IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024DMP0012

Publicized:2025/01/31

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

(t, s)-completely Independent Spanning Trees

Shin-ichi NAKANO[†], ***Member***

SUMMARY In this paper we first define (t, s)-completely independent spanning trees, which is a generalization of completely independent spanning trees. A set of t spanning trees of a graph is (t, s)-*completely independent* if, for any pair of vertices u and v, among the set of t paths from u to v in the t spanning trees, at least $s \le t$ paths are internally disjoint. By (t, s)-completely independent spanning trees, one can ensure any pair of vertices can communicate each other even if at most $s - 1$ vertices break down. We prove that every maximal planar graph has a set of $(3, 2)$-completely independent spanning trees, every tri-connected planar graph has a set of $(3, 2)$-completely independent spanning trees, and every 3D grid graph has a set of $(3, 2)$-completely independent spanning trees. Also one can compute them in linear time.

key words: *Algorithm, Independent Spanning Trees, Spanning Tree*

1. Introduction

Two paths from vertex u to v are *internally disjoint* if they have no common internal vertex.

A set of t spanning trees of a graph is *completely independent* if, for any pair of vertices u and v, the set of t paths from u to v in the t spanning trees are internally disjoint (and edge disjoint) [6]. A necessary and sufficient condition for the existence of a set of t completely independent spanning trees is known [5], [6].

In this paper we generalize the concept of completely independent spanning trees as follows. A set of t spanning trees of a graph is (t, s)-*completely independent* if, for any pair of vertices u and v, among the set of t paths from u to v in the t spanning trees, at least $s \le t$ paths are internally disjoint. By (t, s)-completely independent spanning trees, one can ensure any pair of vertices can communicate each other even if at most $s - 1$ vertices break down. The original completely independent spanning trees are (t, t)-complete spanning trees.

Intuitively, when we have t interconnection (spanning tree) networks, we want to ensure $s \le t$ of separate (independent) routes for each pair of vertices. The original completely spanning tree concept may be too strong for some applications and may fail to construct them, however (t, s)-completely independent tree concept may be a flexible choice for some applications and may increase the chance to construct them.

In this paper, we first design an algorithm to construct a set of $(3, 2)$-completely independent spanning trees in a given maximal planar graph based on the realizer [20], then design an algorithm to construct a set of $(3, 2)$-completely independent spanning trees in a given tri-connected planar graphs based on the canonical decomposition [12], then design an algorithm to construct a set of $(3, 2)$-completely independent spanning trees in a given 3D grid graph. Those algorithms are simple and run in $O(n)$ time, where n is the number of vertices of the given graph.

Another generalization of the independent spanning tree, where each edge is shared by more than one tree and each vertex is shared by more than one tree, is disscussed in [4].

The remainder of this paper is organized as follows. Section 2 gives some definitions and two basic lemmas. In Section 3 we design our first algorithm which constructs a set of $(3, 2)$-completely independent spanning trees in a given maximal planar graph. In Section 4 we design our second algorithm which constructs a set of $(3, 2)$-completely independent spanning trees in a given tri-connected planar graph. In Section 5 we design our third algorithm which constructs a set of $(3, 2)$-completely independent spanning trees in a given 3D grid graph. Finally Section 6 is a conclusion.

A preliminary version of the paper is presented at [19].

2. Preliminaries

A tree is a connected graph with no cycle. *A rooted tree* is a tree with a designated vertex as *the root*. Given a graph G, *a spanning tree* of G is a subgraph of G which is a tree and contains all vertices of G.

A graph is *planar* if it can be embedded on the plane so that no two edges intersect geometrically except at a vertex to which they are both incident. A *plane* graph is a planar graph with a fixed plane embedding.

A graph G with more than k vertices is k-*connected* if removal of any $k - 1$ vertices results in a connected graph.

A 3D *grid graph* with size $L_x \times L_y \times L_z$ is the graph consisting of vertex set $\{(x, y, z) | 0 \le x \le L_x, 0 \le y \le L_y, 0 \le z \le L_z$, and x, y, z are integers $\}$ and edge set $\{\{(x_1, y_1, z_1), (x_2, y_2, z_2)\} \mid |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| = 1\}$.

