3連結グラフと3辺連結グラフに対する3分割アルゴリズム

和田 幸一　川口 喜三男

名古屋工業大学

概要

グラフの3分割問題は、（1）n 点無向グラフ $G=(V,E)$、（2）互いに異なる V の3個の点 a_1,a_2,a_3、（3）$n_1+n_2+n_3=n$ となる自然数 $n_1,n_2,n_3(n_1\le n_2\le n_3)$ の入力に対して、V の3つの互いに素な点集合 V_1,V_2,V_3 で各 V_i が a_i を含み、その点数が n_i となりそれから誘導される部分グラフが連結となるものを求める問題である。グラフの3分割問題は入力グラフを制限しない一般の場合は NP-困難であるが、入力グラフ G が3連結である場合、常に解が存在し、$O(n^2)$ 時間で求めるアルゴリズムが知られている。

本稿では、3連結グラフに対して分割を点の部分集合に一般化した問題に対して $O(n^2)$ 時間のアルゴリズムを与え、このアルゴリズムは元のグラフの3分割問題を $O(m+(n_1+n_2)\cdot n)$ 時間で解くことを示す。ここで、m は G の辺数である。さらに、そのアルゴリズムを利用すると3辺連結グラフ $G=(V,E)$ に対して V の3つの互いに素な点集合 V_1,V_2,V_3 で各 V_i が a_i を含み、その点数が n_i となり V_i を含む部分グラフ G_i がそれぞれ互いに辺を共有しないものを $O(n^2)$ 時間で求まることを示す。

Efficient Algorithms for Tripartitioning Triconnected Graphs and 3-edge-connected Graphs

Koichi Wada and **Kimio Kawaguchi**
Nagoya Institute of Technology
Gokiso-cho, Syowa-ku, Nagoya 466, JAPAN
e-mail:[wada|kawaguchi]@laser.elcom.nitech.ac.jp

Abstract

This paper describes an efficient algorithm for partitioning a triconnected graph into three disjoint connected subgraphs, each of which contains a specified vertex and has a specified number of vertices. By using the algorithm, this paper also shows an algorithm for partitioning a 3-edge-connected graph into three edge-disjoint connected subgraphs which satisfy the conditions same as the case of triconnected graphs.

1 Introduction

The k-partition problem is described as follows.

Input:

(1) an undirected graph $G = (V, E)$ with $n = |V|$ vertices;

(2) k distinct vertices $a_i (1 \leq i \leq k) \in V$, $a_i \neq a_j (1 \leq i < j \leq k)$; and

(3) k natural numbers $n_1, n_2, \ldots, n_k$ such that $\sum_{i=1}^{k} n_i = n$.

Output: a partition $V_1 \cup V_2 \cup \ldots \cup V_k$ of vertex set V such that for each $i (1 \leq i \leq k)$

(a) $a_i \in V_i$;

(b) $|V_i| = n_i$; and

(c) each of V_i induces a connected subgraph of G.

We extend the k-partition problem in the following and we call it *the vertex-subset k-partition problem.*

Input:

(1) an undirected graph $G = (V, E)$ with $n = |V|$ vertices;

(2) a vertex subset $V' (\subseteq V)$ with $n' = |V'| \geq k$;

(3) k distinct vertices $a_i (1 \leq i \leq k) \in V'$, $a_i \neq a_j (1 \leq i < j \leq k)$; and

(4) k natural numbers $n_1, n_2, \ldots, n_k$ such that $\sum_{i=1}^{k} n_i = n'$.

Output: a partition $V_1 \cup V_2 \cup \ldots \cup V_k$ of vertex set V and a partition $V_1' \cup V_2' \cup \ldots \cup V_k'$ of vertex set V' such that for each $i (1 \leq i \leq k)$

(a) $a_i \in V_i'$;

(b) $|V_i'| = n_i$;

(c) $V_i' \subseteq V_i$ and

(d) each of V_i induces a connected subgraph of G.

This problem is extended in the sense that a specified vertex subset is partitioned. If $V' = V$ then it coincides with the original k-partition problem.

