边赋权森林ω-路划分的 $O(n)$算法*

蔡延光[1+], 张新政[1], 钱积新[2], 孙优贤[2]

[1](广东工业大学 自动化学院,广东 广州 510090)
[2](浙江大学 工业控制技术国家重点实验室,浙江 杭州 310027)

An $O(n)$ Algorithm for ω-Path Partition of Edge-Weighted Forests

CAI Yan-Guang[1+], ZHANG Xin-Zheng[1], QIAN Ji-Xin[2], SUN You-Xian[2]

[1](Faculty of Automation, Guangdong University of Technology, Guangzhou 510090, China)
[2](National Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China)
+ Corresponding author: Phn: 86-20-37627023, Fax: 86-20-37626412, E-mail: ygcai@263.net
http://www.gdut.edu.cn

Received 2002-03-28; Accepted 2002-07-08

Cai YG, Zhang XZ, Qian JX, Sun YX. An $O(n)$ algorithm for ω-path partition of edge-weighted forests. *Journal of Software*, 2003,14(5):897~903.
http://www.jos.org.cn/1000-9825/14/897.htm

Abstract: ω-path partition problem is the generalization of path partition problem. It comes of broadcasting communication in parallel computer system, computer network and distributed control system. This problem can be described by graph theory terminology as follows. Let $G=(V,E)$ be an undirected simple graph with nonnegative edge-weights, and $\omega\geq 0$ be a real. $\{P_1, P_2, \ldots, P_m\}$ is called to be an ω-path partition of G if $P_1, P_2, \ldots, P_m$ are m pairwise vertex-disjoint paths of G such that $\bigcup_{i=1}^{m} V(P_i)=V(G)$, and $W(P_i)\leq\omega$ (for all $i=1, 2, \ldots, m$, where $W(P_i)=\sum_{e\in E(P_i)} w(e)$). The ω-path partition number of G is the smallest cardinality of ω-path partition of G. The ω-path partition problem of G is to find a minimum ω-path partition and the ω-path partition number of G. It is evident that ω-path partition problem for any graph is NP-complete by the NP-completeness of Hamiltonian path. In this paper, some properties of ω-path partition have been investigated for paths, trees and forests with nonnegative edge-weights respectively. Linear time algorithms have been presented to solve ω-path partition problem for any path and tree with nonnegative edge-weights respectively. Local realization techniques and complexities of these two algorithms have been discussed in detail. Based above algorithms, an $O(n)$ algorithm has been designed to solve ω-path partition problems for forests with nonnegative edge-weights. The algorithms presented in this paper are concise, and require less time and space.

Key words: broadcasting; path partition; ω-path partition; tree; forest; algorithm; complexity; NP-complete

---

* Supported by the Natural Science Foundation of Guangdong Province of China under Grant No.010060 (广东省自然科学基金); the Key Science-Technology Project of Guangdong Province of China under Grant No.C31801 (广东省科技攻关项目)

第一作者简介: 蔡延光(1963－),男,湖北咸宁人,博士,副教授,主要研究领域为图算法,组合优化,智能优化,智能决策支持系统.

摘 要: ω-路划分问题是路划分问题的一般化,它源于并行计算机系统、计算机网络与分布式控制系统等一类广播通信问题.设置最少的信息源节点,使得在指定的时间内将信息源节点所拥有的信息发送到其余节点,并且保证不同通信线路之间不得相交.从 Hamilton 路的 NP-完全性不难看出,ω-路划分问题属于 NP-完全问题.通过构造性证明技术,获得了边赋非负权路径、树和森林的ω-路划分问题的一些性质.分别提出了求解边赋非负权路径和边赋非负权树的ω-路划分问题的线性时间算法,讨论了算法的局部实现技术,详细地分析了这些算法的复杂度.以这两个算法为基础,提出了一个线性时间算法求解边赋非负权森林的ω-路划分问题.所提出的算法直观简明、操作容易,只需要较少的运行时间和较小的存储空间.

关键词: 广播;路划分;ω-路划分;树;森林;算法;复杂度;NP-完全

中图法分类号: TP301 **文献标识码**: A

并行计算机系统处理器之间和计算机网络节点之间的广播通信提出这样一个问题:设置最少的信息源节点,使得在指定的时间内将信息源节点所拥有的信息发送到其余节点,并且保证不同通信线路之间不得相交,此即为求解边赋非负权图的ω-路划分问题.在自动化领域也可以找到ω-路划分问题的应用,例如,出于建设和维护费用方面的考虑,希望在分布式控制系统中尽可能少地选择局部控制中心,使得各局部控制中心的有关数据和控制信息通过预设的通信线路在规定的时间内传送到全部管理点.

