Journal of Computer Applications
计算机应用,2020,40(2):510 - 517
ISSN 1001-9081
CODEN JYIIDU
2020-02-10
http://www.joca.cn

文章编号:1001-9081(2020)02-0510-08
DOI:10.11772/j.issn.1001-9081.2019091666

低冗余计算的可达性查询保持图压缩策略

赵丹枫[1],林俊辰[1],宋　巍[1],王　建[1],黄冬梅[2*]
(1.上海海洋大学 信息学院,上海 201306;　2.上海电力大学,上海 200090)
(*通信作者电子邮箱dmhuang@shou.edu.cn)

摘　要:针对可达性查询保持图压缩(QPGC)算法存在冗余计算的问题,提出了一种高性能压缩策略。在求解顶点的祖先后代集阶段,针对普通图数据,提出一种基于拓扑排序的求解算法TSB,首先将图数据顶点拓扑排序,然后沿拓扑序列顺序(逆序)求解顶点的祖先(后代)集,避免了求解顺序不明确导致的冗余计算;针对最长路径较短的图数据,提出一种基于图聚合运算的求解算法AGGB,可在确定次数的聚合运算内完成顶点的祖先和后代集的求解。在求解可达性等价类阶段,提出一种分段统计剪枝算法PSP,先对祖先后代集分段统计,再比较统计值以实现粗匹配,剪除了部分不必要的精细匹配。实验结果表明,与QPGC算法相比:在祖先后代集求解阶段,TSB和AGGB在不同数据集上的性能平均提升94.22%和90.00%;在求解可达性等价类阶段,PSP算法在大部分数据集上性能提升超过70%;随着数据集的增大,TSB和AGGB配合PSP算法,性能提升了近28倍。理论分析和模拟实验表明,该策略与QPGC算法相比冗余计算更少、压缩速度更快。

关键词:可达性查询;图压缩;查询保持;图数据;拓扑排序;聚合运算
中图分类号:TP311.12　　**文献标志码:**A

Strategy with low redundant computation for reachability query preserving graph compression

ZHAO Danfeng[1], LIN Junchen[1], SONG Wei[1], WANG Jian[1], HUANG Dongmei[2*]
(1. *College of Information*, *Shanghai Ocean University*, *Shanghai* 201306, *China*;
2. *Shanghai University of Electric Power*, *Shanghai* 200090, *China*)

Abstract: Since some computation in reachability Query Preserving Graph Compression (QPGC) algorithm are redundant, a high-performance compression strategy was proposed. In the stage of solving the vertex sets of ancestors and descendants, an algorithm named TSB (Topological Sorting Based algorithm for solving ancestor and descendant sets) was proposed for common graph data. Firstly, the vertices of the graph data were topological sorted. Then, the vertex sets were solved in the order or backward order of the topological sequence, avoiding the redundant computation caused by the ambiguous solution order. And an algorithm based on graph aggregation operation was proposed for graph data with short longest path, namely AGGB (AGGregation Based algorithm for solving ancestor and descendant sets), so the vertex sets were able to be solved in a certain number of aggregation operations. In the stage of solving reachability equivalence class, a Piecewise Statistical Pruning (PSP) algorithm was proposed. Firstly, piecewise statistics of ancestors and descendants sets were obtained and then the statistics were compared to achieve the coarse matching, and some unnecessary fine matches were pruned off. Experimental results show that compared with QPGC algorithm: in the stage of solving the vertex sets of ancestors and descendants, TSB and AGGB algorithm have the performance averagely increased by 94.22% and 90.00% respectively on different datasets; and in the stage of solving the reachability equivalence class, PSP algorithm has the performance increased by more than 70% on most datasets. With the increasing of the dataset, using TSB and AGGB cooperated with PSP has the performance improved by nearly 28 times. Theoretical analysis and simulation results show that the proposed strategy has less redundant computation and faster compression speed compared to QPGC.

Key words: reachability query; graph compression; query preserving; graph data; topological sorting; aggregation operation

0　引言

近年来,随着计算机技术和网络技术的发展,现实生活中很多领域产生了越来越多的网络数据,如语义网[1]、万维网[2]、道路网[3]、推荐网[4]、社交网[5]以及生物信息网[6]等,并且这些网络中的数据结构趋于复杂,难以用关系数据模型描述。图

收稿日期:2019-08-30;**修回日期:**2019-09-29;**录用日期:**2019-10-09。　**基金项目:**国家重点研发计划项目(2016YFC1401907)。
作者简介:赵丹枫(1982—),女,黑龙江五常人,讲师,博士,CCF会员,主要研究方向:大数据存储、业务流程管理;　林俊辰(1994—),男,吉林白山人,硕士研究生,主要研究方向:图计算;　宋巍(1977—),女,山西大同人,教授,博士,主要研究方向:计算机视觉、机器学习、数据挖掘;　王建(1979—),女,湖北武汉人,讲师,博士,主要研究方向:海洋信息技术;　黄冬梅(1964—),女,河南郑州人,教授,硕士,CCF高级会员,主要研究方向:非线性混沌系统、面向对象数据库、Web数据库、Web GIS数据库。

作为一种能够有效描述事物之间复杂关系的数据结构，在上述领域得到了广泛的研究和应用。

可达性查询是图问题研究中一种基本的查询方法，很多复杂的图查询问题通常建立在可达性查询的基础上，可达性查询一直是数据库领域研究的热点，与其相关的研究已经进行了二十多年[7-11]。随着大数据时代[12-13]的到来，图数据的规模不断扩大，对可达性查询的求解提出了新的挑战。以微信数据[14]为例，据腾讯官方统计，截至2018年9月，全球微信用户已超过10亿8千万，用户平均好友数超过130个。若采用邻接表来存储由微信用户形成的社交网络，需要超过1 TB的存储空间，在研究中难以满足这样的内存需求，因而不能将邻接表全部载入内存后对可达性查询求解。

针对图数据规模增大引发的查询难问题，主要有以下三种解决方法[15]：1）使用硬盘存储图数据，每次载入部分图数据到内存中，分步查询，最后整合结果；2）使用分布式系统存储图数据，每个节点进行部分查询，最后整合结果；3）将图数据进行压缩表示，全部载入内存中，直接查询得到结果。

