Construction of Dual-CISTs on an Infinite Class of Networks

Xiao-Wen Qin, Rong-Xia Hao, and Jie Wu, *Fellow, IEEE*

Abstract—The main method to achieve fault-tolerant network systems is by exploiting and effectively utilizing the edge-disjoint and/or inner-vertex-disjoint paths between pairs of source and destination vertices. Completely independent spanning trees (CISTs for short) are powerful tools for reliable broadcasting/unicasting and secure message distribution. Particularly, it has been shown that two CISTs have an application on configuring a protection routing in IP networks, such as mobile ad hoc networks and relatively large (static) network topologies with scalability in [IEEE/ACM Trans. Netw., 27 (2019) 1112-1123]. Many results focus on CISTs in specific networks in the literature, however, few results are given on an infinite class of networks having common properties. In this article, we prove the existence of dual-CISTs in an infinite number of networks satisfying some Hamilton sufficient conditions. A unique algorithm to construct a CIST-partition is proposed, which can be applied to not only many kinds of networks, but our algorithm can also be implemented very easily in parallel or distributed systems satisfying the conditions. In addition, we make a comparative analysis between the proposed conditions and several known results on an infinite number of networks, the advantage of our result is significant. In particular, the bound in our conditions is sharp. The results will provide a powerful framework for the design of fault-tolerant network topologies and routing protocols for future networks.

Index Terms—Completely independent spanning trees, a protection routing, CIST-partition, Hamilton bipartition sufficient condition, constructive algorithm

1 Introduction

Disjoint multipaths from the source to destination that do not share common vertices and/or edges become an essential technique to improve the *quality-of-service* (QoS) and efficiency of broadcasting in networks. With the enlargement of the network size, failures are more common than we might expect. Disjoint multipaths can provide more alternative and diverse paths which can be selected when fault damages the network structure, which improves the reliability of networks. Designing vertex-disjoint paths and edge-disjoint paths between the source vertex and the destination vertex can tolerate faulty vertices and edges respectively. Fault-tolerant message delivery protocols, reliable broadcasting/unicasting and secure message distribution are also implemented by constructing disjoint multipaths on networks under the existence of such disjoint paths [3].

To guarantee that all messages can be transmitted between any two vertices, a spanning tree or a set of spanning trees is usually used to realize the broadcasting on a network. Itai and Rodeh [4] first introduced the concept of *independent spanning trees* (ISTs for short) to improve load balancing or transmission error tolerance. ISTs are a set of spanning trees share the same root vertex such that the paths from the root to any other vertex of each spanning trees are edge-disjoint and inner-vertex-disjoint. From then on, a lot of efficient construction schemes in different networks are proposed, such as [5] (chordal rings), [6] (multidimensional torus networks), [7] (transposition networks), [8] (alternating group networks), [9] (pancake Networks), [10] (enhanced hypercubes), [11], [12], [13], [14], [15] (hypercube and its variants), and so forth.

By the definition of IST, the application of IST is limited by the setting of the root vertex, which is not conducive to considering the broadcasting between any pair of the source vertex and the destination vertex. Based on this reason, the natation of completely independent spanning trees (CISTs for short) was suggested by Hasunuma [16], which is a generalization of edge-disjoint spanning trees and ISTs. *Completely independent spanning trees* are a set of spanning trees such that the paths joining between any pair of vertices in these trees are pairwise edge-disjoint and inner-vertex-disjoint. If there exist k CISTs in the networks, at least one spanning tree can ensure the message transmission between any pair of unfaulty vertices under at most $k-1$ faulty edges or vertices. In addition, if the network has a large number of data to transmit, we can divide it into k parts and let every spanning tree be responsible for only $\frac{1}{k}$ data to reduce the load pressure on edges and increase the throughput.

Hasunuma [17] had proved that determining whether a graph G admits k CISTs is NP-complete, even for $k=2$. He also conjectured that every $2k$-connected graph admits k CISTs. However, Péterfalvi [18] disproved the conjecture by showing that for every $k \geq 2$, there exists a k-connected graph which does not admit two CISTs. After that, one of the

- *Xiao-Wen Qin and Rong-Xia Hao are with the Department of Mathematics, Beijing Jiaotong University, Beijing 100044, P.R. China. E-mail: {15121545, rxhao}@bjtu.edu.cn.*
- *Jie Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA. E-mail: jiewu@temple.edu.*

Manuscript received 15 Aug. 2021; revised 4 Nov. 2021; accepted 28 Nov. 2021. Date of publication 3 Dec. 2021; date of current version 15 Dec. 2021.
This work was supported in part by the National Natural Science Foundation of China under Grants 11971054 and 11731002.
(Corresponding author: Rong-Xia Hao.)
Recommended for acceptance by D. Yang.
Digital Object Identifier no. 10.1109/TPDS.2021.3132412

important research directions is exploring completely independent spanning trees in some special graphs and interconnection networks. The existence and construction of multiple CISTs in graphs and networks can be referred to [16], [19], [20], [21], [22] for some certain classes of graphs, [23], [24], [25], [26], [27], [28] for hypercube and its variants, [29], [30], [31] for date center networks, and [2], [32] for Cayley graphs.

Constructing more CISTs can enhance the ability of fault tolerance in networks. However, due to the cost and resource considerations, backup hardware and transmission design usually allow only one copy in real life, as do for two CISTs, which is call a *dual-CISTs*. It has been shown that networks with a dual-CIST can be fully protected under a random single element failure in [1]. An interesting fact is that several well-known sufficient conditions for Hamiltonian property ensure the existence of two completely independent spanning trees. For example, Araki [33] confirmed that Dirac's condition implies the existence of two CISTs. Fan *et al.* [34] confirmed that Ore's condition implies the existence of two CISTs. Hong *et al.* and Qin *et al.* [35], [36] recently proved that a neighborhood unions condition of Hamiltonian graphs also implies the existence of two CISTs. Especially, Hong [37] showed the generalization of Dirac's condition implies the existence of k CISTs. For more sufficient conditions for graphs that admit multiple CISTs, see [38], [39]. Recently, Cao *et al.* [40] gave a new kind of sufficient condition, called *bipartition sufficient condition*, of Hamiltonian graphs. That is, if G is a simple connected graph with a vertex bipartition $\{S, T\}$ satisfying the conditions: (1) $G[T]$ is a tree with t leaves; (2) each vertex in T with degree i in $G[T]$ has at least $|S|+1-i$ neighbors in S; and (3) $|S| = t-1$, then G is a Hamiltonian graph.