定义 1. 设 $G=(V,E)$是一个边赋非负权的无向简单图,顶点数为 $n=|V(G)|$,$\omega\geq 0$ 为实数.如果 $P_1,P_2,\ldots,P_m$ 是 G 中 m 条顶点两两不相交的路径,且$\bigcup_{i=1}^{m}V(P_i)=V(G)$,$W(P_i)\leq\omega$(一切 $i=1,2,\ldots,m$; $W(P_i)=\sum_{e\in E(P_i)}w(e)$,即路径 P_i 的所有边的权和),则称$\{P_1,P_2,\ldots,P_m\}$为 G 的一个ω-路划分.

按照图论中的约定,平凡图(即一个顶点的图)也是路径.因此,对于任意边赋非负权的无向简单图 $G=(V,E)$,$V(G)=\{v_1,v_2,\ldots,v_n\}$,如果置 $P_i=\{v_i\}(i=1,2,\ldots,n)$,则有 $W(P_i)=0$;从而不论$\omega\geq 0$ 为何实数,$\{P_1,P_2,\ldots,P_n\}$都是 G 的一个ω-路划分.可见,对于任意边赋非负权的无向简单图都存在ω-路划分.

定义 2. G 的含路径数最少的ω-路划分称为 G 的最小ω-路划分,G 的最小ω-路划分所含的路径数称为 G 的ω-路划分数,记作 $p_\omega(G)$.所谓ω-路划分问题就是求 $p_\omega(G)$和 G 的一个最小ω-路划分.

显然有 $p_\omega(G)\leq n$.当 G 是(边)非赋权图且$\omega\geq n-1$ 时(实际上对路径的长度无限制),ω-路划分问题简称为路划分问题[1~3];当 G 是(边)非赋权图且ω是一个正整数 k 时,ω-路划分问题就是 k-路划分问题[4];ω-路划分问题的一般化就是直径有限的子图划分问题[5].ω-路划分问题与路中心等网络选址问题也有很多联系[6~10].

从 Hamilton 路的 NP-完全性不难看出,路划分问题属于 NP-完全问题.到目前为止,仅对于区间图[11](interval graph)、圆弧图[12](circular arc graph)、树[13,14](tree)、森林[15](forest)、合作图[4](cograph)、二分置换图[16](bipartite permutation graph)、块图[16,17](block graph)等获得了线性时间或多项式时间的算法求解路划分问题.对于 k-路划分问题,仅对树获得了线性时间算法求 $p_k(G)$[4].由于ω-路划分是路划分、k-路划分在赋权图中的一般化,所以ω-路划分问题与实际问题更加贴近但却更加复杂难解.

树是一种结构简单但应用最广泛的通信网络拓扑结构,受到了理论研究者和应用系统设计者的重视.森林是树的超集,即森林的连通片是树.本文研究了边赋非负权路径、树和森林的ω-路划分问题的一些性质,分别提出了求解边赋非负权路径、树、森林的ω-路划分问题的线性时间算法.理论分析和实际计算表明,所提出的算法具有运行速度快、需要的存储空间少等特点.

定义 3. 考虑边赋非负权的无向简单图 $G=(V,E)$.

(1) 设 H,K 是 G 的两个子图,以 $H\ominus K$ 表示在 H 中删除 $V(K)$以及与 $V(K)$相关联的所有边而形成的子图.

(2) 设 $v\in V(G)$,G_1 是 $G-\{v\}$的一个连通片,称导出子图 $G[V(G_1)\cup\{v\}]$为 v(点)的一个分支.

(3) 设$(u,v)\in E(G)$,以 $w(u,v)$表示边(u,v)的权.

(4) 设 $v\in V(G)$,如果 v 的顶点度 $d_G(V)=1$,则称顶点 v 为端顶点.

(5) 设 $u,v\in V(G)$,以 $d_G(u,v)$表示顶点 u,v 之间的距离,在不引起混淆时简记为 $d(u,v)$.

(6) 设 $x,y,z\in V(G)$,以$(x\sim z)$表示顶点 x 到顶点 z 的一条路径,$(x\sim y\sim z)$表示顶点 y 在路径$(x\sim z)$上,顶点 x 也用

$(x\sim x)$表示.

除非特别声明,以下讨论中所提及的树、路径、森林和图分别指边赋非负权的无向的树、路径、森林和简单图.文中未定义的术语和记号与文献[18]一致,诸算法采用 C 语言风格设计.

