (2)$$

例如,在图1中,与节点 a 相连的边有3条,各边权值分别为7、8、10,故其节点的权值 $\omega(a)=7+8+10=25$ 。

定义5(节点平均权值 Aw) 给定一个加权图 $G=(V,E,W)$,图 G 节点平均权值为每个节点的权值之和除以图 G 的节点数,即图 G 节点平均权值为:

$$Aw(G) = \frac{\sum_{u \in V} \omega(u)}{|V|} \quad (3)$$

节点平均权值反映的是整个子图内部的紧密程度,节点平均权值越大则说明子图内部节点之间的交互越多,子图就越紧密。

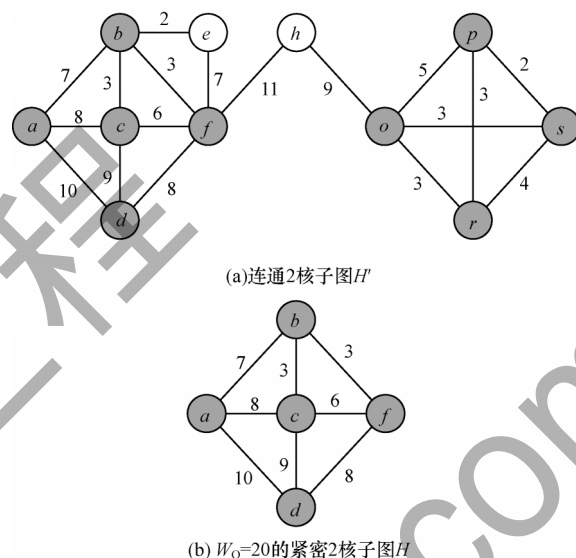
下面给出紧密 k 核子图(Closely Related k -Core Subgraph)的定义。

定义6(紧密 k 核子图) 给定一个加权图 $G=(V,E,W)$,核值 k ,节点平均权值 w_0 ,紧密 k 核子图 $H=(V',E',W)$, H 满足以下条件:1) $H \subseteq G$;2) H 是连通 k 核;3)图 H 的节点平均权值 $Aw(H) \geq w_0$ 。

紧密 k 核子图的定义在 k 核的基础上进一步限制了子图的节点平均权值,该限制将会消除图中权

值较小的节点,使得图整体的紧密性有所提高。

紧密连接 k 核子图查询如图2所示,其中图2(b)为加权图 G (见图1(a))在给定 $k=2, w_0=20$ 时得到的紧密2核子图 H 。图 H 的每个节点都至少有2个邻节点,且图的节点平均权值为21.6,满足给定条件。对比图2(a)和图2(b),可以看出后者节点之间的联系更加紧密,整体也更加紧凑。

图2 紧密连通 k 核子图的查询实例Fig.2 Example of closely related connected k -core subgraph query

定义7(紧密 k 核子图查询) 给定一个加权图 $G=(V,E,W)$,核值 k ,节点平均权值 w_0 ,在图 G 中寻找紧密 k 核子图 H 。

紧密 k 核子图查询(CRKSQ)问题的目的是在图中找到满足条件的紧密 k 核子图。需要注意的是,紧密 k 核子图在图中不一定只有一个,有可能存在多个。

2.2 QCRKS 问题的证明

引理1 给定一个加权图 $G=(V,E,W)$,图 G 中节点权值之和等于图 G 中各边权值之和的两倍,即:

$$\sum_{u \in V} \omega(u) = 2 \times \sum_{e \in E} w(e) \quad (4)$$

证明 由于图 G 中的每条边连接两个节点,而根据节点权值的定义可知图中每条边的权值会被相连的两个节点各计算一次,故上述结论成立。

定理1 CRKSQ问题是NP-难的。

证明 将 clique 问题多项式归约到 CRKSQ 问题。由于 clique 问题是一个 NP-完全问题^[25],进而得出 CRKSQ 问题是 NP-难的。给定 clique 问题的一个实例 I ,其由一个无权图 $G=(V,E)$ 和一个正整数 k 构成。构造 CRKSQ 问题的一个实例 I' :一个加权图 $G'=(V,E,W)$,这里每条边权值都为1;另一个正数 $w_0=k-1$ 。显然,这是一个多项式归约。下面只需证明实例 I 存在一个 k 阶完全子图,当且仅当,实例 I' 存在一个节点平均权值大于等于 w_0 ($w_0=k-1$)的紧密 $(k-1)$ 核子图。

假设 clique 问题存在一个 k 阶完全子图 H 。现证明 H 作为 G' 的子图,其是一个满足条件的紧密 $(k-1)$ 核子图。由完全子图的定义可知, H 中的每个节点具有 $(k-1)$ 个邻接节点,且平均权值为:

$$Aw(H) = \frac{\sum_{u \in V(H)} \omega(u)}{|V(H)|} \quad (5)$$

根据引理 1,式(5)可化为:

$$Aw(H) = \frac{2 \times \sum_{e \in E} \omega(e)}{|V(H)|} \quad (6)$$

因 G' 中每条的权值都为 1,故:

$$Aw(H) = \frac{2 \times |E(H)|}{|V(H)|} = \frac{k(k-1)/2 \times 2}{k} = k-1 \quad (7)$$

得证。

反之,假设 G' 中存在一个满足条件的紧密 $(k-1)$ 核子图 H 。由子图 H 是一个 $(k-1)$ 核知, H 具有至少 k 个节点;由节点平均权值大于等于 w_0 ($w_0 = k-1$),并由引理 1 可知, H 具有 $k(k-1)$ 条边,因此作为 G' 的子图 H 是一个 k 阶完全子图。得证。

3 基于贪婪策略的查询算法

由定理 1 可知,无法为 CRKSQ 问题找到一个多项式时间复杂度的最优算法,为解决该问题并在可接受的时间内找到合适的解,本文基于贪婪策略提出了启发式算法 CRK-G。首先从图中找到候选 k 核子图,然后通过不断地迭代删除候选子图中权值最小的节点,进而得到满足条件的紧密 k 核子图。

3.1 CRK-G 算法

CRK-G 算法步骤如下:

步骤 1 使用 k 核分解算法^[20] 计算每个节点的核值。

步骤 2 使用广度优先搜索,得到候选 k 核子图集合。

步骤 3 判断子图节点平均权值是否小于 w_0 ,若小于则迭代删除图中权值最小的节点。

步骤 4 为了保证剩余子图的 k 核连通性,还需要删除图中节点度小于 k 的节点,随后判断子图是否连通,若不连通则将每个独立的连通子图分隔,并加入候选子图集合等待后续的处理。

步骤 5 重复上述操作,直到剩余子图的节点平均权值大于等于 w_0 ,或节点被全部迭代删除为止。算法最后返回满足条件的紧密 k 核子图集。

例如,在图 2 中,给定图 G (见图 1(a)),核值 $k=2$,节点平均权值 $w_0=20$ 。算法首先查询得到候选 2 核子图 H' ,随后开始判断子图 H' 的节点平均权值是否大于等于 w_0 ,经计算可知节点平均权值小于 20。因此,开始删除最小权值的节点 e 。删除节点 e 后,节点平均权值仍然小于 20,则再依次删除节点 s 、节点 r 。其中,删除节点 s 和节点 r 导致节点 p 的度小于 2,则再将节点 p 删除,此时剩余子图仍然是连通的。重复上述步骤再依次删除节点 o 、节点 h ,最终得到图 2(b)中满足条件的紧密 2 核子图 H 。