Independent spanning trees

Let n be the number of vertices of a given graph G. A set of t rooted spanning trees with a common root r of a graph G is *independent* if, for any vertex v, the set of t paths from r to v in the t spanning trees are internally disjoint. It is conjectured that, for any $k \ge 1$, every k-connected graph G has a

[†]The author is with Gunma University, Maebashi-Shi, 371-8510

set of k independent spanning trees rooted at any vertex [13], [21]. If G is bi-connected then one can find two independent spanning trees in linear time by the st-numbering[1], [11]. If G is tri-connected then one can find three independent spanning trees in $O(n^2)$ time by the ear-decomposition [1], [2]. If G is four-connected then one can find four independent spanning trees in $O(n^3)$ time by the chain-decomposition [3]. If G is a tri-connected planar graph then one can find three independent spanning trees in linear time by the canonical decomposition [1]. If G is a four-connected planar graph then one can find four independent spanning trees in $O(n^3)$ time [9] then in linear time [14], [15]. If G is a five-connected planar graph then one can find five independent spanning trees in polynomial time [10]. If G is a five-connected maximal planar graph then one can find five independent spanning trees in linear time by the 5-canonical decomposition [16], [17].

Completely independent spanning trees

A set of spanning trees is *completely independent* if, for any pair of vertices u and v, the set of paths from u to v in the spanning trees are internally disjoint (and edge disjoint) [6]. A necessary and sufficient condition for the existence of k completely independent spanning trees is known [5], [6].

(t, s)-completely independent spanning trees

A set of t spanning trees is *(t, s)-completely independent* if, for any pair of vertices u and v, among the set of t paths from u to v in the t spanning trees, at least s paths are internally disjoint.

Realizer

Every maximal planar graph with $n \geq 4$ vertices is tri-connected, and has a unique embedding on a sphere only up to mirror copy [7]. In the embedding each face has exactly three vertices on the boundary. Given a maximal planar graph G with n vertices, we can compute a maximal plane graph G' corresponding to G in linear time [8]. Let r_r, r_b, r_y be the three vertices on the outer face of G', and assume that they appear on the outer face clockwise in this order. A partition $\{E_r, E_b, E_y\}$ of inner edges of G' is called a *realizer* of G' if the following conditions (re1)–(re3) are satisfied [20]. See an example in Fig.1(b). Let T_r be the tree induced by all edges in E_r. Similarly, let T_b and T_y be the trees induced by all edges in E_b and E_y, respectively.

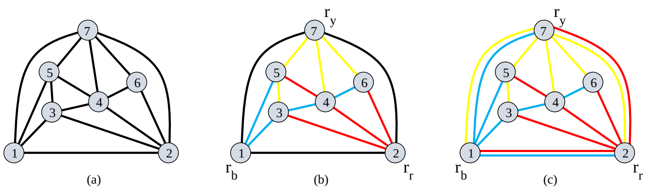

Fig. 1 (a) A maximal plane graph G (b) a realizer of G (c) three spanning thees of G.

(**re1**) T_r is a tree spanning all inner vertices of G and r_r. Similarly, T_b is a tree spanning all inner vertices of G and r_b, and T_y is a tree spanning all inner vertices of G and r_y.
(**re2**) Every inner edge incident to r_r is in T_r. Similarly, every inner edge incident to r_b is in T_b, and every inner edge incident to r_y is in T_y.
(**re3**) Define the orientation of each inner edge as follows. In tree T_r, we regard r_r as the root of T_r, and orient each edge in T_r from a child to its parent. Similarly, we regard r_b and r_y as the roots of T_b and T_y, respectively, and define the orientation of each inner edge in T_b and T_y from a child to its parent.
Then, for each inner vertex v, all edges incident to v appear around v clockwise in the following order. (See Fig. 2)

Exactly one outgoing edge in T_r.
Zero or more incoming edges in T_y.
Exactly one outgoing edge in T_b.
Zero or more incoming edges in T_r.
Exactly one outgoing edge in T_y.
Zero or more incoming edges in T_b.