以上三种解决方法各有优劣。第一种方法适用于访问局部性较好的图数据，若图数据访问局部性较差，则在查询时I/O次数会显著增多，从而导致查询代价增大；第二种方法适用于耦合性较低的图数据，若图数据耦合性较高，则会造成数据难以分割，且分割后各子图相关性较大及通信次数显著增多等问题，从而导致查询代价增大；第三种方法并不能解决所有的问题，对于某些规模特别大的图数据来说，压缩后依然不能全部载入内存中进行计算，但是，它可以和前两种方法有机结合，进一步改善前两种方法的性能。针对第一种方法，若在保持访问局部性的前提下先对图数据进行压缩，可以减少I/O次数，从而提高访问速度；针对第二种方法，若先对图数据进行压缩，则分布式系统可以利用更少的节点完成相同的任务，并且可以减少节点间的通信次数。因此，可达性查询保持图压缩是解决大规模图数据上可达性查询难问题的有效途径。

但现有的可达性查询保持图压缩方法存在冗余计算，因而计算效率较低，随着数据量的增大，时间消耗变得难以接受，所以有必要研究新的压缩策略，提高压缩速度。

本文的主要工作如下：

1）针对求解祖先后代集过程设计了一种基于拓扑排序的祖先后代集求解算法（Topological Sorting Based algorithm for solving ancestor and descendant sets，TSB），能有效地减少冗余计算，加快求解速度；还设计了一种基于图聚合运算的祖先后代集求解算法（AGGregation Based algorithm for solving ancestor and descendant sets，AGGB），对最长路径长度较小的图数据有较好效果。

2）针对求解可达性等价类过程设计了一种分段统计剪枝（Piecewise Statistical Pruning，PSP）算法，实现了可达性等价类求解过程的剪枝，减少了冗余匹配，加快了求解速度。

3）应用不同数据集进行实验，结果表明，本文提出的可达性查询保持图压缩策略是高效、快速的。

1　相关工作

对图数据进行高效、紧凑的表示与压缩是当前图研究的热点之一，同时也为图上的其他基本算法和操作提供了强有力的支持，包括存储空间优化、查询优化及可视化分析等。

1.1　图压缩技术的分类

文献[15]中将图压缩技术分为四类：基于传统存储结构的压缩技术、网页图压缩技术、社交网络图压缩技术和面向特定查询的图压缩技术。文献[16]中对此作了进一步归纳，将图压缩技术分为三类：基于图数据外在特点的压缩技术、基于图数据内在特点的压缩技术和面向特定查询需求的压缩技术。

1）基于图数据外在特点的压缩技术。图数据的外在特点是指图的表示形式或图的存储结构。传统的图数据表现形式和存储结构主要包括邻接矩阵、邻接链表、邻接多重表、十字链表等。文献[17-19]中针对图数据邻接矩阵的稀疏性对其进行压缩，文献[20]中则通过压缩处理图数据的邻接链表中存储的冗余信息来实现图压缩；但是上述这几种压缩方法均改变了图数据的存储结构，通常不利于查询。

2）基于图数据内在特点的压缩技术。图数据的内在特点是指由于应用领域的不同，图数据所具有的不同的内在性质。例如，对社交网络图而言，可以利用社交网络中的社区结构特性对图数据进行压缩[21]，具体表现为社区内相关性强，而社区间相关性弱；再如，对网页图而言，可以利用其本地性和相似性压缩图数据[22-24]；此外，在可视化分析领域，文献[25-27]中针对数据图稠密的特性，开发了相应的稠密有向图压缩算法。

3）面向特定查询需求的压缩技术。查询是图数据研究中的基本操作，一直是图领域中的研究热点。常用的图上查询有可达性查询、内外邻居查询和图模式匹配等。为了达到更低的压缩比[28]，将不可避免地引起一些信息损失。因此，对面向特定查询的压缩图来说，压缩图与原图是一种基于特定查询的等价图[29]。

1.2　可达性查询保持的图压缩技术

可达性查询保持的图压缩是Fan等[30]提出的概念，是一种面向可达性查询的压缩技术，这种压缩技术追求极低的压缩比，与此同时，保证了压缩图对可达性查询保持结果不变，即其压缩图与原图针对可达性查询是等价的。文献[30]提出的可达性查询保持图压缩（Query Preserving Graph Compression，QPGC）算法，使用广度优先遍历（Breadth-First Search，BFS）[31]的方法计算原图中每个顶点的祖先顶点集和后代顶点集，由于具有相同祖先和后代的顶点关于可达性查询是等价的，因此遍历各顶点可确定可达性等价关系。文献[30]同时提出了一种优化策略：先计算原图中的强连通分量，将其压缩为一个虚拟顶点，得到一个有向无环图（Directed Acyclic Graph，DAG）[32]，将此DAG作为算法的输入，最终获得可达性查询保持图。该算法随着图数据规模的增大，表现逐渐不佳，耗时显著增大。因为QPGC算法通过BFS方法计算各顶点的祖先顶点集和后代顶点集的过程中存在冗余计算，而且BFS结果集比较庞大，导致可达性等价类的计算存在一定局限性。文献[16]在QPGC算法的基础上，结合MapReduce[33]的思想，提出了适合于MapReduce的可达性查询保持图计算方法，但是并没有提高计算速度。文献[34]中提出了一种基于BFS结果集的可达性保持图并行计算方法，但是该方法的主体为基于BFS结果集的SCC查找算法，仅考虑压缩原图中存在的强连通分量（Strongly Connected Components，SCC），不能达到QPGC算法的压缩比。

在本文提出的低冗余计算低的可达性查询保持图压缩策

略的压缩过程中，着重考虑避免进行冗余计算，提高计算效率，以实现求解过程的加速，显著提高压缩速度。

2 本文方法

在求解祖先后代集过程和求解可达性等价类过程中，本文分别设计了高效、低冗余的求解算法。

2.1 高效的祖先后代集求解方法

在提出本文方法之前，首先对现有QPGC算法求解祖先集和后代集过程中存在的问题进行分析。

QPGC算法以每个顶点为初始顶点，使用向前或向后的BFS方法，分别求出其祖先顶点集和后代顶点集，要进行$2|V|$次BFS遍历。事实上，每个顶点的祖先（后代）集与其前驱（后继）顶点的祖先（后代）集具有一定的联系，算法在不同的遍历顺序下执行时间大不相同，下面举例说明。

