并一一实现其图相似查询算法.我们从 PubChem 数据集中抽样得到实验数据集,将实现的 4 种算法在抽样的图集上进行实验(实验结果见第 4.1 节),总结分析实验结果发现:已有的规则本身具有优缺点和适用性,这 4 种方法没有一种能够绝对地优于其他的方法.为了最大程度地保留已有方法的优点,避免缺点,本文提出了一种全新的面向关系型数据库的过滤框架,新的过滤框架可以将这 4 种图的编辑距离的近似计算方法进行有效结合,求得更紧的上下界,形成了能够进行非常高效地进行剪枝的方法.

3 面向关系型数据库的图相似查询算法

本节重点阐述面向关系型数据库的图相似查询算法,首先,简要介绍所提出的全新过滤框架以及新的过滤规则.然后,将其在 PostgreSQL 这一开源的关系型数据库中的实现思路进行描述.最后,综合介绍面向关系型数据库的图相似查询框架,并且提出了基于图相似框架的图数据管理方法.

3.1 图结构在关系型数据库中的关系模式

作为一种最为普遍的结构化数据存储工具,关系型数据库与结构化查询语言(SQL)的结合一直在数据操作方面表现良好.但是当前,随着图结构的不断发展,在关系型数据库中对图结构的操作仍然支持不够.因此,本文实现一种基于 SQL 的图相似性过滤方法,充分利用关系型数据库的特性,增强了关系型数据库对图相似性搜索的支持.

我们将所有的图存储在关系数据库管理系统 PostgreSQL 中,存储图的 Graph 表具有以下 3 个基本字段:编号(rid:integer)、节点(vertex:integer[])、边(edge:integer[]),每一个图在关系数据库中存储为一条记录,假设某条记录存储了图 g_p,则该记录可以表示为 $(p,\{V_1,l_1,V_2,l_2,...,V_n,l_n\},\{E_1,E_1',E_2,E_2',...,E_n,E_n'\})$.通过编号唯一地标志一个图,并将顶点和边采用数组的形式存储,(v_k,l_k)代表一个顶点,包括了顶点编号和顶点标签两个属性,(E_k,E_k') 代表一条边,用边的两个端点表示.例如,对于图 1 中的甲烷结构,我们将其存储为表 2 中的数据格式.

Table 2 DataSchema

表 2 图存储模式

Rid	Vertex	Edge
1	{1,6,2,1,3,1,4,1,5,1}	{1,2,1,3,1,4,1,5}

对于过滤方法中需要用到的图特征结构,本文通过如下的关系模式将提取得到的特征结构进行存储,存储图的 Graph 表具有以下 4 个基本字段:编号(rid:integer)、路径(path:integer[])、树结构(tree:integer[])、星型结构(star:integer[]).分别对 Path,Tree,Star 运用一个属性存储下来,并用 rid 标志这些特征所属的图.对于图 1 中的甲烷结构,对其进行特征抽取后,指定 path 的长度为 2,tree 的深度为 1,得到的数据格式见表 3.

Table 3 Graph structure schema

表 3 图特征存储模式

Rid	Path	Tree	Star
1	{1,2,1,3,1,4,1,5}	{1,2,1,3,1,4,1,5,2,1,3,1,4,1,5,1}	{1,2,1,3,1,4,1,5}

3.2 面向关系型数据库的图相似过滤方法

因为在图相似查询过程中,过滤部分是非常重要的环节,所以本文对原有的过滤流程进行了重新设计,提出了一种分层过滤框架,如图 3 所示.框架的整体思想是,希望通过多层过滤较好地规避时间成本和空间成本.因为原有的过滤方法都不是完全最优的,因此,我们考虑将他们结合起来进行使用,即:最先用过滤粒度最粗,但时间复杂度最低的方法进行过滤,得到相应的候选集后,由下一层的过滤方法在所得候选集中进行过滤,以此类推,通过这种层层过滤的方式,使得每个方法都能发挥最好的过滤效果.因此,下文中统一使用 combined-filter 表示本文所使用的过滤方法.