1 路径的ω-路划分

引理 1. 设 $P=(x\sim y)$是一个边赋非负权的路径,如果 $d(x,y)\le\omega$,则$\{P\}$是 P 的一个最小ω-路划分,$p_\omega(P)=1$.

证明:因为对于路径 P,$W(P)=d(x,y)\le\omega$,根据定义 1、定义 2 即证引理. □

引理 2. 设 $P=(x\sim y)$是一个边赋非负权的路径,$d(x,y)>\omega$,取 $z\in V(P)$,使$|E(x\sim z)|$最大但 $d(x,z)\le\omega$,则存在 P 的一个最小ω-路划分 Ω,使$(x\sim z)\in\Omega$,$p_\omega(P)=p_\omega(P\ominus(x\sim z))+1$,$P\ominus(x\sim z)$是一条路径.

证明:设$\{P_1,P_2,\dots,P_m\}$为 P 的一个最小ω-路划分,不妨设 P_1 包含顶点 x.

(1) 首先,如果 $d(x,y)>\omega$,则 P 是非平凡的路径,故 x,y 是 P 的端顶点.其次,因为 P_1 包含顶点 x,故 $P_1\subseteq(x\sim z)$,否则 $P_1\ominus(x\sim z)\neq\varnothing$,与$|E(x\sim z)|$的最大性矛盾.再者,对于一切 $i=2,\dots,m$,P_1 与 P_i 顶点不相交,因此 $x\notin V(P_i)$,即$(x\sim z)\not\subseteq P_i$,故 $P_i\ominus(x\sim z)$或为空或为路径.去掉$\{P_2\ominus(x\sim z),\dots,P_m\ominus(x\sim z)\}$中的空元素,根据定义 1,即知$\{P_2\ominus(x\sim z),\dots,P_m\ominus(x\sim z)\}\cup\{(x\sim z)\}$是 P 的一个ω-路划分.由 $m=p_\omega(P)$可知,$\{P_2\ominus(x\sim z),\dots,P_m\ominus(x\sim z)\}$中不可能有空元素,即$\{P_2\ominus(x\sim z),\dots,P_m\ominus(x\sim z)\}\cup\{(x\sim z)\}$是 P 的一个最小ω-路划分.

(2) 从上述证明过程(1)知,$\{P_2\ominus(x\sim z),\dots,P_m\ominus(x\sim z)\}$是 $P\ominus(x\sim z)$的一个ω-路划分,于是 $p_\omega(P\ominus(x\sim z))\le m-1=p_\omega(P)-1$,即 $p_\omega(P)\ge p_\omega(P\ominus(x\sim z))+1$.另一方面,如果$\{P'_1,P'_2,\dots,P'_{m'}\}$(这里,$m'=p_\omega(P\ominus(x\sim z))$)是 $P\ominus(x\sim z)$的一个最小ω-路划分,则$\{P'_1,P'_2,\dots,P'_{m'}\}\cup\{(x\sim z)\}$是 P 的一个ω-路划分,于是 $p_\omega(P)\le m'+1=p_\omega(P\ominus(x\sim z))+1$.从而有 $p_\omega(P)=p_\omega(P\ominus(x\sim z))+1$.

(3) 因 $P\ominus(x\sim z)$是连通的,故 $P\ominus(x\sim z)$为路径. □

定理 1. 算法 1 在线性时间求解边赋非负权路径的ω-路划分问题.

算法 1. 求解路径的ω-路划分问题.

```
输入:边赋非负权的路径 P=(x~y),常数ω≥0.
输出:pω(P)及 P 的一个最小ω-路划分Ω*(P).
初始:pω(P)=0;Ω*(P)=∅;  //如果是算法 2 调用本算法,
                          则 pω(P),Ω*(P)的初始值由算法 2 传递过来
while V(P)≠∅
{
  如果 d(x,y)≤ω  //片段 1
  {
    Ω*(P)=Ω*(P)∪{P};
    pω(P)=pω(P)+1;
    V(P)=∅;
  }
否则  // d(x,y)>ω
  {
    取 z∈V(P),使|E(x~z)|最大但 d(x,z)≤ω;  //片段 2
    取 z1∈V(P),z1 与 z 相邻但 d(x,z1)>ω;  //片段 3
    Ω*(P)=Ω*(P)∪{(x~z)};
    pω(P)=pω(P)+1;
    P=P⊖(x~z);
    x=z1;  //注意到 V(P)≠∅
  };
};  //循环结束
输出 pω(P),Ω*(P);
结束.
```