例1 考虑图1所示的示例，使用BFS方法求解各顶点的后代集，若遍历顺序为E、A、B、C、D，则：在计算顶点E的后代时，进行了4次访问，后代集为$\{B, C, D\}$；计算顶点A的后代时，进行了4次访问，后代集为$\{B, C, D\}$；计算顶点B的后代时，进行了3次访问，后代集为$\{C, D\}$；计算顶点C的后代时，进行了2次访问，后代集为$\{D\}$；计算顶点D的后代时，进行了1次访问。共进行了14次访问。然而，若按D、C、B、E、A的顺序求解，则只需进行9次访问。这种冗余随着图的规模增大，将会进一步增大。

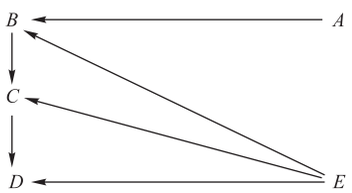


图1 QPGC算法例图

Fig. 1 Example of QPGC algorithm

针对现有方法由冗余计算引起的求解祖先后代集速度慢的问题，本文提出了两种算法TSB和AGGB，分别适用于一般的图数据和最长路径长度较小的图数据，下面分别说明。

2.1.1 基于拓扑排序的祖先后代集求解算法TSB

在描述算法前，首先介绍拓扑排序的定义及拓扑序列具有的性质。

定义1 拓扑排序是定义在DAG上的一种排序，其将图$G = (V, E)$中所有顶点排成一个线性序列，使得对图中任意一对顶点u和v，若$(u, v) \in E$，则u在该线性序列中出现在v之前。

引理1 沿拓扑序列顺序（逆序）求解各顶点的祖先集（后代集），每个顶点与边需要且仅需要访问1次。

证明

以沿拓扑序列顺序求解祖先集为例：

对拓扑序列中任意顶点v，其父顶点集中任意顶点u，满足$(u, v) \in E$，由拓扑序列的性质，u在拓扑序列中出现在v之前，而v的祖先集$A_v = \bigcup_{u \in \{u \mid (u,v) \in E\}} A_u \cup \{u\}$，其中任意$A_u$均已完成祖先集的求解。

因此，求解v的祖先集，仅需访问v的每个父顶点各1次。

综上所述，沿拓扑序列顺序求解各顶点的祖先集，每个顶点与边需要且仅需要访问1次。类似地，可以证明沿拓扑序列逆序求解各顶点的后代集，每个顶点与边需要且仅需要访问1次。

本文提出的基于拓扑排序的祖先后代集求解算法TSB如算法1所示。

首先，将原图转化为DAG。在实际应用中，若图数据不是DAG，则不能进行拓扑排序，因此要对其中的环（也即SCC）进行特殊处理，将图数据转化为DAG。Gabow算法[35]是求解SCC的经典算法，本文使用Gabow算法求解图中的SCC。

然后将每个SCC用一个单顶点代替，此时图数据变为DAG。

接下来应用BFS方法对DAG进行拓扑排序，得到拓扑序列。过程简述如下：取一未访问过的入度为0的顶点，该顶点必为DAG中某一棵树（连通分量）的根顶点，应用BFS方法遍历此树；重复这一步骤，直到不存在未访问过的顶点。顶点的遍历顺序即为拓扑序列。

最后，沿拓扑序列顺序求解各顶点的祖先集，沿拓扑序列逆序求解各顶点的后代集。

TSB和QPGC算法均使用了BFS方法对图数据进行遍历，区别在于：QPGC算法由每个顶点为初始点，应用BFS方法遍历至最底层或最顶层，从而求得其祖先或后代顶点集；而TSB取根节点为初始点，使用一次BFS遍历求得DAG的拓扑序列，沿拓扑序列顺序或逆序求解各顶点的祖先后代顶点集，当求解某顶点时，仅需访问该顶点的邻接顶点即可完成求解。

算法1 TSB。

输入 图$G = (V, E)$；

输出 祖先顶点集A_G和后代顶点集D_G。

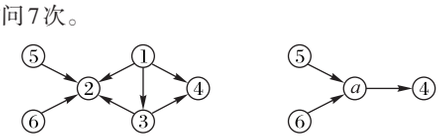
1）使用Gabow算法求解SCC；

2）压缩每个SCC为一个单顶点，得到与原图对应的DAG；

3）应用BFS方法对DAG拓扑排序，得到拓扑序列；

4）沿拓扑序列顺序（逆序）求解祖先集（后代集）。

例2 考虑图2（a）所示的例图，首先使用Gabow算法求得SCC为$\{5\}$，$\{6\}$，$\{1, 2, 3\}$，$\{4\}$；压缩SCC得到DAG如图2（b）；应用BFS方法对DAG拓扑排序，得到拓扑序列5，6，a，4（不唯一）；沿拓扑序顺序求解祖先顶点集，计算5的祖先集为$\varnothing$，访问1次，计算6的祖先集为$\varnothing$，访问1次，计算a的祖先集为$\{5, 6, a\}$，访问3次，计算4的祖先集为$\{5, 6, a\}$，访问2次，共访问7次；沿拓扑序逆序求解后代顶点集，计算4的后代集为$\varnothing$，访问1次，计算a的后代集为$\{a, 4\}$，访问2次，计算6的后代集为$\{a, 4\}$，访问2次，计算5的后代集为$\{a, 4\}$，访问2次，共访问7次。



(a) 原始图 (b) DAG

图2 TSB例图

Fig. 2 Example of TSB algorithm

2.1.2 基于图聚合运算的祖先后代集求解算法

基于图聚合运算的祖先后代集求解算法AGGB如算法2所示。

首先初始化各个顶点的祖先集和后代集为空集；接下来每个顶点分别向其父（子）顶点发送其后代（祖先）集；每个顶点接收到消息后，更新其自身的后代（祖先）集为原后代（祖先）集和其接收到的后代（祖先）集以及其子（父）顶点集的并