本文通过大量的实验得出了一种较为高效的分层过滤顺序,即:先对数据集运用基于 label 的过滤,再对得

到的候选集采用基于 path 的过滤,然后对得到的候选集采用基于 tree 的过滤,基于 star 结构的过滤放到最后进行.这一过滤顺序具有非常好的过滤能力,并且时间复杂度较低,流程图如图 3 所示.

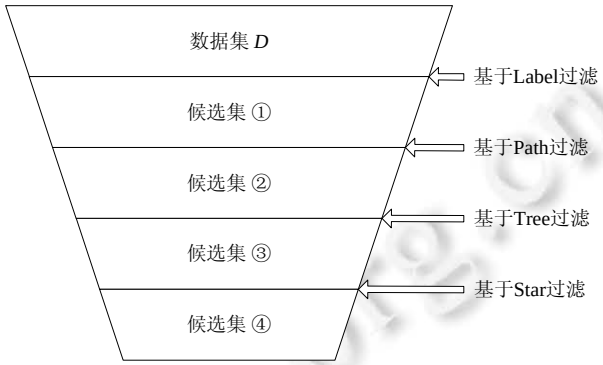


Fig.3 Novel framework of graph similarity search based on RDBMS

图 3 面向关系型数据库的全新图相似过滤框架

下面介绍 combined-filter 在关系型数据库中的实现思路.我们在 PostgreSQL 中使用 PL/SQL 实现了基于 label,star,tree,path 这 4 种图编辑距离过滤算法,并通过函数调用.因为 4 种方法之间是相互独立的,所以这种松耦合的方式使得方法的普适性和稳定性都较强,可以方便地嵌入到各类关系型数据库中,而不需要修改数据库的内核.对于这 4 种方法,我们采用统一的过滤算法框架进行实现,以此保证算法的一致性,保证分层过滤的效率.由此得到的过滤算法框架见算法 1.

算法 1. 过滤算法框架.

```
1   function FILTER(q,D,T,Cand)
2       Input:q(待搜索图),D(数据库中的图结构数据集),T(编辑距离阈值);
3       Onput:Cand(候选集合).
4       Cand←∅
5       for g∈D do
6           if filter-startegy() then
7               Add(g,Cand)
8           end if
9       end for
10      return Cand
11  endfunction
```

3.3 基于新型过滤框架的图相似查询算法

本文提出的新型图相似查询算法流程见算法 2 所示.给出一个待搜索的图 q,并给出图数据集 D,编辑距离阈值 T,在过滤阶段中,通过 label,path,tree,star 的顺序进行层次过滤,得到候选的图集合 *Cand*.然后,通过调用精确的图编辑距离计算方法,将 *Cand* 中的图一一验证得到最终的结果集合 R.因为几个过滤规则间是解耦的,所以混合的过滤规则可以有多种组合,因此提升了图相似查询的灵活性.

算法 2. 基于新型过滤框架的图相似度搜索算法.

```
1   function COMBINED-GRAPH-SEARCH(q,D,T)
2       R←∅
3       CREATE VIEW Candidate AS
4       SELECT*FROM StarFilter(q,TreeFilter(q,PathFilter(q,LabelFilter(q,D))));
```

```
5        for g in SELECT*FROM Candidatedo
6            if ExactGED(g)≤T then    //计算精确编辑距离
7                Add(g,R)
8            end if
9        end for
10   end function
```

因此,在调用时图相似查询方法时,可以通过简单的 SQL 查询语句进行调用.通过对 SQL 函数传入 3 个参数:待查询图的顶点数组(vertex-array)、待查询图的边数组(edge-array)、查询阈值(*T*),即可返回查询得到的图结果集.示例查询语句如下所示:

SELECT *Combined-Graph-Search*(*Vertex-Array*,*Edge-Array*,*T*).

3.4 SQL-Based图特征提取

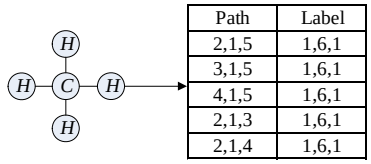
对于特征抽取操作,我们结合 SQL 语言的特性和广度优先搜索的思想,设计了一种效率较高的图遍历算法,见算法 3.

