日志 属于 离散的 算法 7 （2009） 130—146



内容 列表 可获得的 在

科学指导

[t](http://www.ScienceDirect.com/)

日志 属于 离散的 算法

www. elvivi.com

[a](http://www.elsevier.com/locate/jda)



3-边连通性的另一种优化算法

容闳1

*温莎大学计算机学院，温莎，安大略，加拿大*

文章

信息

摘要

|  |  |
| --- | --- |
| *文章历史：*  2007年6月23日收到  2008年4月8日收到修改后的表格  接受2008年4月28日  网上2008年5月22日可用 | 提出了一种三边连通性的优化算法。该算法只对给定的图进行一次传递，以确定一组割对，其移除导致3个边连通的分量。附加的传递决定给定图的所有3-边连通分量。该算法简单、易于实现，并在线性时间和空间上运行。实验结果表明，它优于所有以前 |

|  |  |
| --- | --- |
| *关键词：*  算法分析  深度优先搜索  图连通性  3-边连通图  边连通度  3-边连通分量  割对 | 已知3个边连通度的线性时间算法，用于确定给定图形是否是3个边连通的，并且在确定切割对时。它的性能也是最好的确定三边连接的组件。  专利权2008爱思唯尔B.V.保留所有权利。 |

# 介绍

图连通度（边连通度和顶点连通度）是图论中的一个基础课题，得到了广泛的研究。〔1—6、10—13〕. 一*k*-边连通的*k*图的顶点连通图是一个不可分割的连通图。*K*边缘（分别为顶点）。图连通性在网络可靠性、计算机视觉、VLSI电路设计等领域有着广泛的应用。

对于3-边连通性，除了上述应用之外，它还具有物理和量子化学的应用，其中使用费曼图。[7，8]. Fyman图由一组顶点和一组边组成。FEYMAN图中的边可分为两类：*V*-边缘和*G*-边缘。这个*V*边是无向的，而*G*-边缘是有向的。图中的每个顶点都有正好三个边的事件：一个*V*-边，一*G*-顶点是尾部的边，一个是顶点。*G*顶点是头的边缘。费曼图*f*是*不可约的*如果不能通过拆除小于三的连接来断开*G*-边缘。让*F* 得到的无向图*f*通过承包*V*边缘与治疗*G*边缘为无向边。然后*f*既约约当且仅当*F* 是三边连通的。在量子Monte Carlo模拟中，有必要确定FEYMAN图是否是不可约的。

在计算生物学中，3-边缘连通已被用于分析从微阵列数据获得的蛋白质-蛋白质网络的FPT（固定参数可追踪）算法。〔9〕. 我们的想法是确定切割对（一对边缘的删除断开给定的图表）在一个基于蛋白质-蛋白质网络图，以便编辑集群，发现不相交的集团，可能代表相关蛋白质组。在这种情况下，切对代表两个潜在派系之间的边缘编辑（删除）。

已经给出了3个边缘连通性的线性时间算法。[2，6，10，13]. 第一个是由于

加利尔和Italiano〔2〕. 它们的方法是在线性时间内减少3-边连通度到3-顶点连通性，然后使用。

*电子邮件地址：*彼得耶尔.

一部分支持研究**美国国家安全委员会**授予NSECC-78103。

157~8667～2$见前面的问题专利权2008爱思唯尔B.V.保留所有权利。[DOI:101016.~2J.JDA.83.04.00](http://dx.doi.org/10.1016/j.jda.2008.04.003)

一百三十一

基于HopcRoFT和Tarjan的线性时间-深度优先搜索的3维连通性算法〔3〕解决这个问题。因此，它们的算法是基于约简和深度优先搜索的。由于3-顶点连通性算法相当复杂，因此加利尔和意大利语的算法虽然很优雅，但却相当复杂。

