生成有向图中全部简单回路的一种有效算法

王玉英[1,2] 陈 平[2] 苏 旸[2]
([1](西安建筑科技大学理学院 陕西 西安 710055)
[2](西安电子科技大学软件工程研究所 陕西 西安 710071)

摘 要 在研究有向图中全部简单回路现有算法的基础上,综合各算法的优点提出一种新的算法。算法的主要思想是对图中顶点进行深度优先搜索,在搜索过程中采取措施避免同一回路的重复出现,同时记录有用信息避免对顶点的重复访问,从而提高算法的效率,还给出了算法的证明和实例应用。算法结构清晰简单,方便转化为计算机程序。

关键词 有向图 简单有向回路 深度优先搜索 算法

AN EFFICIENT ALGORITHM OF GENERATING ALL ELEMENTARY CIRCUITS IN DIRECTED GRAPH

Wang Yuying[2] Chen Ping[2] Su Yang[2]
([1] (School of Science, Xi'an University of Architecture and Technology, Xi'an 710055, Shaanxi, China)
[2] (Software Engineering Institute, Xidian University, Xi'an 710071, Shaanxi, China)

Abstract Based on studying existing algorithms with regard to all elementary circuits in directed graph, a new algorithm is proposed by integrating the advantages of these algorithms. The main thought of it is the depth-first search of vertex in graph with special measure taken to insure no circuit is repeatedly found during the searching process, meanwhile some available information is recorded for avoiding duplicated visits to same vertex, in this way it improves the efficiency of the algorithm. The proof of the algorithm and its application are given as well. It is convenient to transform this algorithm to a computer program, for its structure is clear and simple.

Keywords Directed graph Elementary directional circuit Depth-first search Algorithm

0 引 言

在计算机科学中,经常需要用图的理论解决问题,需要找出图的全部回路。例如在面向对象软件的测试中,为了确定类的测试次序,需要求出类图的全部回路,寻找一条关联多个回路的边,打破这条边,将类图从网状结构变为树形结构,从而在测试成本最低的条件下,得到类的测试次序。

对于图的回路的研究由来已久,多数算法的核心思想都是对图进行深度优先搜索,多数研究是针对于无向图的。无向图中回路的生成算法不能简单地应用于有向图中[1]。文献[1—6]中分别给出了有向图中全部简单有向回路的算法。本文在综合研究已有的比较好的有向回路求解算法的基础上提出了一种新的算法。算法的主要思想是对图中顶点进行深度优先搜索,在搜索过程中避免同一回路的重复出现,同时记录有用信息尽量避免对顶点的重复访问,从而提高算法的效率。

1 求有向图中所有简单回路的基本算法

设图 $G=(V,E)$是一简单有向图,$V=\{v_1, v_2, \cdots, v_n\}$为图的顶点集,$E$为边集,$n$为正整数。图 G中,我们称路经$<v_1, v_2, \cdots, v_k>$构成一个回路,如果 $v_1=v_k$且该路经中至少包含一条边。更进一步,如果 $v_2, v_3, \cdots, v_k$均不相同,我们说这条回路是简单回路。

1.1 算法的主要思想

算法是基于深度有限搜索的。

将图中顶点从小到大给出序号,不妨将顶点记作 1,2,…,N,每条回路的表示统一为从回路中最小顶点开始的字符串,且首尾相同的顶点只出现一次。例如,回路<3,5,2,4,3>表示为:<2,4,3,5>;算法从每个顶点 u出发,按照深度优先搜索法生成从顶点 u出发,后续顶点都大于 u的路径,将路径存在数组 P中,最后考虑这条路径以及这条路径的前部分能否和顶点 u一起形成回路,这样就找到了包含顶点 u的回路上其它顶点都大于 u的所有简单回路。

1.2 算 法

算法 1 求有向图简单回路基本算法

数据描述:有向图 $G=(V,E)$采用邻接表表示,Adj[v]表示顶点 v的邻接表;对于图中的每个顶点 v,其色彩存于变量 color[v]中;长度为 N的数组 P中存放图中后续顶点均大于起始点 p[1]的路径;链表 Circs其中每个结点包含一个长度为 N的数组,这个数组存放一个回路。