算法 3. 图的深度优先搜索算法.

```
1    function BREADTH-FIRST-SEARCH(edge,node)
2        //edge 边集合,node 顶点集合
3        WITH RECURSIVE transitive-closure(a,b,distance,paths,labels) AS
3        (
4        SELECT fromnode, tonode, 1 AS distance,
5        array[fromnode] $||$ array[tonode] AS paths,
6        array[n1.nname] $||$ array[n2.nname] AS labels
7        FROM edge left join node n1 on fromnode=n1.nid left join node n2 on tonode=n2.nid
8        UNION ALL
9        SELECT tc.a, e.tonode, tc.distance +1,
10       tc.paths $||$ array[e.tonode] AS paths,
11       tc.labels $||$ array[n3.nname] AS labels
12       FROM edge AS e left join node n3 on n3.nid=e.tonode
13       JOIN transitive-closure AS tc ON e.fromnode=tc.b
14       WHERE tc.paths::text NOT LIKE '%'||e.tonode||'%'
15       )
16       SELECT tc1.paths,tc1.labels FROM transitive-closure as tc1 where tc1.distance=q
17       and tc1.a$<$tc1.b ORDER BY labels;
18   end function
```

运用 WITH,UNION 关键字在 SQL 语言中实现了高效的递归查询,从而查询得到指定长度的图中所有路径集合,distance 字段代表路径长度,paths 代表顶点编号组合的路径,labels 代表顶点标签组合的路径,如图 4 所示,即为对甲烷结构进行长度为 3 的路径提取示例.根据此图搜索算法,也可以方便地处理得到 tree,star 等结构.

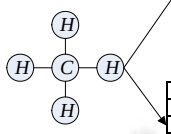
对于递归查询所需要的 Edge 和 Node 表,我们通过临时表的形式进行构建,Node 表具有两个字段:顶点编号(nid:integer)、顶点标签(nname:integer);Edge 表具有 3 个字段:开始顶点(fromnode:integer),结束顶点(tonode:integer),边标签(ename:integer).两张表结合起来,可以完全表示出一张图具有的结构和特征.在数据处理时,先从 graph 表中查询某个具体图,然后将其用 Node 表和 Edge 表组织起来,因为关系型数据库的支撑,在其基础上进行了查询等操作就能够具有较高的效率.例如,甲烷结构可以用图 5 的形式在数据库中表示出来.



Path	Label
2,1,5	1,6,1
3,1,5	1,6,1
4,1,5	1,6,1
2,1,3	1,6,1
2,1,4	1,6,1

Fig.4 Path extract example

图 4 路径提取示例



Nid	Name
1	6
2	1
3	1
4	1
5	1

Fromnode	Tonode	Ename
1	2	1
1	3	1
1	4	1
1	5	1

Fig.5 Node and Edge relation schema

图 5 Node 和 Edge 关系模式

4 实验与结果分析

本节对图相似查询算法进行了实验验证,并对实验结果和分析进行了展示.我们所有的实验都在 Intel(R) Xeon(R) CPU E5-4607@2.20GHz,8GB 内存的服务器上完成,操作系统为 CentOS Linux 7.0.所有的算法都通过 PL/SQL 语言进行实现,并在 PostgreSQL 9.4 版本上执行通过.

我们选用了公开的 PubChem 数据集作为实验数据.本数据集中包含 100 万个真实的化学结构,平均的顶点个数为 23.98,平均边数为 25.76,属于较为稀疏的简单图结构,在图中不存在环等复杂结构,符合本文实验的开展条件.在每次实验中,本文将数据集中编号最小的图作为待查询图,将数据集中剩余的图结构作为图集合,通过调用不同的过滤方法查询图集合中与待查询图相似的图结构,即可得到结果集以及过滤所需要的时间.通过结果集大小以及过滤时间,可以很好地衡量不同算法的过滤效果.下面给出了其对应的定义.

- 结果集(candidate size),指所有图经过过滤规则作用,过滤掉无用结果之后得到的候选集.结果集大小即为结果集所包含图结构的个数;
- 过滤时间(response time),指过滤算法执行的总时间,这里,我们忽略候选集进行具体图编辑距离计算的验证时间,因为验证时间直接与候选集大小成正比,从候选集大小就可以衡量出验证时间.