In this paper, we will study the existence of a dual-CISTs in an infinite class of networks by using the characterization of CISTs given in [33]. The most significant contribution of our work is that there exists an infinite number of graphs which only satisfy our conditions in known results. The main results of the paper are summarized as follows:

1. Proving the existence of a dual-CISTs in an infinite class of networks.
2. Proposing a unique algorithm to construct a CIST-partition in an infinite class of networks.
3. Giving a comparison with the known results based on the analysis of applications.

The remaining work is organized as follows. Section 2 presents some useful lemmas. Section 3 proves the existence of two completely independent spanning trees under the bipartition sufficient condition in the general graph. In Section 4, we make a comparison between our results and some known conclusions and analyze the optimality of the parameter bound. Finally, a conclusion is given in Section 5.

2 Notation and Preliminaries

It is a common method to regard the interconnection network as a loopless undirected graph $G = (V(G), E(G))$, where $V(G)$ denotes the set of processors and $E(G)$ denotes the set of communication links. In this paper, we use graphs and networks interchangeably. The value of $|V(G)|$ is called the *order* of G. Two vertices v_1, v_2 in $V(G)$ are said to be *adjacent* if and only if $(v_1, v_2) \in E(G)$ and v_1, v_2 are said to be *incident* with the edge (v_1, v_2), and v_1, v_2 are called *ends* of the edge (v_1, v_2). Two edges which are incident with a common vertex are also said to be *adjacent*. The *neighbor* of a vertex $u \in V(G)$ is a vertex adjacent to u in G. The number of neighbors of u is called the *degree* of u in G, denoted by $d_G(u)$. For two disjoint nonempty vertex subsets S and T of G, we use $E(T, S)$ to denote the set of edges between S and T, and *the subgraph of G induced by S* is denoted by $G[S]$ which is a graph whose vertex set is S and whose edge set consists of all edges of G which have both ends in S. For two vertices x and y in $V(G)$, the *distance* between x and y, denoted $d_G(x, y)$, is the length of the shortest path connecting them. A *spanning tree* T of a graph G is an acyclic connected subgraph of G such that $V(T) = V(G)$. A vertex is said to be a *leaf* of a tree T if it has degree 1 in T, and an *inner-vertex* otherwise. *A rooted tree* is a tree with a specified vertex, called the *root*. For a vertex x in a rooted tree T, we use $T(x)$ to denote the *subtree rooted at x*, the *ancestors of x* are the vertices on the unique path from x to the root except itself, *descendants of x* are vertices in $T(x)$ except itself. Let x and y be two vertices of G. Two paths joining x and y in G are *openly disjoint* if they have no common vertex except for the two ends x and y. Table 1 shows the notations needed for the discussion.

TABLE 1
Notations Needed for the Discussion

Notation	*Meaning*
$V(G)$	The vertex set of a graph G
$E(G)$	The edge set of a graph G
$\|V(G)\|$	The value of $\|V(G)\|$
(u, v)	An edge with two ends u and v
$N_G(v)$	The neighborhood of the vertex v in G
$d_G(v)$	The degree of the vertex v in G
$\delta(G)$	The minimum degree of the graph G
$E(T, S)$	The set of edges between S and T
$G[S]$	The subgraph induced by edge or vertex set S in G
$d_G(x, y)$	The distance between vertices x and y in G
$T(x)$	A subtree of T rooted at the vertex x
$B(S, T, G)$	A bipartite subgraph of G induced by the edge set $E(S, T)$

To prove the main result, two important equivalent conditions for k completely independent spanning trees provide tools.

Theorem 1. *(See [16]) Spanning trees $T_1, T_2, \ldots, T_k$ of a graph G are completely independent spanning trees if and only if they are edge-disjoint in G and for any $v \in V(G)$, there is at most one T_i such that v is an inner-vertex.*

Theorem 2. *(See [33]) A connected graph G has k completely independent spanning trees if and only if there is a partition $\{V_1, V_2, \ldots, V_k\}$ of $V(G)$ such that: (i) the induced subgraph $G[V_i]$ is connected for every $i \in \{1, 2, \ldots, k\}$ and (ii) the bipartite subgraph of G induced by the edge set $E(V_i, V_j)$ has no tree component for any $1 \le i < j \le k$, denoted by $B(V_i, V_j, G)$. Moreover, the partition $\{V_1, V_2, \ldots, V_k\}$ of $V(G)$ which satisfies above conditions (i) and (ii) is called a CIST-partition of $V(G)$.*

In Section 1, we have mentioned that several well-known sufficient conditions for Hamiltonian property ensure the existence of two completely independent spanning trees. The specific conclusions are listed in Theorems 3, 4 and 5.

Theorem 3. *(See [33]) Let G be a graph with n vertices for $n \geq 7$. If $\delta(G) \geq \frac{n}{2}$, then G has two completely independent spanning trees.*

Theorem 4. *(See [34]) Let G be a graph with n vertices for $n > 8$. If $d(u) + d(v) \geq n$ for every pair of non-adjacent vertices u and v, then G has two completely independent spanning trees.*

Theorem 5. *(See [35], [36]) Let G be a graph with n vertices for $n \geq 8$. If $|N(x) \cup N(y)| \geq \frac{n}{2}$ and $|N(x) \cap N(y)| \geq 3$ for every pair of non-adjacent vertices x and y, then G has two completely independent spanning trees.*

To prove the main result, we give the following useful lemmas.

Lemma 2.1. *(See [22]) There are $\lfloor \frac{n}{2} \rfloor$ completely independent spanning trees in the complete bipartite graph $K_{m,n}$ for all $m \geq n \geq 4$.*

Lemma 2.2. *Let T be a tree with t leaves and $t \geq 4$. Then*

(1) *there is no vertex in $V(T)$ with degree more than t.*
(2) *there is at most one vertex in $V(T)$ with degree t. Moreover, if $d_T(x) = t$ with $x \in V(T)$, each vertex in $V(T) \setminus \{x\}$ is a leaf or with degree two in T.*
(3) *there are at most two vertices in $V(T)$ with degree $t-1$.*

Proof. Let x be a vertex with maximum degree in T and $N_T(x) = \{x_1, x_2, \ldots, x_s\}$. Regard T as a tree rooted at x.

