一种改进的 K-Reach 索引的快速创建算法

刘凯洋

(深圳职业技术学院计算机工程学院 深圳 518055)

摘 要 基于经典网络可达性问题的 k 可达性问题对于无线网络、社交网络等新型网络具有重要意义。最新提出的 K-Reach[1] 可以快速地计算任意两个顶点之间是否存在长度小于为 k 的路径。注意到随着 K-Reach 索引的逐步建立，已经计算过的顶点包含其所有可达顶点的路径信息。因此，提出一种改进的 K-Reach 索引创建算法，其充分利用了这些路径信息，避免了重复访问大量顶点，从而提升了算法效率。通过对不同实际数据进行的实验表明，改进算法所用时间明显小于原始算法。

关键词 可达性，K-Reach 索引，优化算法

中图法分类号 TP391 **文献标识码** A

Improved Construction Algorithm for K-Reach Index

LIU Kai-yang

(Department of Computer Engineering, Shenzhen Polytechnic, Shenzhen 518055, China)

Abstract The k-rechability problem is of great importance for newly emerged wireless and social network. The latest proposed index K-Reach can quickly answer the problem of whether there exists a path of length less or equal than k between any pair of vertices. Notice that during the construction of the index, all vertices which are reachable by a computed vertex have been traversed, and the path information is recorded in the index. Therefore, an improved construction algorithm is proposed to avoid the repeated visiting of a large amount of vertices by making fully use of the path information of the computed vertices. Experiments based on a set of real-life data have demonstrated that the time complexity of the improved algorithm is significantly lower than the original construction algorithm.

Keywords Rechibility, K-Reach index, Algorithm optimization

1 概述

图论是计算机科学领域里面一个重要的分支，被广泛应用到社会生活的许多方面，如运筹学、地理信息科学和交通运输等领域。其中一个重要研究课题是可达性问题，即如何快速地计算两点之间是否可达。目前，这方面已有大量研究工作，如最新的 Path-tree[3]、3-hop[4] 等。

随着移动网络的快速扩张以及社交网络的大量涌现，一些基于可达性算法的新型查询需求得到更多的研究。例如，在一个社交网络里面，我们可能更关心从一个节点出发，经过最多 k 条边可以到达的节点，即计算两个节点之间是否存在一条长度为 k 的最短可达路径，我们称为 k 可达性问题。k 可达性问题对于社交网络具有重要含义，因为在社交网络里面，人与人之间的熟悉程度随着两者之间的距离增加而快速降低，因此一个顶点的影响辐射范围可能更局限于其最短可达路径为 k 的那些顶点。另外一个例子是在路径规划中，如自驾游服务，基于油量的考虑，一个驾驶者可能关注从源点出发，k 个距离之内的景点或者加油站、服务站等服务设施。传统意义上的可达性问题可以被认为是 k 可达性问题的一个特例：$k=\infty$。

适用于可达性问题的索引结构也能支持 k 可达性问题的求解，但是存在查询效率低、预处理时间过长等问题。最近提出的基于顶点包(Vertex Cover)[2] 的 K-Reach[1] 索引可以较好地解决 k 可达性问题。但是，对于顶点包里面的顶点，K-Reach 索引的创建需要计算所有其可达距离在 k 之内的所有顶点。这是制约 K-Reach 索引创建所需时间的最大影响因素。注意到在此计算过程中有大量的最短路径被重复计算，本文提出一种改进的 K-Reach 快速创建算法，其可以最大程度地利用已有的最短路径信息，从而大为减少 K-Reach 索引的创建时间。

2 K-Reach 索引及其相关定义

定义 1 给定 $G=(V,E)$ 是一个无权有向图，V 和 E 分别是 G 中顶点和边的集合。$(u,v)\in E$ 表示从 u 到 v 存在一条有向边。

定义 2 给定 G 的一对顶点 u 和 v，如果从 u 到 v 存在一条简单有向路径 $P=\langle s,\cdots,t\rangle$，我们称 u 到 v 可达，表示为 $u\rightarrow v$。如果路径的长度 $|P|\leqslant k$，则称 u 通过 k-hop 可达 v，表示为 $u\rightarrow_k v$。

定义 3(顶点包(Vertex Cover)和最小顶点包) 对于一个顶点集合 S，如果使得图 $G=(V,E)$ 中的每条边 $(u,v)\in E$ 都有 $\{u,v\}\cap S\neq\Phi$，我们称 S 为 G 的顶点包。如果一个顶点