算法 1 紧密 k 核子图查询算法 CRK-G

输入 图 $G=(V, E, W)$,核值 k ,节点平均权值 w_0

输出 满足条件的紧密子图集 $S_H = \{H | H \subseteq G \text{ 且 } H \text{ 为 } G \text{ 的紧密 } k \text{ 核子图}\}$

```

1.k-Core(); /*k 核分解算法计算每个节点的核值*/
2. $S_H = \emptyset$ ;
3.FOR 每个节点  $u \in V$  DO
4.IF 节点  $u$  还未被访问过 THEN
5. $H = \text{BFS\_Core}(u, k)$ ; /*使用广度优先搜索,找到从节点  $u$  开始的连通  $k$  核子图,并将其中的节点标记为已访问*/
6. $S_H = S_H \cup H$ ;
7.END IF
8.END FOR
9.FOR 每个子图  $H \subseteq S_H$  DO
10.IF 子图  $H$  的节点平均权值  $\geq w_0$  THEN
11.CONTINUE;
12.ELSE THEN
13.WHILE 子图  $H$  的节点平均权值  $\leq w_0$  DO
14.删除子图  $H$  中节点权值最小的节点;
15.删除剩余子图中度小于  $k$  的节点;
16.IF 子图  $H$  的节点数等于 0 DO
17.从  $S_H$  中移除  $H$ ;
18.BREAK;
19.END
20.IF Disconnected( $H$ ) DO /*检查  $H$  是否连通*/
21. $D_H = \text{Divide}(H)$ ; /*将  $H$  分割为独立的连通图,得到集合  $D_H$ */
22. $S_H = S_H \cup D_H$ ;
23. $H = D_H \text{ get}(0)$ ; /*将  $D_H$  中的第一个元素赋给  $H$ */
24.END IF
25.END WHILE
26.END IF
27.END FOR
28.RETURN  $S_H$ ;

```

3.2 CRK-G 算法的复杂度分析

CRK-G 算法采用贪婪思想,通过迭代删除节点而使得子图满足条件,其主要时间花费包括 k 核候选子图查询、节点平均权值计算、子图连通性判断以及子图 k 核的维护。首先,在 k 核候选子图的查询方面,使用的是广度优先搜索算法,其时间复杂度为 $O(|V| + |E|)$ 。其次,在节点平均权值计算和子图连通性判断方面,计算节点平均权值需要耗时 $O(|E|)$,寻找最小权值的节点需要耗时 $O(|V|)$,若算法需要迭代 t 次,则计算节点平均权值耗时 $O(t(|V| + |E|))$ 。连通性判断使用的是广度优先搜索,耗时也为 $O(t(|V| + |E|))$ 。最后,在子图 k 核的维护方面,查询节点的度耗时为 $O(|E|)$,若算法需要迭代 t 次,则这一步总耗时为 $O(t(|E|))$ 。最终 CRK-G 算法的时间复杂度为 $O(t(n + m))$,其中 n 和 m 分别表示原图的节点数和边数。

下文分析 CRK-G 算法的空间复杂度,对于原图中的每个节点,需要存储其邻节点以及其与邻节点之间边的权值,所以空间消耗为 $O(2|E|)$ 。另外,还需要存储候选子图中节点的邻节点以及节点与邻节点之间边的权值,空间消耗为 $O(2|E|)$ 。总共空间消耗为 $O(4|E|)$,故 CRK-G 算法的空间复杂度为 $O(m)$ 。

4 CRK-G 算法的优化

CRK-G 算法对于 CRKSQ 问题的求解是有效的,它能在 $O(t(n+m))$ 时间内找到一个解,但对于在大规模图上的查询而言,该算法求解需要花费大量的时间,效率较低。

根据对 CRK-G 算法时间复杂度的分析,影响其效率的主要因素是图的规模和迭代的次数,因此可以通过使用降低图规模或减少迭代次数的方法,使得算法的效率提高。基于该思路,本节提出了 CRK-G 算法的两种改进算法:1)使用图压缩策略的 CRK-C 算法,其适用于节点权值差异性较小的图数据;2)使用单次多节点删除策略的 CRK-F 算法,该算法实现简单,效率提高明显,适用于差异性较大或无法确定差异性的图数据。

4.1 基于图压缩的优化策略

目前关于面向特定查询的图压缩已经有了广泛的研究,常见的如面向保持可达查询的图压缩^[26]、面向邻节点查询的压缩^[27]以及面向 k 核查询的图压缩^[21]。与大部分面向特定查询的压缩技术类似,本文的压缩方法遵循查询等价原则,通过将权值相近的节点合并得到超节点,判断超节点所包含的节点在原图中是否存在边来构建超边以及超边的权值。

定义 8 (超节点和超边) 给定图 $G=(V,E)$,其经过压缩后得到 $G_c=(V',E')$,其中, $V'=\{v'_1, v'_2, \dots, v'_n\}$ 由原图节点集 V 分割得到,即 $v'_i \subset V (i=1, 2, \dots, n)$, 并且 $v'_i \cap v'_j = \emptyset (i \neq j)$, 则节点 $v'_i \in V'$ 被称为超节点,边 $e' \in E'$ 被称为超边。

例如,图 3(b)的图 H'_c 表示的是一个压缩图,其每个节点表示的即为超节点,如超节点 s_1 包含了原图 H' 中的节点 a 、节点 c 和节点 d 。压缩图 H'_c 中的边即为超边。

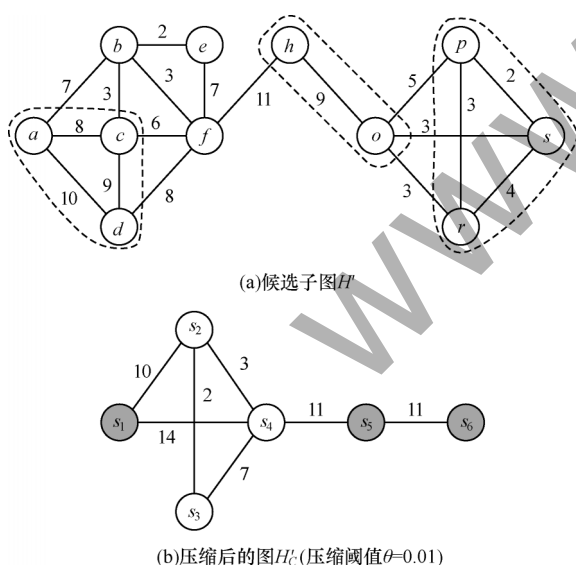


图3 图压缩实例

Fig.3 Example of graph compression

定义 9 (压缩比 cr) 给定图 $G=(V,E)$,其经过压缩得到 $G_c=(V',E')$,则压缩比为:

$$cr = \frac{|E'|}{|E|} \quad (8)$$

压缩比是衡量图压缩是否有效的指标,根据压缩比的定义可知,压缩比越小,则压缩后的图规模也越小。

4.1.1 CRK-C 算法

CRK-C 算法在 CRK-G 算法的基础上进行,查询到候选 k 核子图集后,对每个候选子图进行压缩,得到压缩后的图 G_c ,在后续的迭代删除过程中直接对 G_c 进行操作。压缩算法步骤如下:

