参数为 k 的几乎树中的染色多路割

李曙光[1] 辛 晓[2]

(山东工商学院计算机科学与技术学院 烟台 264005)[1] (山东工商学院外国语学院 烟台 264005)[2]

摘 要 染色多路割问题源于对等网络中的数据分片，是传统多路割问题的推广。给定颜色相关边赋权图 G 和 G 上若干特异顶点的局部染色，将该局部染色扩展到所有顶点上，使得两端点染不同颜色的边的权和最小。对于参数为 k 的几乎树，给出了多项式时间精确算法。也就是说，染色多路割问题是固定参数可解的，其中的参数 k 是使得 G 中任意双连通分支 C 成为树所要拿掉的最大边数。

关键词 算法，染色多路割，固定参数可解，参数为 k 的几乎树

中图法分类号 TP301 **文献标识码** A

Colored Multiway Cuts in Almost Trees with Parameter k

LI Shu-guang[1] XIN Xiao[2]

(College of Computer Science and Technology, Shandong Institute of Business and Technology, Yantai 264005, China)[1]

(College of Foreign Studies, Shandong Institute of Business and Technology, Yantai 264005, China)[2]

Abstract The colored multiway cut problem generalizes the traditional multiway cut problem. It is motivated by data partitioning in Peer-to-Peer networks. Given a graph G with color dependent edge weights and a partial coloration on some distinguished vertices, the colored multiway cut problem is to extend the partial coloration such that all the vertices are colored and the total weight of edges that have different colored endpoints is minimized. A polynomial time exact algorithm was presented for almost trees with parameter k. That is to say, the colored multiway cut problem is fixed-parameter tractable with respect to the parameter k, which is defined as the maximum, over all biconnected components C of the graph G, of the number of edges that must be removed from C to obtain a tree.

Keywords Algorithms, Colored multiway cuts, Fixed-parameter tractable, Almost trees with parameter k

1 引言

设 $G=(V,E)$ 是一个简单图，$|V|=n$。$C=\{1,2,\cdots,d\}$ 是 d 种颜色的集合。对于 $S\subseteq V(G)$，映射 $\chi:S\rightarrow C$ 称为一个局部染色，它定义了 S 的一个剖分 $S_i=\{v\in S:\chi(v)=i\}$。映射 $\bar{\chi}:V(G)\rightarrow C$ 称为一个染色，如果对于所有的 $v\in S$ 均有 $\chi(v)=\bar{\chi}(v)$ 成立。染色 $\bar{\chi}$ 定义了 $V(G)$ 的一个剖分 $V_i=\{v\in V(G):\bar{\chi}(v)=i\}$，使得对于所有的 $i\in\{1,2,\cdots,d\}$，都有 $S_i\subseteq V_i$。

颜色相关权函数 w 分配给边 (u,v) 和颜色 i,j 一个非负实数 $w(u,v;i,j)$，表示在满足条件 $\bar{\chi}(u)=i$ 且 $\bar{\chi}(v)=j$ 的染色 $\bar{\chi}$ 中，边 (u,v) 所获得的权重。设 $w(u,v;i,i)=0$，$w(u,v;i,j)=w(v,u;j,i)$，若 $u\in S$ 但 $\chi(u)\neq i$ 或者 $v\in S$ 但 $\chi(v)\neq j$，则有 $w(u,v;i,j)=\infty$。若对于任意边 (u,v)，当 $i_1\neq j_1$ 且 $i_2\neq j_2$ 时都有 $w(u,v;i_1,j_1)=w(u,v;i_2,j_2)$，则称 w 是颜色无关的。

染色 $\bar{\chi}$ 的权定义为 $\sum_{(u,v)\in E(G)}w(u,v;\bar{\chi}(u),\bar{\chi}(v))$。染色多路割问题的目标是找到一个最优染色，即最小权染色。