刘凯洋(1978－)，男，博士，讲师，主要研究方向为图论、网络最短路径，E-mail：liukaiyang@szpt.edu.cn。

包 S 是 G 所有顶点包中的最小集合，我们称 S 为 G 的最小顶点包。

定义 4 (K-Reach)给定一个无权有向图 $G=(V,E)$、G 的一个顶点包 S 和一个整数 k，K-Reach 索引则被定义为有权有向图 $I=(V_I,E_I,\omega_I)$：

- $V_I=S$
- $E_I=\{(u,v):u,v\in S,u\rightarrow_k v\}$
- ω_I 为每条 $e=(u,v)\in E_I$ 分配一个权重，分配规则如下：

①如果 $u\rightarrow_{k-2}v$，则 $\omega_I(e)=(k-2)$；

②如果 $u\rightarrow_{k-1}v$，则 $\omega_I(e)=(k-1)$；

③如果 $u\rightarrow_k v$，则 $\omega_I(e)=k$。

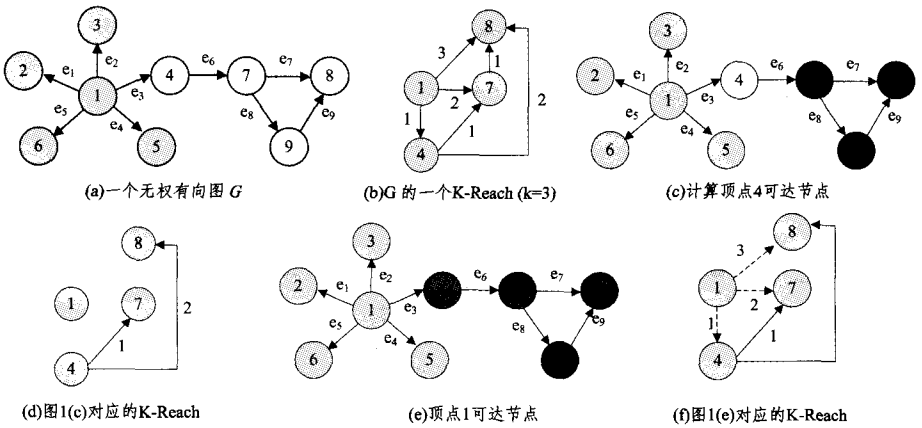

图 1

图 1(a)和图 1(b)演示了一个无权有向图及其对应的 K-Reach 索引($k=3$，即对一个顶点可以通过最多 3 条边可达的顶点进行索引)。为计算图 1(b)，首先需要计算图 1(a)对应的一个顶点包(注意，一个无权有向图可以有多个顶点包，因此其对应的 K-Reach 索引不是唯一的)。一个较快的计算顶点包的算法[1]如下：任意选定一条边，将这条边上的两个顶点加入 K-Reach 索引，并且删除与这两个顶点相关联的所有边；重复以上过程，直到所有边均被删除。假定算法首先选择边 e_3，将顶点 1 和 4 添加到索引，删除和顶点 1 和 4 相关联的所有边(即 e_1,e_2,e_3,e_4,e_5 和 e_6)。第二次选择边 e_7，将顶点 7 和 8 加入索引，并且删除 e_7,e_8 和 e_9。G 中所有边均被删除，因此 G 对应的一个顶点包为(1,4,7,8)。

需要说明的是定义 4 中的 ω_I。根据顶点包 S 的定义，对于 G 中的任意顶点对 u 和 v，只能存在以下 4 种情形：

1. $u,v\in S$；
2. $u\in S,v\notin S$；
3. $u\notin S,v\in S$；
4. $u\notin S,v\notin S$。

对于情形 2，必定存在另外一个顶点 $w,(u,v)\in G$，并且 $w\in S$。如果 u 可以通过不长于 $k-1$ 的路径到达 w，则 u 一定可以通过不长于 k 的路径到达 w(定理 1)。同理，对于情形 4，存在两个顶点 w 和 x，使得 $(u,w)\in G$ 和 $(x,v)\in G$，并且 $w,x\in S$。如果 $w\rightarrow_{k-2}x$，则 $u\rightarrow w\rightarrow x\rightarrow v$ 的路径长度不超过 k。因此，ω_I 只需要记录 $\geqslant k-2$ 的路径长度。对于 $\leqslant k-2$ 的路径长度，ω_I 统一用 $k-2$ 替代。