步骤 1 遍历图中的所有节点,将权值相近的节点合并,从而得到超节点。

步骤 2 超节点包含的节点若在原图中存在边,则在超节点之间构建超边。

步骤 3 超边的权值设为两个超节点所包含的原节点在原图中边权值之和。

其中,步骤 1 的节点合并是通过设置压缩阈值 θ ($\theta \in (0, 1)$),根据起始节点的权值 ω ,将节点权值在 $\pm\theta\omega$ 之间的连通节点进行合并。

例如,图 3(a)表示的是一个 2 核候选子图,节点 a 、节点 c 和节点 d 的权值分别为 $\omega(a)=25$ 、 $\omega(c)=26$ 、 $\omega(d)=27$,节点之间的最大差值为 2,当压缩阈值 $\theta=0.01$ 时,最大允许的权值差值为 $2.5(\theta\omega(a)=2.5)$,所以节点 a 、节点 c 和节点 d 可以被压缩为一个超节点 s_1 。由于图 H' 中节点 a 、节点 c 与节点 b 存在边,而节点 b 压缩后由超节点 s_2 表示,因此超节点 s_1 和超节点 s_2 之间存在边,边的权值为原图中边 (a,b) 和边 (c,b) 的权值之和,即 $7+3=10$ 。重复上述压缩步骤,最终该候选子图 H' 经过压缩得到了图 H'_c ,压缩后图的规模与原图 H' 对比明显有所降低,压缩比 $cr=7/18 \approx 0.39$ 。

压缩算法 GC-W 见算法 2,算法最后返回压缩后的图 G_c 和映射表 M , M 中记录了原图 G 中每个节点所对应的超节点。借助 M ,可以在 $O(n)$ 时间内将压缩后的图解压, n 表示的是原图的节点数。

需要注意的是,压缩阈值 θ 越大,算法查询速度越快,但对应着查询结果的误差也会增大。根据算法特点,压缩阈值 θ 的取值误差主要取决于候选子图节点权值的波动或偏移程度,故可以使用子图中节点权值的差异系数 (Coefficient of Variation, CV) 作为 θ 的取值依据,其反映了子图节点权值的离散程度,计算公式为:

$$C_{cv} = \frac{\sigma_w}{Aw} \quad (9)$$

其中: σ_w 为子图权值的标准差; Aw 为子图平均节点权值。

算法2 压缩算法 GC-W

输入 需要压缩的子图 $G = (V, E, W)$, 压缩阈值 θ

输出 压缩后的图 $G_c = (V_c, E_c, W_c)$, 映射表 M

```

1.FOR 每个节点  $u \in V$  DO
2.IF 节点  $u$  还未被访问过 THEN
3. $B = \text{BFS\_weight}(u, \theta)$ ; /*找到与节点  $u$  在阈值范围内的
  连通节点集合*/
4.FOR 每个节点  $v \in B$  DO
5. $M[v] = [u]_c$ ;
6.将节点  $v$  设为已访问;
7.END FOR
8. $V_c = V_c \cup [u]_c$ ;
9. $M[u] = [u]_c$ ;
10.将节点  $u$  设为已访问;
11.END IF
12.END FOR
13.FOR 每个节点对  $[u]_c \in V_c, [v]_c \in V_c$  DO
14.IF  $(u, v) \in E$  THEN
15. $E_c = E_c \cup ([u]_c, [v]_c)$ ;
16. $W_c([u]_c, [v]_c) = W_c([u]_c, [v]_c) + W(u, v)$ ;
17.END IF
18.END FOR
19.RETURN  $G_c = (V_c, E_c, W_c), M$ ;
```

4.1.2 CRK-C算法的复杂度分析

CRK-C算法在原有贪婪策略的基础上新增了图压缩的步骤,其主要的时间花费包括 k 核候选子图查询、图压缩、节点平均权值计算、子图连通性判断、子图 k 核维护以及最后压缩图的解压缩。

1)在 k 核候选子图查询方面,其时间复杂度与 CRK-G算法一致,都为 $O(|V| + |E|)$ 。

2)在图压缩方面,其主要由超节点划分和创建超边两部分组成。其中超节点划分是通过广度优先搜索算法,不断地扩展与节点 u 权值相近的节点,直至无法扩展为止,该步可以在 $O(|V| + |E|)$ 时间内完成;而超边的创建是通过依次访问原图 G 中的每一条边,从而确定压缩图的边集,则其时间复杂度为 $O(|E|)$ 。因此,整个图压缩算法 GC-W 的耗时为 $O(|V| + 2|E|)$ 。

3)在节点平均权值计算和子图连通性判断方面,假设图压缩的压缩比为 cr ,则压缩后节点和边的数量大致可以表示为 $(cr)|V|$ 和 $(cr)|E|$ 。此外,由于节点和边的数量减少,算法迭代的次数也会相应减少。若压缩前需要的迭代次数为 t ,则压缩后需要的迭代次数大致为 $(cr)t$ 。因此,节点平均权值的计算需要耗时 $O(t(|V| + |E|)(cr)^2)$ 。子图连通性判断使用的是广度优先搜索,所以耗时也为 $O(t(|V| + |E|)(cr)^2)$ 。

4)在子图 k 核的维护方面,由于压缩图没有保留原节点的度信息,因此查询节点度的耗时没有变,仍然为 $O(|E|)$,算法需要迭代 $(cr)t$ 次,则此步的总耗时为 $O(t(cr)|E|)$ 。

5)在压缩图的解压缩方面,由于压缩算法 GC-W 会返回一个映射表 M , M 中记录了原图 G 中每个节点所对应的超节点。借助 M ,可以在 $O(|V|)$ 时间内将压缩后的图解压缩。

最终 CRK-C 算法的时间复杂度为 $O(t(cr)m)$, 其中, t 表示迭代次数, cr 表示压缩比, m 表示原图的边数。

接下来分析 CRK-C 算法的空间复杂度,由于该算法在 CRK-G 算法的基础上引入了图压缩过程,因此在计算过程中除了原有的空间消耗外,还需要额外存储压缩图的信息,压缩图信息包括每个超节点的邻节点和超边的权值,假设图压缩的压缩比为 cr ,则其空间消耗为 $O(2(cr)|E|)$ 。原有的空间消耗为 $O(4|E|)$,故 CRK-C 算法的空间复杂度为 $O(m)$ 。

4.1.3 CRK-C算法的误差分析

CRK-C 算法虽然提高了效率,但是其相对于 CRK-G 算法存在一定的误差。CRK-C 算法可能导致的误差是其多迭代删除的节点数。假设经过压缩算法压缩后的压缩比为 cr ,在压缩图中平均一个节点对应原图 $1/cr$ 个节点。在最坏的情况下,某次迭代删除中正常迭代删除一个节点即可达到紧密 k 核子图的要求,而由于采用的是压缩图进行迭代,则该次删除的节点数为 $1/cr$,误差为 $(1/cr) - 1 < 1/cr$,即 CRK-C 算法的平均误差小于 $1/cr$,压缩比 cr 越大则误差越小。