桃卡等。〔10〕在三个阶段计算3个边连通的分量，并在给定的图上执行四个深度优先搜索。他们把切割对分成两种类型：1型和2型。在第一阶段，所有类型-1切割对被确定。在第2阶段，三步确定2型切割对。在第一步中，给定图的深度优先搜索生成树被划分为不相交的路径，使得每个切割对中的边缘位于同一路径上。在第二步中，对于每个路径，它们构造给定的图的子图，该子图由路径本身和所有的后退路径组成，这些路径要么来自路径，要么终止于路径（A）。*后路*是由属于深度优先搜索生成树的边序列组成的，此后称为*树边*s，跟随一个不属于生成树的单个边，此后称为*后边缘*）然后，他们将每个这样的子图转换成一个由路径和一些新边组成的图（他们称之为图）。*在边缘*或*出边*其顶点都位于路径上。计算每个顶点的两个重要参数对2型切割对的检测至关重要。在第三步骤中，确定每个路径上的2型切割对。在第三阶段，在给给定的图添加一些新的边之后，确定3个边连通的分量。

长沼等。〔6〕在给定的图上执行深度优先搜索，然后在深度优先搜索树上确定每个树边缘，如果边缘和后缘通过计算有多少个后边缘绕过该树边缘而形成1型切割对。在确定所有类型1对的情况下，使用三种类型的变换将给定的图形转换成更小的图形。然后将相同的方法递归地应用到后者的每个非平凡连接成分。在本质上，他们发现所有的2型切割对，通过将它们转换成1型切割对，通过逐步修改给定的图表。结果，所涉及的所有图中的边的总数可以是给定图的三倍。然而，总的时间和空间复杂度是线性的。

最近，Toka等人的多遍算法相反。那嘎莫迟等，Tsin〔13〕提出了一个简单的算法，只对给定图进行一次遍历。该算法基于观察到，如果两个边具有共2度的顶点，则这两个边形成切割对。显然，并非每个切割对都具有这种特性。所以一个转变，叫做*吸收弹射*引入了对给定图的变换，使得如果两个边形成割对并且它们不相邻，则当深度优先搜索回溯到深度优先搜索树中较高的一个端顶点时，该图最终将被转换为具有两个图的图。边共享2度的公共端点。此外，公共端顶点是一个超顶点，在这个意义上，它是由3个边连通分量中的所有顶点组成的。

在本文中，我们提出了另一种简单的线性时间算法，它只在给定的图上执行一个深度优先搜索，以确定所有割边的集合和第二遍来确定所有的3个边连通的分量。该算法不区分类型1和类型2对，也不使用任何变换。实验结果表明，我们的算法比其他算法具有更好的性能，主要关注的是确定给定的图是3个边连通的还是确定切割对。它优于所有其他，除了Tsin〔13〕当3个边连通的分量也要被确定时。因此，我们的算法是用于确定一个给定的图形是否是3个边连接或确定图的所有切割边（属于切割对的边）的应用的选择。前面提到的物理、化学和生物信息学的应用就是这样的应用。

# 基本定义

我们假设读者有图论的基本知识。让*G*=(*V*，*E*）表示一个无向图*V*是顶点集和*e*是一组边。图表可能包含***平行边***（两个或多个具有相同端点的边）但不是自循环（端点相同的边）。让*u*，*V*γ*V*. 安***U-V路径***是一个边序列*e*1,*e*2…*eK*这样*e我*=(*ui*−1,*ui*)，1 *我* *k*在哪里*u我*γ*V*，0 *我* *k*，*u*零=*U*和*uK*=*v*. 两顶点*U*和*V*是***有联系的***在里面*G*有一个*U*-*V*路径在*G*.*G*是***有联系的***IF每两个顶点连接在一起。让*G*=(*V*，*E*）是一个无向连通图γ*V*γ2。边是***桥***在里面*G*如果它的移除导致一个断开的图。一对边是***割对***在里面*G*如果它们的删除结果是一个断开的图，它们都不是桥。一***切边***是一对一对边。一***3-边连通分量***属于*G*是一个顶点的极大集合*G*这样，在集合中的每两个不同顶点之间，至少有三个边缘不相交的路径连接它们。显然，如果两个顶点属于同一个三边连通分量，则没有删除桥或割对。*G*可能导致它们断开连接。