分布式数据库系统或对等系统(Peer-to-Peer system)中的数据放置问题可以抽象为染色多路割问题。在对等系统中，信息存储在服务器上。用户提交查询时，请求被定向到查询数据项所在的存储服务器。对某数据项的查询请求可能会产生对其他数据项的查询请求。如何将数据合理放置使得总通讯费用最低，是对等系统中的一个重要问题。染色多路割问题中，V 中顶点代表数据项，d 种颜色代表 d 个存储服务器。G 中若有边 (u,v)，则表明对数据项 u 的查询会产生对数据项 v 的查询。权函数 w 是预期的通讯费用函数。S 中顶点对应的数据项已预先放置在指定服务器上(给定映射 $\chi:S\rightarrow C$)。目标是在存储服务器上合理放置数据(确定映射 $\bar{\chi}:V(G)\rightarrow C$)，使得实际产生的总通讯费用最低。

若权函数 w 是颜色无关的，并且所有的 $|S_i|=1$，则染色多路割问题退化为传统的多路割问题。该问题在无向图和有向图中都是 APX 难解的[1,2]。无向图上的多路割问题有 $(1.5-1/d)$-近似算法[3]和 $(1.3438-\varepsilon_d)$-近似算法[4]；有向图上的多路割问题有 2-近似算法[5]。当图 G 是稠密图并且所有边的权都为 1 时，多路割问题有多项式时间近似方案

到稿日期：2009-03-08 返修日期：2009-06-21 本文受国家自然科学基金(60673153，60970105)，山东省信息产业发展专项资金项目(2008X00039)资助。

李曙光(1970－)，男，博士，副教授，主要研究方向为组合最优化等，E-mail：sgliytu@hotmail.com；辛 晓(1970－)，女，学士，主要研究方向为计算机网络。

(PTAS)[6]。$d=2$ 时，多路割问题就是著名的最大流最小割问题。该问题在组合最优化领域中占据了重要位置，因为它有相当多的应用，并且可在多项式时间之内求解[7,8]。当图 G 是平面图、d 为常数并且所有边的权都为 1 时，多路割问题是多项式时间之内可解的[1]。

Erdos 和 Szekely 提出并研究了染色多路割问题[9,10]，当图 G 是树时，给出了运行时间为 $O(nd^2)$ 的精确算法[10]。这一算法经过修改之后，可用来求解树环和广义树网络中的染色多路割问题[11,12]。

对于 NP 难解问题，可从参数复杂性的角度来研究。参数复杂性的观点是寻找固定参数可解算法(Fixed-Parameter Tractable algorithms，FPT 算法)，从而将组合爆炸性(指数运行时间因子)限制在某个参数 k 问题上(该参数一般来说很小)[13,14]。如果一个参数化问题存在运行时间为 $O(f(k)poly(n))$ 的精确算法，其中，$f(k)$ 是仅依赖于参数 k 的任意可计算函数，$poly(n)$ 是输入规模为 n 的多项式函数，则称该参数化问题是固定参数可解的。若该类算法存在，则意味着只要参数 k 的值适当小，该参数化问题的较大实例也可高效求解。对于 NP 难解问题来说，固定参数可解算法可以看作是近似算法的有效替代。

本文给出了染色多路割问题的一个固定参数可解算法，所选参数 k 是使得 G 中任意双连通分支 C 成为树所要拿掉的最大边数。由于树和树环分别是参数为 0 和 1 的几乎树，广义树网络中每个双连通分支都有常数个顶点(故有常数条边)，因此本文推广了文献[10-12]中的结果。

本文第 2 节介绍了一些基本概念；第 3 节研究仙人掌图(cactus graphs)中的染色多路割问题，给出了多项式时间精确算法；第 4 节说明如何简化和推广第 3 节的结果，得到了求解参数为 k 的几乎树中染色多路割问题的精确算法；最后指出了可供进一步研究的工作。

2 预备知识

设 $G=(V,E)$ 是无向连通图。如果去掉 G 的任意顶点 v 及该顶点所关联的边，所得图 G-v 仍然连通，则称 G 是双连通图(按照这一定义，单边也是双连通图)。G 的极大双连通子图称为 G 的双连通分支(或称为块)。若 G 不是双连通图，则可以拿掉它的某个顶点(及该顶点所关联的边)，使其成为非连通图。这样的顶点称为 G 的割点[15]。