证明:由引理 1 和引理 2,算法 1 可以正确求解边赋非负权路径的ω-路划分问题.现在分析算法 1 的复杂度.算法 1 的运行时间主要是循环、判断是否 $d(x,y)\le\omega$(片段 1)、寻找 z 和 z_1(片段 2、片段 3)以及计算 $d(x,v)$的时间.片段 1、片段 2 与片段 3 和计算 $d(x,v)$可以通过下述方法实现:从 x 出发,x 的惟一邻居为 x_1(如果不存在这样的 x_1,则表明 $x=y$,也就是 $d(x,y)\le\omega$,这对应于片段 1),$d(x,x_1)=w(x,x_1)$;如果 $d(x,x_1)\le\omega$,则对 x_1 的邻居 $x_2(x_2\neq x$,于是 x_2 惟一.如果不存在这样的 x_2,则表明 $x_1=y$,也就是 $d(x,y)\le\omega$,此对应片段 1.以下有类似的解释,不再重复说明)有 $d(x,x_2)=d(x,x_1)+w(x_1,x_2)$,否则 $z=x,z_1=x_1$;如果 $d(x,x_2)\le\omega$,则对 x_2 的邻居 $x_3(x_3\neq x_1$,于是 x_3 惟一)有 $d(x,x_3)=d(x,x_2)+w(x_2,x_3)$,否则 $z=x_1,z_1=x_2;\dots$;继续进行这一过程直至找到了 x_k,x_{k+1} 使 $d(x,x_{k+1})>\omega$而 $d(x,x_k)\le\omega$,取 $z=x_k$,$z_1=x_{k+1}$;如果找不到这样的 x_k,x_{k+1},则表明 $d(x,y)\le\omega$.因此,片段 1、片段 2 和片段 3 只搜索了 $x,x_1,\dots,x_k,x_{k+1}$,但在进行 $P\ominus(x\sim z)$运算后,只留下了 x_{k+1} 是搜索过的顶点.考虑到循环,算法 1 在执行过程中类似于 x_{k+1} 这样的顶点被重复搜索 1 次(x_{k+1} 作为下次迭代的开始顶点还要被考虑 1 次),其他顶点恰好被搜索 1 次,但重复搜索的顶点为 $p_\omega(P)-1$ 个,因此算法 1 执行时间为 $O(p_\omega(P)+|V(P)|)$即 $O(n)$. □

2 树的ω-路划分

标号 A. 设 T 是一个边赋非负权的树,$\Delta(T)\geq 3$(即 T 不是路径).任选顶点 $v_0\in V(T),d(v_0)\geq 3.v_0$ 标号为 0;T 中的每一个顶点 v 标号为 v 到 v_0 的路径边数(称为 v 的标号值),即$|E(v\sim v_0)|$.

定义 4. 设 T 是一个边赋非负权的树,$\Delta(T)\geq 3$.如果 T 的某个顶点 v 使用标号 A 进行了标号,则称顶点 v 已标号,或者称顶点 v 是一个已标号的顶点;如果 T 的所有顶点使用标号 A 进行了标号,则称树 T 按标号 A 已标号,或者简称树 T 已标号,或者称树 T 是已标号树.

注 1. 标号 A 可在线性时间内完成.事实上,在用标号 A 对树的顶点进行标号时,还可建立一个包含 T 的所有使 $d(v)\geq 3$ 顶点 v 的有序集 Q,Q 的元素按其标号值从大到小顺序排列.方法是:初始,从顶点 v_0 开始,v_0 位于 Q 的末尾;第 1 步,依次将标号值为 0 的顶点(此时只有 v_0)的每个邻居 v 标号为 1(已标号的顶点不再标号),如果 $d(v)\geq 3$,则将 v 置于 Q 之首;第 2 步,依次将标号值为 1 的顶点的每个邻居 v 标号为 2(已标号的顶点不再标号),如果 $d(v)\geq 3$,则将 v 置于 Q 之首;…;第 k 步,依次将标号值为 $k-1$ 的顶点的每个邻居 v 标号为 k(已标号的顶点不再标号),如果 $d(v)\geq 3$,则将 v 置于 Q 之首.继续这一过程,直至 T 的所有顶点获得标号.因为 Q 的元素个数小于$|V(T)|$,因此,可以在 $O(n)$时间完成标号 A 并获得有序集 Q.此外,标号值最大的且 $d(v)\geq 3$ 的顶点 v 就是 Q 的第 1 个元素.

引理 3. 设 T 是一个边赋非负权的已标号树, $\Delta(T)\geq 3$,则

(1) 如果 v 是标号值最大的使 $d(v)\geq 3,v\neq v_0$ 的顶点,则除含顶点 v_0 的分支以外,v 点的其余 $d(v)-1$ 个分支均为路径.