1) Let the number of leaves in the subtree $T(x_i)$ of T be y_i for every $i \in \{1, 2, \ldots, s\}$. If $d_T(x_i) \geq 3$, then x_i is not a leaf in $T(x_i)$, $y_i \geq 2$, and $T(x_i)$ provides y_i leaves to T. If $d_T(x_i) = 2$, then x_i is a leaf in $T(x_i)$, $y_i \geq 2$, and $T(x_i)$ provides $y_i - 1$ leaves to T. If $d_T(x_i) = 1$, then $T(x_i)$ is an isolated vertex x_i and $T(x_i)$ provides a leaf to T. Thus each $T(x_i)$ for $i \in \{1, 2, \ldots, s\}$ provides at least a leaf to the tree T, and the number of leaves in a tree is at least the maximum degree of it. Thus (1) holds.
2) If $d_T(x) = t$ with $x \in V(T)$, then $s = t$. Since T is a tree with t leaves, each subtree $T(x_i)$ provides only a leaf to T for $i \in \{1, 2, \ldots, t\}$. Thus $T(x_i)$ is a path or $T(x_i)$ is an isolated vertex x_i. It implies that each vertex in $V(T) \setminus \{x\}$ is a leaf or with degree two in T. The result (2) holds.
3) If $d_T(x) = t-1$, then there exists a subtree $T(x_i)$ which provides two leaves to T and each of the others provides a leaf. Without loss of generality, assume that $T(x_1)$ provides two leaves to T. It implies that each $T(x_j)$ is a path (maybe one vertex) which provides only one leaf to T for $j \in \{2, \ldots, t-1\}$. Then $d_T(x_1) = 3$ or $d_T(x_1) = 2$, and $d_T(x_j) = 2$ or $d_T(x_j) = 1$ for any $j \in \{2, \ldots, t-1\}$.

If $d_T(x_1) = 3$, then $d_T(v) = 2$ or $d_T(v) = 1$ for any $v \in V(T) \setminus \{x, x_1\}$. If $d_T(x_1) = 2$, the subtree $T(x_1)$ has three leaves including x_1. Thus there exists only one vertex in $V(T(x_1))$ with degree 3 in $T(x_1)$, denoted by y. And $d_T(v) = 2$ or $d_T(v) = 1$ for any $v \in V(T) \setminus \{x, y\}$, which is less than $t-1$. Thus there are at most two vertices with degree $t-1$ if $t = 4$, otherwise there is at most one vertex x with degree $t-1$. The result (3) holds. □

3 The Existence of a Dual-CISTs on an Infinite Class of Networks

The following Theorem 6 is our main result.

Theorem 6. *Let G be a graph. If the vertex set of G has a bipartition $\{S, T\}$ satisfying the following conditions:*

(1) *$G[T]$ is a tree with t leaves for $t \geq 4$;*
(2) *each vertex in T with degree i in $G[T]$ has at least $|S| + 1 - i$ neighbors in S;*
(3) *$|S| = t - 1$,*

then G has a dual-CISTs.

Proof. Let $\{S, T\}$ be a bipartition of $V(G)$ satisfying the conditions (1)-(3) in Theorem 6. By deleting some edges in $E(T, S)$, we can obtain a spanning subgraph H of G such that $\{S, T\}$ is a bipartition of $V(H)$ (where $V(H) = V(G)$) which still satisfies the conditions (1) and (3). In addition, $\{S, T\}$ satisfies the condition $(2)^*$ that each vertex in T with degree i in $H[T]$ has *exactly* $|S| + 1 - i$ neighbors in S. One knows that a dual-CISTs of H under the conditions (1), $(2)^*$ and (3) is also a dual-CISTs of G under the conditions (1), (2) and (3). Thus we will show that H has a dual-CISTs hereinafter.

Let L be the set of leaves in $H[T]$. By the condition $(2)^*$, the following Facts 1 and 2 can be obtained directly. □

Fact 1. *In the graph H, each vertex in L is adjacent to all vertices in S. That is, the graph $H[E(L, S)]$ is a complete bipartite graph.*

Fact 2. *In the graph H, each vertex in T with degree two in $H[T]$ is adjacent to all but one vertex in S.*

From Theorem 2, it needs to construct a CIST-partition $\{V_1, V_2\}$ of $V(H)$. The following two parts are considered.

Part I. $d_{H[T]}(v) < t$ for every vertex $v \in T$.

We first show that there is a vertex $s^* \in S$ such that $N_S(v) \neq \{s^*\}$ for each $v \in T$. By the contrary, for every vertex $s \in S$, there exists one vertex in T, say v_s, such that $N_S(v_s) = \{s\}$. By conditions $(2)^*$ and (3), one has that $d_{H[T]}(v_s) = |S| = t - 1$. For different vertices s and r in S, let v_s and v_r be vertices in T such that $N_S(v_s) = \{s\}$ and $N_S(v_r) = \{r\}$. Then v_s and v_r are different. Otherwise, $N_S(v_s) = \{s, r\}$, which is a contradiction. Thus $|\{v_s, s \in S\}| = t - 1$. It derives that there are at least $t-1$ vertices in T with degree $t-1$ in $H[T]$. By Lemma 2.2, there are at most two vertices in T with degree $t-1$ in $H[T]$. Thus $t - 1 \leq 2$, which contradicts with $t \geq 4$.

Let $V_1 = (T \setminus \{t^*\}) \cup \{s^*\}$ and $V_2 = (S \setminus \{s^*\}) \cup \{t^*\}$, where t^* is an arbitrary vertex in L. See Fig. 1. We will show that $\{V_1, V_2\}$ is a CIST-partition of $V(H)$. (The definition of CIST-partition is shown in Theorem 2.)

Since $t^* \in L$, $H[T \setminus \{t^*\}]$ is a subtree of $H[T]$. By $t \geq 4$, $L \setminus \{t^*\}$ is not empty. From Fact 1, s^* is connected with all vertices in $L \setminus \{t^*\}$. Thus $H[V_1]$ is connected. Similarly, by Fact 1, t^* is connected with all vertices in the non-empty set $S \setminus \{s^*\}$, which derives the connectedness of $H[V_2]$.

For the bipartite graph $B(V_1, V_2, H)$, it contains a bipartite subgraph induced by $E(L \setminus \{t^*\}, S \setminus \{s^*\})$, denoted by B.