设 $C_1,C_2,\cdots,C_t$ 和 $s_1,s_2,\cdots,s_q$ 分别表示图 G 的双连通分支和割点。图 G 的双连通分支 $C_1,C_2,\cdots,C_t$ 产生了一个树状结构，在该结构中，属于多个双连通分支的顶点一定是割点。因此，我们可以定义图 G 的超结构 T_G。T_G 是一棵树，其顶点集和边集分别是：

$V(T_G)=\{C_1,C_2,\cdots,C_t,s_1,s_2,\cdots,s_q\}$

$E(T_G)=\{(C_i,s_j)\mid s_j \text{ 是 } C_i \text{ 的顶点}\}$

适当修改 Tarjan[16] 的双连通算法，则可以在 $O(|V(G)|+|E(G)|)$ 时间之内找到图 G(任意无向连通图)的所有双连通分支、割点以及超结构 T_G。文献[17]给出了这样一个子程序 $SEP(G,v,T_G)$。输入图 G 和图 G 的一个顶点 v，$SEP(G,v,T_G)$ 就能得出图 G 的超结构 T_G。T_G 是一棵根树：若 v 是 G 的割点，则树根就是 v，否则树根是 v 所在的(唯一)双连通分支 C。

令 $k(C_i)$ 表示要使 G 中双连通分支 C_i 成为树所要拿掉的边数，则有 $k(C_i)=|E(C_i)|-|V(C_i)|+1$。令 $k(G)=\max\limits_{1\leqslant i\leqslant t}\{k(C_i)\}$，若 $k(G)=k$，则称 G 是参数为 k 的几乎树[17]。

含任何两圈无公共边的连通图称为仙人掌图。一个图是仙人掌图当且仅当它的每个双连通分支是圈或者单边，即当且仅当它是树或参数为 1 的几乎树。由于考虑的是边赋权图，故只需考虑简单图。此时，仙人掌图上每个圈的长度不小于 3。树环是一种特殊的仙人掌图，可由树按如下方式得到：将树的每个顶点替换为一个环，然后收缩树的边，使得与任一边两端点相对应的两个环恰有一个公共点。一个图是树环当且仅当它的所有双连通分支都是圈。

3 仙人掌图

本节研究仙人掌图中的染色多路割问题，将给出多项式时间精确算法。图 G 在本节中表示仙人掌图。算法核心是一个递归程序 $COST(G,M(G),s,c(s))$。给定图 G 中染色多路割问题的一个实例 $M(G)$ 和图 G 的一个顶点 s，$COST(G,M(G),s,c(s))$ 能够计算出点 s 的费用函数 $c(s)=(c_1(s),c_2(s),\cdots,c_d(s))$，其中 $c_i(s)$ 表示点 s 必须染颜色 i 时图 G 的最优顶点染色的费用(即此时 G 中两端点染不同颜色边的权和)，$i\in\{1,2,\cdots,d\}$。进一步可得到图 G 的最优顶点染色的费用为 $\min\limits_{1\leqslant i\leqslant d}\{c_i(s)\}$。应用回溯法可以确定问题的最优解。

设点 v 是图 G 的任一顶点。点 v 的初始费用函数 $c(v)$ 定义如下：对 $i\in\{1,2,\cdots,d\}$ 置 $c_i(v)=\begin{cases}0, & v\in S_i \text{ 或者 } v\notin S\\ +\infty & v\in S\backslash S_i\end{cases}$。

算法第一步是调用文献[17]中的子程序。输入图 G 和顶点 $s\in V(G)$，$SEP(G,s,T_G)$ 得出图 G 的超结构 T_G。T_G 是一棵根树：若 s 是 G 的割点，则树根就是 s，否则树根是 s 所在的唯一双连通分支 C。然后要区分两种情形。

情形 1　G 是双连通图。

若图 G 是单边 (s,t)，则

$c_i(s)\leftarrow c_i(s)+\min\limits_{j=1,2,\cdots,d}\{w(s,t;i,j)+c_j(t)\}$

若图 G 是一个圈，则按如下方法得到点 s 的费用函数 $c(s)=(c_1(s),c_2(s),\cdots,c_d(s))$。

取 G 的一个顶点 $u\neq s$，称点 u 为裂开点。将点 u 裂成两个点 u_1 和 u_2，得到图 $Split(G,u)$。$Split(G,u)$ 由以下几部分组成：图 G 的导出子图 G-u，两个新的顶点 u_1 和 u_2，两条新边 (v_1,u_1) 和 (v_2,u_2)。这里的 v_1 和 v_2 是图 G 中点 u 的两个邻点。