需要说明的是,压缩比 cr 取决于压缩阈值 θ 的取值, θ 和 cr 的具体关系则还要结合候选子图中节点权值的情况。但若在同一个图数据中, θ 越大则会有更多的节点被合并,伴随着压缩比 cr 会越小。

4.2 基于单次多节点删除的优化策略

提高 CRK-G 算法效率的改进策略除降低图规模外,另一种方法是减少算法的迭代次数,而减少迭代次数的一个有效且直接的方法是批量删除节点。通过在单次迭代中一次性删除多个节点来减少迭代次数,从而提高算法的效率。

4.2.1 CRK-F算法

基于单次多节点删除策略,对 CRK-G 算法进行改进,得到效率更高的快速贪婪算法 CRK-F。CRK-F 算法设置了每次迭代时的删除比率 γ , 其中 $\gamma \in (0, 1)$ 。若候选子图 H 的节点数为 $|V(H)|$, 设迭代时删除的节点集为 S , 则节点集 S 的数量 $|S| =$

$\gamma|V(H)|$ 。节点集 S 的取值是通过每次迭代后剩余子图的节点按节点权值进行排序,权值最小的前 $\gamma|V(H)|$ 个节点组成的集合即为节点集 S 。在下次迭代删除时, S 即作为需要从候选子图中删除的节点集。

需要说明的是, γ 越大则每次迭代删除的节点数就越多,总的迭代次数就越少,可能导致的误差就越大。 γ 的参考取值可经过计算得到,假设迭代删除一个节点后,子图的节点权值平均上升 $\Delta \overline{Aw}$,则:

$$\Delta \overline{Aw} = \frac{\sum_{u \in V'} \omega(u) - Aw}{n-1} - \frac{\sum_{u \in V'} \omega(u)}{n} \quad (10)$$

目标节点平均权值 w_Q 和当前候选子图的节点平均权值 Aw 之间的差值 $\Delta w_Q = w_Q - Aw$,假设需要删除 K 个节点后子图节点平均权值为 w_Q ,则:

$$w_Q - Aw = \Delta \overline{Aw} \times K \quad (11)$$

而迭代删除的节点数 $K = \gamma|V|$,则:

$$\gamma = \frac{K}{|V|} = \frac{w_Q - Aw}{\Delta \overline{Aw} \times |V|} \quad (12)$$

在实际应用时,式(12)可作为 γ 的取值依据。一般来讲,取值时考虑目标节点平均权值 w_Q 和当前候选子图的节点平均权值 Aw 之间的差值。当差值较大时, γ 取值可适当增大,使其能更快地迭代到目标值附近;反之, γ 取值可适当减小。

4.2.2 CRK-F算法的复杂度分析

CRK-F算法克服了CRK-G算法效率低下的问题,通过利用单次多节点删除的方法,极大地减少了算法的迭代次数,接下来对该算法进行时间复杂度与空间复杂度分析。

CRK-F算法与CRK-G算法相比,前者的迭代次数减少。首先分析改进后算法的迭代次数,初始时候选子图 H 的节点数为 $|V(H)| \leq |V(G)|$,其中 G 为给定需要查询的原图。若每次迭代删除的节点集为 S ,则经过 i 次迭代后, $|S| = \gamma|V(H_i)|$,即此时节点集 S 将从剩余子图 H_i 中删除。这表明在经过第 $i+1$ 次迭代后,剩余子图 H_{i+1} 中至多有 $(1-\gamma)(|V(H_i)|)$ 个节点,即 $|V(H_{i+1})| \leq (1-\gamma)(|V(H_i)|)$ 。假设需要进行 t' 次迭代后算法才结束,则:

$$1 \leq |V(H_{t'})| \leq (1-\gamma)(|V(H_{t'-1})|) \leq \dots \leq (1-\gamma)^t (|V(H)|) \leq (1-\gamma)^t |V(G)|$$

上式经过化简可得 $t' \leq \left\lceil \log_{(1-\gamma)} \frac{1}{|V(G)|} \right\rceil$,因为CRK-G算法的时间复杂度为 $O(t(n+m))$,所以最终CRK-F算法的时间复杂度为 $O\left((n+m) \log_{(1-\gamma)} \frac{1}{n}\right)$ 。

CRK-F算法与CRK-G算法在空间上的消耗没有区别,所以CRK-F算法的空间复杂度也是 $O(m)$ 。

4.2.3 CRK-F算法的误差分析

根据CRK-F算法特点可知,其相对于CRK-G算法的误差是由于批量删除时多删除的节点导致。

假设给定删除比率 γ ,当算法迭代到第 t 次时结束,那么在 $t-1$ 次时候选子图剩余的节点数为 $(1-\gamma)^{t-1}(|V|)$ 。随后进行第 t 次迭代,在最坏情况下,本次迭代原本只需要删除一个节点即可达到紧密 k 核子图的要求,而由于采用批量删除,实际删除的节点数为 $(1-\gamma)^{t-1}(|V|)\gamma$,误差为 $(1-\gamma)^{t-1}(|V|)\gamma - 1 < (1-\gamma)^{t-1}(|V|)\gamma$,即CRK-F算法误差不超过 $(1-\gamma)^{t-1}(|V|)\gamma$ 。

4.3 CRK-C与CRK-F算法对比分析

通过对CRK-C与CRK-F算法的原理分析,对两者的特点总结如下:

1) CRK-C算法使用了图压缩策略,通过将节点权值在阈值范围内的连通节点进行合并,从而降低子图的规模,也相应减少了迭代次数,算法效率相对CRK-G算法而言有了较大的提升。根据算法的特点,若图中节点权值较为相近时,节点压缩策略将会起较大的作用。在节点压缩时,更多节点权值相近的节点将会被合并,并且由于被合并的节点权值相近,其在迭代删除过程中所造成的误差也较小,因此CRK-C算法更适用于图中节点权值差异性较小的数据。

2) CRK-F算法利用单次多节点删除策略,每次迭代删除多个节点,显著减少了迭代次数。由于CRK-F算法仅需要在迭代删除时设置删除比率 γ ,从而进行节点的批量删除,其实现相较于CRK-C算法而言更加简单,效率提升也较为明显。并且根据算法的特点,其误差能够较好地被预测,有利于控制算法的误差。当图中节点权值差异性较大或无法确定差异性时,可优先考虑使用CRK-F算法。

5 实验结果与分析

本文在3个真实数据集上进行了实验,对比了算法在不同查询条件下和不同规模图上查询的耗时,并分析压缩算法GC-W和紧密 k 核子图模型的有效性,最后结合实例进一步说明模型的优越性。

5.1 数据集

本文实验使用的数据集主要有以下3个:

1) 生物通用交互数据集Bio-GRID,其中节点表示基因或蛋白质,边表示基因或蛋白质之间存在交互,边的权值表示节点之间交互的得分,得分越高说明节点之间的联系越紧密。

2)美国安然公司的邮件网络 Email-Enron, 其中顶点表示邮箱,边表示一个邮箱向另一邮箱发送过邮件,边的权值表示邮箱之间邮件的互通次数,次数越高说明邮箱之间的联系越紧密。