基于 G 的顶点包，算法 1 创建了对应的 K-Reach 索引。

算法 1 原始的 K-Reach 索引创建算法

```
1. 计算 G 的顶点集 S
2. 初始化一个有权有向图 I=(V_I,E_I,ω_I)
3. V_I←S
4. for each u∈S do
5.    使用 BFS 算法计算 S_k(u)={v:v∈S,u→_k v}
6.      for each v∈S_k(u) do
7.      E_I←(E_I∪{(u,v)}
8.      if(u→_{k-2} v)
9.        ω_I(u,v)=(k-2)
10.     else if(u→_{k-1} v)
11.       ω_I(u,v)=(k-1)
12.     else
13.       ω_I(u,v)=k
14.     end for
15. end for
```

其基本思想是对于顶点集 S 里面的每一个顶点(u,v)，计算 u 到 v 的最短路径。图 1(c)－(f)演示了算法中对应顶点 4 和顶点 1 的计算流程及其对应的 K-Reach。图 1(c)中，为计算顶点 4 可达的所有顶点，BFS 算法将遍历 G 中顶点 7，8，9，并在 K-Reach 索引中创建两条边(4,7)和(4,8)(见图 1(d))。因为顶点 9 不属于图 1(b)(即其不在 G 的顶点包中)，所以在图 1(d)中不需要创建边(4,9)。图 1(e)中 BFS 算法计算顶点 1 可达的顶点，并遍历 4,7,8,9，其最终添加的边如图 1(f)中虚线边所示。

3 改进的 K-Reach 索引创建算法

通过对比图 1(d)和图 1(e)，可以得知在已经计算顶点 4 可达顶点的情况下，在计算顶点 1 时不需要再次重新访问顶点 7，8，9。因为顶点 7，8，9 可以被顶点 4 至多通过 2 条边可达，并且顶点 1 可以通过一条边直达顶点 4，因此顶点 1 可以通过不多于 3 条边到达顶点 7，8，9。下面的定理证明了其普遍性。

定理 1 给定一个无权有向图 $G=(V,E)$和 3 个顶点 u，$v,w\in V$，如果 $u\rightarrow_k v,v\rightarrow_l w$，则 $u\rightarrow_m w,m\leqslant k+l$。

证明：假定结论不成立，即 $u\rightarrow_m v,m>k+l$。我们可以通过路径 $u\rightarrow v\rightarrow w$ 从顶点 v 到达 w，并且其路径长度为 $k+l$。证明完毕。

基于以上观察和定理 1，本文提出 BFS-skip 算法和改进的 K-Reach 索引创建算法，其基本思想为充分利用已经获取的路径信息，在进行 BFS 算法时跳过不必要访问的子树，从而提高算法性能。

算法 2 BFS-cut 算法

输入：无权有向图 $G=(V,E)$，$u\in V$，K-Reach 索引 $I=(V_I,E_I,\omega_I)$，整数 k

输出：$S_k(u)=\{v: v\in S, u\rightarrow_k v\}$

```
1. for each v∈V do
2.   设置 v 为未访问
3.   v.distance=0
4. end for
5. 初始化堆栈 stack，并且将 u 压入 stack
6. while stack 不为空 do
7.   v←top(stack)
8.   if v.distance >= k then
9.     continue
10.  end if
11.  for each w,(v,w)∈E do
12.    if w 未被访问 then
13.      w.distance=v.distance+1
14.      w→Sk(u)
15.      if w∈VI，并且状态为已计算 then
16.        skip w
17.      else
18.        将 w 压入 stack
19.      end if
20.    end if
21.  end for
22. end while
```

BFS-skip 算法与 BFS 算法的区别在于第 15 行的判断语句。如果顶点 w 已经计算过其可达顶点，算法将跳过位于 w 的子树。

算法 3 改进的 K-Reach 索引创建算法

```
1. 计算 G 的顶点集 S
2. 初始化一个有权有向图 I=(VI,EI,ωI)
3. VI←S
4. for each u∈S do
5.   使用 BFS-cut 算法计算 Sk(u)={v: v∈S, u→k v}
6.   for each v∈Sk(u) do
7.     EI←(EI∪{(u,v)}
8.     ωI(u,v)=k
9.     if v 的状态为已计算 then
10.      for each w∈Sk(v) do
11.      if ωI(u,v)+ωI(v,w)≤k then
12.        EI←(EI∪{(u,w)}
13.        ωI(u,w)=min(ωI(u,v)+ωI(v,w),ωI(u,w))
14.      end if
15.    end for
16.  end if
17.  end for
18.  设置 u 的状态为已计算
19. end for
```