本文首先对 4 种已有过滤方法的过滤效果进行了对比实验分析.通过在 PubChem 数据集中随机抽取 0.1k 个、1k 个、10k 个有机物结构,组成了 4 组实验数据集,通过实验对比了 4 种方法的过滤效果.然后,通过随机生成多组 100k 的数据集,对本文提出的分层过滤模型进行了实验验证,并与 4 种已有过滤方法的过滤效果进行对比,从而分析本文方法的过滤效率.

4.1 4种过滤方法效能分析

基于 path 的过滤算法需要指定 path 长度 q,因此,我们在 0.1k,1k,10k 这 3 个数据集对取不同阈值、不同 path 长度环境下的过滤效果进行了测试.结果如图 6 所示.图 6(a)~图 6(c)分别代表 0.1k,1k,10k 数据集,指定不同阈值所得到的过滤结果集大小(candidate size);图 6(d)指出了在 10k 数据集,path 提取的时间(response time)随 path 的增长有增长趋势,所以 q 值应该在保证过滤能力的情况下尽可能小.可以看到:当 path 长度取 3 时,在不同大小的数据集,不同阈值下的过滤表现都较为良好,并且因为有机物结构大小不统一,并且当 path 长度超过 3 时,会出现结构丢失的情况.综上,在后续实验中,我们指定 path 的长度统一取 3.

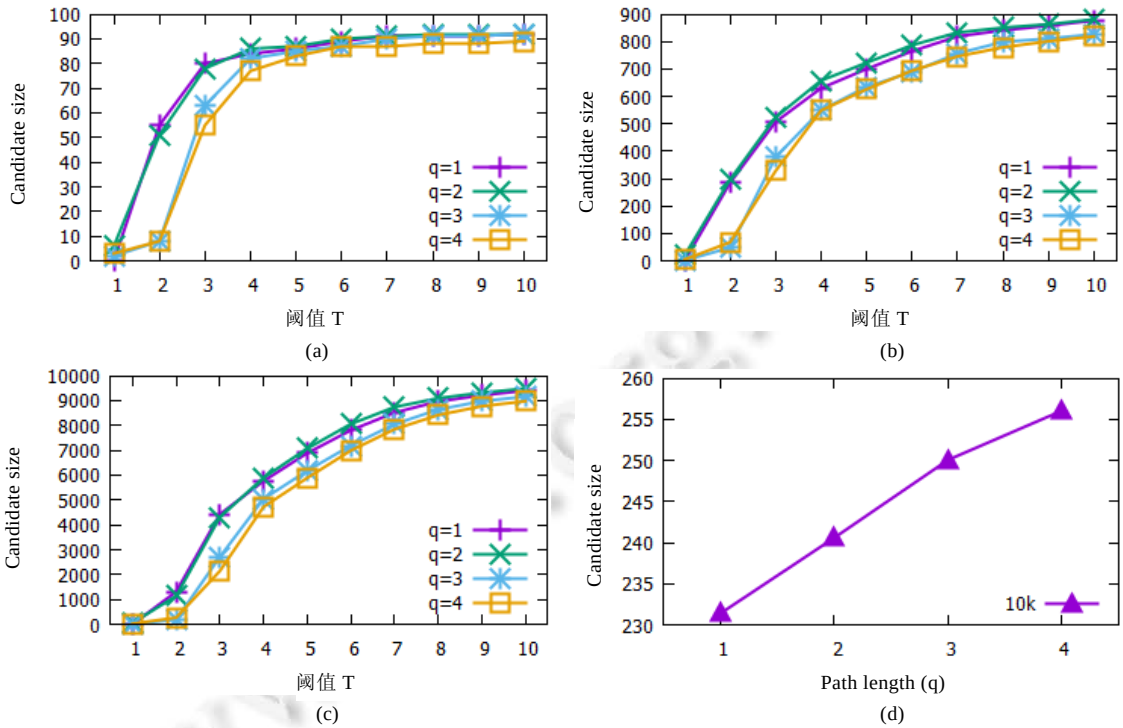


Fig.6 Path-Based filtering method analysis

图 6 基于 path 的过滤方法分析

基于 tree 的过滤方法中,需要指定生成树的深度 *k*,通过实验综合考虑过滤结果集以及响应时间两个影响因素,本文设定 *k*=1,具体过程不再赘述.