让*T*=(*V*，*ET*）成为一棵生成树*G*通过深度优先搜索遍历创建〔11〕.*T*被称为***DFS树***属于*G*. 让*V*γ*V*. 这个***V的子树***，用*T*(*v*)是最大的子树*T*谁的根是*v*. 让*u*，*V*γ*V*. 顶点*U*是一个***祖先***顶点的*V*敌我识别*V*是子树中的顶点。*u*. 顶点*U*是一个***固有祖先***顶点的*V*敌我识别*U*是一个祖先*V*和*U*=*v*. 顶点*V*是一个（***适当的***）***后裔***顶点的*U*敌我识别*U*是一个（正确的）祖先*v*.

边缘*G*躺在*T*被称为***树边***那些躺在外面的*T*被称为***后边缘***. 如果(*u*，*v*）是一个后缘，那么*U*是一个祖先*V*或*V*是一个祖先*U*〔11〕. 每个顶点*V*分配一个不同的数字，由*DFS*(*v*)称之为***深度优先搜索数***哪个是排名*V*在遍历中访问顶点的顺序中。让(*u*，*v*）是树的边缘（分别是后缘），顶点*U*是***尾***顶点时*V*是***头***边缘的IF*DFS*(*u*<*DFS*(*v*）(*DFS*(*u*>*DFS*(*v*)，分别）。在本文的其余部分中，每当我们用(*u*，*v*)我们假定*U*是尾巴和*V*是头。让(*u*，*v*）做树边，*U*是***起源***属于*V*和*V*是一个***小孩***属于*u*;

一百三十二

(*u*，*v*）是***父边***属于*V*和A***子边缘***属于*u*. 安***输入后缘***顶点的*V*是它的后缘*V*是头部和一个***输出后缘***顶点的*V*是它的后缘*V*是尾巴。安*x*–*Y****树形路径***连接顶点的路径是*X*和*Y*在里面*T*. 这个***水平***顶点的*V*在里面*T*是边上的个数*r*–*V*树路径*R*是根*T*.

# 预赛

让*G*=(*V*，*E*）是一个无向图*T*=(*V*，*ET*）是一个DFS树*G*. 为清晰起见，如那嘎莫迟等。〔6〕，桃加等。〔10〕Tsin〔13〕我们假设不丧失一般性*G*在本文中是连接的和无桥的。

如果*e*和*e* 形成一对剪辑*G*然后，其中一个是后缘。这是因为去除任何两个后缘*G*不存在不连通图，因为存在*DFS*树。

**引理3.1。***让**在G中是一对*=(*u*，*v*）*e*=(*x*，*y*)*.*

1. *如果e* *是后缘，Y是U的祖先，而X是后裔。*瓦伊*在T*;
2. *如果E和E* *是树边，E和E* *躺在树根上，把树根和叶子连接起来。*

**证明。**割对的定义和DFS树的性质的直接结果。 我们的算法是基于以下特征的割对定理。

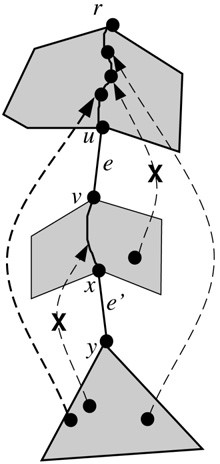
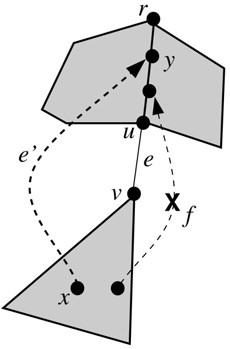
**定理3.2。***让e*,*e*γ*E是这样的E* *DFS*(*x*)*E是树的边缘。*