假定当前计算顶点为 u，对于 $\forall v\in S_k(u)$，算法 3 添加边 (u,v) 到 E_I（第 7 行）。与定义 4 中的 ω_I 函数相比，因为改进算法需要保留所有的路径信息，以便后续计算所使用，因此算法 3 中第 8 行定义了一个不同的 ω_I 函数。

与算法 1 比较，算法 3 需要判断 v 的状态，从而决定是否需要进一步处理（第 9 行）。如果 v 的状态为已计算，则表示以 v 为出发点的所有长度$\leqslant k$ 的路径及顶点均已被找到，也即 $S_k(v)$ 已知。对于 $\forall w\in S_k(v)$，需要计算 $u\rightarrow w\rightarrow v$ 的路径长度，判断 w 是否属于 $S_k(u)$（第 11 行）。如果 $w\in S_k(u)$，算法 3 添加边 (u,w)。因为从 u 到 w 可能存在多条路径，第 13 行计算 u 到 w 的最短路径长度。

4 实验结果与分析

本文中采用斯坦福大学实验室[5]提供的真实网络数据集：Wiki-Vote 和 soc-Epinions1。Wiki-Vote 描述 Wiki 网站上的投票关系，soc-Epinions1 记录了大型社交网站 Epinions.com 的社交关系。两个数据集涵盖了中型网络和大型网络，基本数据如表 1 所列。

表 1 数据集统计信息

数据集	顶点数	边数
Wiki-Vote	7115	103689
soc-Epinions1	75879	508837

实验采取的运行环境为 Intel T7600 2.33GHz，内存为 2G。操作系统为 Debian 5.0，编译器使用 GNU C 4.7。

图 2 分别描述了两个数据集对应的 K-Reach 索引的大小。因为 K-Reach 索引的顶点数对于不同的 k 值是不变的，因此 K-Reach 索引的大小取决于边的数目（见图 2）。对比图 2 和表 1 可以看出，K-Reach 的边数远远超过原始数据集的边数。另外，注意对于较小的 k 值（$k\leqslant 5$）时，K-Reach 索引大小显著增加，因为路径长度增加，导致 $|S_k(u)|$ 增加，从而增加需要添加的边。当 k 值继续增大，超过网络中大部分路径的长度，则 $|S_k(u)|$ 保持稳定。

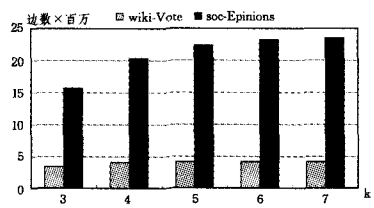

图 2 K-Reach 大小

图 3 显示，本文提出的改进算法优于原始算法，对于不同的 k 值效率提升保持在 20%左右。因为 wiki-Vote 相对数据量小，因此两种算法的运行时间对于 $k\geqslant 5$ 保持稳定，这和图 2 中的 K-Reach 索引大小变化趋势保持一致。

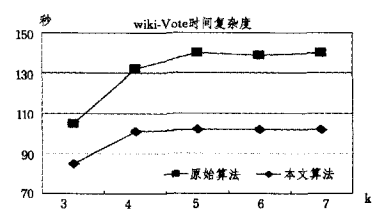

图 3 wiki-Vote 时间复杂度

soc-Epinions 数据集的大小提高了一个量级，并且路径长度也因此增加，因此两种算法的时间复杂度显著上升（见图 4）。与原始算法相比，改进算法能够较好地适应大数据量的计算要求。另外注意到随着 k 值的增加，改进算法取得了更高的效率提升（>30%）。对于较大的 k 值，由于需要访问的路径长度增加，原始算法必须重复访问更多的顶点，而本文提出的改进算法可以较好地避免这种重复访问问题，因此效率

（下转第 310 页）

与协作感知时的 ROC 曲线及性能比较。介绍了频谱感知数据篡改攻击、分类以及目前对该攻击的研究现状，针对可能发送的攻击信号的情况进行了理论分析和仿真。仿真结果表明遭受 SSDF 攻击后系统检测性能明显下降。

参 考 文 献

[1] Spectrum policy task force report [R]. Technical Report 022 135. Federal Communications Commision, Nov. 2002

[2] Akyildiz I F, Lee W-Y. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey[J]. Computer Networks, 2006, 50(13): 2127-2159

[3] 温志刚. 认知无线电频谱检测理论与实践[M]. 北京：北京邮电大学出版社，2011

[4] Akyildiz I F, Lo B F, Balakrishnan R. Cooperative spectrum sensing in cognitive radio networks: A survey[J]. Physical Communication, 2011, 4(1): 40-62