我们从 PubChem 数据集中随机抽取了 5 组 10k 的数据,来作为对 4 种过滤方法进行对比分析的实验数据.通过将 5 组实验结果取平均值的方法,得到的结果集大小随阈值变化如图 7(a)所示,响应时间如图 7(b)所示.

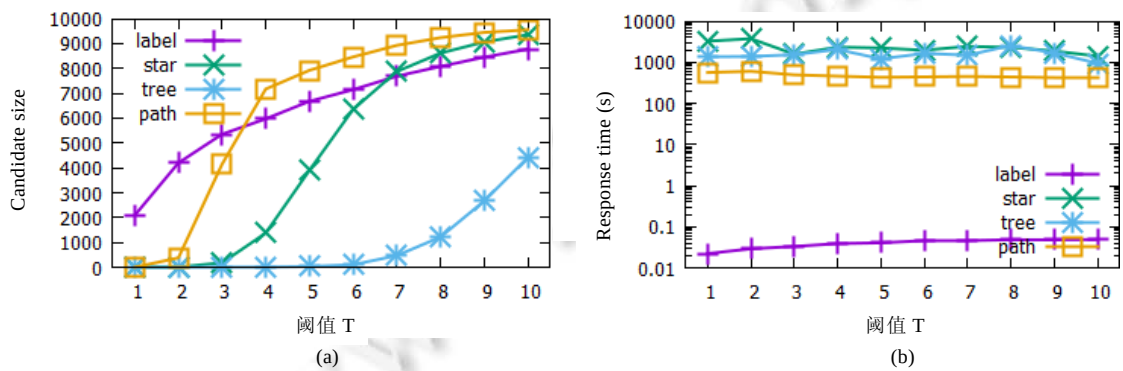


Fig.7 Four filtering methods' performance

图 7 4 种方法的过滤效果对比分析

从图 7(a)可以分析得出:4 种方法过滤得到的候选集大小都会随着阈值的增大而增大,但都会表现出不同的特性,即,随着阈值的变化,结果集最小的方法会发生改变.例如:当阈值小于 7 时,基于 label 的过滤方法较 star 方法较差;而当阈值大于 7 时,则相反.因此,很难给出一种方法在任何情况下能够优于另一种方法的结论.通过对图 7(b)所示相应时间的分析,基于 label 的过滤耗费时间相较于其他 3 种方法较少,所以作为一种初步筛选的方

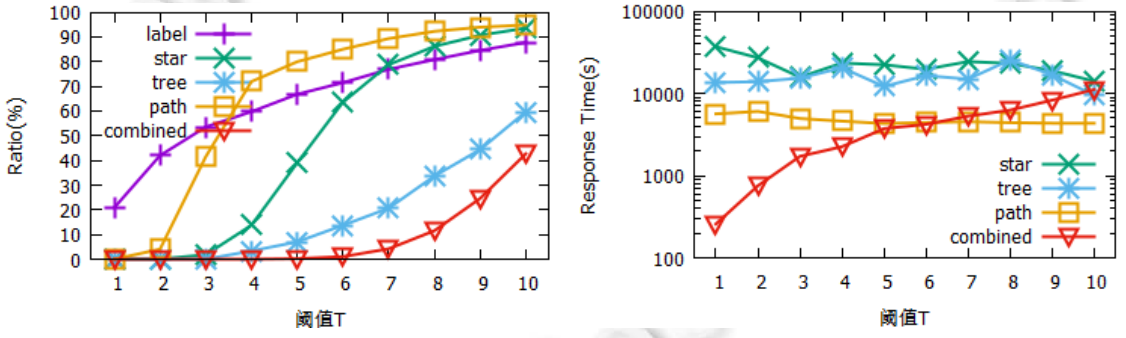
法较为合适.基于 path 的方法过滤时间较基于 star 和 path 的方法较少,且结合过滤效果来看,其在阈值较小的情况下表现较好,阈值较大的情况下表现较差,较于剩余两种方法过滤能力不够稳定,因此作为第 2 步的过滤较为合理.最后,基于 star 和基于 tree 的过滤方法都需要较长的过滤时间,根据实验结果中 star 方法过滤时间较长的结论,因此先进行基于 tree 的过滤后再进行基于 star 的过滤是效果较好的.需要说明的是:本次实验中,因为每次实验的输入数据集大小相同,都为 10k 个图,每个阈值下都会对数据集中的图进行遍历操作,而响应时间主要由数据集的大小决定,因此相应时间随阈值的变化基本维持不变.