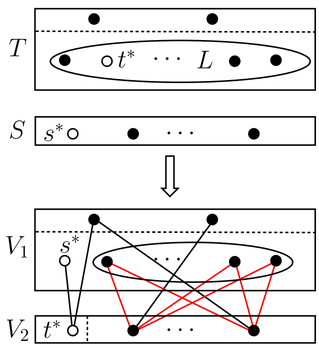

Fig. 1. The illustration of the bipartition $\{V_1, V_2\}$ in Part I.

By Fact 1, $|L| \geq 4$ and $|S| \geq 3$, one has that B is a complete bipartite graph and each part has at least two vertices, which implies that B is a connected graph with at least one cycle (see red lines in Fig. 1). Since $d_{H[T]}(v) < t$ for any $v \in T$, by the condition $(2)^*$ and the choice of s^*, each vertex in $T \setminus L$ has a neighbor in $S \setminus \{s^*\}$. Thus each vertex in $T \setminus L$ can be connected to B in $B(V_1, V_2, H)$. Moreover, since $(s^*, t^*) \in E(H)$ and t^* is adjacent to some vertices in $T \setminus L$, the vertices s^* and t^* are in the component containing B of $B(V_1, V_2, H)$. Thus $B(V_1, V_2, H)$ is a component containing B which is not a tree. See Fig. 1. Thus $\{V_1, V_2\}$ is a CIST-partition of $V(H)$ for Part I.

Part II. There exists a vertex $v \in T$ such that $d_{H[T]}(v) \geq t$.

By Lemma 2.2, there is only one such vertex in T, denoted by v, such that $d_{H[T]}(v) = t$. Moreover, each vertex in $T \setminus \{L \cup \{v\}\}$ has degree 2 in $H[T]$. The tree $H[T]$ is regarded as a tree rooted at v. Let $S = \{s_1, s_2, \ldots, s_{t-1}\}$, $N_{H[T]}(v) = \{x_1, x_2, \ldots, x_t\}$ and $H[T](x_i)$ be the subtree of $H[T]$ rooted at x_i (T_i in short) with $i \in \{1, 2, \ldots, t\}$. Note that T_i is a path. Assume the ends of T_i are x_i and y_i, where $y_i \in L$ for $i \in \{1, 2, \ldots, t\}$. The vertices y_i and x_i are the same if $|V(T_i)| = 1$.

Case 1. $t \geq 5$.

Assume $V_1 = V(T_1) \cup V(T_2) \cup (\bigcup_{i=3}^{t-1}\{s_i\})$ and $V_2 = V(H) \setminus V_1$, that is $\{v\} \cup (\bigcup_{i=3}^{t} V(T_i)) \cup \{s_1, s_2\}$. See (a)-(c) in Fig. 2. We will show that $\{V_1, V_2\}$ is a CIST-partition of $V(H)$ in this case.

From Fact 1, the induced subgraph $H[V_1]$ consists of two paths T_1, T_2 and a complete bipartite subgraph induced by $E(\{y_1, y_2\}, \bigcup_{i=3}^{t-1}\{s_i\})$, where y_1 and y_2 are ends of T_1 and T_2 respectively. See Fig. 2b. Let $Y = H[T \setminus (V(T_1) \cup V(T_2))]$. As we can see, Y is a tree with the leaf set $\bigcup_{i=3}^{t}\{y_i\}$. The induced subgraph $H[V_2]$ consists of the tree Y and a complete bipartite subgraph induced by $E(\bigcup_{i=3}^{t}\{y_i\}, \{s_1, s_2\})$. See Fig. 2$c$. Thus the subgraphs $H[V_1]$ and $H[V_2]$ are connected.

For the bipartite graph $B(V_1, V_2, H)$, by Fact 1, it contains two complete bipartite subgraphs, where one is induced by $E(\{y_1, y_2\}, \{s_1, s_2\})$, denoted by B_1, and the other is induced by $E(\bigcup_{i=3}^{t}\{y_i\}, \bigcup_{j=3}^{t-1}\{s_j\})$, denoted by B_2. Since $t \geq 5$, both subgraphs B_1 and B_2 contain a cycle which are shown in Fig. 2d by green and red lines respectively. By Fact 2 and $t \geq 5$, every vertex in $(V(T_1) \cup V(T_2)) \setminus \{y_1, y_2\} \subseteq V_1$ is adjacent to at least one vertex in $\{s_1, s_2\} \subseteq V_2$ and every vertex in $\bigcup_{i=3}^{t}(V(T_i) \setminus \{y_i\}) \subseteq V_2$ is adjacent to some vertices in $\bigcup_{i=3}^{t-1}\{s_i\} \subseteq V_1$. Thus $(V(T_1) \cup V(T_2)) \setminus \{y_1, y_2\}$ and B_1 are in the same component of $B(V_1, V_2, H)$ and $\bigcup_{i=3}^{t}(V(T_i) \setminus \{y_i\})$ and B_2 are in the same component of $B(V_1, V_2, H)$.

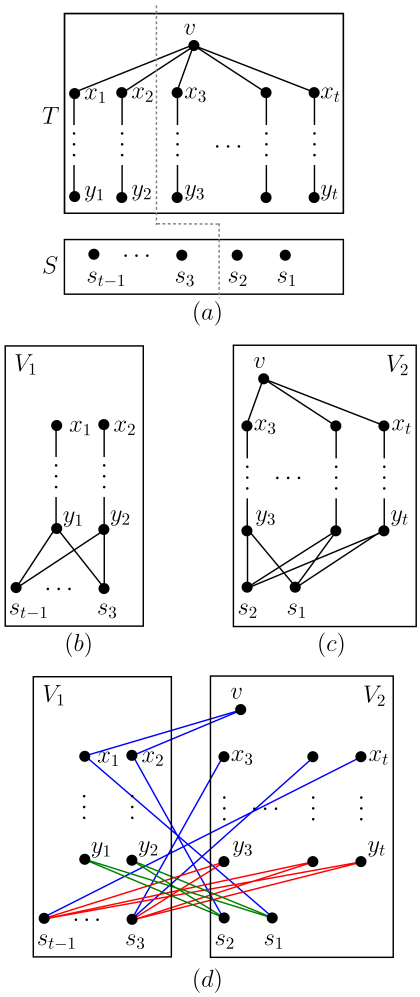

Fig. 2. The local illustration of (a) H; (b) $H[V_1]$; (c) $H[V_2]$ and (d) $B(V_1, V_2, H)$, where H satisfies Case 1 in Part II.