输入:邻接表表示的有向图 $G=(V,E)$。

收稿日期:2007—12—18。国家自然科学基金项目(60473063)。王玉英,副教授,主研领域:逆向工程,面向对象技术和软件体系结构。

输出：G中的所有简单回路链表 Circs。

过程：

```
EnumAllCircs(G)
1  begin
2  将 G中顶点排序，不妨设顶点为 1，2，…，N
3  数据初始化：
4      每个顶点 v的 color[v]←WHITE
5      链表 Circs为空
6      数组 P中每个元素置 0
7  for G中每个顶点 v
8  { k←1
9    P[1]←v
10   DFS-Circuits(v)
11 }
12 输出回路链表 Circs
13 end

DFS-Circuits(v)
1  Begin
2  color[v]←GRAY
3  for每个顶点 u∈Adj[v]
4  { if(u>P[1] and color[u]=WHITE)
5      { k←k+1
6      P[k]←u
7      DFS-Circuits(u)
8    }
9      if(P[1]=u)                          //形成回路
10     { 复制数组 P，并将复制的添加到 Circs中
11                                         //记录回路
12   }
13 }
14 color[P[k]]←WHITE
15 P[k]←0
16 k←k-1
17 end
```

1.3 算法证明

算法的证明分为三个部分。

（1）图中的任一条满足 $v_1<v_2,\cdots,v_m$的简单路径$<v_1,v_2,\cdots,v_k>$都会在数组 P中出现。

采用归纳法证明：

当 k=1时，算法中调用 DFS-Circuits(v_1)的过程中，$P[1]\leftarrow v_1$，结论成立。

假设当 k=m-1时，结论成立。

对于任一条满足 $v_1<v_2,\cdots,v_m$的简单路径$<v_1,v_2,\cdots,v_{m-1},v_m>$，$<v_1,v_2,\cdots,v_{m-1}>$必然在 P中出现，并且 $v_m\in$ Adj[v_{m-1}]。在执行算法 DFS-Circuits(v_1)的过程中调用 DFS-Circuits(v_{m-1})时，在第 6行执行 $P[k]\leftarrow v_m$，此时路径$<v_1,v_2,\cdots,v_{m-1},v_m>$在 P中。所以当 k=m时结论成立。

（2）通过此算法可以获得图中任一条简单回路。

任一条简单回路$<a_1,a_2,\cdots,a_k,a_1>$都可以表示为：$<v_1,v_2,\cdots,v_k>$，其中 $v_1<v_2,\cdots,v_k$（这条回路从最小顶点 v_1开始，v_1不重复出现）。

根据（1），路径$<v_1,v_2,\cdots,v_k>$在算法的执行过程中在数组 P中出现，这时在 DFS-Circuits(v_k)的第 10—12行，得到回路$<v_1,v_2,\cdots,v_k>$，并且将这条回路写入到链表中。

（3）每条简单回路在算法中只得到一次。

任一条简单回路$<a_1,a_2,\cdots,a_k,a_1>$都可以表示为：$<v_1,v_2,\cdots,v_k>$，其中 $v_1<v_2,\cdots,v_k$（这条回路从最小顶点 v_1开始，v_1不重复出现）。这条回路只能在调用 DFS-Circuits(v_1)的过程中得到。在 DFS-Circuits(v_1)执行过程中，递归调用到 DFS-Circuits(v_k)时，算法中的第 11—15行只在 $v_1\in$ Adj[v_k]时执行一次，而且在算法的其余部份都不可能生成回路，所以这条回路只能得到一次。

1.4 算法的效率分析

对图 1中的有向图 G实施此算法时，执行 DFS-Circuits(v_1)时的递归调用序列为：

$$v_1\rightarrow v_2\rightarrow v_3\rightarrow v_5\multimap v_3\multimap v_2\rightarrow v_4$$
$$\rightarrow v_3\rightarrow v_5\multimap v_3\multimap v_4\multimap v_2\multimap v_1$$

（其中“→”表示“递推”，“⊸”表示“回归”）

可以看出，在顶点 v_3和 v_5上重复执行了 DFS-Circuits，这样必然降低了效率。

为提高效率，我们改进此算法。主要思想是：在求最小顶点为 v的简单回路过程中，对每个顶点至多调用一次 DFS-Circuits，调用过的顶点颜色注明为黑色。对每个顶点 v，用一个链表 path[v]表示从 v到 v的所有路径，在递归调用过程中，如果遇到黑色顶点，则将数组 P中的路径（表示从顶点 v到当前顶点的路径）和 path中的路径合并到一起，便形成了回路。