The k-partition problem is NP-hard in general even if k is limited to 2 [2]. Györi and Lovász independently showed that the k-partition problem has a solution if the input graph is k-connected [5, 8]. The bipartition problem($k = 2$) can be solved in $O(m)$ time [10] and the tripartition problem($k = 3$) can be solved in $O(n^2)$ time [11]. Throughout this paper, n and m denote the number of vertices in G and the number of edges in G, respectively. It is an open problem whether or not we can solve the k-partition problem for $k \geq 4$ in polynomial time if the input graph is k-connected [11].

In this paper, we present an algorithm to solve the vertex-subset tripartition problem for triconnected graphs. Our algorithm utilizes a nonseparating ear decomposition for a triconnected graph and it runs in $O(n^2)$ time. We also show that this algorithm solves the tripartition problem for triconnected graphs in $O(m + (min_{1 \leq i < j \leq 3}(n_i + n_j)) \cdot n)$ time.

Furthermore, we characterize k-edge-connected graphs by using a vertex-partition. In general the k-partition problem does not have a solution for k-edge-connected graphs. However by using Györi and Lovász's result the following edge-partition can be done[5]:

Given a k-edge-connected graph $G = (V, E)$, k edges $e_1, e_2, \ldots, e_k \in E$ and k positive integers $m_1, m_2, \ldots, m_k$ such that $\sum_{i=1}^{k} m_i = |E|$, there exists an edge-partition $E = E_1 \cup E_2 \cup \ldots \cup E_k$

such that $e_i \in E_i$, $|E_i| = m_i$ and $G_i = (V(E_i), E_i)$ is connected for each $i(1 \leq i \leq k)$, where $V(E')$ denotes the set of vertices incident with at least one member of E'.

In this paper, we propose the k-partition problem obtained by replacing the condition (c) with the following condition (c-v)((c-e)).

(c-v)((c-e)) For any pair (v_i, u_i) of vertices in any V_i and any pair (v_j, u_j) of vertices in any $V_j(1 \leq i < j \leq k)$, there exist a path P_i between v_i and u_i and a path P_j between v_j and u_j such that P_i and P_j are vertex-disjoint(edge-disjoint).

We refer to this problem as *the k-partition problem with respect to vertex-disjointness(edge-disjointness)*. Since a solution for the k-partition problem is also a solution for the k-partition problem with respect to vertex-disjointness from the definition, the k-partition problem with respect to vertex-disjointness has a solution if the input graph is k-connected. Although we conjecture that the k-partition problem with respect to edge-disjointness for k-edge-connected graphs has a solution for any $k(\geq 2)$, it is not known so far in general. We show that if we can solve the vertex-subset k-partition problem for k-connected graphs, we can solve the k-partition problem with respect to edge-disjointness. By using the result and the algorithms shown here, we also show the cases in which $k = 2$ and $k = 3$ can be solved in $O(m)$ and $O(n^2)$ time, respectively.

2 Preliminary

We deal with an undirected graph $G = (V, E)$ with vertex set V and edge set E. For a graph G, the vertex set is denoted by $V(G)$. A graph G is *k-connected(k-edge-connected)* if there exist k node-disjoint(edge-disjoint) paths between every pair of distinct nodes in G. Usually 2-connected graphs are called *biconnected graphs* and 3-connected graphs are called *triconnected graphs*. The *distance* between nodes x and y in G is the length of the shortest path between x and y and is denoted by $dis_G(x, y)$. For a graph $G = (V, E)$ and a vertex subset V', the induced subgraph is denoted by $G[V']$. For two graphs $G = (V, E)$ and $G' = (V', E')$, the graph $(V \cup V', E \cup E')$ is denoted by $G \cup G'$.

3 Nonseparating Ear Decomposition and s-t Numbering

In order to solve the tripartition problem efficiently, we utilize the concepts of a nonseparating ear decomposition which characterizes triconnected graphs and an s-t numbering for biconnected graphs.