集；重复发送—接收—更新的过程，直至所有顶点的后代集和祖先集均不再发生变化，算法结束。

AGGB的聚合运算次数与图数据最长路径长度具有正相关关系，因此适用于最长路径长度较小的图数据。

算法2　AGGB。

输入　图$G=(V,E)$；

输出　祖先顶点集A_G和后代顶点集D_G。

1）初始化每个顶点的祖先集和后代集为空集；

2）每个顶点向其父顶点和子顶点分别发送其后代集和祖先集；

3）每个顶点接收其父顶点和子顶点发送的后代集和祖先集，更新自身的祖先集和后代集；

4）重复步骤2）、3），直到每个顶点的祖先集和后代集均不再发生变化。

例3　仍考虑图2(a)所示的例图。应用AGGB：第1次聚合运算得到各顶点的祖先集（括号内，下同）为5(∅)，6(∅)，2({3,5,6})，1({2})，3({1})，4({1,3})，后代集（括号内，下同）为5({2})，6({2})，2({1})，1({3,4})，3({2,4})，4(∅)；第2次聚合运算得到的祖先集为5(∅)，6(∅)，2({1,3,5,6})，1({2,3,5,6})，3({1,2})，4({1,2,3})，后代集为5({1,2})，6({1,2})，2({1,3,4})，1({2,3,4})，3({1,2,4})，4(∅)；第3次聚合运算得到的祖先集为5(∅)，6(∅)，2({1,2,3,5,6})，1({1,2,3,5,6})，3({1,2,3,5,6})，4({1,2,3,5,6})，后代集为5({1,2,3,4})，6({1,2,3,4})，2({1,2,3,4})，1({1,2,3,4})，3({1,2,3,4})，4(∅)；第4次聚合运算得到的祖先集为5(∅)，6(∅)，2({1,2,3,5,6})，1({1,2,3,5,6})，3(1,2,3,5,6)，4({1,2,3,5,6})，后代集为5({1,2,3,4})，6({1,2,3,4})，2({1,2,3,4})，1({1,2,3,4})，3({1,2,3,4})，4(∅)。祖先顶点集和后代顶点集均不再发生变化，求解结束，共进行了4次聚合运算。

2.1.3　算法性能对比分析

1）时间复杂度。

QPGC：对使用邻接表存储的图数据，使用BFS方法将顶点遍历一次的时间复杂度为$O(|V|+|E|)$，而QPGC算法求解祖先后代集时需对每个顶点分别进行一次向前（向后）的BFS遍历，从而获得当前顶点的祖先（后代）集，因此时间复杂度为$O(|V|\cdot(|V|+|E|))$。

TSB：使用Gabow算法求解SCC时，对每个顶点和边访问1次，时间复杂度为$O(|V|+|E|)$。压缩SCC为单顶点的时间复杂度为$O(n)$，其中n为SCC的个数（$n\leqslant|V|$）。使用BFS方法对顶点拓扑排序时，对每个顶点和边访问1次，时间复杂度为$O(|V|+|E|)$。沿拓扑序列顺序求解祖先集时，对每个顶点和边访问1次，求解后代集亦然，时间复杂度为$O(|V|+|E|)$。因此TSB的时间复杂度为$O(|V|+|E|)$。

AGGB：算法需要最多不超过图数据最长路径长度次数的聚合运算，每次聚合运算访问顶点和边各两次，时间复杂度为$O(|V|+|E|)$，因此，AGGB的时间复杂度为$O(|V|+|E|)$。

2）空间复杂度。

QPGC：算法使用BFS方法遍历图数据，需要使用一个队列，空间复杂度为$O(|V|)$。

TSB：算法使用Gabow算法求解SCC时，需使用两个栈，此步骤空间复杂度为$O(|V|)$；算法求得的DAG需使用邻接表存储，此步骤空间复杂度为$O(|V|+|E|)$；算法使用BFS方法求拓扑序列，需使用一个队列，此步骤空间复杂度为$O(|V|)$；算法存储拓扑序列需使用一个数组，此步骤空间复杂度为$O(|V|)$。因此，TSB空间复杂度为$O(|V|+|E|)$。

AGGB：算法可直接在祖先后代结果集上进行操作，因此空间复杂度为$O(1)$。

分析三种算法的时间复杂度和空间复杂度，可以得出以下结论：与QPGC算法相比，TSB和AGGB均将时间复杂度由$O(|V|\cdot(|V|+|E|))$降至$O(|V|+|E|)$，其中TSB以牺牲部分空间为代价，将空间复杂度由$O(|V|)$提高到$O(|V|+|E|)$，而AGGB能够进一步将空间复杂度降至$O(1)$。AGGB的特性决定其仅适用于最长路径较短的图数据，而TSB的普适性更强。

2.2　分段统计剪枝的可达性等价类求解算法

针对现有算法由BFS结果集（即顶点祖先后代结果集）计算可达性等价类效率低的问题，本文提出了一种分段统计剪枝（PSP）算法（见算法3），可以有效提高可达性等价类的求解速度。首先，确定分段大小S，每S个顶点划分为一个分段；接着对每一个顶点，分别统计其祖先集和后代集落在对应分段内的顶点个数。这个过程可以在计算祖先和后代集的同时进行，每求得当前顶点的一个祖先顶点，就为这一祖先顶点所在的分段的统计值加1，这一过程每次要计算一次除法（计算所在分段）和一次加法。接下来，对每一个顶点对，首先判断二者的统计值是否相同，若不相同，则二者祖先集和后代集必然不同，因此不属于同一可达性等价类，无需进行精细比较，从而达到快速剪枝的效果；若相同，则二者存在祖先集和后代集相同的可能，进一步精细比较二者的祖先集和后代集，若二者具有相同的祖先和后代，则属于同一可达性等价类，否则不属于同一可达性等价类。最后，将所有的可达性等价类构成的集合输出。