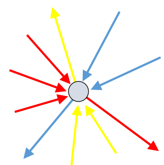

Fig. 2 Illustration for the condition of a realizer.

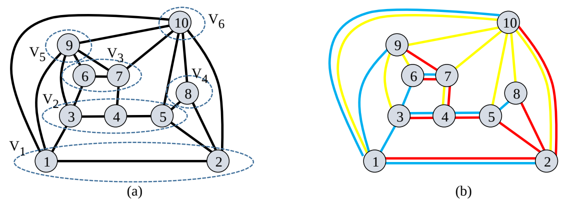

Fig. 3 An illustration for (cd3).

We sometimes regard the set of three rooted trees T_r, T_b, T_y a realizer of G. The above explanation is from [18]. The following lemma is known.

Lemma 1. *[20] Every maximal plane graph has a realizer. One can find it in linear time.*

Canonical decomposition

Every tri-connected planar graph has a unique embedding on a sphere only up to mirror copy [7]. Given a tri-connected planar graph G with n vertices, we can compute a plane graph G' corresponding to G in linear time [8]. Let v_1, v_2, v_n be the three consecutive vertices on the outer face of G', and they appear on the outer face counterclockwise in order (v_1, v_2, v_n). A partition $V_1, V_2, \cdots, V_h$ of vertices of

G' is called a *canonical decomposition* of G' if the following conditions (cd1)–(cd4) are satisfied [12]. See an example in Fig. 6(a). Let G_i be the subgraph of G' induced by $V_1 \cup V_2 \cup \cdots \cup V_i$, and Let $\overline{G_i}$ be the subgraph of G' induced by $V_{i+1} \cup V_{i+2} \cup \cdots \cup V_h$.

(**cd1**) $V_1 = \{v_1, v_2\}$.
(**cd2**) For each $i = 2, 3, \cdots, h$, G_i is bi-connected.
(**cd3**) For each $i = 2, 3, \cdots, h-1$, V_i is either (1) a vertex u on the outer face of G_i having at least one neighbor in $\overline{G_i}$ (See Fig.3(a)), or (2) consecutive vertices $\{u_\ell, u_{\ell+1}, \cdots, u_r\}$ on the outer face of G_i such that each vertex has degree two in G_i and has at least one neighbor in $\overline{G_i}$ (See Fig.3(b)).
(**cd4**) $V_h = \{v_n\}$.

One can regard the canonical decomposition of a maximal plane graph is a realizer.

The following lemma is known.

Lemma 2. *[12] Every tri-connected plane graph has a canonical decomposition. One can find it in linear time.*

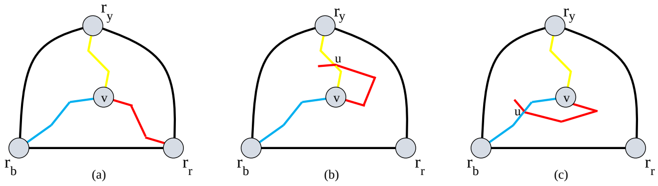

Fig. 4 Illustration for Lemma 3.

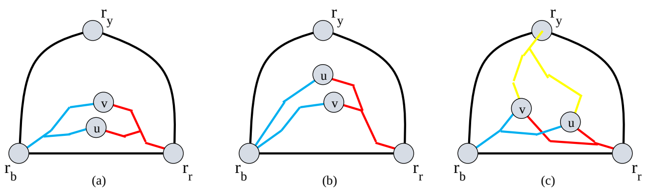

Fig. 5 Illustration for Theorem 1.

3. Algorithm I

In this section we design a linear time algorithm to construct a set of $(3, 2)$-completely independent spanning trees in a given maximal planar graph with n vertices. The algorithm is based on the realizer [20].

Let T_r, T_b, T_y be a realizer of a maximal planar graph. We have the following lemma.