3.1 Nonseparating Ear Decomposition

An *ear decomposition* of a biconnected graph $G = (V, E)$ is a decomposition $G = P_0 \cup P_1 \cup \ldots \cup P_k$, where P_0 ia a cycle and $P_i(1 \leq i \leq k)$ is a path whose end vertices are distinct and only end vertices are in common with $P_0 \cup \ldots \cup P_{i-1}$. Each P_i is called an open ear.

Given an ear decomposition $P_0 \cup P_1 \cup \ldots \cup P_k$ of G, let $V_i = V(P_0) \cup \ldots \cup V(P_i)$, let $G_i = G[V_i]$ and let $\overline{G_i} = G[V - V_i]$ for each $i(1 \leq i \leq k)$.

We say that $G = P_0 \cup P_1 \cup \ldots \cup P_k$ is an *ear decomposition through edge* (a, b) *and avoiding vertex* c, if the cycle P_0 contains the edge (a, b) and the last ear of length greater than one, say P_q, has c as its only internal vertex.

An ear decomposition $P_0 \cup P_1 \cup \ldots \cup P_k$ of a graph G through edge (a, b) and avoiding vertex c is a *nonseparating ear decomposition* if for all $i(1 \leq i < m)$, each graph $\overline{G_i}$ is connected and each internal vertex of the ear P_i has a neighbour in $\overline{G_i}$.

Proposition 1 [1] *For a triconnected graph $G = (V, E)$, any edge $(a, b) \in E$ and any vertex $c(\neq a, b) \in V$, a nonseparating ear decomposition $P_0 \cup P_1 \cup \ldots \cup P_q$ through edge (a, b) and avoiding vertex c can be constructed in $O(|V| \cdot |E|)$, where the path P_q has the vertex c as its only internal one. In particular, the cycle P_0 and each path $P_i(1 \leq i \leq q)$ can be constructed in $O(|E|)$.*

3.2 s-t Numbering

Given an edge (s, t) of a biconnected graph $G = (V, E)$, a bijective function $g : V \rightarrow \{1, 2, \ldots, |V| = n\}$ is called an *s-t numbering* if the following conditions are satisfied:

- $g(s) = 1$, $g(t) = n$ and
- Every node $v \in V - \{s, t\}$ has two adjacent nodes u and w such that $g(u) < g(v) < g(w)$.

Proposition 2 [3] *Let $G = (V, E)$ be a biconnected graph. For any edge $(s, t) \in E$, an s-t numbering can be computed in $O(|E|)$ time.*

The following lemma holds from the definition of the s-t numbering.

Lemma 1 *Let g be an s-t numbering for a biconnected graph $G = (V, E)$ and an edge (s, t). For any $i(1 \leq i \leq |V|)$, the two induced subgraphs $G[\{g^{-1}(j) | 1 \leq i\}]$ and $G[\{g^{-1}(j) | i+1 \leq j \leq |V|\}]$ are connected, where g^{-1} is the inverse function of g.*

The following theorem is easily shown from Lemma 1 and Proposition 2.

Theorem 1 *The vertex-subset bipartition problem can be solved in $O(m)$.*

4 Tripartition of Triconnected Graphs

In this section, we present an efficient algorithm PART3 for solving the vertex-subset tripartition problem for a triconnected graph G. We may assume without loss of generality that $n_1 \leq n_2 \leq n_3$ for a given input.

The algorithm is as follows:

Algorithm PART3$(G = (V, E); V'; a_1, a_2, a_3; n_1, n_2, n_3)$

begin
if $(a_1, a_2) \notin E$ **then** $E \leftarrow E \cup \{(a_1, a_2)\}$;