3)由 DBLP 数据构成的作者协作网络,其中节点表示作者,边表示两个作者共同合作发表过文章,边的权值表示作者合作过的次数,次数越高说明作者之间的联系越紧密。

上述数据集经预处理转化为图数据,各个图的基本特性如表 1 所示。其中:节点数和边数反映了图的规模;最大核值为 k 核分解后图中节点最大的核值,反映了图中最稠密社团的稠密度;平均核值反映了图整体的稠密程度;节点平均权值反映了图节点之间整体联系的紧密程度。

表 1 数据集的基本特性

Table 1 Basic characteristics of dataset

数据集	节点数	边数	最大核值	平均核值	节点平均权值
Bio-GRID	7 777	70 774	79	6.38	6.29
Email-Enron	38 378	410 296	230	7.42	666.84
DBLP	1 086 874	9 156 074	286	5.31	1 600.11

5.2 实验分析

本文在上述 3 个数据集上进行实验,对比了算法 CRK-G、CRK-C、CRK-F 在不同条件下的查询效率和图压缩算法 GC-W 的压缩效率,并验证了模型的有效性。实验使用 C++ 实现,在 2.90 GHz Intel® Core™ i5-9400 CPU、8.0 GB 内存、Windows10 系统的台式机上运行。

5.2.1 算法效率分析

在查询紧密 k 核子图时,需要给定核值 k 和节点平均权值 w_0 。为验证算法的效率,本文设计了控制变量的对比实验,记录 3 个算法随着给定值变化时其查询所消耗的时间。在下文实验中,CRK-C 算法

的压缩阈值 $\theta=0.01$,CRK-F 算法的删除比率 $\gamma=0.1$ 。

由于查询参数的取值不是本文的研究重点,实验中的参数值是通过前期的初步实验并结合数据集的特性得到。其中,参数 w_0 的给定是通过初步的实验,大致计算得到各个数据集子图的最大节点平均权值,将该值取整十或整百,然后等差递减得到 w_0 的取值范围。在实际应用中可以根据对子图紧密程度的需求来选择核值 k 和节点平均权值 w_0 ,以及对误差允许的范围来选择合适的压缩阈值 θ 和删除比率 γ 。

本文分别控制 k 和 w_0 的变化,在 3 个数据集上进行以下两组实验:

1) 随着给定节点平均权值 w_0 的变化,实验并记录在 3 个数据集上算法 CRK-G、CRK-C、CRK-F 查询所需要的时间。对于数据集 Bio-GRID,给定 $k=20$,变量 w_0 的取值范围为 $\{20,30,40,50,60\}$;对于数据集 Email-Enron,给定 $k=30$,变量 w_0 的取值范围为 $\{9\ 000,10\ 000,11\ 000,12\ 000,13\ 000\}$;对于数据集 DBLP,给定 $k=40$,变量 w_0 的取值范围为 $\{200,225,250,275,300\}$ 。

3 种算法随 w_0 变化的运行效率对比结果如图 4 所示,随着节点平均权值 w_0 增大,各算法在不同数据集上的查询耗时总体均增加。之所以如此,是因为当给定节点平均权值 w_0 增大时,算法所需要的迭代次数增多,故耗时增加。在 3 种算法的效率对比中,CRK-G 算法的查询耗时明显高于另两种算法。在 Bio-GRID 数据集上,CRK-C 算法的速度略快于 CRK-F 算法,通过对数据集特性的分析,发现 Bio-GRID 数据集中存在较多权值相近的节点,使得该数据集的候选子图经过压缩后的压缩比较低,故 CRK-C 算法的查询速度较快。反之,在 DBLP 数据集上,CRK-F 算法速度快于 CRK-C 算法是由于该数据集上权值相近的节点较少。

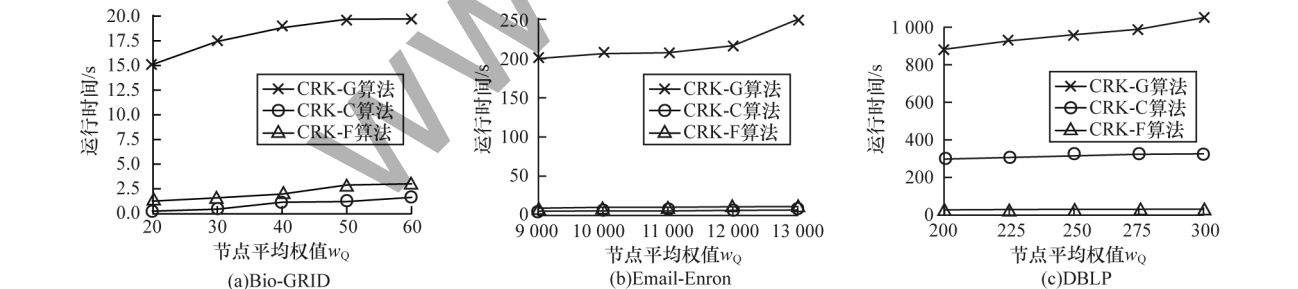


图 4 3 种算法随 w_0 变化的运行效率对比

Fig.4 Comparison of the operating efficiency of the three algorithms with the change of w_0

2)随着给定核值 k 的变化,实验并记录在 3 个数据集上算法 CRK-G、CRK-C、CRK-F 查询所需要的

时间。对于数据集 Bio-GRID,给定 $w_0=40$,变量 k 的取值范围为 $\{10,15,20,25,30\}$;对于数据集 Email-

Enron, 给定 $w_0=11\ 000$, 变量 k 的取值范围为 $\{10, 20, 30, 40, 50\}$; 对于数据集 DBLP, 给定 $w_0=250$, 变量 k 的取值范围为 $\{20, 30, 40, 50, 60\}$ 。

3种算法随 k 变化的运行效率对比结果如图5所示,随着核值 k 增大,各算法在不同数据集上的查询

耗时总体均减少。之所以如此,是因为当给定核值 k 增大时,得到的候选 k 核子图规模减小,算法迭代次数减少,故耗时减少。另外,在3种算法的效率对比中,当 k 值较小时,CRK-G算法的查询耗时明显高于另两种算法。

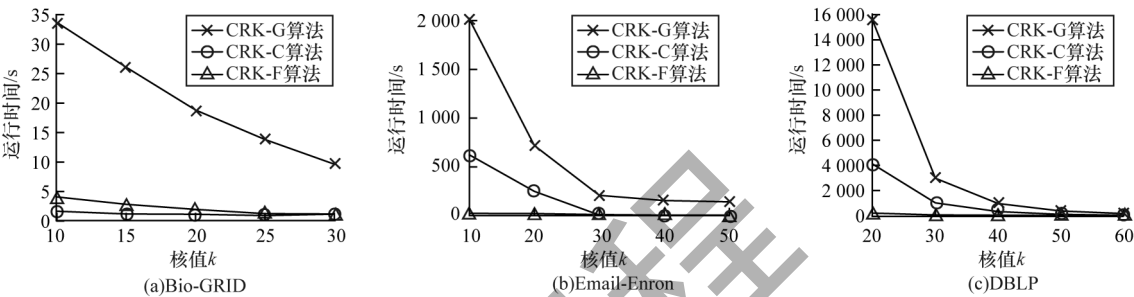


图5 3种算法随 k 变化的运行效率对比

Fig.5 Comparison of the operating efficiency of the three algorithms with the change of k