Lemma 3. *[20][Theorem 4.6] For any inner vertex v, let S be the set of three paths consisting of (1) the path from v to r_r in T_r, (2) the path from v to r_b in T_b and (3) the path from v to r_y in T_y. Then any two paths in S share only v.*

Proof. Assume otherwise for a contradiction. If the path from v to r_r in T_r and the path from v to r_y in T_y share a vertex except v, then let $u \neq v$ be the first such vertex in the path from v to r_r in T_r (See Fig.4 (b)), then, by the planarity, (re3) is not satisfied at u. (A red path never crosses a yellow path from right to left.) A contradiction.

Similar for the other cases. See Fig.4 (c). □

Given a realizer of a plane graph G' corresponding to a maximal planer graph G, let T'_r be the spanning tree of G' rooted at r_r consisting of T_r and two edges (r_y, r_r) and (r_b, r_r). Similarly, let T'_b be the spanning tree of G' rooted at r_b consisting of T_b and two edges (r_r, r_b) and (r_y, r_b), and T'_y be the spanning tree of G' rooted at r_y consisting of T_y and two edges (r_b, r_y) and (r_r, r_y). See an example in Fig. 1(c).

We have the following theorem.

Theorem 1. *T'_r, T'_b, T'_y are $(3, 2)$-completely independent spanning trees.*

Proof. For an inner vertex v, let $Y(v)$ be the region surrounded by the path from v to r_r in T'_r, the path from v to r_b in T'_b and edge (r_r, r_b). Similarly, let $R(v)$ be the region surrounded by the path from v to r_b in T'_b, the path from v to r_y in T'_y and edge (r_b, r_y) and $B(v)$ be the region surrounded by the path from v to r_y in T'_y, the path from v to r_r in T'_r and edge (r_y, r_r).

Given two vertices u and v in G', let S be the set of three paths consisting of the path from u to v in T'_r, the path from u to v in T'_b and the path from u to v in T'_y. Then we show that some pair of paths in S are internally disjoint.

If $\{u, v\} \subset \{r_r, r_b, r_y\}$ then the claim holds. Assume otherwise.

We have the following three cases to consider.

Case 1: $Y(v)$ contains u. See Fig. 5(a).

The path from u to v in T'_r and the path from u to v in T'_b are internally disjoint. (If the path from u to v in T'_r and the path from u to v in T'_b are not internally disjoint, then, similar to the proof of Lemma 3, we can show that there is a vertex where (re3) does not satisfied. A contradiction.)

Case 2: $Y(u)$ contains v. See Fig. 5(b).

The path from u to v in T'_r and the path from u to v in T'_b are internally disjoint.

Similar to Case 1.

Case 3: Otherwise.

Then either $B(v)$ contains u (See Fig. 5(c)) or $R(v)$ contains u.

The path from u to v in T'_b and the path from u to v in T'_y are internally disjoint. (Also the path from u to v in T'_r and the path from u to v in T'_y are internally disjoint.) Similar to the proof of Lemma 3. □

4. Algorithm II

In this section we design a linear time algorithm to construct a set of $(3, 2)$-completely independent spanning trees in a

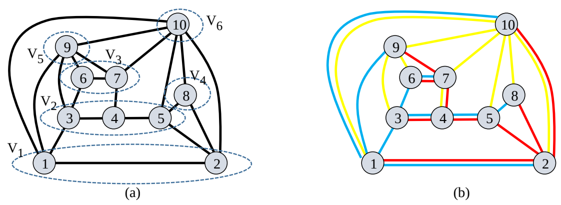

Fig. 6 (a) A canonical decomposition of a tri-connected plane graph. (b) Three spanning trees.

given tri-connected planar graph with n vertices. The algorithm is based on the canonical decomposition[12].

Given a tri-connected planar graph G, let G' be the corresponding plane graph, and $V_1, V_2, \cdots V_h$ its canonical decomposition. We define, for each vertex v of G, three outgoing edges $ll(v)$, $rl(v)$ and $h(v)$ from v, as follows. We call those edges left leg, right leg and head of v, and intuitively each left leg points lower left, each right leg points lower right and each head points upward.

For vertex in $V_1 = \{v_1, v_2\}$ we define as follows. $rl(v_1) = (v_1, v_2)$ and $ll(v_2) = (v_2, v_1)$. And v_1 has no left leg and v_2 has no right leg.