1. *如果e* *是后缘，则是G中的割对，当且仅当不存在后缘F时*=(*s*,*t*）*如此f*=*e* *而S是一个后裔*瓦伊*而T是U的祖先*(*图*1（a）；
2. *如果e* *是树边，则是G的割对，当且仅当不存在后缘时*(*s*,*t*）*这样要么*(*图*1（b）；（a）*S是一个后裔*瓦伊*而不是Y，而T是U的祖先，或*（b）*S是Y的后裔，而T是后裔。*瓦伊*而不是Y。*

**证明。**（i）假设*e*和*e* 形成一对剪辑*G*. 然后*e*必须是一座桥梁*G*-*e* 这意味着不存在一个循环。*G*-*e* 包含*e*. 由此可见，不存在后缘。(*s*,*t*）在里面*G*-*e* 这样*S*是一个后裔*V*虽然*T*是一个祖先*U*在里面*T*. 因此，不存在后缘。*f*=(*s*,*t*）在里面*G*这样*f*=*e* 和*S*是一个后裔*V*虽然*T*是一个祖先*U*(图1（a）相反，因为*G*-*e* 不包含后缘(*s*,*t*）这样*S*是一个后裔*V*虽然*T*是一个祖先*u*没有周期*G*-*e* 包含*e*. 因此，*e*是一座桥*G*-*e* 这意味着是一对剪辑*G*.

（二）见

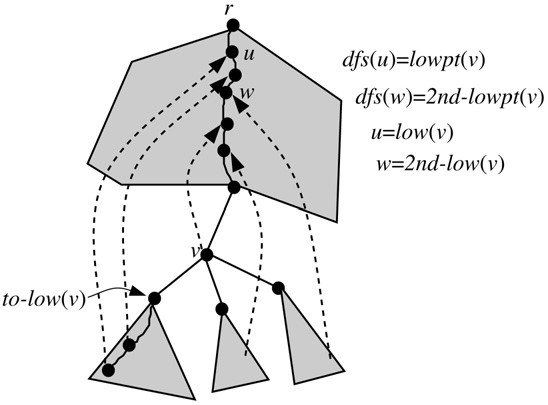
让*E*切是切边的集合*G*. 确定所有3-边连通分量*G*我们首先要确定*E*切.



（a） （b）

**图1。**插图引理3.1和定理3.2.

一百三十三



**图2。***洛普特*(ν）和*第二低音*(ν).

**定义。**让*e*=(*x*，*y*）成为一个锋利的人。如果任一*e*是后缘还是*e*是树边，没有树边*T*(*y*）或后端有一个端点*T*(*y*）形成一对剪辑*e*然后*e*被称为***发电机***.

注意，我们对发电机的定义比Toka等人更一般。〔10〕因为我们允许后边是生成器。

**引理3.3。***让一对这样的切割*(*x*，*y*)*DFS*(*v*） *DFS*(*x*）*E和E* *是树的边缘。IF-是一对割，这样F是生成器，那么*{*f*,*e*}*是一对剪刀。*

**证明。**立即从定理3.2.

作为直接后果引理3.3每一个切边都属于一个包含生成器的切割对。因此，要确定*E*切它足以确定切割对的子集包含生成器。

下面的引理在检查顶点之间的祖先- ~2-后代关系时是有用的。

**引理3.4。***（见〔14〕让你*,瓦伊*是DFS树中的两个顶点。顶点*瓦伊*是U的祖先，当且仅当DFS时*(*v*） *DFS*(*u*） *DFS*(*v*）^ ^ ^*钕*(*v*）-1*其中钕*(*v*）*是后代的数量*瓦伊*在树上。*

当执行深度优先搜索时*G*在每个顶点计算以下两个值。

**定义。**让*V*γ*V*. 一*v*–*T****后路***是一条路径*G*由A组成*v*–*S*树路径*S*是一个后裔*v*和后缘(*s*,*t*）（注意，当*v*–*S*树路径是空路径，*v*–*T*后路是后缘(*v*,*t*)）然后***洛普特***(*v*）=**闽**({*DFS*(*t*）γ*A V*–*T背*-*路径*}{*DFS*(*v*)}）〔11〕.