上述实验结果与本文对各个算法的时间复杂度分析结果相符合,CRK-G算法的查询耗时总体均高于CRK-C和CRK-F算法。而CRK-C算法和CRK-F算法两者耗时大致相近,具体耗时高低主要取决于数据集的特点,当数据集中存在较多权值相近的节点时,图压缩算法压缩后得到的图规模较小,使得CRK-C算法的查询耗时低于CRK-F算法。

5.2.2 算法误差分析

由于CRKSQ问题是NP-难的,无法在多项式时间复杂度内找到最优解,CRK-G算法虽然能够找到一个可行解,但其与最优解之间的偏离程度无法被预计。CRK-C和CRK-F算法是在CRK-G算法基础上的改进,效率有了很大的提升,但两者相对于CRK-G算法存在一定的误差。接下来以CRK-G算法为参考,在不同 k 值条件下,对CRK-C和CRK-F算法查询结果的误差进行分析。

根据算法的原理,评判误差的依据是CRK-C和CRK-F算法相对于CRK-G算法是否过多删除了节点。实验在3个数据集上进行:对于数据集 Bio-GRID, 给定 $w_0=40$, 变量 k 的取值范围为 $\{10, 15, 20, 25, 30\}$; 对于数据集 Email-Enron, 给定 $w_0=11\ 000$, 变量 k 的取值范围为 $\{10, 20, 30, 40, 50\}$; 对于数据集 DBLP, 给定 $w_0=250$, 变量 k 的取值范围为 $\{20, 30, 40, 50, 60\}$ 。

实验记录了各个算法在上述条件下查询得到的子图节点数,计算了CRK-C和CRK-F算法相对于CRK-G算法的误差百分比,结果分别如表2~表4所示,多次实验的平均误差均在8%以内,误差较小。

表2 Bio-GRID数据集的查询结果与误差

Table 2 Query results and errors of Bio-GRID dataset					
核值 k	CRK-G 算法	CRK-C 算法	CRK-F 算法	CRK-C 算法误差	CRK-F 算法误差
10	120	117	114	0.025	0.050
15	120	118	119	0.017	0.008
20	120	116	105	0.033	0.130
25	109	95	105	0.130	0.037
30	115	98	100	0.150	0.130
平均误差	—	—	—	0.071	0.071

表3 Email-Enron数据集的查询结果与误差

Table 3 Query results and errors of Email-Enron dataset					
核值 k	CRK-G 算法	CRK-C 算法	CRK-F 算法	CRK-C 算法误差	CRK-F 算法误差
10	234	232	222	0.009	0.051
20	234	232	220	0.009	0.060
30	234	232	214	0.009	0.086
40	234	232	214	0.009	0.086
50	234	232	214	0.009	0.086
平均误差	—	—	—	0.009	0.074

表4 DBLP数据集的查询结果与误差

Table 4 Query results and errors of DBLP dataset					
核值 k	CRK-G 算法	CRK-C 算法	CRK-F 算法	CRK-C 算法误差	CRK-F 算法误差
20	622	625	637	0.056	0.038
30	577	526	565	0.088	0.021
40	537	526	538	0.020	0.002
50	535	527	535	0.015	0.000
60	287	287	287	0.000	0.000
平均误差	—	—	—	0.036	0.012

5.2.3 图压缩算法有效性分析

为验证图压缩算法 GC-W 的有效性,实验记录了不同 k 值条件下,GC-W 算法在 3 个数据集上压缩候选子图的耗时。

对于数据集 Bio-GRID,变量 k 的取值范围为

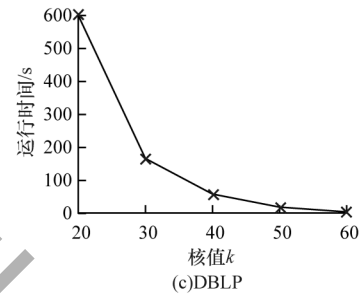
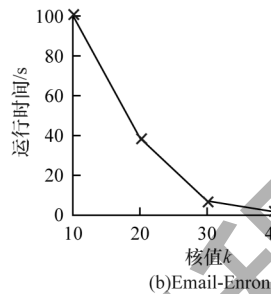
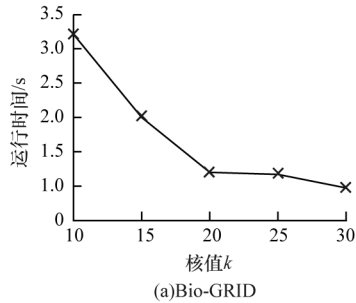


图6 GC-W 算法随 k 变化的压缩耗时

Fig.6 Compression time of GC-W algorithm with the change of k

压缩耗时相较于查询耗时低了 1 或 2 个数量级,其可以在相对短的时间内对候选子图进行压缩,有效地减少了查询时间,压缩算法是有效的。

5.2.4 模型有效性分析

为验证紧密 k 核子图 (CRKS) 模型的有效性,本文实验对比了 CRK-F 算法与忽略权值的 k 核 (k -core) 算法,记录两者在不同 k 值条件下查询得到的子图节点平均权值。

实验分别在 3 个数据集上进行,考虑到 CRK-F 算法和 k -core 算法每次查询得到的子图都有可能存在多个,并且前者会由于给定的 w_Q 不同而得到不

{10, 15, 20, 25, 30}; 对于数据集 Email-Enron, 变量 k 的取值范围为 {10, 20, 30, 40, 50}; 对于数据集 DBLP, 变量 k 的取值范围为 {20, 30, 40, 50, 60}。实验结果如图 6 所示,当 k 值增大时,压缩时间减少。由于当 k 值增大时候选 k 核子图的规模减小,因此压缩算法 GC-W 的耗时减少。

同的结果。因此,本文将 CRK-F 算法在不同 w_Q 条件下查询得到的子图最大节点平均权值与 k -core 算法查询得到的所有子图最大节点平均权值进行对比。

对于数据集 Bio-GRID,变量 k 的取值范围为 {10, 15, 20, 25, 30}; 对于数据集 Email-Enron, 变量 k 的取值范围为 {10, 20, 30, 40, 50}; 对于数据集 DBLP, 变量 k 的取值范围为 {20, 30, 40, 50, 60}。实验结果如图 7 所示。从实验结果可以看出,CRK-F 算法得到的子图节点平均权值均大于 k -core 算法,验证了 CRKS 模型的有效性和优越性。

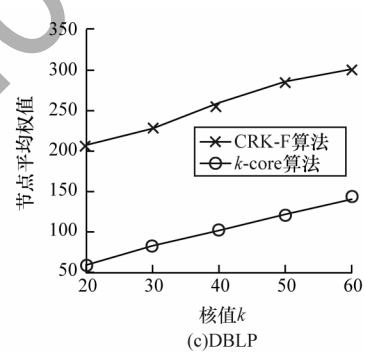
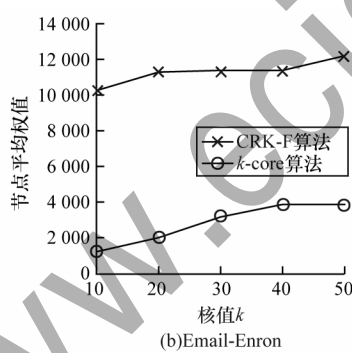
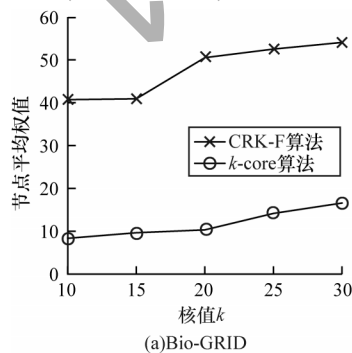