将点 u 染颜色 j，$j\in\{1,2,\cdots,d\}$。对于 $a\in\{1,2\}$，$i\in\{1,2,\cdots,d\}$，置 $w(u_a,v_a;j,i)=w(u,v_a;j,i)$。置 $c_j(u_1)=c_j(u)$，$c_j(u_2)=0$。

接下来，调用程序 $COST(Split(G,u),M(Split(G,u)),s,c^j(s))$，得到 $c^j(s)=(c_1^j(s),c_2^j(s),\cdots,c_d^j(s))$，其中 $c_i^j(s)$ 表示点 s 必须染颜色 i 并且点 u_1 和 u_2 必须染颜色 j 时图 $Split(G,u)$(图 G)的最优顶点染色的费用，$i\in\{1,2,\cdots,d\}$。

点 s 必须染颜色 i 时图 G 的最优顶点染色的费用为：

$c_i(s)=\min\limits_{j=1,2,\cdots,d}\{c_i^j(s)\}$，$i\in\{1,2,\cdots,d\}$。

情形 2　G 不是双连通图。

程序 $COST(G,M(G),s,c(s))$ 从 T_G 的叶子开始向上依次处理图 G 的双连通分支。设当前正在处理的分支是 C，它

是 T_G 的一片叶子。C 与图 G 其余部分的唯一公共点设为 q。设图 G 中染色多路割问题的实例 $M(G)$ 在 C 上的限制为 $M(C)$。调用程序 $COST(C,M(C),q,c(q))$，对点 q 的费用函数 $c(q)$ 进行更新。完成之后，将分支 C 从 T_G 上删除。若 T_G 上有尚未处理的双连通分支，则重复这一过程，否则程序 $COST(G,M(G),s,c(s))$ 终止。

现将程序 $COST(G,M(G),s,c(s))$ 扼要描述如下。

for $i\in\{1,2,\cdots,d\}$，$v\in V(G)$，do //初始化

置 $c_i(v)=\begin{cases}0, & v\in S_i \text{ 或者 } v\notin S\\ +\infty, & v\in S\backslash S_i\end{cases}$

程序 $COST(G,M(G),s,c(s))$：

```
begin
    SEP(G,s,T_G);
    if G 是双连通图 then begin
        if G 是单边(s,t) then
            for i∈{1,2,…,d}do
    c_i(s)←c_i(s)+ min_{j=1,2,…,d} {w(s,t;i,j)+c_j(t)};
        else begin //G 是一个圈
            找到 G 的一个顶点 u≠s;
            构造 Split(G,u);
            for j∈{1,2,…,d}do
    COST(Split(G,u),M(Split(G,u)),s,c^j(s))
            for i∈{1,2,…,d}do
                c_i(s)← min_{j=1,2,…,d} {c_i^j(s)}
        end
    end
    else begin
        处理 T_G 的任一片叶子;
        if 该叶子是 G 的割点 then 从 T_G 删除
        else begin
            设该叶子是 G 的一个双连通分支 C;
            if C 是 T_G 的树根 then q←s
            else q←C 在 T_G 中的父亲;
            COST(C,M(C),q,c(q));
            从 T_G 中删除 C
        end
    end
end
```

程序 $COST(G,M(G),s,c(s))$ 处理一个圈所用时间为 $O(n_i d^3)$，其中 n_i 表示圈上的顶点数。因此总运行时间为 $O(nd^3)$。这样就得到：

定理 1 存在求解仙人掌图中染色多路割问题的精确算法，其运行时间为 $O(nd^3)$。

该算法实质是将图 G 中任意双连通分支 C 的顶点染色费用函数浓缩到一个点 q 上，称该点为 C 的质心。若 C 是子程序 $SEP(G,s,T_G)$ 所产生的 T_G 的树根，则质心就是 s，否则质心是 C 在 T_G 中的父亲。处理完 C 之后，费用函数 $c(q)$ 得到更新，C 随之从 T_G 中删除。文献[10]求解树中染色多路割问题的算法正是这一思路，实际上它给出了程序 $COST(T,M(T),s,c(s))$，这里 T 表示一棵树。