让ρ*V*是一个*v*–*T*这样的反向路径*DFS*(*t*）=*洛普特*(*v*). 然后，***第二低音***(*v*）=**闽**({*DFS*(*t*）γ(∃*A V*–*T背*-*帕特克*）*Q*和ρ*V*是*边缘*-*不相交*}{*DFS*(*v*)}）〔13〕.

让***低的***(ν）(***第二低***(ν)（分别）是这样的顶点*DFS*(*低的*(*v*）=*洛普特*(*v*）(*DFS*(2*钕*-*低的*(*v*）=2*钕*-*洛普特*(*v*)，分别）。

给出了一个例子。图2.

**定义。**对于每一个*V*γ*V*, the***低端***属于*V*是后缘(*v*，*w*）这样*DFS*(*w*）=*洛普特*(*v*）还是第一个孩子*W*属于*V*遇到深度优先搜索时这样*洛普特*(*w*）=*洛普特*(*v*）如果没有这样的后缘(*v*，*w*). 顶点*W*用***到低***(ν). 在这种情况下(*v*，*w*）是树边，顶点*W*被称为***低龄儿童***属于*v*.

请注意，如果*低端*由后缘定义，图包含平行边，然后*低端*不是唯一的。在这种情况下，我们任意选择其中之一(*v*，*w*）后边缘满足*DFS*(*w*）=*洛普特*(*v*）作为*到低*边缘。

**定义。**这个***低位路径***顶点的*V*最长的路径是从*V*组成*到低*后代的边缘*v*.

**引理3.5。***顶点V的低路径是V低。*(*v*）*返回路径。*

一百三十四

**证明。**通过归纳的层次*V*在*DFS*树。

**引理3.6。***让VIS切割成一对这样的E* *是树的边缘*(*v*）*或在T中有一个端点的非树边*(*v*）*当E在T的外面**必须躺在V的低路径上，而E位于低处*(*v*)*-V树路径。*

**证明。**引理3.1意味着*e*必须躺在*R*-*V*树路径定理3.2（ii）（b）暗示*e*必须躺在*低的*(*v*)–*V*树路径。

假设相反*e* 不在于*低位路径*属于*v*.

让*e*=(*u*，*w*）和*e*=(*x*，*y*). 此外，由于引理3.5, the*到低*路径*V*是一个*v*–*低的*(*v*）后路，让*f*,*低的*(*v*）是后面的边缘*到低*路径*v*.

1. 假设*e* 是后边。通过引理3.1，*e*必须躺在*y*–*X*树路径。但是*e*躺在外面*T*(*v*)因此，它必须躺在*y*–*V*树路径。如果*X*是一个顶点*到低*路径*v*然后作为*e* 不是*到低*边缘*x*我们必须拥有*DFS*(*y*） *洛普特*(*x*）=*洛普特*(*v*). 另一方面，如果*X*不是一个顶点*到低*路径*v*让*一*是最亲近的祖先*X*上*到低*路径*V*和*乙*是它的子顶点*a*–*X*树路径。自从*乙*不是*到低*儿童*a*我们必须拥有*DFS*(*y*） *洛普特**洛普特*(*b*） *洛普特*(*a*）=*洛普特*(*v*). 在任何一种情况下，我们都有*DFS*(*y*） *洛普特*(*v*). 但是边缘*f*是这样的后缘*低的*(*v*）是一个祖先*Y*因此，一个祖先*U*虽然*z* 是一个后裔*w*. 这种矛盾定理3.2（i）。
2. 假设*e* 是一棵树的边缘。让*一*是最亲近的祖先*X*上*到低*路径*V*和*乙*是它的子顶点*a*–*Y*树路径。自从*乙*不是*到低*儿童*a*，*洛普特*(*y*） *洛普特*(*b*） *洛普特*(*a*）=*洛普特*(*v*）这意味着*低的*(*v*）是一个祖先*低的*(*y*). 让是后面的边缘*到低*路径*y*. 然后通过定理3.2（ii）（b），边缘*e*必须躺在*低的*(*y*)–*V*树路径。但是边缘*f*是这样的后缘*低的*(*v*）是一个祖先*低的*(*y*）因此*U*虽然*z* 是一个后裔*W*而不是*Y*哪些矛盾定理3.2