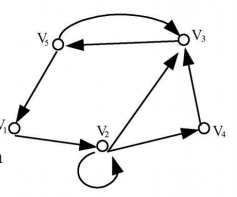

图 1有向图 G

2 改进的求有向图中所有简单回路的算法

2.1 算 法

算法 2 改进的求有向图简单回路算法

数据描述：

有向图 G=(V,E)采用邻接表表示，Adj[v]表示顶点 v的邻接表；对于图中的每个顶点 v，其色彩存于变量 color[v]中；长度为 N的数组 P中记录图中后续顶点均大于起始点 P[1]的路径；长度为 N的指针数组 Path，Path[i]指向一个链表，这个链表的每个结点包含一个长度为 N的数组 data，对于一个特定的顶点 v，这个数组中存放从顶点 i开始到达顶点 v的路径，但采用逆序存放（例如路径$<3,5,2,4>$在这个数组中的存放次序为：4,2,5,3）；链表 Circs，其中每个结点包含一个长度为 N的数组，这个数组存放一个回路。

输入：邻接表表示的有向图 G=(V,E)。

输出：G中的所有简单回路链表 Circs。

过程：

```
ImprovedEnumAllCircs(G)
1  begin
2  将 G中顶点排序，不妨设顶点为 1，2，…，N
3  数据初始化：
4      链表 Circs为空
5  for G中每个顶点 v
6  { 数组 P中每个元素置 0
7    G中每个顶点 u的 color[u]←WHITE
8    k←1
9    P[1]←v
```

```
10        Improved-DFS-Circuits(v)
11    }
12    输出回路链表 Circs
13    end

Improved-DFS-Circuits(v)
1   Begin
2   color[v]=GRAY
3   for每个顶点 u∈Adj[v]
4   { if(u≥P[1])
5     { if(color[u]=BLACK)
6       { for path[u]中每个结点 node
7         {生成新的数组 temp, temp=P,然后从 temp的第一个非零位置开始,将 node的数组中的非零元素逆序拷贝到 temp中,这样 temp中表示一条回路,将这条回路连接到链表 Circs中
8          复制 node中的数组,并将复制的数组的第一个非零位置置为 v,将这个数组添加到链表 path[v]
9         }
10      }
11
12      if(color[u]=WHITE)
13      { k=k+1
14        P[k]=u
15        Improved-DFS-Circuits(u)
16      }
17
18        if(P[1]=u) //形成回路
19        {复制数组 P并将复制的添加到 Circs中
20                                    //记录回路
21      为链表 path[v]构造一个新结点,结点中的数组 data={v},并把这个结点添加在链表的尾部
22      }
23    }
24 }
25
26  color[v]=BLACK
27  P[k]=0
28  k=k-1
29  for每个顶点 w∈Adj[v]
30  { 如果 path[w]不空,且 w≠v,复制 path[w]中的每个数组,并将复制的数组的第一个非零位置置为 v,将这些复制的数组添加到链表 path[v]
31  }
32  end
```

算法 Improved-DFS-Circuits中的第 5—10行,表明在深度优先搜索中遇到了已经搜索过的顶点 u,不需要再重新进行深度优先搜索了,将 P中记录的从顶点 P[1]到 u的父结点 v的路径,path[u]中记录的从 u到 P[1]的路径二者合并到一起,形成了最小顶点为 P[1]的简单回路。根据 path[u]修改 path[v]。第 12—16行对没有搜索过的顶点作深度优先搜索。

2.2 算法应用

下面将改进后的算法 Improved-EnumAllCircs应用于图 1中的有向图 G,求其所有简单回路。

对顶点进行排序,假设从小到大依次为 v_1, v_2, …, v_k,简单记作:1,2,3,4,5。初始化完成后,依次对每个顶点调用 Improved-DFS-Circuits。执行 Improved-DFS-Circuits(1)详细过程见表 1所示。