4 参数为 k 的几乎树

本节将给出求解参数为 k 的几乎树中染色多路割问题的精确算法。图 G 在本节中表示参数为 k 的几乎树。与上节类似，算法核心是一个递归程序。不同之处在于：上节算法是将仙人掌图转化为一系列单边来实现，而本节算法较为宏观，是将参数为 k 的几乎树转化为一系列树来实现。这种简化十分有助于分析算法的时间复杂性。

设 $D_1,D_2,\cdots,D_l$ 和 $s_1,s_2,\cdots,s_q$ 分别表示图 G 的非单边双连通分支和割点。将树 T_G 上对应于 $D_1,D_2,\cdots,D_l$ 的顶点删除，若某割点（图 G 的割点）对应顶点成为孤立点，则将其删除。设所得连通分支为 $T_1,T_2,\cdots,T_h$。记 $G(V(T_i))=D_{l+i}$，$i\in\{1,2,\cdots,h\}$。显然 $D_{l+1},D_{l+2},\cdots,D_{l+h}$ 都是树，为叙述方便，不妨称 $D_1,D_2,\cdots,D_{l+h}$ 为图 G 的分块。

定义图 G 的修正超结构 $\overline{T}_G$ 如下。$\overline{T}_G$ 是一棵树，其顶点集和边集分别是：

$V(\overline{T}_G)=\{D_1,D_2,\cdots,D_{l+h},s_1,s_2,\cdots,s_q\}$

$E(\overline{T}_G)=\{(D_i,s_j)\mid s_j \text{ 是 } D_i \text{ 的顶点}\}$

文献[17]的子程序 $SEP(G,v,T_G)$ 略加修改后得到子程序 $\overline{SEP}(G,v,\overline{T}_G)$。输入图 G 和图 G 的一个顶点 v，$\overline{SEP}(G,v,\overline{T}_G)$ 就能得出图 G 的修正超结构 $\overline{T}_G$。$\overline{T}_G$ 是一棵根树：若 v 是 G 的割点，则树根就是 v，否则树根是 v 所在的（唯一）分块 D。

与上一节类似，算法核心是递归程序 $COST(G,M(G),s,c(s))$。给定图 G 中染色多路割问题的一个实例 $M(G)$ 和图 G 的一个顶点 s，$COST(G,M(G),s,c(s))$ 能够计算出点 s 的费用函数 $c(s)=(c_1(s),c_2(s),\cdots,c_d(s))$。进一步可得到图 G 的最优顶点染色的费用为 $\min\limits_{1\leqslant i\leqslant d}\{c_i(s)\}$。应用回溯法可以确定问题的最优解。

设点 v 是图 G 的任一顶点。点 v 的初始费用函数 $c(v)$ 定义如下：对 $i\in\{1,2,\cdots,d\}$ 置 $c_i(v)=$
$\begin{cases}0, & v\in S_i \text{ 或者 } v\notin S\\ +\infty, & v\in S\backslash S_i\end{cases}$。

算法第一步是调用子程序 $\overline{SEP}(G,s,\overline{T}_G)$ 得到图 G 的修正超结构 $\overline{T}_G$。然后要区分两种情形。

情形 1 $\overline{T}_G$ 有唯一顶点。

若 G 是树 T，则调用程序 $COST(T,M(T),s,c(s))$。

若 G 是非单边双连通图，则按如下方法得到点 s 的费用函数 $c(s)=(c_1(s),c_2(s),\cdots,c_d(s))$。

设点 u 是 $V(G)\backslash\{s\}$ 中的度最大点，其度数为 δ。选点 u 为裂开点。将点 u 裂成 δ 个点 $u_1,u_2,\cdots,u_\delta$，得到图 $Split(G,u)$。$Split(G,u)$ 由以下几部分组成：图 G 的导出子图 G-u，δ 个新点 $u_1,u_2,\cdots,u_\delta$，δ 条新边 $(v_1,u_1),(v_2,u_2),\cdots,(v_\delta,u_\delta)$。这里的 $v_1,v_2,\cdots,v_\delta$ 是图 G 中点 u 的 δ 个邻点。