For each vertex $v \in V_2 \cup V_3 \cup \cdots \cup V_h$, we have the following two cases.

If $|V_i| = 1$ then let $V_i = \{v\}$ and $u_\ell, u_{\ell+1}, \cdots, u_r$ be the neighbor of v on the outer face of G_{i-1} and assume that they appear in this order clockwise. We define $ll(v) = (v, u_\ell)$ and $rl(v) = (v, u_r)$.

If $|V_i| > 1$ then let $V_i = \{v_1, v_2, \cdots, v_k\}$, and they appear in this order clockwise on the outer face of G_i and v_0 and v_{k+1} be the neighbor of v_1 and the neighbor of v_k on the outer face of G_{i-1}, respectively. Then, for each $j = 1, 2, \cdots k$, we define $ll(v_i) = (v_i, v_{i-1})$ and $rl(v_i) = (v_i, v_{i+1})$.

For $v_n \in V_h$ we define $ll(v_n) = v_1$ and $rl(v_n) = v_2$.

Also, for each vertex $v \in V_1 \cup V_2 \cup \cdots \cup V_{h-1}$, let $u \in V_{h'}$ be the neighbor of v with the maximum h'. (For tie we choose the rightmost vertex.) We define $h(v) = (v, u)$. For $v_n \in V_h$, v_n has no head.

We regard v_1, v_2, v_n as the three roots r_b, r_r, r_y, respectively.

Let T_r be the tree rooted at r_r consisting of all right legs. Similarly, let T_b be the tree rooted at r_b consisting of all left legs and let T_y be the tree rooted at r_y consisting of all heads. The set of those three trees is called a realizer of a triconnected plane graph[1].

We have the following lemma.

Lemma 4. *[1] Each of trees T_r, T_b, T_y is a spanning tree of G'. For each inner vertex v of G' all edges incident to v appear around v clockwise in the following order.*

- *Exactly one outgoing edge in T_r. (Optionally it is shared with either one incoming edge in T_b or one incoming edge in T_y)*
- *Zero or more incoming edges in T_y.*
- *Exactly one outgoing edge in T_b. (Optionally it is shared with either one incoming edge in T_y or one incoming edge in T_r)*
- *Zero or more incoming edges in T_r.*
- *Exactly one outgoing edge in T_y. (Optionally it is shared with either one incoming edge in T_r or one incoming edge in T_b).*
- *Zero or more incoming edges in T_b.*

Proof. We denote the claim by (cd).

We can prove (cd) by induction on V_i, that is, for each i, the following (1)–(5) holds. (1) (cd) holds on each vertex of G_i having no neighbor in $\overline{G_i}$, (2) a relax version of (cd) holds on each vertex of G_i having a neighbor in $\overline{G_i}$, (3) the right legs induce a spanning tree of G_i rooted at r_r, (4) the left legs induce a spanning tree of G_i rooted at r_b, (5) the heads induce a spanning forest of G_i with each root on the outer face of G_i and each root has a neighbor in $\overline{G_i}$. □

Lemma 5. *[1][Lemma 6] For any vertex v, let S be the set of three paths consisting of (1) the path from v to r_r in T_r, (2) the path from v to r_b in T_b and (3) the path from v to r_y in T_y. Then any two paths in S share only v.*

Proof. Assume otherwise for a contradiction. If the path from v to r_r in T_r and the path from v to r_y in T_y share vertex $u \neq v$, then, by the planarity, the condition (cd) of lemma 4 is not satisfied at u. A contradiction.

Similar for other cases. □

Theorem 2. *T_r, T_b, T_y are $(3,2)$-completely independent spanning trees.*

Proof. Similar to Theorem 1 we can prove the following.

Given two vertices u and v in G, let S be the set of three paths consisting of the path from u to v in T_r, the path from u to v in T_b and the path from u to v in T_y. Then some pair of paths in S are internally disjoint.