Since v is adjacent to both x_1 and x_2, the bipartite graph $B(V_1, V_2, H)$ has no tree component, see Fig. 2d. Based on the discussions and Theorem 2, $\{V_1, V_2\}$ is a CIST-partition of $V(H)$ for $t \geq 5$ in Part II.

Case 2. $t = 4$.

Let $I = \{i : |V(T_i)| = 1 \text{ for } i \in \{1, 2, 3, 4\}\}$.

Subcase 2.1. $|I| = 4$.

The graph H is isomorphic to a complete bipartite graph $K_{4,4}$ with the bipartition $\{\{v, s_1, s_2, s_3\}, \{x_1, x_2, x_3, x_4\}\}$ of $V(H)$ and the edge set $E(\{v, s_1, s_2, s_3\}, \{x_1, x_2, x_3, x_4\})$. By Lemma 2.1, the graph H admits a dual-CISTs.

Subcase 2.2. $2 \leq |I| \leq 3$.

Without loss of generality, assume $1 \notin I$ and $\{3, 4\} \subseteq I$, which implies that $x_1 \neq y_1$, $x_3 = y_3$ and $x_4 = y_4$. Let $V_1 = \{v\} \cup \{V(T_1) \setminus \{x_1\}\} \cup V(T_2) \cup \{s_1\}$ and $V_2 = V(H) \setminus V_1$, that is $\{x_1\} \cup V(T_3) \cup V(T_4) \cup \{s_2, s_3\} = \{x_1, y_3, y_4, s_2, s_3\}$. See Figs. 3$a$, 3$b$, and 3$c$.

The subgraph $H[V_1]$ contains two paths $T_1 \setminus \{x_1\}$ and T_2. By Fact 1, the ends y_1 and y_2 of these two paths are adjacent to s_1. In addition, $x_2 \in V(T_2)$ is adjacent to v, thus $H[V_1]$ is connected shown in Fig. 3b. For the subgraph $H[V_2]$, by Fact 1, it contains two paths T_3, T_4 and a complete bipartite graph induced by $E(\{y_3, y_4\}, \{s_2, s_3\})$, where y_3 and y_4 are ends of T_3 and T_4 respectively. Since $1 \notin I$, we have $d_{H[T]}(x_1) = 2$. By Fact 2, x_1 has two neighbors in S, thus x_1 is

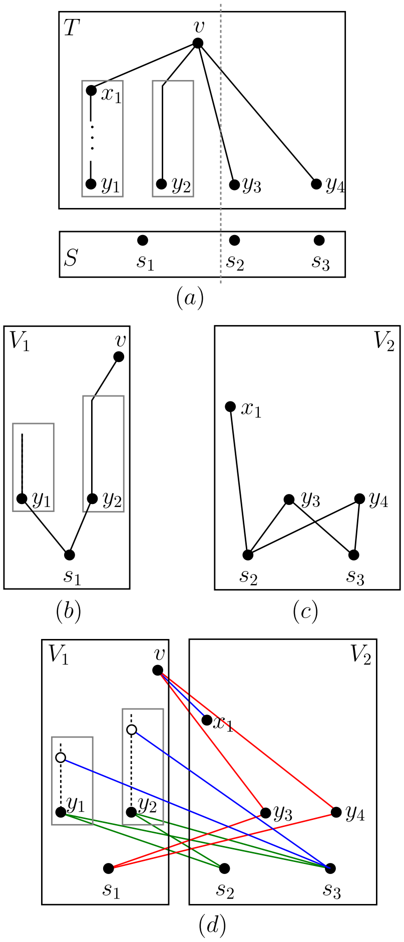

Fig. 3. The local illustration of (a) H; (b) $H[V_1]$; (c) $H[V_2]$ and (d) $B(V_1, V_2, H)$, where H satisfies Subcase 2.2 in Part II.

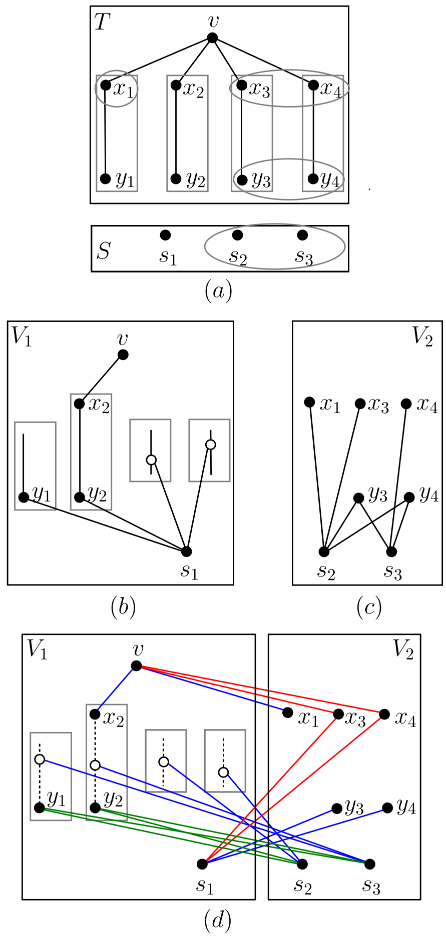

Fig. 4. The local illustration of (a) H; (b) $H[V_1]$; (c) $H[V_2]$ and (d) $B(V_1, V_2, H)$, where H satisfies Subcase 2.3.1 with $|I| = 0$.

adjacent to at least one vertex in $\{s_2, s_3\}$. Then $H[V_2]$ is connected which is shown in Fig. 3c.

For the bipartite graph $B(V_1, V_2, H)$, by Fact 1, there are two complete bipartite subgraphs, where one is induced by $E(\{y_1, y_2\}, \{s_2, s_3\})$, denoted by B_1, and the other is induced by $E(\{y_3, y_4\}, \{s_1, v\})$, denoted by B_2. Each of the subgraphs B_1 and B_2 is a cycle with four vertices shown in Fig. 3d by green and red lines respectively. Since subtrees T_1 and T_2 are paths, by Fact 2, each vertex in $(V(T_1) \setminus \{x_1, y_1\}) \cup (V(T_2) \setminus \{y_2\})$, that is $V_1 \setminus \{v, y_1, y_2\}$, is adjacent to at least one vertex in $\{s_2, s_3\} \subseteq V_2$. Thus $V_1 \setminus \{v\}$ and B_1 are contained in the same component of $B(V_1, V_2, H)$. Since $(v, x_1) \in E(V_1, V_2)$, the vertex x_1 is in the component of $B(V_1, V_2, H)$ which containing B_2. Thus, the bipartite graph $B(V_1, V_2, H)$ has no tree component, see Fig. 3d. Thus $\{V_1, V_2\}$ is a CIST-partition of $V(H)$.