将点 u 染颜色 j，$j\in\{1,2,\cdots,d\}$。对于 $a\in\{1,2,\cdots,\delta\}$，$i\in\{1,2,\cdots,d\}$，置 $w(u_a,v_a;j,i)=w(u,v_a;j,i)$。置 $c_j(u_1)=c_j(u),c_j(u_2)=\cdots=c_j(u_\delta)=0$。

接下来，调用程序 $COST(Split(G,u),M(Split(G,u)),s,c^j(s))$，得到 $c^j(s)=(c_1^j(s),c_2^j(s),\cdots,c_d^j(s))$。则点 s 必须染颜色 i 时图 G 的最优顶点染色的费用为：

$c_i(s)=\min\limits_{j=1,2,\cdots,d}\{c_i^j(s)\}$，$i\in\{1,2,\cdots,d\}$

情形 2 $\overline{T}_G$ 的顶点数不小于 2。

程序 $COST(G,M(G),s,c(s))$ 从 $\overline{T}_G$ 的叶子开始向上依次处理图 G 的分块。设当前正在处理的分块是 D，它是 $\overline{T}_G$ 的一片叶子。D 与图 G 其余部分的唯一公共点设为 q。设图 G 中染色多路割问题的实例 $M(G)$ 在 D 上的限制为 $M(D)$。

调用程序 $COST(D,M(D),q,c(q))$，对点 q 的费用函数 $c(q)$ 进行更新。完成之后，将分块 D 从 $\overline{T}_G$ 上删除。若 $\overline{T}_G$ 上有尚未处理的分块，则重复这一过程，否则程序 $COST(G,M(G),s,c(s))$ 终止。

现将程序 $COST(G,M(G),s,c(s))$ 扼要描述如下。

for $i\in\{1,2,\cdots,d\}$，$v\in V(G)$，do //初始化

置 $c_i(v)=\begin{cases}0, & v\in S_i \text{ 或者 } v\notin S\\ +\infty, & v\in S\backslash S_i\end{cases}$

```
程序 COST(G,M(G),s,c(s))：
begin
    SEP(G,s,T̄_G)；
    if |V(T̄_G)|=1 then begin
        if G 是树 then
            COST(T,M(T),s,c(s))；
        else begin //G 是非单边双连通图
            找到 V(G)\{s}中的度最大点 u；
            构造 Split(G,u)；
            for j∈{1,2,…,d}do
    COST(Split(G,u),M(Split(G,u)),s,c^j(s))
            for i∈{1,2,…,d}do
                c_i(s)← min_{j=1,2,…,d} {c_i^j(s)}
        end
    end
    else begin
            处理 T_G 的任一片叶子；
            if 该叶子是 G 的割点 then 从 T_G 删除
            else begin
                设该叶子是 G 的一个分块 D；
                if C 是 T_G 的树根 then q←s
                else q←D 在 T_G 中的父亲；
                COST(D,M(D),q,c(q))；
                从 T_G 中删除 D
            end
        end
end
```

下面来分析算法的时间复杂性。我们需要文献[17]中的一个引理。

引理 1[17] 设 G 是非单边双连通图，点 u 是 $V(G)\backslash\{s\}$ 中的度最大点，其度数为 δ。则有：

(1) $k(Split(G,u))\leqslant k(G)-\delta+1$。

(2) 若 $k(G)=1$，则 $\delta=2$；若 $k(G)\geqslant 2$，则 $\delta\geqslant 3$。

设算法运行过程中准备对树 T 调用程序 $COST(T,M(T),s,c(s))$。我们有：

引理 2 程序 $COST(T,M(T),s,c(s))$ 的运行时间是 $O(m_T d^{\lceil\frac{k}{2}\rceil+2})$，这里的 m_T 表示树 T 的边数。