算法3　PSP算法。

输入　祖先集和后代集；

输出　可达性等价类集。

1）确定分段大小S；

2）遍历顶点，对每个顶点分别统计其祖先集和后代集在每个分段内的个数；

3）遍历顶点对，若顶点对的统计值相同，则进一步比较顶点对的祖先集和后代集是否相同，否则无需进一步比较；

4）具有相同祖先集和后代集的顶点构成一个可达性等价类，不存在相同祖先集和后代集的顶点单独作为一个可达性等价类。

例4　有顶点A、B、…、I，设A的祖先集为$\{C,D,E,G,H\}$，B的祖先集为$\{C,D,G,H,I\}$，为比较A、B的祖先集，传统方法需对两个集合取交集得$\{C,D,G,H\}$，其交集元素个数小于原集合元素个数，因此不等价。PSP算法考虑将顶点进行分段，以分段大小$S=3$为例，此时各分段的顶点为$\{A,B,C\}$、$\{D,E,F\}$、$\{G,H,I\}$，对A的祖先集统计结果为[1,2,2]，B的统计结果为[1,1,3]，对A、B的统计结果进行比较，若不同，可以断言A、B的祖先集不同，无需进行精细比较；若A的祖先集不变，B的祖先集改为$\{C,D,F,G,I\}$，则B的统计结果变为[1,2,2]，A、B的统计结果一致，需采用传统方法进一步进行精细

比较。

在顶点对的比较过程中，QPGC 算法每次都进行精细比较，设其用时为 $f(|V|)$，由于 PSP 算法先对长度为 $L=|V|/S$ 的统计值进行比较，如果统计值相同，再进行精细比较，因此，其用时可表示为 $f(L+p\cdot|V|)$，其中 p 为进行精细比较的概率，则 PSP 算法与 QPGC 算法的用时比为 $\frac{f(L+p\cdot|V|)}{f(|V|)}$，若 f 为线性函数，则用时比为 $\frac{1}{|S|}+p$。

3 实验结果与分析

3.1 实验数据集与环境

本文使用了 6 组实验数据集，数据集名称、大小、来源及在文中的简称如表 1 所示；实验环境配置如表 2 所示。

表 1 实验数据集

Tab. 1 Experimental datasets

数据集	顶点数(|V|)	边数(|E|)	数据来源文献	在文中的缩写
Youtube_0	3 356	3 615	[36]	Y0
Web-California	6 175	16 150	[37]	WebC
Wiki-Vote	7 115	103 689	[37]	WikiV
Youtube_1	26 845	60 603	[36]	Y1
Cit-HepTh	27 769	352 768	[37]	CHT
Cit-HepPh	34 546	421 534	[37]	CHP

表 2 实验环境配置表

Tab. 2 Configuration of experimental environment

软、硬件环境	配置
操作系统	Ubuntu Server 18.04
CPU	Intel Xeon Gold 6130 CPU @ 2.10 GHz
内核数	2
内存	256 GB
硬盘	1 TB
Spark	2.4.3
Hadoop	2.7
Python	3.7.0
Worker 个数	4

3.2 实验结果与分析

进行 2 组实验来验证本文提出的可达性查询保持图压缩策略的可行性和高效性。第 1 组实验对改进的祖先后代集求解算法 TSB 和 AGGB 与原始算法 QPGC 的性能进行了对比；第 2 组实验对改进的可达性等价类求解算法 PSP 和原始算法 QPGC 的性能进行了对比，并对分段大小 S 与 PSP 算法性能的关系进行了实验分析。接下来，实验分析了 PSP 算法由分段统计引发的额外存储空间的消耗。最后，对比了 QPGC 算法和本文算法的整体耗时。为保证实验结果的稳定性，本文所有实验结果均为 5 次实验取平均值。

本文方法与 QPGC 算法采用相同的压缩理念，根据可达性等价顶点集具有相同的可达性关系，对其进行压缩，得到的压缩图与原图是可达性等价的，压缩图中可达性查询的结果就是原图上的结果。本文提出的压缩策略与 QPGC 算法得到的压缩结果完全一致，压缩效果如表 3 所示。文献[30]第 6 章中对不同数据集应用可达性查询保持图压缩算法达到的效果进行了较为详细的对比和分析，本文不再赘述。

表 3 本文算法的压缩率

Tab. 3 Compression ratio of the proposed algorithm

数据集	顶点数(|V|)	边数(|E|)	压缩图顶点数(|Vr|)	压缩图边数(|Er|)	压缩率/%
Y0	3 356	3 615	350	248	8.58
WebC	6 175	16 150	3 802	11 826	70.00
WikiV	7 115	103 689	1 016	2 666	3.32
Y1	26 845	60 603	3 717	7 331	12.63
CHT	27 769	352 768	18 821	127 280	38.39
CHP	34 546	421 534	18 683	109 583	28.12

3.2.1 祖先后代集求解算法性能对比实验

本文算法相对 QPGC 算法的加速比如表 4 所示。从表中可以看出，与 QPGC 算法相比，本文提出的两种算法执行时间显著减少，其中 TSB 平均加速 94.22%，AGGB 平均加速 90.00%。

TSB 和 AGGB 间互有优劣。在数据集 Y0 上，AGGB 执行速度较 TSB 快；在数据集 WebC 和 WikiV 上，TSB 执行速度较 AGGB 略快；在数据集 CHT 和 CHP 上，AGGB 较 TSB 执行速度慢近 50%，这是因为这两个数据集为引文数据集，其数据图中路径较长，聚合运算次数较多。

表 4 TSB、AGGB 与 QPGC 三种算法的性能对比

Tab. 4 Performance comparison of TSB, AGGB and QPGC

数据集	运行时间/s			加速比/%	
	QPGC	TSB	AGGB	TSB	AGGB
Y0	1.34	0.06	0.05	95.28	96.41
WebC	4.67	0.39	0.54	91.60	88.37
WikiV	24.92	1.95	2.25	92.17	90.98
Y1	74.53	1.20	4.03	98.39	94.60
CHT	309.85	15.40	52.23	95.03	83.14
CHP	705.15	50.41	95.33	92.85	86.48

TSB、AGGB 和 QPGC 算法的执行时间随数据集规模变化的趋势如图 3 所示。从图中可以看出，随着数据集规模的扩大，三种算法求解祖先集与后代集过程的执行时间均呈现上升趋势，其中 QPGC 算法受数据集规模影响较大，上升速度较快，TSB 和 AGGB 受数据集规模影响较小，上升速度缓慢。