表 1　对图 G上执行 Improved-DFS-Circuits(1)的结果

搜索路径	k	P[1]=1	color数组	path	Circs中的数组
开始时	1	1	G,W,W,W,W (G:GRAY, W:WHITE, B:BLACK, 按顶点次序)	path[i]=null (i=1,2,…,5) (此列中没有指明的 path[i]默认为 null)	空
1→2	2	1 2	G,G,W,W,W	同上	同上
2→2	同上	同上	同上	同上	同上
2→3	3	1 2 3	G,G,G,W,W	同上	同上
3→1 (P[1]=1)	同上	同上	同上	Path[3]={[3]}	[1 2 3]
3→5	4	1 2 3 5	G,G,G,W,G	同上	同上
5→1 (P[1]=1)	同上	同上	同上	Path[3]={[3]} Path[5]={[5]}	[1 2 3] [1 2 3 5]
5→3(color[3]=G)	同上	同上	同上	同上	同上
搜索完 5	3	1 2 3	G,G,G,W,B	同上	同上
搜索完 3	2	1 2	G,G,B,W,B	Path[3]={[3],[5 3]} Path[5]={[5]}	同上
2←3 2→4	3	1 2 4	G,G,B,G,B	同上	同上
4→3(color[3]=B)	同上	同上	同上	Path[3]={[3],[5 3]} Path[5]={[5]} Path[4]={[3 4],[5 3 4]}	[1 2 3] [1 2 3 5] [1 2 4 3] [1 2 4 3 5]
搜索完 4 2←4	2	1 2	G,G,B,B,B	同上	同上
搜索完 2	1	1 2	G,B,B,B,B	Path[3]={[3],[5 3]} Path[5]={[5]} Path[4]={[3 4],[5 3 4]} Path[2]={[3 4 2],[5 3 4 2],[3 2],[5 3 2]}	同上
搜索完 1	0	0	B,B,B,B,B		同上

执行 Improved-DFS-Circuits(1)后,Circs中数组为:[1 2 3 1],[1 2 3 5 1],[1 2 4 3 1],[1 2 4 3 5]。

执行算法 Improved-DFS-Circuits(2)后,Circs中增加了数组[2]。

执行算法 Improved-DFS-Circuits(3)后,Circs中增加了数组[3 5]。

(下转第 33页)

很少有模型能达到 AUC＝1。由图 6看出，本文实验 AUC达到了 0.9614 这已经是一个很高的数值。

为了验证模型的稳定性，即预测结果不受某一次随机实验的影响，笔者对本实验做了 10次。实验结果见图 7。

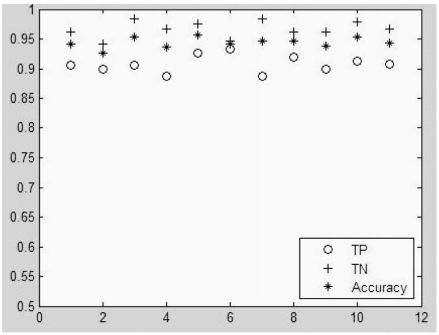

图 7　稳定性实验结果

图 7中，横坐标代表实验序号，前 10列为所做的 10次实验，最后一列为这 10次实验结果的平均值。“○”代表 TP，“＋”代表 TN，“＊”代表 Accuracy。很容易看出，TP稳定在 90.8%左右，TN稳定在 96.6%左右，Accuracy稳定在 94.4%左右。可以看出，本文提出的方法具有很高的稳定性。

4　结　论

本文利用自然图像和计算机生成图像在统计特性上的不同，采用块离散余弦变换和统计矩特征量对图像特征进行表征，详细讨论了统计矩特征量的提取和 168维模型算法的建立，通过支持向量机的训练和验证，对自然图像和计算机生成图像进行了有效鉴别。实验结果表明，本文所提出的方法对自然图像和计算机生成图像具有很高的鉴别率，与文献[1,2]中提到的方法相比，理论依据充分，数学模型清晰，正确率更高，因此，本文所提出的基于统计矩特征量的自然图像和计算机生成图像的鉴别方法具有广阔的应用前景。

参　考　文　献

[1] Lyu S, Farid H. How realistic is photorealistic? [C]// IEEE Transactions on Signal Processing, 2005, 53(2): 845-850.

[2] Chen W, Shi Y Q, Xuan G. Identifying Computer Graphics Using HSV Color Model [C]// IEEE International Conference, 2007.

[3] Chang C C, Lin C J. LIBSVM: A library for support vector machines [DB/OL]. http://www.csie.ntu.edu.tw/cjlin/libsvm.