□

5. Algorithm III

In this section we design a linear time algorithm to construct a set of $(3,2)$-completely independent spanning trees in a given 3D grid graph with size $L_x \times L_y \times L_z$. We assume $L_x \geq 1, L_y \geq 1, L_z \geq 1$.

Let G be a grid graph with size $L_x \times L_y \times L_z$. For a vertex (x, y, z) with $x < L_x$ we define its parent vertex as $(x+1, y, z)$, and for a vertex (x, y, z) with $x = L_z$ and $y > 0$ we define its parent vertex as $(L_z, y-1, z)$, and for a vertex (x, y, z) with $x = L_x, y = 0$ and $z > 0$ we define its parent vertex as $(L_x, 0, z-1)$. The root r_x is the vertex at $(L_x, 0, 0)$ and it has no parent. Then for each vertex of G except the root r_x we append the edge connecting v and its parent. Those edges induces the spanning tree of G and we denote it as T_{xyz}. The path from a vertex (x, y, z) to r_x in T_{xyz} consists of three line segments, those are (1) the line segment from (x, y, z) to (L_x, y, z), (2) the line segment from (L_x, y, z) to $(L_x, 0, z)$, and (3) the line segment from

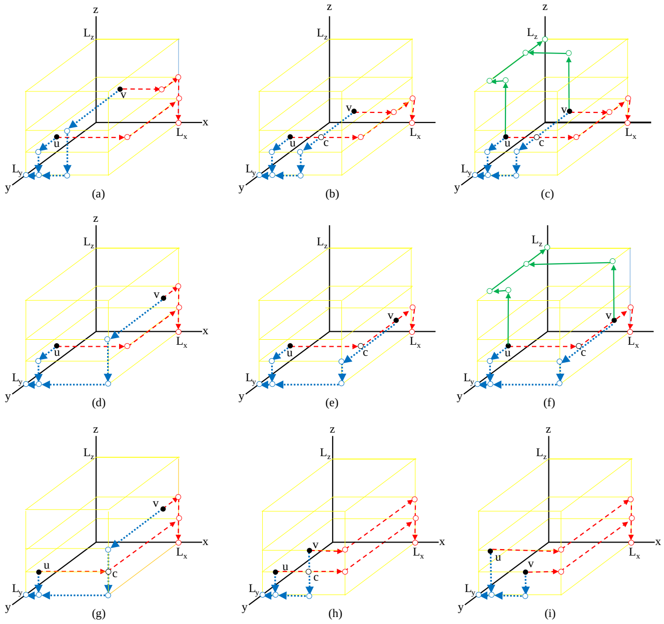

Fig. 7 Illustration for the cases.

$(L_x, 0, z)$ to $(L_x, 0, 0)$.

Similarly, we define the spanning tree T_{yzx} with the root r_y at $(0, L_y, 0)$ and the spanning tree T_{zxy} with the root r_z at $(0, 0, L_z)$. The path from a vertex (x, y, z) to the root r_y in T_{yzx} consists of three line segments, those are (1) the line segment from (x, y, z) to (x, L_y, z), (2) the line segment from (x, L_y, z) to $(x, L_y, 0)$, and (3) the line segment from $(x, L_y, 0)$ to $(0, L_y, 0)$. Similarly the path from a vertex (x, y, z) to the root r_z in $Tzxy$ consists of three line segments, those are (1) the line segment from (x, y, z) to (x, y, L_z), (2) the line segment from (x, y, L_z) to $(0, y, L_z)$, and (3) the line segment from $(0, y, L_z)$ to $(0, 0, L_z)$.

Note that the second part and the third part of $P_{xyz}(u, r_x)$, which is the path from u to r_x in T_{xyz}, locate on the plane with $x = L_x$, and the second part and the third part of $P_{yzx}(u, r_y)$ locate on the plane with $y = L_y$.

We have the following theorem.