(2) 如果 Q 的第 1 个元素是 v_0,则 v_0 点的所有分支均为路径.

证明:(1) 设 $T_1(v)$是 v 点的不包含 v_0 的分支,如果 $T_1(v)$不是路径,即在 $T_1(v)$中存在顶点 v_1 使 $d_{T_1}(v_1)\geq 3$(于是有 $d_T(v_1)\geq 3$).注意到 $v_1\neq v$(因为 v 在 $T_1(v)$中是端顶点),在 T 中 v_1 的标号值为$|E(v_0\sim v_1)|=|E(v_0\sim v)|+|E(v\sim v_1)|\geq |E(v_0\sim v)|+1>|E(v_0\sim v)|$,即 v_1 的标号值大于 v 的标号值,矛盾.

(2) 类似于(1)的证明. □

注 2. 在一个边赋非负权的已标号树 $T(\Delta(T)\geq 3)$中,如果 v 是标号值最大的使 $d(v)\geq 3,v\neq v_0$ 的顶点,则除了包含顶点 v_0 的分支以外,v 点的其余 $d(v)-1$ 个分支均是路径且有一个端顶点为 v;如果标号值最大的顶点的标号值为 0(对应顶点即是 v_0),则 v_0 点的所有分支均是路径且有一个端顶点为 v_0.

引理 4. 设 T 是一个边赋非负权的已标号树,$\Delta(T)\geq 3$,v 是标号值最大的使 $d(v)\geq 3,v\neq v_0$ 的顶点,除了包含顶点 v_0 的分支以外,v 点的其余 $d(v)-1$ 个分支分别记为$(v_1\sim v),(v_2\sim v),\ldots,(v_k\sim v)$.相应地,非 v 的端顶点分别记为 $v_1,v_2,\ldots,v_k$,则

(1) 如果 $d(v_1,v)>\omega$,取 $z\in V(v_1\sim v)$,使$|E(v_1\sim z)|$最大但 $d(v_1,z)\leq\omega$,则存在 T 的一个最小ω-路划分Ω,使$(v_1\sim z)\in\Omega$,$p_\omega(T)=p_\omega(T\ominus(v_1\sim z))+1$,$T\ominus(v_1\sim z)$是树;

(2) 如果对一切 $i(i=1,2,\ldots,k)$有 $d(v_i,v)\leq\omega$,则

(2.1) 如果 $k\geq 3$,且 $d(v_1,v)=\max\limits_{1\leq i\leq k} d(v_i,v)$,$z$ 是$(v_1\sim v)$上与 v 相邻的顶点,则存在 T 的一个最小ω-路划分Ω,使$(v_1\sim z)\in\Omega$,$p_\omega(T)=p_\omega(T\ominus(v_1\sim z))+1$,$T\ominus(v_1\sim z)$是树;

(2.2) 如果 $k=2$,且 $d(v_1,v_2)\leq\omega$,则存在 T 的一个最小ω-路划分Ω,使$(v_1\sim v_2)\in\Omega$,$p_\omega(T)=p_\omega(T\ominus(v_1\sim v_2))+1$,$T\ominus(v_1\sim v_2)$是树;

(2.3) 如果 $k=2$,且 $d(v_1,v_2)>\omega,d(v_1,v)\geq d(v_2,v)$,$z$ 是$(v_1\sim v)$上与 v 相邻的顶点,则存在 T 的一个最小ω-路划分Ω,使$(v_1\sim z)\in\Omega$,$p_\omega(T)=p_\omega(T\ominus(v_1\sim z))+1$,$T\ominus(v_1\sim z)$是树.

证明:设$\{P_1,P_2,\ldots,P_m\}$为 T 的一个最小ω-路划分.

(1) 不妨设 P_1 包含 v_1.类似引理 2 的证明,我们有$\{P_2\ominus(v_1\sim z),\ldots,P_m\ominus(v_1\sim z)\}\cup\{(v_1\sim z)\}$是 T 的一个最小ω-路划分,$p_\omega(T)=p_\omega(T\ominus(v_1\sim z))+1$,$T\ominus(v_1\sim z)$是树.

(2-1) 不妨设 P_1 包含 v_1.

情形 1.P_1 不包含 v,则必有 $P_1\subseteq(v_1\sim z)$,$P_i\ominus(v_1\sim z)(i=2,\ldots,m)$是路径,从而$\{P_2\ominus(v_1\sim z),\ldots,P_m\ominus(v_1\sim z)\}\cup\{(v_1\sim z)\}$

是 T 的一个 ω-路划分(因而是 T 的一个最小 ω-路划分).