Subcase 2.3. $0 \leq |I| \leq 1$.

Without loss of generality, assume $1, 2, 3 \notin I$, which implies that $d_{H[T]}(x_i) = 2$ and x_i, y_i represent different vertices for $i \in \{1, 2, 3\}$. By Facts 1, 2 and $|S| = t - 1 = 3$, the vertex x_3 has two neighbors in S and the vertex x_4 has at least two neighbors in S, then $|N_S(x_3) \cap N_S(x_4)| \geq 1$. Without loss of generality, assume $s_1 \in N_S(x_3) \cap N_S(x_4)$.

Subcase 2.3.1. $N_S(u) \neq \{s_2, s_3\}$ for every $u \in (V(T_3) \setminus \{x_3, y_3\}) \cup (V(T_4) \setminus (\{x_4\} \cup \{y_4\}))$.

Let $V_1 = \{v\} \cup (V(T_1) \setminus \{x_1\}) \cup V(T_2) \cup (V(T_3) \setminus \{x_3, y_3\}) \cup (V(T_4) \setminus (\{x_4\} \cup \{y_4\})) \cup \{s_1\}$ and $V_2 = V(G) \setminus V_1$, that is $\{x_1, x_3, x_4\} \cup \{y_3, y_4\} \cup \{s_2, s_3\}$. Note that if $4 \in I$, that is $x_4 = y_4$, then $|V_2| = 6$; otherwise, $|V_2| = 7$. Figs. 4 and 5 are illustrations for $4 \notin I$ and $4 \in I$ respectively.

The subgraph $H[V_1]$ contains paths $T_1 \setminus \{x_1\}$, T_2, $T_3 \setminus \{x_3, y_3\}$ (if it exists) and $T_4 \setminus (\{x_4\} \cup \{y_4\})$ (if it exists). Fact 1 implies that $(y_1, s_1), (y_2, s_1) \in E(H[V_1])$, then $T_1 \setminus \{x_1\}$, T_2 and s_1 are in a component of $H[V_1]$. By the assumption of Subcase 2.3.1, $T_3 \setminus \{x_3, y_3\}$ (if it exists), $T_4 \setminus (\{x_4\} \cup \{y_4\})$ (if it exists) and s_1 are in a component of $H[V_1]$. Since $(v, x_2) \in E(H[V_1])$, the subgraph $H[V_1]$ is connected. See Figs. 4b and 5b. For the subgraph $H[V_2]$, by Fact 1, it contains a complete bipartite graph induced by $E(\{y_3, y_4\}, \{s_2, s_3\})$. Since $d_{H[T]}(x_j) \leq 2$ for $j \in \{1, 3, 4\}$, by Facts 1 and 2, x_j has at least two neighbors in S. Then x_1, x_3 and x_4 are adjacent to some vertices in $\{s_2, s_3\}$. Then $H[V_2]$ is connected. See Figs. 4c and 5c.

For the bipartite graph $B(V_1, V_2, H)$, by Fact 1, it contains two complete bipartite subgraphs, where one is induced by $E(\{y_1, y_2\}, \{s_2, s_3\})$, denoted by B_1, and the other is induced by $E(\{x_3, x_4\}, \{s_1, v\})$, denoted by B_2. The subgraphs B_1 and B_2 are cycles with four vertices shown in Figs. 4d and 5d by green and red lines respectively. Since every vertex y in $V_1 \setminus \{v, y_1, y_2, s_1\}$, that is $(V(T_1) \setminus \{x_1, y_1\}) \cup (V(T_2) \setminus \{y_2\}) \cup (V(T_3) \setminus \{x_3, y_3\}) \cup (V(T_4) \setminus (\{x_4\} \cup \{y_4\}))$, has degree two in $H[T]$, by Fact 2, the vertex y is adjacent to some vertices in $\{s_2, s_3\}$, which implies that y and B_1 are contained in the same component of $B(V_1, V_2, H)$. Since

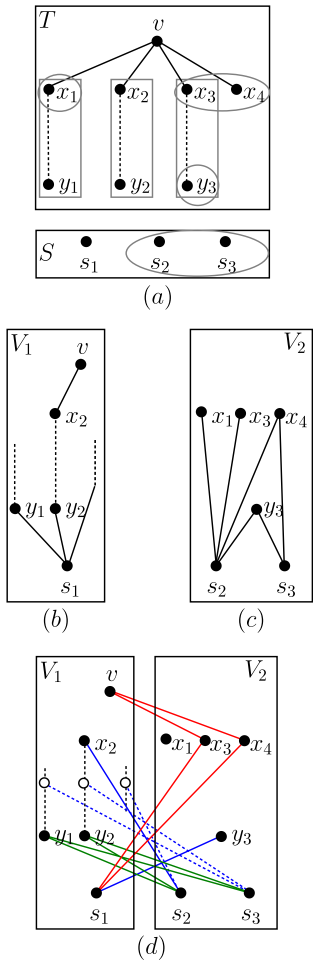

Fig. 5. The local illustration of (a) H; (b) $H[V_1]$; (c) $H[V_2]$ and (d) $B(V_1, V_2, H)$, where H satisfies Subcase 2.3.1 with $|I| = 1$.

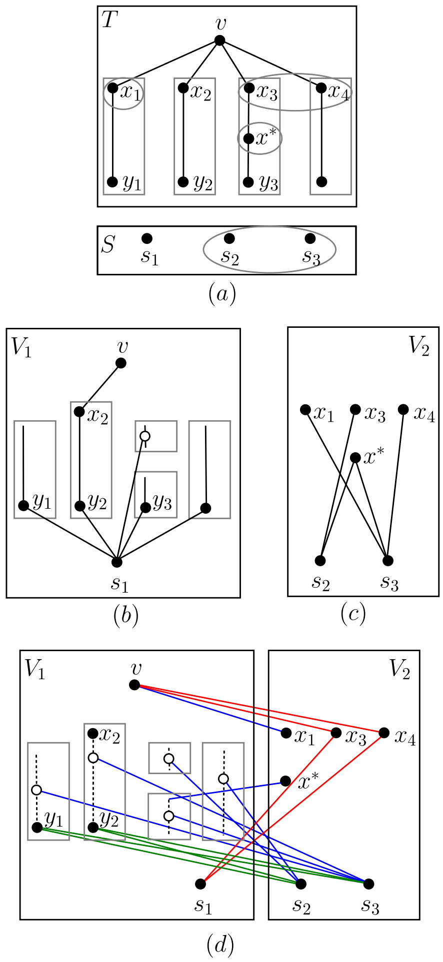

Fig. 6. The local illustration of (a) H; (b) $H[V_1]$; (c) $H[V_2]$ and (d) $B(V_1, V_2, H)$, where H satisfies Subcase 2.3.2.