Theorem 3. *$T_{xyz}, T_{yzx}, T_{zxy}$ are $(3, 2)$-completely independent spanning trees.*

Proof. For any pair of two vertices u and v in G we show if the path $P_{xyz}(u, v)$ connecting u and v in T_{xyz} and the path $P_{yzx}(u, v)$ connecting u and v in T_{yzx} are not internally disjoint, then either (a) $P_{xyz}(u, v)$ and the path $P_{zxy}(u, v)$ connecting u and v in T_{zxy} are internally disjoint, or (b) $P_{yzx}(u, v)$ and the path $P_{zxy}(u, v)$ connecting u and v in T_{zxy} are internally disjoint.

Assume that $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ are not internally disjoint.

We have the following four cases.

Case 1: $x(u) \leq L_x, x(v) < L_x, y(u) < L_y$ and $y(v) \leq L_y$ hold.

If $z(u) \neq z(v)$ then $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ are internally disjoint. A contradiction. See Figure 7 (a). If $z(u) = z(v)$ and either $x(u) = x(v)$ or $y(u) = y(v)$ then $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ are internally disjoint. A contradiction.

So assume otherwise. Now $z(u) = z(v)$, $x(u) \neq x(v)$ and $y(u) \neq y(v)$ hold. See Figure 7 (b). If $P_{xyz}(u, r)$ and $P_{yzx}(v, r)$ cross at a vertex c on the plane $z = z(u)$. Then $P_{xyz}(u, v)$ and $P_{zxy}(u, v)$ are internally disjoint. See Figure 7 (c).

Case 2: $x(u) \leq L_x, x(v) = L_x, y(u) < L_y$ and $y(v) \leq L_y$ hold.

If $z(u) \neq z(v)$ then $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ are internally disjoint. See Figure 7 (d). So assume otherwise.

If $z(u) = z(v)$ and $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ share some vertex c on the plane $z = z(u)$. See Figure 7 (e), then $P_{xyz}(u, v)$ and $P_{zxy}(u, v)$ are internally disjoint. See Figure 7 (f).

Case 3: $x(u) \leq L_x, x(v) = L_x, y(u) = L_y$ and $y(v) \leq L_y$ hold.

If $x(u) = L_x$ then $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ are internally disjoint. So assume otherwise.

If $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ cross at a vertex c on the line with $x = L_x$ and $y = L_y$. See Figure 7 (g). Then $P_{xyz}(u, v)$ and $P_{zxy}(u, v)$ are internally disjoint.

Case 4: $x(u) \leq L_x, x(v) \leq L_x, y(u) = L_y$ and $y(v) = L_y$ hold.

If $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ share some vertex c on the plane $y = L_y$ then $P_{xyz}(u, v)$ and $P_{zxy}(u, v)$ are internally disjoint. See Figure 7 (h).

Otherwise $P_{xyz}(u, v)$ and $P_{yzx}(u, v)$ are internally disjoint. See Figure 7 (i).

Each of other cases is symmetric to one of above cases. □

6. Conclusion

In this paper we have defined (t, s)-completely independent spanning trees which is a generalization of completely independent spanning trees. Then we have designed an algorithm to construct a set of $(3, 2)$-completely independent spanning trees in a given maximal planar graph, an algorithm to construct a set of $(3, 2)$-completely independent spanning trees in a given tri-connected planar graph, and an algorithm to construct a set of $(3, 2)$-completely independent spanning trees in a given 3D grid graph. Those algorithms are simple and run in $O(n)$ time, where n is the number of vertices of the given graph.

Can we design an algorithm to construct a set of (t, s)-completely independent spanning trees for other classes of graphs and some other choices of t and s?

References

[1] Giuseppe Di Battista, Roberto Tamassia and Luca Vismara, Output-Sensitive Reporting of Disjoint Paths, Algorithmica, 23, 4, pp. 302–340 (1999)

[2] Joseph Cheriyan and S. N. Maheshwari, Finding Nonseparating Induced Cycles and Independent Spanning Trees in 3-Connected Graphs, J. Algorithms, 9, 4, pp. 507–537 (1988)