Let $P_0 \cup P_1 \cup \ldots \cup P_q$ be a nonseparating ear decomposition
through edge (a_1, a_2) and avoiding vertex a_3 for the graph G;
{Note that this algorithm does not construct all paths of the ear decomposition.}
$i \leftarrow 1$;
while $|V(G_i) \cap V'| < n_1 + n_2$ **do** $i \leftarrow i + 1$;
if $|V(G_i) \cap V'| = n_1 + n_2$ **then**
    begin
        Let g be an a_1-a_2 numbering of G_i and
        let n_1' satisfy $n_1 = |\{g^{-1}(j) | 1 \leq j \leq n_1'\} \cap V'|$ and $g^{-1}(n_1') \in V'$;
        $V_1 \leftarrow \{g^{-1}(j) | 1 \leq j \leq n_1'\}$;
        $V_2 \leftarrow \{g^{-1}(j) || G_i| = n_1 + n_2 \geq j \geq n_1' + 1\}$;
        $V_3 \leftarrow V(\overline{G_i})$;
        return(V_1, V_2, V_3)
    end
    else $\{|V(G_i) \cap V'| > n_1 + n_2 > |V(G_{i-1}) \cap V'|\}$
        begin
            Let g be an a_1-a_2 numbering of G_{i-1},
            let $P_i = (x_0, \ldots, x_r)$ such that $g(x_0) < g(x_r)$,
            let n_1' satisfy $n_1 = |\{g^{-1}(j) | 1 \leq j \leq n_1'\} \cap V'|$ and $g^{-1}(n_1') \in V'$ and
            let n_2' satisfy $n_2 = |\{g^{-1}(j) || G_{i-1}| \geq j \geq |G_{i-1}| - n_2' + 1\} \cap V'|$ and
            $g^{-1}(|G_{i-1}| - n_2' + 1) \in V'$;
            $U_1 \leftarrow \{g^{-1}(j) | 1 \leq j \leq n_1'\}$;
            $U_2 \leftarrow \{g^{-1}(j) || G_{i-1}| \geq j \geq |G_{i-1}| - n_2' + 1\}$;
            $I \leftarrow U_1 \cap U_2$;
            if $x_0 \in U_1 - I$ **then**
                begin
                    Let $j = |I \cap V'|$ and
                    let j' satisfy $j = |\{x_1, \ldots, x_{j'}\} \cap V'|$ and $x_{j'} \in V'$;
                    $V_1 \leftarrow (U_1 - I) \cup \{x_1, \ldots, x_{j'}\}$;
                    $V_2 \leftarrow U_2$;
                    $V_3 \leftarrow V(\overline{G_i}) \cup \{x_{j'+1}, \ldots, x_{r-1}\}$;
                    return(V_1, V_2, V_3)
                end
            else if $x_r \in U_2 - I$ **then**
                begin
                    Let $j = |I \cap V'|$ and
                    let j' satisfy $j = |\{x_{r-j'}, \ldots, x_{r-1}\} \cap V'|$ and $x_{r-j'} \in V'$;
                    $V_1 \leftarrow U_1$;
                    $V_2 \leftarrow (U_2 - I) \cup \{x_{r-j'}, \ldots, x_{r-1}\}$;
                    $V_3 \leftarrow V(\overline{G_i}) \cup \{x_1, \ldots, x_{r-j'-1}\}$;
                    return(V_1, V_2, V_3)
                end

else $\{x_0, x_r \in I\}$
begin
Let $I = \{z_1, \ldots, z_{|I|}\}$ and let $x_0 = z_s$ and $x_r = z_t$ $(1 \leq s < t \leq |I|)$,
let $j_1 = |\{z_{s+1}, \ldots, z_{|I|}\} \cap V'|$,
let $j_2 = |\{z_1, \ldots, z_s\} \cap V'|$,
let j'_1 satisfy $j_1 = |\{x_1, \ldots, x_{j'_1}\} \cap V'|$ and $x_{j'_1} \in V'$ and
let j'_2 satisfy $j_2 = |\{x_{r-j'_2}, \ldots, x_{r-1}\} \cap V'|$ and $x_{r-j'_2} \in V'$;
$V_1 \leftarrow (U_1 - I) \cup \{z_1, \ldots, z_s\} \cup \{x_1, \ldots, x_{j'_1}\}$;
$V_2 \leftarrow (U_2 - I) \cup \{z_{s+1}, \ldots, z_{|I|}\} \cup \{x_{r-1}, \ldots, x_{r-j'_2}\}$;
$V_3 \leftarrow V(\overline{G_i}) \cup \{x_{j'_1+1}, \ldots, x_{r-j'_2-1}\}$;
return(V_1, V_2, V_3)
end
end

The correctness of the algorithm is derived from the definition of the nonseparating ear decomposition and Lemma 1. Since a spanning triconnected subgraph $G' = (V, E')$ with $|E'| = O(|V|)$ can be computed in $O(|E|)$ time for a triconnected graph $G = (V, E)$ [9, 11], the next theorem is obtained from Propositions 1 and 2.