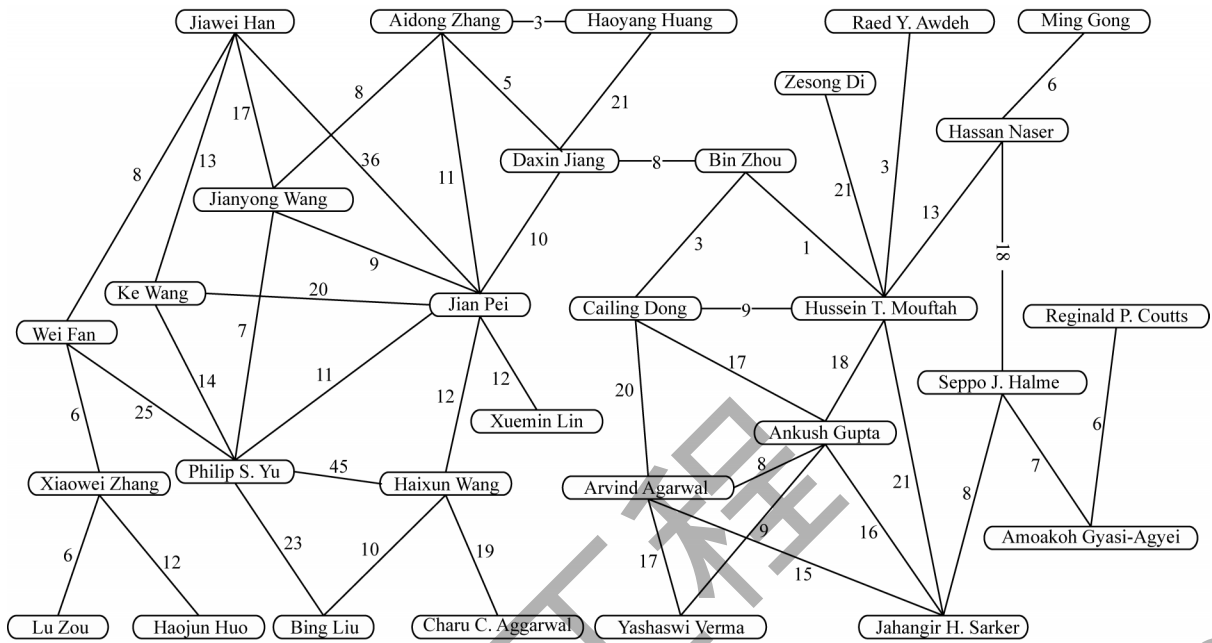
图7 CRK-F 与 k -core 算法的子图节点平均权值对比

Fig.7 Comparison of average weights of subgraph nodes obtained between CRK-F and k -core algorithms

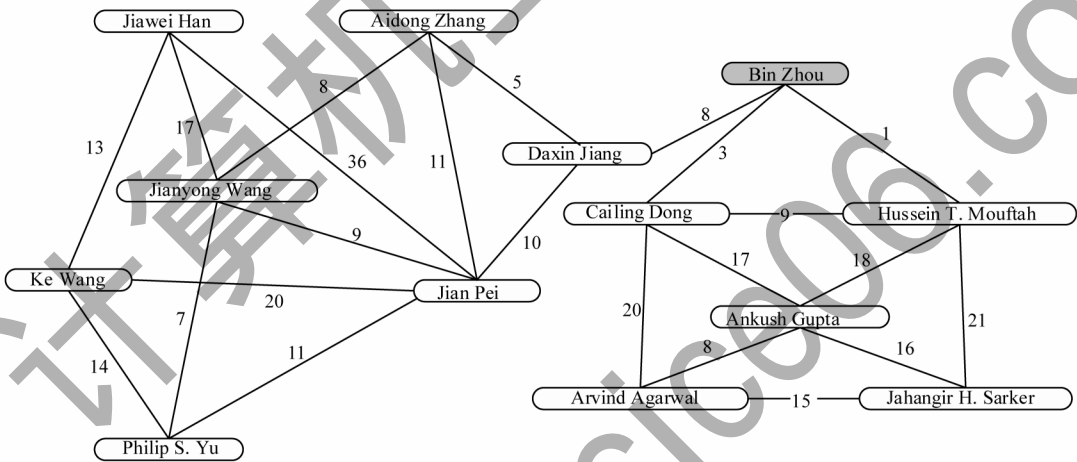
5.2.5 实例分析

为进一步说明 CRKS 模型的合理性,本文选取一个实例进行分析。图 8(a)所示是作者协作网络中选取的部分数据,其包含了 30 个节点和 46 条边,图的节点平均权值为 40.67,存在一些联系较为松散的节点。若忽略权值,仅给定 $k=3$,进行 3 核查询,查询结果如图 8(b)所示。从图中可以看出,虽然所有节点的度均大于等于 3,对应着至少 3 个邻接节点,但结合边的权值就可发现节点“Bin Zhou”所对应的权

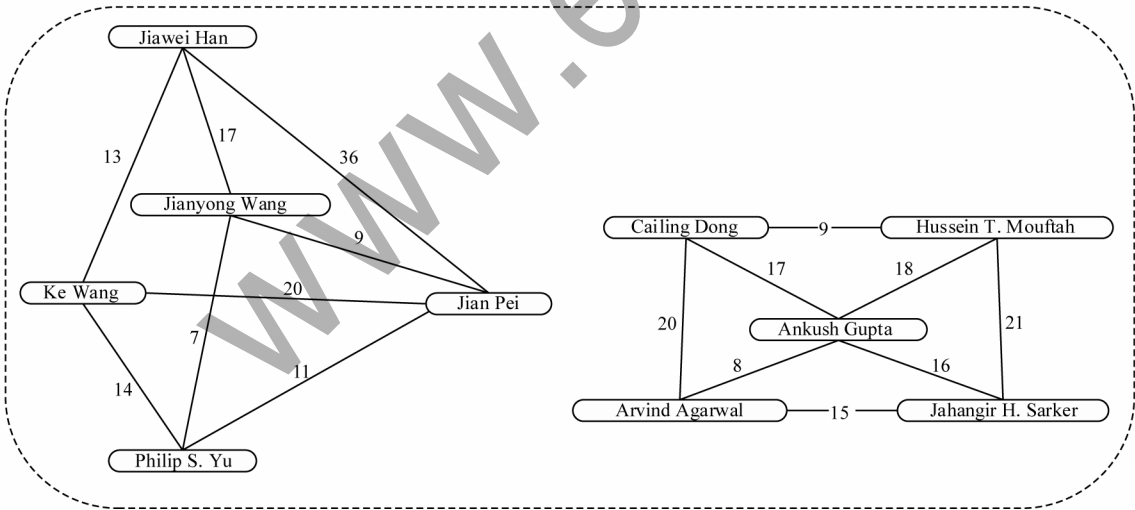
值较小,该节点将左右两部分的图连接成一体,使得图整体结构并不紧凑,左右两部分节点之间联系的紧密程度不足。若考虑权值,给定 $w_Q=50$,对图 8(a)所示的图进行紧密 3 核子图查询,得到图 8(c)所示的结果。该结果包含两个子图,每个图相较于图 8(b),内聚性更强,节点之间的联系也更加紧密。两个子图的节点平均权值分别为 50.8 和 53.2,相较于图 8(b)的 47.08,两个子图整体的联系也更加紧密。