# 算法的高级描述

**引理4.1。***让e*=(*x*，*y*）*和f*=(*u*，*v*）*E是发电机，*{*e*，*f*}*是一对剪刀。让e*=(*w*，*z*）*是另一个发电机躺在V-X树路径。然后每边形成一对E* *必须位于V-W树路径上。*

**证明。**我们将考虑这种情况。*e*是一棵树的边缘。情况下*e*非树边缘相似但比较简单。

自从*e* 是发电机，它不会形成切割对。*e*. 通过定理3.2（ii）有后缘(*s*,*t*）这样要么*T*躺在*z*–*X*树路径*S*是一个后裔*y*或*T*躺在*r*–*W*树路径*S*是一个后裔*Z*但不是*y*. 但是{*e*，*f*}割对意味着前一种情况是不可能的。定理3.2（ii）（b），对于后一种情况*T*必须躺在*v*–*W*树路径定理3.2（ii）（a）。随之而来的是任何边缘形成一个切割对。*e* 必须躺在*t*–*W*树路径定理3.2（ii）（b）。引理由此而来。

上述引理表明切割对具有嵌套结构。由于这个原因，堆栈是用于确定切割对的自然数据结构。具体地说，在算法的执行过程中，在每个顶点*v*堆栈，***堆栈***(ν)由条目组成，[(*xi*，*yi*）*p我* *q我*]，1 *我* *k*是这样创建的：

1. *DFS*(*xi*） *DFS*(*yi*+1)，*q我*=*pi*+1，1 *我*<*k*，*p*一=*低的*(*v*）和*DFS*(*qk*） *DFS*(*v*） *DFS*(*xk*);
2. 每个(*xi*，*yi*）是一个发电机或一个潜在的发电机，并有潜力产生切割对边缘。*p我* *q我*树形路径；
3. 外边*T*(*v*）形成一对剪辑(*xi*，*yi*）必须躺在*p我* *q我*树路径和无边*p我* *q我*树路径可以形成任何边缘的切割对。*qi*–*V*树路径。

为了便于解释，在本文的其余部分中，有一个端点的后缘。*T*(*v*)，*V*γ*V*被认为是一个边缘*T*(*v*).

**定义。**让*V*γ*V*. 让(*x*，*y*）是一个谎言的边缘*T*(*v*）和*磷* *Q*躺在树上*R* *V*这样外面的每一个边缘*T*(*v*）形成一对剪辑(*x*，*y*）必须躺在*磷* *Q*没有边缘*磷* *Q*可以形成任何边缘的切割对*q*–*V*树路径。然后，*磷* *Q*被称为***势路***属于(*x*，*y*）在*v*.

注意，尽管每个条目[(*xi*，*yi*）*p我* *q我*]，1 *我* *k*上*堆栈*(*v*)，*p我* *qi*是潜在的路径(*xi*，*yi*）在*v*边缘及其潜在路径*V*可能没有相应的条目*堆栈*(*v*）如果结果表明，在电位路径上没有边缘可以形成与边缘的切割对。此外，通过引理3.6，*我* *k*都躺在*到低*路径*V*虽然*p我* *qi*，1 *我* *k*都躺在*低的*(*v*)–*V*树路径。类似的堆栈结构用于[3，10].

如果事实证明没有边缘*p我* *q我*树形路径形成切割对(*xi*，*yi*)然后进入[(*xi*，*yi*）*p我* *q我*]（可能与*p我* *q我*当深度优先搜索回溯到某个祖先时，将被从堆栈中弹出。*V*上*pi*–*V*树路径。否则，让(*s*,*t*）是一个边缘*p我* *q我*树形路径，形成剪切对(*xi*，*yi*). 然后当深度优先搜索回溯到*S*从*t*上面的条目*堆栈*(*t*）必须是[(*xi*，*yi*）*p我* *T*].