证明：根据引理 1，树 T 中的新点（不在 $V(G)$ 中的点）由 G 中至多 $\left\lceil\frac{k}{2}\right\rceil$ 个点裂开而得。这些新点的不同染色的可能性共有 $d^{\lceil\frac{k}{2}\rceil}$ 种，故程序 $COST(T,M(T),s,c(s))$ 要 $d^{\lceil\frac{k}{2}\rceil}$ 次调用文献[17]中求解树中染色多路割问题的算法，该算法的运行时间为 $O(m_T d^2)$。故引理得证。

由于图 G 和图 $Split(G,u)$ 的边数相等，因此在图 G 的多次转换过程中，边数始终不变（这也正是我们选择用边数而不是用顶点数来分析时间复杂性的原因）。另一方面，设几乎树的顶点数和边数分别为 n 和 m，则对于固定参数 k，有 $m=O(n)$。于是我们得到：

定理 2 存在求解参数为 k 的几乎树中染色多路割问题的精确算法，其运行时间为 $O(nd^{\lceil\frac{k}{2}\rceil+2})$。

结束语 本文研究了染色多路割问题，给出了固定参数可解算法，所选参数 k 是使得图中任意双连通分支成为树所要拿掉的最大边数。可以继续研究这一问题的其他固定参数可解算法，或者对任意图给出好的近似算法。

参考文献

[1] Dahlhaus E, Johnson D S, Papadimitriou C H, et al. The complexity of multiterminal cuts[J]. SIAM Journal on Computing, 1994, 23: 864-894

[2] Garg N, Vazirani V V, Yannakakis M. Multiway cuts in directed and node weighted graphs[C]// Proceedings of ICALP' 94. 1994: 487-498

[3] Calinescu G, Karloff H, Rabani Y. An improved approximation algorithm for MULTIWAY CUT[J]. Journal of Computer and System Sciences, 2000, 60(3): 564-574

[4] Karger D R, Klein P N, Stein C, et al. Rounding algorithms for a geometric embedding of minimum multiway cut[J]. Mathematics of Operations Research, 2004, 29(3): 436-461

[5] Naor J, Zosin L. A 2-approximation algorithm for the directed multiway cut problem[C]// Proceedings of the 38th Annual Symposium on Foundations of Computer Science. New York: IEEE Computer Society, 1997: 548-553

[6] Frieze A, Kannan R. The regularity lemma and approximation schemes for dense problems[C]// Proceedings of the 37th Annual Symposium on Foundations of Computer Science. New York: IEEE Computer Society, 1996: 12-20

[7] Ford L R, Fulkerson D R. Flows in Networks[M]. New Jersey: Princeton University Press, 1962

[8] Papadimitriou C H, Steiglitz K. Combinatorial optimization : algorithms and complexity[M]. New Jersey: Prentice-Hall, 1982

[9] Erdos P L, Szekely L A. Evolutionary trees : an integer multi-commodity max-flow-min-cut theorem[J]. Advances in Applied Mathematics, 1992, 13: 375-389

[10] Erdos P L, Szekely L A. On weighted multiway cuts in trees[J]. Mathematical Programming, 1994, 65: 93-105

[11] Xin Xiao, Li Shuguang. Colored multiway cuts in trees of rings[C]// Proceedings of the 2009 International Forum on Information Technology and Applications. Chengdu, China: IEEE Computer Society, 2009: 236-239

[12] Xin Xiao, Li Shuguang. Colored multiway cuts in generalized tree networks[C]// Proceedings of the 2009 International Conference on Information Engineering. Taiyuan, China: IEEE Computer Society, 2009: 299-301

[13] Downey R, Fellows M. Parameterized Complexity (Monographs in Computer Science)[M]. Berlin: Springer, 1999

[14] Niedermeier R. Invitation to Fixed-Parameter Algorithms[M]. USA: Oxford University Press, 2006

[15] Aho A V, Hopcroft J E, Ullman J D. The design and analysis of computer algorithms[M]. Reading, MA: Addison-Wesley, 1974

[16] Tarjan R E. Depth first search and linear graph algorithms[J]. SIAM Journal on Computing, 1972, 1(2): 146-160

[17] Gurevich Y, Stockmeyer L, Vishkin U. Solving NP-hard problems on graphs that are almost trees and an application to facility location problems[J]. Journal of the ACM, 1984, 31(3): 459-473