Theorem 2 *The vertex-subset tripartition problem for triconnected graphs can be solved in* $O(m + i \cdot n)$*, where* i *denotes the number of ears constructed in PART3.*

The next corollaries are easily obtained from the construction of the algorithm.

Corollary 1 *The vertex-subset tripartition problem for triconnected graphs can be solved in* $O(n^2)$*.*

Corollary 2 *The tripartition problem for triconnected graphs can be solved in* $O(m + (n_1 + n_2) \cdot n)$*, where* $n_1 \leq n_2 \leq n_3$ *for the given input.*

Corollary 3 *If* $n_1 + n_2 = O(1)$ *or* $dis_G(a_1, a_2) \geq n_1 + n_2$ *then the tripartition problem for triconnected graphs can be solved in linear time.*

5 The k-partition with respect to Edge-disjointness

In this section, instead of solving the k-partition problem with respect to edge-disjointness directly, we compute mutually k edge-disjoint subgraphs $G_i = (U_i, E_i)(1 \leq i \leq k)$ such that

(a) $a_i \in V_i$,
(b) $V_i \subseteq U_i$ and
(c) G_i is connected.

If we can find the mutually k edge-disjoint subgraphs stated above, we can easily show that the vertex-partition $V = V_1 \cup V_2 \cup \ldots \cup V_k$ is a solution for the k-partition problem with respect to edge-disjointness.

We utilize the method transforming k-edge-connected graphs into k-connected graphs [4].

Let $k \geq 2$. Given a graph $G = (V, E)$, define the graph $\varphi_k(G) = (\varphi(V), \varphi(E))$ as follows. For every vertex $v \in V$, there are $k-2$ vertices $\varphi(v_1), \varphi(v_2), \ldots, \varphi(v_{k-2})$ in $\varphi(V)$. These vertices are called *node-vertices* of $\varphi_k(G)$. For every edge $e \in E$, there is a vertex $\varphi(e)$ in $\varphi(V)$. This vertex is called *arc-vertex* of $\varphi_k(G)$. Note that if $k = 2$, there is no node-vertex in $\varphi(V)$.

The edge set $\varphi(E)$ is defined as follows: Let v be any vertex in V and $u_0, u_1, \ldots, u_{d-1}$ be the vertices adjacent to v. Let $e_i = (v, u_i)(0 \leq i \leq d-1)$. Then there are edges $(\varphi(e_i), \varphi(e_{(i+1) mod\, d}))(0 \leq i \leq d-1)$ and $(\varphi(e_i), \varphi(v_j))(0 \leq i \leq d-1, 1 \leq j \leq k-2)$ in $\varphi(E)$. Note that if $d = 2$, there is an edge $(\varphi(e_0), \varphi(e_1))$ in $\varphi(E)$.

From the definition $\varphi_k(G)$ has $(k-2)|V| + |E|$ vertices and $O(k|E|)$ edges and it can be computed in $O(k(|V| + |E|))$.

Proposition 3 [4] *For any $k(\geq 2)$, G is k-edge-connected if and only if $\varphi_k(G)$ is k-connected.*

Let U be a vertex subset of $\varphi_k(G)$. The subgraph $\varphi^{-1}(U) = (\varphi_v^{-1}(U), \varphi_e^{-1}(U))$ of $G = (V, E)$ is defined to be
$\varphi_v^{-1}(U) = \{v | \varphi(v) \in U \text{ and } v \in V\} \cup \{ \text{ endpoints of } e | \varphi(e) \in U \text{ and } e \in E\}$ and
$\varphi_e^{-1}(U) = \{e | \varphi(e) \in U \text{ and } e \in E\}$.