$(v, x_1), (y_3, s_1), (y_4, s_1) \in E(V_1, V_2)$ and $s_1, v \in V(B_2)$, the bipartite graph $B(V_1, V_2, H)$ has no tree component. Thus $\{V_1, V_2\}$ is a CIST-partition of $V(H)$.

Subcase 2.3.2. $N_S(u) = \{s_2, s_3\}$ for some $u \in (V(T_3) \setminus \{x_3, y_3\}) \cup (V(T_4) \setminus (\{x_4\} \cup \{y_4\}))$.

Without loss of generality, let x^* be the vertex in $V(T_3) \setminus \{x_3, y_3\}$ such that $N_S(x^*) = \{s_2, s_3\}$ and the distance $d_{T_3}(x^*, x_3)$ is minimal subject to $N_S(x^*) = \{s_2, s_3\}$. Let $V_1 = \{v\} \cup (V(T_1) \setminus \{x_1\}) \cup V(T_2) \cup (V(T_3) \setminus \{x_3, x^*\}) \cup (V(T_4) \setminus \{x_4\}) \cup \{s_1\}$ and $V_2 = V(H) \setminus V_1 = \{x_1, x_3, x_4, x^*\} \cup \{s_2, s_3\}$. See Figs. 6*a*, 6*b*, and 6*c*.

The subgraph $H[V_1]$ contains paths $T_1 \setminus \{x_1\}$, T_2, two paths obtained by $T_3 \setminus \{x_3, x^*\}$ (if they exist), and $T_4 \setminus \{x_4\}$ (if it exists). Fact 1 brings out $(y_1, s_1), (y_2, s_1), (y_4, s_1) \in E(H)$, thus paths $T_1 \setminus \{x_1\}$, T_2 and $T_4 \setminus \{x_4\}$ (if it exists) are connected with s_1. Let the two paths obtained by $T_3 \setminus \{x_3, x^*\}$ be P_1 and P_2 (if they exists), where $V(P_1)$ is the set of ancestors of x^* in $V(T_3) \setminus \{x_3, x^*\}$ and $y_3 \in V(P_2)$. By the choice of x^*, each vertex, say w, in $V(P_1)$ satisfy that $N_S(w) \neq \{s_2, s_3\}$. Also by $|N_S(w)| = 2$, the vertex w is adjacent to s_1. By Fact 1, y_3 in $V(P_2)$ is adjacent to s_1. Because $(v, x_2) \in E(H[V_1])$, the subgraph $H[V_1]$ is connected shown in Fig. 6*b*. In the subgraph $H[V_2]$, the vertex x^* is adjacent to s_2 and s_3. Since $d_{H[T]}(x_j) \leq 2$ for $j \in \{1, 3, 4\}$, by Facts 1 and 2, x_j is adjacent to at least one of s_2 and s_3. Then $H[V_2]$ is connected shown in Fig. 6*c*.

For the bipartite graph $B(V_1, V_2, H)$, by Fact 1, it contains two complete bipartite subgraphs, where one is induced by $E(\{y_1, y_2\}, \{s_2, s_3\})$, denoted by B_1, and the other is induced by $E(\{x_3, x_4\}, \{s_1, v\})$, denoted by B_2. The subgraphs B_1 and B_2 are cycles with four vertices shown in Fig. 6*d* by green and red lines respectively. By Facts 1 and 2, every vertex in $V_1 \setminus \{v, y_1, y_2, s_1\}$, that is $(V(T_1) \setminus \{x_1\}) \cup (V(T_2) \setminus \{y_2\}) \cup (V(T_3) \setminus \{x_3, x^*\}) \cup (V(T_4) \setminus \{x_4\})$, is adjacent to at least one vertex in $\{s_2, s_3\}$. Then $V_1 \setminus \{v, s_1\}$ and B_1 are in the same component of $B(V_1, V_2, H)$. In addition, $d_{T_3}(x^*) = 2$ and $V(P_2)$ is the set of descendants of x^* in T_3, thus x^* has a neighbor in $V(P_2)$. Since $(v, x_1) \in E(V_1, V_2)$ and $v \in V(B_2)$, the vertex x_1 and B_2 are in the same component of $B(V_1, V_2, H)$. As a result, the bipartite graph $B(V_1, V_2, H)$ has no tree component, see Fig. 6*d*. Thus $\{V_1, V_2\}$ is a CIST-partition of $V(H)$.

Based on the discussions, by Theorem 2, there exists a dual-CISTs in the graph H, so does G.

Theorem 6 converts readily into the CIST-partition algorithm, see Algorithm 1. The algorithm runs as follows. Some notations are defined in lines 1-2. By lines 3-8, a new graph G satisfying the condition $(2)^*$ is obtained by deleting some edges in the original graph G. Lines 11-21 will perform if there exists a vertex v_i in T having degree at least t in $G[T]$, which is corresponding to Part II of Theorem 6. The CIST-partition is given by the Subroutine Part II. The cases for $t \geq 5$ and $t \leq 4$ are treated separately in Subroutine Part

II. Line 2 in Subroutine Part II gives a CIST-partition for $t \geq 5$, which is corresponding to Case 1 in Part II in Theorem 6. For $t \leq 4$, lines 3-23 in Subroutine Part II perform. The CIST-partition is constructed in different ways based on the cardinality of the variable I defined in line 5. For $|I| = 4$, the CIST-partition is given by lines 5-6, which is corresponding to Subcase 2.1 in Part II. For $2 \leq |I| \leq 3$, the CIST-partition is given in lines 7-10, which is corresponding to Subcase 2.2 in Part II in Theorem 6. For $0 \leq |I| \leq 1$, the CIST-partition is given in lines 11-22, which is corresponding to Subcase 2.3 in Part II in Theorem 6. Lines 23-24 will perform if there does not exist a vertex v_i in T having degree at least t in $G[T]$. This process is corresponding to the analysis in Part I of Theorem 6. Subroutine Part I is called and returns a CIST-partition by line 14 in Subroutine Part I.