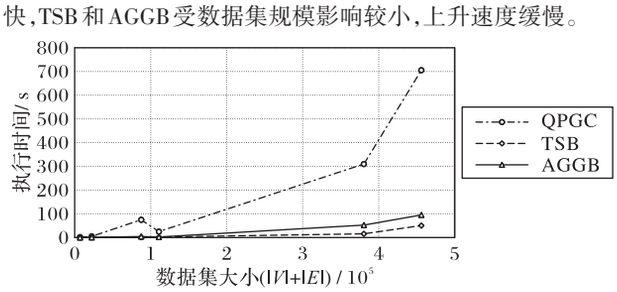


图 3 三种算法执行时间随数据集规模变化趋势

Fig. 3 Trend of the execution time of three algorithms with increasing dataset

最后对 TSB、AGGB 和 QPGC 算法求解祖先后代集的存储开销进行对比。由于 3 种算法的输入输出一致，仅考虑中间变量的存储占用情况，如表 5 所示。从表中可以看出，与 QPGC 算法相比，TSB 中间变量内存使用明显增大，但是其值在可以接受的范围内(KB 级)，而 AGGB 仅使用固定的中间变量，内存占用较小且不变。因此，本文方法在大数据上较 QPGC 算法有更优异的表现。

表5　三种算法中间变量的存储占用对比

Tab. 5　Comparison of memory usage of intermediate variables of three algorithms

数据集	中间变量占用的存储空间大小		
	QPGC/KB	TSB/KB	AGGB/B
Y0	10.27	21.73	6
WebC	18.01	36.53	6
WikiV	19.85	38.48	6
Y1	79.98	165.83	6
CHT	76.00	144.61	6
CHP	89.24	158.67	6

3.2.2　可达性等价类求解算法性能对比实验

首先对PSP算法在不同分段大小S取值下相对QPGC算法的加速效果进行对比。

取分段大小S为5、10、50、100、500、1 000、2 000,PSP算法相对QPGC算法的加速比如图4所示。可以看出,不同的S取值下,PSP算法较QPGC算法计算速度均有明显提升,显著提高了可达性等价类的计算速度,除Y0数据集外,加速效果均在70%以上,其中对数据集WebC、CHT和CHP,加速效果达到了90%以上。

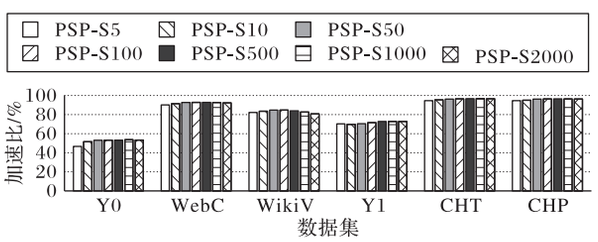


图4　PSP算法相对QPGC算法的加速比

Fig. 4　Speedup of PSP algorithm compared to QPGC algorithm

接下来,对PSP算法在不同数据集上较QPGC算法的加速效果进行分析。

通过分析对比,PSP算法与QPGC算法的用时比(用时比与加速比的和为1)与数据集压缩后得到的等价压缩图的点边比间的关系如图5所示,可以看出,两种算法的用时比与压缩图的点边比呈正相关关系,因此,精细比较的概率p与压缩图的点边比呈正相关关系。

此外,分段大小S也会影响p的取值,随着S取值的变化,p也会发生变化,从而导致PSP算法较QPGC算法的加速比发生变化。从图4中可以看出,当S取值为100时,加速比基本达到最高值,此后,WikiV数据集上加速比开始下降,其他数据集上加速比有极小幅度的上升。鉴于此,本文对S取值在[5,200]区间内加速比的变化做了进一步研究,结果如图6所示。从图中可以看出,当S取值很小时,加速比较小;S在[50,100]区间内,加速比基本达到最高值;此后,加速比有一定的波动,并逐渐开始下降。这是因为,如果S取值过小,分段数将增大,粗粒度比较的计算成本升高;如果S取值过大,将有更多的顶点被误分为备选可达性等价顶点集,从而导致精细比较次数增多,提高了计算成本。另外,加速比的变化不是平滑的,具有一定的波动性,这与分段统计过程中,分段大小、编号顺序、祖先后代分布均有一定的关系。

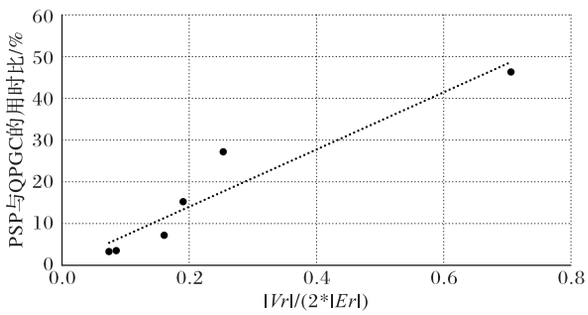


图5　用时比与数据集压缩后得到的等价压缩图的点边比间的关系

Fig. 5　Relationship between time cost ratio and point-edge ratio of equivalent compression graph obtained after data set compression

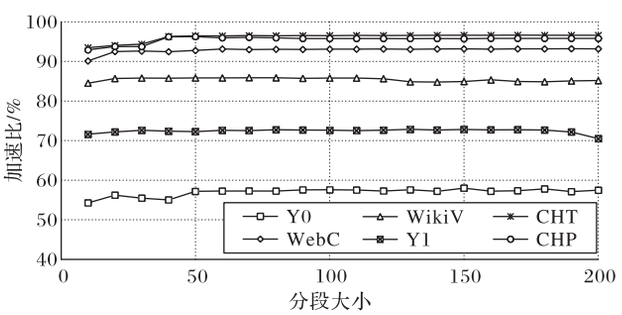


图6　PSP算法相对QPGC算法加速比与分段大小S的关系

Fig. 6　Relationship between speedup of PSP algorithm compared to QPGC algorithm and segment size S