(a)作者协作网络中选取的部分数据



(b)不考虑权值情况下仅给定 $k=3$ 的查询结果



(c) $k=3$ 、 $w_0=50$ 时CRKS模型的结果

图 8 紧密 k 核子图的查询实例分析结果

Fig.8 Query example analysis results of closely related k -core subgraph

综合上述分析,CRKS模型可以检测出加权图中紧密联系的子图,相较于现有的方法更加合理有效。

6 结束语

本文提出一种在加权图中查询联系紧密的连通 k 核子图问题,并证明了该问题是NP-难的。为解决CRKSQ问题,设计基于贪婪策略的启发式算法CRK-G,其能在可接受的时间内为CRKSQ问题找到一个近似解,分别从降低图规模和减少迭代次数两个方面出发,提出两个改进算法CRK-C和CRK-F,其在查询速度上有明显提升。在不同规模数据集上进行多次实验,结果表明,CRK-C和CRK-F算法的平均误差都在8%以内。下一步将研究本文算法参数的取值和优化问题,并探索其他更加高效的紧密 k 核子图查询算法,以满足超大图的查询要求。

参考文献

- [1] FANG Y X, HUANG X, QIN L, et al. A survey of community search over big graphs[J]. The VLDB Journal, 2020, 29(1): 353-392.
- [2] 胡开先,梁英,苏立新,等. 基于完全子图的社交网络用户特征识别方法[J]. 模式识别与人工智能, 2016, 29(8): 698-708.
HU K X, LIANG Y, SU L X, et al. Method for social network user feature recognition based on clique[J]. Pattern Recognition and Artificial Intelligence, 2016, 29(8): 698-708. (in Chinese)
- [3] LAPPAS T, LIU K, TERZI E. Finding a team of experts in social networks[C]//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2009: 467-476.
- [4] YAN X, ZHOU X J, HAN J. Mining closed relational graphs with connectivity constraints[C]//Proceedings of the 21st International Conference on Data Engineering. Washington D. C., USA: IEEE Press, 2005: 357-358.
- [5] ASTHANA S, KING O D, GIBBONS F D, et al. Predicting protein complex membership using probabilistic network reliability[J]. Genome Research, 2004, 14(6): 1170-1175.
- [6] 杜梅,胡学钢,李磊,等. 基于网络结构极值优化的半监督社团检测方法[J]. 模式识别与人工智能, 2015, 28(2): 162-172.
DU M, HU X G, LI L, et al. Network structure-enhanced extremal optimization based semi-supervised algorithm for community detection[J]. Pattern Recognition and Artificial Intelligence, 2015, 28(2): 162-172. (in Chinese)
- [7] FANUEL M, ALAÍZ C M, SUYKENS J A K. Magnetic eigenmaps for community detection in directed networks[J]. Physical Review E, 2017, 95(2): 022302.
- [8] 李国朋,潘志松,姚清,等. 融合先验信息的非负矩阵分解社区发现算法[J]. 模式识别与人工智能, 2016, 29(7): 608-615.
LI G P, PAN Z S, YAO Q, et al. Nonnegative matrix factorization algorithm with prior information for community detection[J]. Pattern Recognition and Artificial Intelligence, 2016, 29(7): 608-615. (in Chinese)
- [9] SEIDMAN S B. Network structure and minimum degree[J]. Social Networks, 1983, 5(3): 269-287.
- [10] MEDYA S, MA T, SILVA A, et al. K-core minimization: a game theoretic approach[EB/OL]. [2021-08-08]. <https://arxiv.org/pdf/1901.02166v1.pdf>.
- [11] FANG Y X, WANG Z, CHENG R, et al. On spatial-aware community search[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(4): 783-798.
- [12] CHU D M, ZHANG F, LIN X M, et al. Finding the best k in core decomposition: a time and space optimal solution[C]//Proceedings of the 36th IEEE International Conference on Data Engineering. Washington D. C., USA: IEEE Press, 2020: 685-696.
- [13] LI R H, QIN L, YU J X, et al. Finding influential communities in massive networks[J]. The VLDB Journal, 2017, 26(6): 751-776.
- [14] FANG Y X, WANG Z R, CHENG R, et al. Effective and efficient community search over large directed graphs[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(11): 2093-2107.
- [15] CHEN Y K, FANG Y X, CHENG R, et al. Exploring communities in large profiled graphs[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(8): 1624-1629.
- [16] HABIB W M A, MOKHTAR H M O, EL-SHARKAWI M E. Weight-based K-truss community search via edge attachment[J]. IEEE Access, 2020, 8: 148841-148852.
- [17] AKBAS E. Index based efficient algorithms for closest community search[C]//Proceedings of 2019 IEEE International Conference on Big Data. Washington D. C., USA: IEEE Press, 2020: 701-710.
- [18] YUAN L, QIN L, ZHANG W J, et al. Index-based densest clique percolation community search in networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(5): 922-935.
- [19] BATAGELJ V, ZAVERNIK M. An $O(m)$ algorithm for cores decomposition of networks[J]. Computer Science, 2003, 1(6): 34-37.
- [20] KHAOUID W, BARSKY M, SRINIVASAN V, et al. K-core decomposition of large networks on a single PC[J]. Proceedings of the VLDB Endowment, 2015, 9(1): 13-23.
- [21] 李鸣鹏,高宏,邹兆年. 基于图压缩的最大Steiner连通 k 核查询处理[J]. 软件学报, 2016, 27(9): 2265-2277.
LI M P, GAO H, ZOU Z N. Maximum steiner connected k-core query processing based on graph compression[J]. Journal of Software, 2016, 27(9): 2265-2277. (in Chinese)
- [22] BARBIERI N, BONCHI F, GALIMBERTI E, et al. Efficient and effective community search[J]. Data Mining and Knowledge Discovery, 2015, 29(5): 1406-1433.
- [23] CUI W Y, XIAO Y H, WANG H X, et al. Local search of communities in large graphs[C]//Proceedings of 2014 ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 2014: 991-1002.
- [24] ZHENG D, LIU J Q, LI R H, et al. Querying intimate-core groups in weighted graphs[C]//Proceedings of the 11th IEEE International Conference on Semantic Computing. Washington D. C., USA: IEEE Press, 2017: 156-163.
- [25] KARP R M. Reducibility among combinatorial problems[C]//Proceedings of IEEE International Conference on Complexity of Computer Computations. Washington D. C., USA: IEEE Press, 1972: 85-103.
- [26] LIANG Y Z, CHEN C, WANG Y K, et al. Reachability preserving compression for dynamic graph[J]. Information Sciences, 2020, 520: 232-249.
- [27] MASERRAT H, PEI J. Neighbor query friendly compression of social networks[C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2010: 533-542.