[4] Columbia DVMM Research Lab. Columbia Photographic Images and Photorealistic Computer Graphics Dataset [DB/OL]. http://www.ee.columbia.edu/ln/dvmm/downloads/PIM_PRCG_dataset/.

[5] Shi Y Q, Chen C H, Chen W. A Natural Image Model Approach to Splicing Detection [C]// Proceedings of the 9th Workshop on Multimedia & Security, 2007.

[6] 崔霞，童学锋，宣国荣，等. 基于直方图频域矩的自然图像和计算机图形的鉴别[C]//第七届全国信息隐藏暨多媒体安全学术大会会议论文集.

[7] Cutzu F, Hammoud R, Leykin A. Distinguishing paintings from photographs [J]. Computer Vision and Image Understanding, 2005, 100: 249-273.

[8] 吴琼，李国辉，涂丹，等. 基于真实性鉴别的数字图像盲取证技术综述[J]. 自动化学报，2008, 34(12): 1458－1466.

（上接第 29页）

执行算法 Improved-DFS-Circuits(4)、Improved-DFS-Circuits(5)后，Circs中没有增加新数组。

执行完算法 EmuAllCircs(G)后，得到回路：[1 2 3 1]，[1 2 3 5 1]，[1 2 4 3 1]，[1 2 4 3 5]，[2 2]，[3 5 3]。

从链表 Circs出发，可以很容易得到图中每条边所关联的回路数。

2.3　算法复杂度分析

求有向图中全部简单回路问题是 NP问题，算法复杂度是指数级的。本算法的主要耗费有两点：一是对顶点的深度优先搜索；二是将到某一顶点 v的路径记录到 Path中。需要往 path中记录的路径和图的回路数（最小顶点为 v的回路）直接相关。因此算法的耗费直接和图中的顶点数、图中的边数、图中的回路数直接相关。图中的回路数越少这个算法的效率越高。对于边数也一样。

算法总的执行时间为：

$$k\times[(n-1)\times L_1\times\sum_{v>1}|Adj(v)|+(n-2)\times L_2\times\sum_{v>2}|Adj(v)|+\cdots+1\times L_{n-1}\times\sum_{v>n-1}|Adj(v)|]\leqslant k(n\times e\times L)$$

（k为一个常数，n为顶点数，L_v为最小顶点为 v的回路数，v＝1，2，…，n－1，L为图中的回路数）

最坏情况下，即有向图为完全图时时间复杂度为 $O(n^2 2^n)$，这时的复杂度仍然优于文献[6]中一般情况下的复杂度 $O(n!)$。

和目前现有的基于深度优先搜索的算法（如文献[1]）相比，本算法在搜索过程中记录了有用信息，在搜索过程中不需要对以前搜索过的结点再进行搜索，所以提高了效率，尤其是在回路数较少的情况下，效果更加明显。

3　结　论

本算法的特点主要有三点：

（1）将所有回路表示为从最小顶点开始的顶点序列，用这种方法保证了算法中得到的回路不重复。

（2）在求最小顶点为 v的回路过程中，对大于 v的顶点只做一次深度优先搜索，所以提高了效率。

（3）本算法结构清晰简单，方便转化为计算机程序。

参　考　文　献

[1] James C T. An Efficient Search Algorithm to Find the Elementary Circuits of a Graph [J]. Communications of the ACM, 1970, 13(12): 273-276.

[2] 徐兵，贾仁安. 有向圈的 SD计算方法[J]. 数学的实践与认识，2002, 36(7): 329-333.

[3] 刘耀年. 从有向图的通路矩阵生成有向图的全部有向回路的一个算法[J]. 电工技术学报，1992(2): 58-60.

[4] 毕双艳，郭金梅，齐明. 有向图中初级有向回路的求解算法及实例[J]. 长春邮电学院学报，1999, 17(2): 41-45.

[5] 马军，岩间一雄，马绍汉. 寻找无向图中回路的并行算法[J]. 软件学报，1997, 8(6): 475-480.

[6] 徐兵，贾仁安. 有向圈的矩阵算法及其有关性质[J]. 南昌大学学报：自然科学版，2002, 26(1): 5-11.

[7] Thomas H C, Charles E L, Ronald L R, et al. Introduction to Algorithms [M]. 2th ed. Cambridge, MA: The MIT Press, 2002: 466-467.