[5] Kaligineedi P, Khabbazian M, Bhargava V. Secure cooperative sensing techniques for cognitive radio systems[C]// IEEE International Conference on Communications. 2008: 3406-3410

[6] Cabric D, Brodersen R W. Physical layer design issues unique to cognitive radio system[C]// IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. 2005: 759-763

[7] 周来秀. 感知无线网络中频频检测与动态接入技术研究[D]. 长沙：中南大学，2007

[8] Edward P, Liang Ying-chang. Optimization for cooperative sensing in cognitive radio networks[C]// IEEE Wireless Communications and Networking Conference. 2007: 27-32

[9] Quan Zhi, Cui Shu-guang, Vincent Poor H, et al. Collaborative Wideband Sensing for Cognitive Radios[J]. IEEE Signal Processing Magazine, 2008, 25(6): 60-73

[10] Barkat M. Signal detection and estimation(2nd ed)[M]. Artech House Inc, 2005

[11] Advanced Television Systems Committee[Z]. ATSC Standard: Digital television standard(A/53), Revision D, Including amendment, 2005

[12] Fragkiadakis A G, Tragos E Z, Askoxylakis I G. A survey on security threats and detection techniques in cognitive radio networks[J]. IEEE Communications Surveys & Tutorials, 2013, 15(1): 428-445

[13] Chen Y. Collaborative spectrum sensing in the presence of secondary user interferences for lognormal shadowing[J]. Wireless Communications and Mobile Computing, 2012, 12(5): 463-472

[14] Shen B, Kwak K, Bai Z. Optimal linear soft fusion schemes for cooperative sensing in cognitive radio networks [C] // IEEE Global Telecommunications Conference. 2009: 1-6

[15] Meng J, Yin W, Li H, et al. Collaborative spectrum sensing from sparse observations using matrix completion for cognitive radio networks[C] // IEEE International Conference on Acoustics Speech and Signal Processing. 2010: 3114-3117

(上接第 301 页)

提升更加显著。

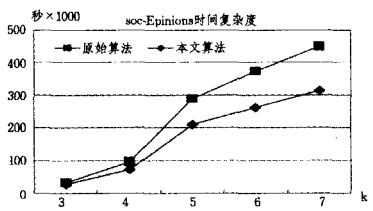

图 4 soc-Epinions 时间复杂度

实验表明本文提出的改进算法能较好地适应各种网络，并对于大、中型网络均能取得较好的运行效率提升。

结束语 针对 K-Reach 索引的创建算法中存在的重复访问路径和顶点的问题，本文提出一种改进算法，其充分利用了已计算顶点的路径信息，避免了顶点的重复访问，提高了算法效率。实验表明，针对中、大网络，改进算法均能明显地降低运行时间，并能更好地适应大型网络的索引创建。

参 考 文 献

[1] Cheng J, Shang Ze-chao, Cheng Hong, et al. K-Reach: Who is in Your Small World [C]// Proceedings' of the 38th International Conference on Very Large Databases. 2012: 1292-1303

[2] Cheng J, Ke Yi-ping, Chu Shu-mo, et al. Efficient processing of distance queries in large graphs: a vertex cover approach [C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. 2012: 457-468

[3] Jin Ruo-ming, Yang Xiang, Ruan Ning, et al. Path-Tree: An Efficient Reachability Indexing Scheme for Large Directed Graphs [J]. ACM Transactions on Database Systems, 2011(36): 1-7

[4] Jin Ruo-ming, Yang Xiang, Ruan Ning, et al. 3-HOP: a high compression indexing scheme for reachability query [C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. SIGMOD 2009: 813-826

[5] Stanford Large Network Dataset Collection[OL]. http://snap.stanford.edu/data/index.html

[6] Chen Yang-jun. An Efficient Algorithm for Answering Graph Rechability Queries [C]// Proceedings of the IEEE 24th International Conference on Data(ICDE). 2008: 893-902

[7] Cheng J, Ke Yi-ping, Fu A W-C, et al. Graph Query Processing with a Low-cost Index [J]. VLDB Journal, 2011, 20(4): 521-539

[8] Nutanonong S, Jacox E H, Samet J H. Distance-constraint Reachability Computation in Uncertain Graphs [C]// Proceedings of the 37th International Conference on Very Large Databases. 2011, 4: 506-517

[9] van Schaik S, de Moor O. A Memory Efficient Reachability Data Structure Through Bit Vector Compression [C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. 2011: 913-924

[10] Jin Ruo-ming, Ruan Ning, Dey S, et al. SCARAB: Scaling Reachability Computation on Large Graphs [C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. 2012: 169-180