4.2 新型图相似查询方法效能分析

在本次实验中,我们通过候选集和与总数据集合大小的比值来反映过滤能力,比值的计算见公式(8)(比值与过滤能力成反比关系,过滤能力越强,比值越小):

$$Ratio=|Cand|/|D| \tag{8}$$

我们基于 100k 数据集进行对比实验,并通过随机提取 3 组 100k 的数据集进行重复实验的方法,使得实验的结果相对准确.实验结果通过取平均值的方式给出,*Ratio* 值随阈值的变化如图 8(a)所示,平均响应时间如图 8(b)所示.在图 8(a)中可以看到:本文提出的过滤方法可以得到较小的结果集,过滤效率基本都超过 50%,分层混合过滤要优于任何一种单一过滤方法.当阈值为 10 时,分层混合过滤比单一的 label 过滤效果好 50%,比单一的基于 tree 的过滤效果好 10%.在过滤时间方面,如图 8(b)所示:由于本文层次过滤的思路,每次过滤得到的结果集都能得到很好的收敛,因此对于响应时间较长的方法,传入的候选图集合大小相对于响应时间短的方法较小,从而极大地减小了响应时间.因为基于 label 的过滤作为一种较为粗略且用时很短的过滤方法,因此在图 8(b)中将其图像省略.综上所述,本文所提出的分层过滤方法大幅地缩小了候选集,从而减小精确计算编辑距离的次数,也较好地缩短了过滤时间,因此具有较好的过滤效果,对提升图查询操作效率有一定的借鉴意义.



(a) (b)

Fig.8 Original filtering methods' performanceanalysis

图 8 新型图过滤算法效果分析

5 结束语

本文重点研究基于编辑距离的图相似查询问题.首先,通过对现有具有代表性的 4 种算法进行分析,发现现有算法存在过滤效果不稳定和适用性有限的问题;其次,针对已有方法在过滤阶段存在的问题,提出面向关系型数据库的全新过滤策略.该策略可以在过滤阶段灵活结合已有过滤规则来获取更加紧的编辑距离的上下界,从而过滤更多的无用结果,减少需要验证的候选集合的大小.本文经过调研,选取具有代表性的关系型数据库系统 PostgreSQL 来实现所提出的新策略,设计并实现了 4 种已有过滤规则,并实现不同过滤规则松耦合结合策略;最后,基于 PubChem 数据集,通过比较算法在求解查询结果的时间消耗,验证本文提出算法的高效性及可扩展性.实验结果表明,本文提出的策略优于现有方法.

References:

[1] Cai D, Shao Z, He X, Yan X, Han J. Community mining from multi-relational networks. In: Proc. of the Knowledge Discovery in Databases (PKDD 2005). Berlin: Springer-Verlag, 2005. 445−452. [doi: 10.1007/11564126_44]

[2] Yang Q, Sze S H. Path matching and graph matching in biological networks. Journal of Computational Biology, 2007,14(1):56−67. [doi: 10.1089/cmb.2006.0076]

[3] Willett P, Barnard JM, Downs GM. Chemical similarity searching. Journal of Chemical Information and Computer Sciences, 1998, 38(6):983−996. [doi: 10.1021/ci9800211]

[4] Shasha D, Wang JTL, Giugno R. Algorithmics and applications of tree and graph searching. In: Proc. of the twenty-first ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems. New York: ACM Press, 2002. 39−52. [doi: 10.1145/543613.543620]

[5] Bunke H, Allermann G. Inexact graph matching for structuralpattern recognition. Pattern Recognition Letter, 1983,1(4):245−253. [doi: 10.1016/0167-8655(83)90033-8]

[6] Sanfeliu A, Fu KS. A distance measure between attributedrelational graphs for pattern recognition. IEEE Trans. on Systems, Man, and Cybernetics, 1983,13(3):353−362. [doi: 10.1109/TSMC.1983.6313167]