情形 2.P_1 包含 v,记 $P_1=(v_1\sim z\sim v\sim x)$.注意到$(v_1\sim v),(v_2\sim v),\dots,(v_k\sim v)(k\geq 3)$不可能都在 P_1 中(P_1 最多只能包含它们中的两个),不妨设 P_1 不包含 v_2,P_2 包含 v_2,因而 P_2 不包含 v.设 z_2 是$(v_2\sim v)$上与 v 相邻的顶点.易证:如果 x 在$(v_2\sim z_2)$上,则$\{(v_1\sim z),(v_2\sim v),P_3,\dots,P_m\}$是 T 的一个最小 ω-路划分;如果 x 不在$(v_2\sim z_2)$上,此时由于 $d(v_1,v)\geq d(v_2,v)$,故 $d(v_2,x)=d(v_2,v)+d(v,x)\leq d(v_1,v)+d(v,x)=d(v_1,x)\leq\omega$,于是$\{(v_1\sim z),(v_2\sim v\sim x),P_3,\dots,P_m\}$是 T 的一个最小 ω-路划分.

综合情形 1、情形 2,即知存在 T 的一个最小 ω-路划分 Ω,使$(v_1\sim z)\in\Omega$.结合上述讨论,类似引理 2 的证明(2)、(3),易证 $p_\omega(T)=p_\omega(T\ominus(v_1\sim z))+1$,$T\ominus(v_1\sim z)$是树.

(2-2) 不妨设 P_1 包含 v_1.

情形 1.P_1 包含 v.如果 P_1 包含 v_2,即$(v_1\sim v_2)\subseteq P_1$,因而 $P_1=(v_1\sim v_2)$,故$\{(v_1\sim v_2),P_2,\dots,P_m\}$是 T 的一个最小 ω-路划分.如果 P_1 不包含 v_2,不妨设 P_2 包含 v_2,则$\{P_1\ominus(v_1\sim v),P_3,\dots,P_m\}\cup\{(v_1\sim v_2)\}$是 T 的一个最小 ω-路划分.

情形 2.P_1 不包含 v.显然 P_1 不包含 v_2,不妨设 P_2 包含 v_2.此时必有 P_2 包含 v.如果这样,则类似情形 1 的证明,可得存在 T 的一个最小 ω-路划分 Ω,使$(v_1\sim v_2)\in\Omega$.否则,如果 P_1,P_2 都不包含 v,则必有 $P_1\subset(v_1\sim v)$,$P_2\subset(v_2\sim v)$,取 z_1 是$(v_1\sim v)$上与 v 相邻的顶点、z_2 是$(v_2\sim v)$上与 v 相邻的顶点.易知,$P_i\ominus(v_1\sim z_1)\ominus(v_2\sim z_2)$(一切 $i=3,\dots,m$)是路径,从而$\{(v_1\sim z_1),(v_2\sim z_2),P_3\ominus(v_1\sim z_1)\ominus(v_2\sim z_2),\dots,P_m\ominus(v_1\sim z_1)\ominus(v_2\sim z_2)\}$是 T 的一个 ω-路划分.注意到 $k=2$,故 $P_i\ominus(v_1\sim z_1)\ominus(v_2\sim z_2)\ominus\{v\}=P_i\ominus(v_1\sim v_2)$(一切 $i=3,\dots,m$)是路径,因此 $\Omega_1=\{(v_1\sim v_2),P_3\ominus(v_1\sim v_2),\dots,P_m\ominus(v_1\sim v_2)\}$是 T 的一个 ω-路划分,但$|\Omega_1|=m-1$,矛盾.

综合情形 1、情形 2,即知存在 T 的一个最小 ω-路划分 Ω,使$(v_1\sim v_2)\in\Omega$.结合上述讨论,类似引理 2 的证明(2)、(3),易证 $p_\omega(T)=p_\omega(T\ominus(v_1\sim v_2))+1$,$T\ominus(v_1\sim v_2)$是树.

(2-3) 不妨设 P_1 包含 v_1.因 $d(v_1,v_2)>\omega$,故 P_1 不包含 v_2,不妨设 P_2 包含 v_2.以下类似于(2-1)的证明,从略. □

注 3. 设 T 是一个边赋非负权的已标号树,$\Delta(T)\geq 3$,v_0 是标号值最大的顶点,记 $v=v_0$,对于 v 点的任意 $d(v)-1$ 个分支$(v_1\sim v),(v_2\sim v),\dots,(v_k\sim v)(k=d(v)-1)$,相应的,非 v 的端顶点分别记为 $v_1,v_2,\dots,v_k$,则引理 4 之(1)、(2)的所有结论也是成立的.