另一方面,分段统计带来了额外的存储开销,且空间占用与S的大小有关,如表6所示。可以看出,当S取值较小时,存储空间占用较大,且随着数据集的增大,存储空间也在增大。当S取1 000时,CHT和CHP数据集上存储空间占用不足1 MB,是当前硬件条件下可以接受的。

表6　不同分段大小S下的存储空间占用

Tab. 6　Memory usage under different segment size S

S	不同数据集的存储空间占用大小/KB					
	Y0	WebC	WikiV	Y1	CHT	CHP
2	2 749.71	9 310.76	12 360.94	175 947.49	188 267.87	291 363.81
50	111.44	373.89	496.81	7 038.96	7 538.86	11 655.92
100	55.73	186.95	250.15	3 526.04	3 769.44	5 836.40
200	27.87	93.48	125.08	1 769.58	1 884.73	2 918.21
500	11.48	39.21	52.13	707.84	759.32	1 180.79
1 000	6.57	21.12	27.81	353.93	379.67	590.40

综上所述,S取值过小,存储空间占用会显著增大,S取值过大,精细比较的概率p随之增大,因此,分段大小S取值应适中,并结合具体应用环境进行分析,达成时间和空间的协调。例如,在应用中,若存储空间较为充足,则可适当的减小S,以空间换时间,否则应适当增大S。

接下来,对两个阶段的执行时间占比进行统计分析。以

Y0和WikiV数据集为例，取分段大小S = 100，求解祖先后代集阶段（第1阶段）和求解可达性等价类阶段（第2阶段）的占比如图7所示。从图中可以看出，第2阶段用时所占比重更大。

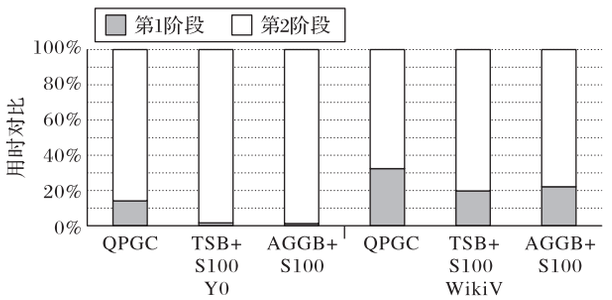


图7 两阶段执行时间占比

Fig. 7 Proportion of execution time in two stages

最后，在6组数据集上分别应用TSB+S100、AGGB+S100和QPGC算法，总体用时对比如表7所示。从表中可以看出，本文提出的TSB和AGGB配合PSP算法，时间性能提升显著，随着数据量的增大，性能提升近28倍。

表7 总体用时对比

Tab. 7 Total time cost comparison

数据集	压缩过程两阶段总体用时/s		
	QPGC	TSB+S100	AGGB+S100
Y0	9.48	3.87	3.86
WebC	500.97	36.17	36.32
WikiV	76.83	9.87	10.16
Y1	10 465.31	2 954.20	2 957.03
CHT	41 377.63	1 469.68	1 506.51
CHP	50 557.20	1 824.11	1 869.03

4 结语

本文提出了一种低冗余计算的可达性查询保持图压缩策略。其中：TSB通过拓扑排序确定祖先后代集求解顺序，避免了重复计算；AGGB可在不超过最长路径长度次数的聚合运算内完成求解，对最长路径长度较小的数据图有较好的效果；PSP算法预分段统计BFS结果集，利用统计值对匹配过程剪枝，剪除了大量的不必要匹配。实验结果表明，本文提出的可达性查询保持图压缩策略，较现有方法显著减少了冗余计算，在不同的数据集上均取得了较好的加速效果。因此，本文策略提高了压缩速度，冗余计算少，是可行且高效的。

下一步，我们将研究大规模图数据可达性查询的应答策略，从查询求解的角度减少应答时间，提高应答速度。

参考文献（References）

[1] 柴瑜晗. 基于语义图的中文领域概念及关系抽取方法研究与实现[D]. 石家庄：河北科技大学，2019：17.（CHAI Y H. Research and implementation on the method of Chinese domain concept and relation extraction based on semantic graph [D]. Shijiazhuang: Hebei University of Science and Technology, 2019: 17.）

[2] LI F, ZHANG Q, GU T, et al. Optimal representation for Web and social network graphs based on K-2-tree[J]. IEEE Access, 2019, 7: 52945-52954.

[3] 马超，孙群，陈换新，等. 加权网页排序算法在道路网自动选取中的应用[J]. 武汉大学学报（信息科学版），2018，43(8)：1159-1165.（MA C, SUN Q, CHEN H X, et al. Application of weighted PageRank algorithm in road network auto-selection[J]. Geomatics and Information Science of Wuhan University, 2018, 43(8): 1159-1165.）

[4] 王婧璇，闫强. 移动应用商店中的平台推荐网络分析[J]. 北京邮电大学学报（社会科学版），2014，16(1)：60-64，72.（WANG J X, YAN Q. Platform recommendation network of mobile application store[J]. Journal of Beijing University of Posts and Telecommunications (Social Sciences Edition), 2014, 16(1): 60-64, 72.）

[5] 谢婧璘. 面向Web社会网络的分析工具[D]. 上海：复旦大学，2011：6-8.（XIE J L. Analysis tool for Web social network[D]. Shanghai: Fudan University, 2011: 6-8.）

[6] 许睿，李琳芳，白林峰. 基于节点关联性的关键蛋白质识别算法研究[J]. 河南科技学院学报（自然科学版），2016，44(2)：68-72，78.（XU R, LI L F, BAI L F. Identification of essential proteins based on network node correlation[J]. Journal of Henan Institute of Science and Technology (Natural Science Edition), 2016, 44(2): 68-72, 78.）

[7] AGRAWAL R, BORGIDA A, JAGADISH H V. Efficient management of transitive relationships in large data and knowledge bases [J]. ACM SIGMOD Record, 1989, 18(2): 253-262.

[8] CHEN Y. On the evaluation of large and sparse graph reachability queries[C]// Proceedings of the 2008 International Conference on Database and Expert Systems Applications, LNCS 5181. Berlin: Springer, 2008: 97-105.

[9] FAN W, LI J, MA S, et al. Adding regular expressions to graph reachability and pattern queries[C]// Proceedings of the IEEE 27th International Conference on Data Engineering. Piscataway: IEEE, 2011: 39-50.