[3] Sean Curran, Orlando Lee and Xingxing Yu: Finding Four Independent Trees, SIAM J. Comput., 35, 5, pp. 1023-1058 (2006)
[4] Benoît Darties, Nicolas Gastineau, Olivier Togni: Almost disjoint spanning trees: Relaxing the conditions for completely independent spanning trees, Discret. Appl. Math., 236, pp. 124-136 (2018)
[5] Toru Hasunuma, Completely independent spanning trees in the underlying graph of a line digraph, Discrete Math., 234, pp. 149–157 (2001)
[6] Toru Hasunuma, Completely Independent Spanning Trees in Maximal Planar Graphs, Proc. of Graph-Theoretic Concepts in Computer Science, WG 2002, Lecture Notes in Computer Science, 2573, pp. 235–245, Springer (2002)
[7] John E. Hopcroft and K. Wong, Linear Time Algorithm for Isomorphism of Planar Graphs, Proc. of the 6th Annual ACM Symposium on Theory of Computing, pp.172–184 (1974)
[8] John E. Hopcroft and Robert Endre Tarjan, Efficient Planarity Testing, J. ACM, 21, 4, pp. 549–568 (1974)
[9] Andreas Huck, Independent trees in graphs, Graphs and Combinatorics, 10, 1, pp. 29–45 (1994)
[10] Andreas Huck, Independent Trees in Planar Graphs Independent trees, Graphs Comb., 15, 1, pp. 29–77 (1999)
[11] Alon Itai and Michael Rodeh, The Multi-Tree Approach to Reliability in Distributed Networks, Information and Computation, 79, 1, pp. 43–59 (1988)
[12] Goos Kant, Drawing Planar Graphs Using the Canonical Ordering, Algorithmica, 16, 1, pp. 4–32 (1996)
[13] Samir Khuller and Baruch Schieber, On Independent Spanning Trees, Inf. Process. Letters, 42, 6, pp. 321–323 (1992)
[14] Kazuyuki Miura, Daishiro Takahashi, Shin-Ichi Nakano and Takao Nishizeki, A Linear-Time Algorithm to Find Four Independent Spanning Trees in Four-Connected Planar Graphs, Proc. of Graph-Theoretic Concepts in Computer Science, WG '98, Lecture Notes in Computer Science, 1517, pp. 310–323, Springer (1998)
[15] Kazuyuki Miura, Daishiro Takahashi, Shin-Ichi Nakano and Takao Nishizeki, A Linear-Time Algorithm to Find Four Independent Spanning Trees in Four Connected Planar Graphs, Int. J. Found. Comput. Sci., 10, 2, pp. 195–210 (1999)
[16] Sayaka Nagai and Shin-Ichi Nakano, A Linear-Time Algorithm to Find Independent Spanning Trees in Maximal Planar Graphs, Proc. of Graph-Theoretic Concepts in Computer Science, WG 2000, Lecture Notes in Computer Science, 1928, pp. 290–301, Springer (2000)
[17] Sayaka Nagai and Shin-ichi Nakano, A Linear-Time Algorithm for Five-Partitioning Five-Connected Internally Triangulated Plane Graphs, IEICE TRANS. FUNDAMENTALS, E84-A, 9, pp. 2330-2337 (2001)
[18] Shin-ichi Nakano, Planar Drawings of Plane Graphs, IEICE TRANS. INF.& SYST., E83-D, 3, pp. 384-391 (2000)
[19] Shin-ichi Nakano, (t, s)-completely Independent Spanning Trees, Proc. of WALCOM 2024, Lecture Notes in Computer Science, 14549, pp.365–376, Springer (2024)
[20] Walter Schnyder, Embedding Planar Graphs on the Grid, Proc. of SODA, pp. 138-148 (1990)
[21] Avram Zehavi and Alon Itai, Three tree-paths, J. Graph Theory, 13, 2, pp. 175–188 (1989)

Shin-ichi Nakano received his B.E. and M.E. degrees from Tohoku University, Sendai, Japan, in 1985 and 1987, respectively. In 1987 he joined Seiko Epson Corp. and in 1990 he joined Tohoku University. In 1992, he received Dr. Eng. degree from Tohoku University. Since 1999 he has been a faculty member of Gunma University. His research interests are algorithms. He is a member of IPSJ and ACM.

Ver. Wednesday 22nd January, 2025