[7] Zeng Z, Tung AKH, Wang J, Feng J, Zhou L. Comparingstars: On approximating graph edit distance. Proc. of the VLDB Endowment, 2009,2(1):25−36. [doi: 10.14778/1687627.1687631]

[8] Khan A, Li N, Yan X, Guan Z, Chakraborty S, Tao S. Neighborhood based fast graph search in large networks. In: Proc. of the 2011 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2011). New York: ACM Press, 2011. 901−912. [doi: 10.1145/1989323.1989418]

[9] Zhao X, Xiao C, Lin X., Wang W. Efficient graph similarity joins with edit distance constraints. In: Proc. of the 2012 IEEE 28th Int'l Conf. on Data Engineering (ICDE 2012). Washington: IEEE Computer Society, 2012. 834−845. [doi: 10.1109/ICDE.2012.91]

[10] Wang X, Ding X, Tung AKH, Ying S, Jin H. An efficient graph indexing method. In: Proc. of the 2012 IEEE 28th Int'l Conf. on Data Engineering (ICDE 2012). Washington: IEEE Computer Society, 2012. 210−221. [doi: 10.1109/ICDE.2012.28]

[11] Wang G, Wang B, Yang X, Yu G. Efficiently indexing large sparse graphs for similarity search. IEEE Trans. on Knowledge and Data Engineering, 2010,24(3):440−451. [doi: 10.1109/TKDE.2010.28]

[12] Le TH, Elghazel H, Hacid MS. A relational-based approach for aggregated search in graph databases. In: Proc. of the Database Systems for Advanced Applications (DASFAA 2012). Berlin: Springer-Verlag, 2012. 33−47. [doi: 10.1007/978-3-642-29038-1_5]

[13] Tian Y, McEachin RC, Santos C, States DJ, Patel JM. SAGA: A subgraph matching tool for biological graphs. Bioinformatics, 2007,23(2):232−239. [doi: 10.1093/bioinformatics/btl571]

[14] Tong H, Faloutsos C, Gallagher B, Eliassi-Rad T. Fastbest-Effort pattern matching in large attributed graphs. In: Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2007). New York: ACM Press, 2007. 737−746. [doi: 10.1145/1281192.1281271]

[15] Tian Y, Patel JM. Tale: A tool for approximate largegraph matching. In: Proc. of the 2008 IEEE 24th Int'l Conf. on Data Engineering (ICDE 2008). Washington: IEEE Computer Society, 2008. 963−972. [doi: 10.1109/ICDE.2008.4497505]

[16] Yan X, Yu PS, Han J. Graph indexing: A frequent structure-basedapproach. In: Proc. of the 2004 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD 2004). New York: ACM Press, 2004. 335−346. [doi: 10.1145/1007568.1007607]

[17] Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. on Systems Science & Cybernetics, 2007,4(2):100−107. [doi: 10.1109/TSSC.1968.300136]

[18] Zheng W, Zou L, Lian X, Wang D, Zhao D. Graph similarity search with edit distance constraint in large graph databases. In: Proc. of the 22nd ACM Int'l Conf. on Information & Knowledge Management (CIKM 2013). New York: ACM Press, 2013. 1595−1600. [doi: 10.1145/2505515.2505723]

[19] Kuhn HW. The Hungarian method for the assignment problem. Naval Research Logistics (NRL), 1955,2(1−2):83−97. [doi: 10.1002/nav.3800020109]

[20] Lu W, Hou J, Yan Y, Zhang M, Du Y, Thomas M. MSQL: Efficient similarity search in metric spaces using SQL. The VLDB Journal, 2017,26(6):829−854. [doi: 10.1007/s00778-017-0481-6]

赵展浩(1995－),男,浙江诸暨人,硕士生,主要研究领域为数据库,大数据管理系统.

黄斐然(1992－),男,硕士生,主要研究领域为数据库.

王晓黎(1985－),女,博士,助理教授,CCF 专业会员,主要研究领域为医疗健康大数据分析,图搜索,文本自动注解.

卢卫(1981－),男,博士,副教授,CCF 专业会员,主要研究领域为云计算与大数据管理,空间与文本数据库管理,索引技术.

杜小勇(1963－),男,博士,教授,博士生导师,CCF 会士,主要研究领域为数据库系统,智能信息检索.