[10] YUNG D, CHANG S K. Fast reachability query computation on big attributed graphs[C]// Proceedings of the 2016 IEEE International Conference on Big Data. Piscataway: IEEE, 2016: 3370-3380.

[11] ZHANG K, LI K. The optimization reachability query of large scale multi-attribute constraints directed graph[J]. Computer Systems Science and Engineering, 2018, 33(2): 71-85.

[12] LABRINIDIS A, JAGADISH H V. Challenges and opportunities with big data[J]. Proceedings of the VLDB Endowment, 2012, 5(12): 2032-2033.

[13] JAGADISH H V, GEHRKE J, LABRINIDIS A, et al. Big data and its technical challenges[J]. Communications of the ACM, 2014, 57(7): 86-94.

[14] 微信. 2018微信数据报告[EB/OL]. [2019-06-01]. https://support.weixin.qq.com/cgi-bin/mmsupport-bin/getopendays.（WeChat. Wechat Data Report of 2018[EB/OL]. [2019-06-01]. https://support.weixin.qq.com/cgi-bin/mmsupport-bin/getopendays.

[15] 张宇，刘燕兵，熊刚，等. 图数据表示与压缩技术综述[J]. 软件学报，2014，25(9)：1937-1952.（ZHANG Y, LIU Y B, XIONG G, et al. Survey on succinct representation of graph data[J]. Journal of Software, 2014, 25(9): 1937-1952.）

[16] 马绪军. 基于MapReduce的可达性保持图研究[D]. 沈阳：沈阳航空航天大学，2016：6-44.（MA X J. Research on reachability preserving graph based on MapReduce [D]. Shenyang: Shenyang Aerospace University, 2016: 6-44.）

[17] SIMECEK I, LANGR D, TVRDÍK P. Space efficient formats for structure of sparse matrices based on tree structures[C]// Proceedings of the 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. Piscataway: IEEE, 2013:

344-351.

[18] SIMECEK I, LANGR D. Tree-based space efficient formats for storing the structure of sparse matrices[J]. Scalable Computing: Practice and Experience, 2014, 15(1): 1-20.

[19] BRISABOA N R, LADRA S, NAVARRO G. K^2-trees for compact Web graph representation[C]// Proceedings of the 16th International Symposium on String Processing and Information Retrieval, LNCS 5721. Berlin: Springer, 2009: 18-30.

[20] CLAUDE F, NAVARRO G. Fast and compact Web graph representations[J]. ACM Transactions on the Web, 2010, 4(4): 1-31.

[21] YUAN C, CHAI Y, WEI S. Feature analysis and modeling of the network community structure[J]. Communications in Theoretical Physics, 2012, 58(4): 604-612.

[22] ASANO Y, MIYAWAKI Y, NISHIZEKI T. Efficient compression of Web graphs[C]// Proceedings of the International Computing and Combinatorics Conference, LNCS 5092. Berlin: Springer, 2008: 1-11.

[23] BOLDI P, VIGNA S. The WebGraph framework I: compression techniques[C]// Proceedings of the 13th International Conference on World Wide Web. New York: ACM, 2004: 595-602.

[24] BOLDI P, VIGNA S. The WebGraph framework II: Codes for the World-Wide Web[C]// Proceedings of the 2004 Conference on Data Compression. Piscataway: IEEE, 2004: 528.

[25] DWYER T, RICHE N H, MARRIOTT K, et al. Edge compression techniques for visualization of dense directed graphs[J]. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(12): 2596-2605.

[26] DINKLA K, WESTENBERG M A, VAN WIJK J J. Compressed adjacency matrices: untangling gene regulatory networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2012, 18(12): 2457-2466.

[27] HERNÁNDEZ C, NAVARRO G. Compressed representation of Web and social networks via dense subgraphs[C]// Proceedings of the 2012 International Symposium on String Processing and Information Retrieval, LNCS 7608. Berlin: Springer, 2012: 264-276.

[28] CHENG J, YU J X, LIN X, et al. Fast computing reachability labelings for large graphs with high compression rate[C]// Proceedings of the 11th International Conference on Extending Database Technology. New York: ACM, 2008: 193-204.

[29] AHO A V, GAREY M R, ULLMAN J D. The transitive reduction of a directed graph[J]. SIAM Journal on Computing, 1972, 1(2): 131-137.

[30] FAN W, LI J, WANG X, et al. Query preserving graph compression[C]// Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2012: 157-168.

[31] BUNDY A, WALLEN L. Breadth-first search[M]// Catalogue of Artificial Intelligence Tools. Berlin: Springer, 1984: 13-13.

[32] VASSEUR J P, AGARWAL N, THUBERT P, et al. Dynamic directed acyclic graph (DAG) adjustment: US8489765[P]. 2013-07-16.

[33] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

[34] 谢羿. 基于BFS结果集的可达性保持图并行计算[J]. 中国新技术新产品, 2016(11):35-36. (XIE Y. Parallel computing of reachability preserving graph based on BFS result set[J]. Chinese New Technologies and New Products, 2016(11):35-36.)

[35] HAGERUP T, MAAS M. Generalized topological sorting in linear time[J]. Nordic Journal of Computing, 1994, 1(1): 38-49.

[36] CHENG X, DALE C, LIU J. Dataset for "Statistics and social network of YouTube videos"[EB/OL]. [2019-06-01]. http://netsg.cs.sfu.ca/youtubedata/.

[37] LESKOVEC J. Stanford large network dataset collection[EB/OL]. [2019-06-01]. http://snap.stanford.edu/data/.

This work is partially supported by the National Key Research and Development Program of China (2016YFC1401907).

ZHAO Danfeng, born in 1982. Ph. D., lecturer. Her research interests include big data storage, business process management.

LIN Junchen, born in 1994, M. S. candidate. His research interests include graph computation.

SONG Wei, born in 1977. Ph. D., professor. Her research interests include computer vision, machine learning, data mining.

WANG Jian, born in 1979. Ph. D., lecturer. Her research interests include marine information technology.

HUANG Dongmei, born in 1964. M. S., professor. Her research interests include nonlinear chaotic system, object-oriented database, Web database, Web GIS database.