The next lemma can be easily shown.

Lemma 2 *If the induced subgraph $\varphi_k(G)[U]$ is connected, then the graph $\varphi^{-1}(U)$ is connected.*

From the Proposition 3 and Lemma 2, the following theorem can be proved.

Theorem 3 *If the vertex-subset k-partition problem can be solved, the k-partition problem with respect to edge-disjointness can be solved.*

Combining Theorem 3 with Theorem 1 and Corollary 1, we can prove the following theorems.

Theorem 4 *The bipartition problem with respect to edge-disjointness can be solved in $O(m)$ time.*

Theorem 5 *The tripartition problem with respect to edge-disjointness can be solved in $O(n^2)$ time.*

6 Concluding Remarks

We present an algorithm which solves the vertex-subset tripartition problem for triconnected graphs. This algorithm solves the tripartition problem for triconnected graphs efficiently. Compared with the previous result, the worst case of our algorithm is the same as that in [11], since there is a case that $n_1 + n_2 = \Omega(n)$. However, we have applications which satisfies $n_1 + n_2 = o(n)$. For example, the tripartition is necessary in order to define efficient fault-tolerant routings for triconnected graphs and in that case it satisfies that $n_1 + n_2 = O(1)$ or $n_1 + n_2 = O(\log n)$ [6, 7]. Thus, the tripartition can be solved in $O(m)$ or $O(m + n \log n)$ time.

The k-partition problem with respect to edge-disjointness is related to fault-tolerant routings for k-edge-connected graphs [12].

It still remains as a further study to solve the k-partition problem($k \geq 4$) for k-connected graphs in polynomial time and to solve the vertex-subset k-partition problem and/or the k-partition problem with respect to edge-disjointness($k \geq 4$).

Acknowledgement This research is partly supported by a Scientific Research Grant-In-Aid(04750320,1992) from the Ministry of Education, Science and Culture, Japan .

References

[1] J. Cherian and S.N. Maheshwari : "Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs," *Journal of Algorithms*,9,507–537(1988).

[2] M.E Dyer and A.M. Frieze : "On the complexity of partitioning graph into connected subgraphs," *Discrete Appli. Math.*,10,139–153(1985).

[3] S.Evens: Graph algorithms, Computer Science Press, Potomac, MD(1979).

[4] Z. Galil and G.F. Italiano: "Reducing edge connectivity to vertex connectivity," *SIGACT NEWS*, 22,1,57-61(1991).

[5] E. Györi: "On division of connected subgraphs," in: Combinatorics(Proc. 5th Hungarian Combinational Coll., 1976, Keszthely) North-Holland, Amsterdam,485–494(1978).

[6] M.Imase and Y.Manabe: "Fault tolerant routings in a κ-connected network," *Information Processing Letters*, 28,4,171-175(1988).

[7] K.Kawaguchi and K.Wada: "New results in graph routing," to appear in *Information and Computation*(1992).

[8] L. Lovász: "A homology theory for spanning trees of a graph," *Acta math. Acad. Sci. Hunger*,30,241–251(1977).

[9] H.Nagamochi and T.Ibaraki : "A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph," *Algorithmica*,7,5/6,583–596(1992).

[10] H.Suzuki, N.Takahashi and T.Nishizeki: "A linear algorithm for bipartition of biconnected graphs," *Information Processing Letters*,33,5(1990).

[11] H.Suzuki, N.Takahashi T.Nishizeki, H. Miyano and S.Ueno: "An algorithm for tripartitioning 3-connected graphs," *Information Processing Society of Japan*, 31,5,584–592(1990),in Japanese.

[12] K.Wada, T.Shibuya, E.Shamoto and K.Kawaguchi: "Fault-tolerant routings for k-edge-connected graphs," *Paper of Technical Group*, COMP91-1,IECE(1990),in Japanese.