注 4. 设 v 是标号值最大且 $d(v)\geq 3$ 的顶点,建立对应于 v 的临时记忆 Q_1,Q_1 的元素至多 2 个.如果 $v\neq v_0$,则 Q_1 的元素为 v 点的不含 v_0 的长度不大于 ω 的最短的 2 个分支(如果存在的话);如果 $v=v_0$,则 Q_1 的元素为 v 点的任意 $d(v)-1$ 个分支中长度不大于 ω 的最短的 2 个分支(如果存在的话).Q_1 的第 1 个元素记作$(v'_1\sim v)$,第 2 个元素记作$(v'_2\sim v)$.

算法 2. 求解树的 ω-路划分问题.

输入:边赋非负权的树 T,常数 $\omega\geq 0$.
输出:$p_\omega(T)$及 T 的一个最小 ω-路划分 $\Omega^*(T)$.
初始:$p_\omega(T)=0$; $\Omega^*(T)=\varnothing$;$Q=\varnothing$;$Q_1=\varnothing$;
如果 $\Delta(T)\geq 3$,则将树 T 按标号 A 标号并构造 Q; //片段 1
while $V(T)\neq\varnothing$
{
  如果 $Q=\varnothing$ //$0\leq\Delta(T)\leq 2$,即 T 是路径
  {
    调用算法 1;
    结束;
  }
  否则 //$Q\neq\varnothing$,即 $\Delta(T)\geq 3$
  {
    取 Q 的首元素 v;
    如果 $d(v)<3$
    {
      $Q=Q-\{v\}$; //$\ominus$运算如果删去了 v 的邻居将导致 $d(v)$减少
      $Q_1=\varnothing$;
      continue;
    };
    如果 v 点存在至少两个不在 Q_1 中的分支
    {
      如果 $v\neq v_0$,则任取 v 点不含 v_0 的分支$(v_i\sim v)$
      否则任取 v_0 点的一个分支$(v_i\sim v)$;

```
        如果 d(v_i,v)>ω   //对应于引理 4(1),片段 2
        {
          取 z∈V(v_i~v),使|E(v_i~z)|最大但 d(v_i,z)≤ω;   //片段 3
          Update(v_i,z,∅);
          continue;
        };
        如果|Q_1|<2   //以下隐含 d(v_i,z)≤ω
        {
          Q_1=Q_1∪{(v_i~z)};
          continue;
        }
        否则   //对应于引理 4(2.1)
        {
          如果 d(v_i,v) < max_{1≤j≤2} d(v_j,v) ,则将(v_i~v)把 Q_1 中的较大者置换出来,置换出来的元素仍然用(v_i~v)表示;
          寻找(v_i~v)上与 v 相邻的顶点 z;
          Update(v_i,z,∅);
        };
      }
      否则   //对应于引理 4(2.2,2.3)
      {
        如果 d(v'_1,v'_2)≤ω,则 Update(v'_1,v'_2,v)
        否则
        {
          在 Q_1 中选取(v_i~v),使 d(v_i,v) = max_{1≤j≤2} d(v'_j,v) ;
          在 (v_i~v)上寻找与 v 相邻的顶点 z;
          Update(v_i,z,∅);
        };
        Q_1=∅;
      };
    };
};   //循环结束
输出 p_ω(T), Ω*(T);
结束.

function Update(x,y,z)   //更新 Ω*(T),p_ω(T),T 和 Q 的值
{
   Ω*(T)= Ω*(T)∪{(x~y)};
   p_ω(T)=p_ω(T)+1;
   T=T⊖(x~y);
   Q=Q−{z};
}
```

定理 2. 算法 2 在线性时间求解边赋非负权树的ω-路划分问题.

证明:由引理 4、注 3、注 4 和定理 1,使用算法 2 可以正确求解边赋非负权树的ω-路划分问题.算法 2 的执行时间主要是循环、标号 A 及建立 Q(片段 1)、判断是否 $d(v_i,v)>\omega$(片段 2)、寻找 $z(z\in V(v_i\sim v))$使$|E(v_i\sim z)|$最大但 $d(v_i,z)\le\omega$(片段 3)以及计算 $d(x,v)$的时间.根据注 1,片段 1 具有线性时间复杂度;由定理 1 的证明,片段 2、片段 3 和计算 $d(x,v)$可在线性时间内完成.

结合循环和⊖算子(在 *Update* 函数内),仿照定理 1 的证明,易知重复搜索过的顶点只有 $O(p_\omega(T))$个,因此算法 2 执行时间为 $O(p_\omega(T)+|V(T)|)$,即 $O(n)$. □

3 森林的ω-路划分

容易证明:

引理 5. 设边赋非负权的简单图 $G=(V,E)$的连通片为 $G_1,G_2,\ldots,G_k,\Omega^*(G_i)$为 G_i 的一个最小ω-路划分 $(i=1,2,\ldots,k)$,则

(1) $p_\omega(G)=\sum_{i=1}^{k} p_\omega(G_i)$;

(2) $\bigcup_{i=1}^{k}\Omega^*(G_i)$ 是 G 的一个最小ω-路划分.

对于森林 F,我们也可使用标号 A 对 F 中的每个顶点进行标号,方法是对 F 的每个不是路径的连通片分别使用标号 A 进行标号.类似于定义 4,可定义森林 F 已标号、已标号森林等概念.由定理 2 及引理 5,有如下定理.

定理 3. 可以在线性时间求解边赋非负权森林的ω-路划分问题.

References:

[1] Farley AM, Pelc A, Proskurowski A. Minimum-Time multidrop broadcast. Discrete Applied Mathematics, 1998,83(1):61~77.

[2] Klasing R. The relationship between the gossip complexity in vertex-disjoint paths mode and the vertex bisection width. Discrete Applied Mathematics, 1998,83(1):229~246.

[3] Hedetniemi SM, Hedetniemi ST, Liestman AL. A survey of gossiping and broadcasting in communication networks. Networks, 1988,18:319~349.

[4] Yan J-H, Chung GJ, Hedetniemi SM, Hedetniemi ST. *K*-Path partition in trees. Discrete Applied Mathematics, 1997,78(1):227~233.

[5] Arikati SR. Graph clustering: complexity, sequential and parallel algorithms [Ph.D. Thesis]. Edmonton: Department of Computer Science, University of Alberta, 1994.

[6] Liu S, Cai YG. The *m*-path centers in trees. Journal of Chongqing University, 1991,14(4):104~110 (in Chinese with English abstract).

[7] Liu S, Cai YG. Disjoint *m*-path cneters. Chinese Journal of Operations Research, 1992,11(2):63~66 (in Chinese with English abstract).

[8] Cai YG, Shi QZ. The constrained *p*-centers in trees. Journal of Hubei Industries Institute, 1992,7(2):20~28 (in Chinese with English abstract).

[9] Cai YG, Qian JX, Sun YX. A parallel iterative algorithm for constrained *p*-centers. Systems Engineering-Theory and Practice, 2000, 20(7):1~6 (in Chinese with English abstract).

[10] Slater PJ. Locating central path in a graph. Transportation Science, 1982,16(1):1~18.

[11] Arikati SR, Rangan CP. Linear algorithm for optimal path cover problem on interval graphs. Information Processing Letters, 1990, 35(3):149~153.

[12] Bonuccelli MA, Bovet DP. Minimum node disjoint path covering for circular-arc graph. Information Processing Letters, 1979, 8(4):159~161.

[13] Goodman SE, Hedetniemi ST. On the Hamiltonian completion problem. In: Graph Theory and Combinatorics 1973. Berlin: Springer, 1974. 262~272.

[14] Misra J, Tarjan RE. Optimal chain partitions of trees. Information Processing Letters, 1975,4(1):24~26.

[15] Skupien Z. Path partitions of vertices and hamiltonicity of graphs. In: Proceedings of the 2nd Czechoslovakian Symposium on Graph Theory. Prague, 1974.

[16] Srikant R, Sundaram R, Singh KS, Rangan CP. Optimal path cover problem on block graphs and bipartite permutation graphs. Theoretic Computer Science, 1993,115(2):351~357.

[17] Yan J-H, Chung GJ. The path partition problem in block graphs. Information Processing Letters, 1994,52(6):317~322.

[18] Bondy JA, Murty USR. Graph Theory with Applications. New York: MacMillan Press Ltd., 1976.

附中文参考文献:

[6] 刘松,蔡延光.树的 *m*-路中心.重庆大学学报,1991,14(4):104~110.

[7] 刘松,蔡延光.不相交的 *m*-路中心.运筹学杂志,1992,11(2):63~66.

[8] 蔡延光,石庆忠.树的受限 *p*-中心.湖北汽车工业学院学报,1992,7(2):20~28.

[9] 蔡延光,钱积新,孙优贤.受限 *p*-中心的并行迭代算法.系统工程理论与实践,2000,20(7):1~6.