Algorithm 1. CIST-Partition

Input: A graph G whose vertex set has a bipartition $\{S, T\}$ satisfying consitions $(1) - (3)$ in Theorem 6
Output: A CIST-partition $\{V_1, V_2\}$ of $V(G)$
Let $p = |T|$, $T = \{v_1, \ldots, v_p\}$ and $S = \{s_1, \ldots, s_{t-1}\}$;
Let L be the set of leaves of $G[T]$;
for $i \leftarrow 1$ to p **do**
  $a_i \leftarrow d_G(v_i) - t$;
  if $a_i > 0$ **then**
    $G \leftarrow$ Delete a_i edges in $E(\{v_i\}, S)$ from G;
    end
end
$\ell \leftarrow 1$;
while $\ell \leq p$ **do**
  if $d_{G[T]}(v_\ell) \geq t$ **then**
    Label v_ℓ as v;
    Regard $G[T]$ as a tree rooted at v;
    Let $N_{G[T]}(v) = \{x_1, x_2, \ldots, x_k\}$;
    for $\ell \leftarrow 1$ to k **do**
      $T_\ell \leftarrow$ the subtree of $G[T]$ rooted at x_ℓ;
      $y_\ell \leftarrow V(T_\ell) \cap L$;
    end
    Call Subroutine Part II;
    break
  end
  $\ell \leftarrow \ell + 1$;
  if $\ell > p$ **then**
    Call Subroutine Part I ;
  end
end

We will give an example to show how Algorithm 1 to find its CIST-partition. Let G be a graph with ten vertices satisfying the conditions (1)-(3) in Theorem 6 shown in Fig. 7a. By lines 3-8 in Algorithm 1, by deleting the edge (3,9), we can obtain a spanning subgraph H of G such that $\{S, T\}$ is a bipartition of $V(H)$ which still satisfies the conditions (1), $(2)^*$ and (3) in Theorem 6 shown in Fig. 7b. Since $d_H(1) = t = 4$, lines 11-21 will perform and call Subroutine Part II. Regard $G[T]$ as a tree rooted at the vertex 1. Let T_1 (resp. T_2, T_3 and T_4) be the subtree rooted at the vertex 2 (resp. 3, 4 and 5). In Subroutine Part II, the variable $I = \{1, 4\}$, thus $|I| = 2$. Then lines 8-11 in Subroutine Part II will perform. Choose the elements 1 and 4 in I and 3 not in I. Thus we can get a CIST-partition with two parts $V_2 = \{2, 4, 5, 9, 10\}$ and $V_1 = \{1, 3, 6, 7, 8\}$ shown in Fig. 7c by red and blue vertices respectively. The induced subgraphs $H[V_1]$ and $H[V_2]$ are shown in Fig. 7c by blue and red edges respectively and Fig. 7d shows the bipartite graph $B(V_1, V_2, H)$.

Subroutine. Part I

$i \leftarrow 1$;
while $i < t$ **do**
  $j \leftarrow 1$;
  while $N_S(v_j) \neq \{s_i\}$ and $j \leq p$ **do**
    $j \leftarrow j + 1$;
  end
  if $j > p$ **then**
    $s^* \leftarrow s_i$;
    break
  end
  $i \leftarrow i + 1$;
end
$t^* \leftarrow$ an element in L;
$V_1 \leftarrow (T \setminus \{t^*\}) \cup \{s^*\}$, $V_2 \leftarrow V \setminus V_1$;
return $\{V_1, V_2\}$

Subroutine. Part II

if $t \geq 5$ **then**
  $V_1 \leftarrow V(T_1) \cup V(T_2) \cup (\bigcup_{j=3}^{t-1} \{s_j\})$; $V_2 \leftarrow V(G) \setminus V_1$;
else
  $I \leftarrow \{i : |V(T_i)| = 1$ for $i \in \{1, 2, 3, 4\}\}$;
  if $|I| = 4$ **then**
    $V_1 \leftarrow \{v\} \cup S$; $V_2 \leftarrow V(G) \setminus V_1$;
  else if $2 \leq |I| \leq 3$ **then**
    $r_1, r_2 \leftarrow$ two elements from I;
    $r \leftarrow$ an element from $\{1, 2, 3, 4\} \setminus I$;
    $V_2 \leftarrow \{x_r\} \cup V(T_{r_1}) \cup V(T_{r_2}) \cup \{s_2, s_3\}$;
    $V_1 \leftarrow V \setminus V_2$;
  else
    $r_1, r_2, r_3 \leftarrow$ three elements from $\{1, 2, 3, 4\} \setminus I$;
    $r_4 \leftarrow$ the element in $I \setminus \{r_1, r_2, r_3\}$;
    $s^* \leftarrow$ an element from $N_S(x_{r_3}) \cap N_S(x_{r_4})$;
    $X \leftarrow \{x | x \in (V(T_3) \cup V(T_4)) \setminus (\{x_3\} \cup \{y_3\} \cup \{x_4\} \cup \{y_4\})\}$ and $N_S(x) = S \setminus \{s^*\}$;
    if $X \neq \emptyset$ **then**
      $x^* \leftarrow$ the vertex such that $\min_{x \in X} d_{T_{r_3}}(x, x_{r_3})$;
      $V_2 \leftarrow \{x_{r_1}, x_{r_3}, x_{r_4}, x^*\} \cup S \setminus \{s^*\}$;
      $V_1 \leftarrow V \setminus V_2$;
    else
      $V_2 \leftarrow \{x_{r_1}, x_{r_3}, x_{r_4}\} \cup \{y_{r_3}, y_{r_4}\} \cup S \setminus \{s^*\}$;
      $V_1 \leftarrow V \setminus V_2$;
    end
  end
end
return$\{V_1, V_2\}$

4 The Advantages and Applications of Main Results

The differences among our results and some known conclusions are analyzed as follows. Moreover, we will show that the bound of the parameter t in Theorem 6 is tight.

Note 1. *There exists an infinite number of graphs which satisfy the conditions in Theorem 6 and fail the conditions in each of Theorems 3, 4 and 5.*

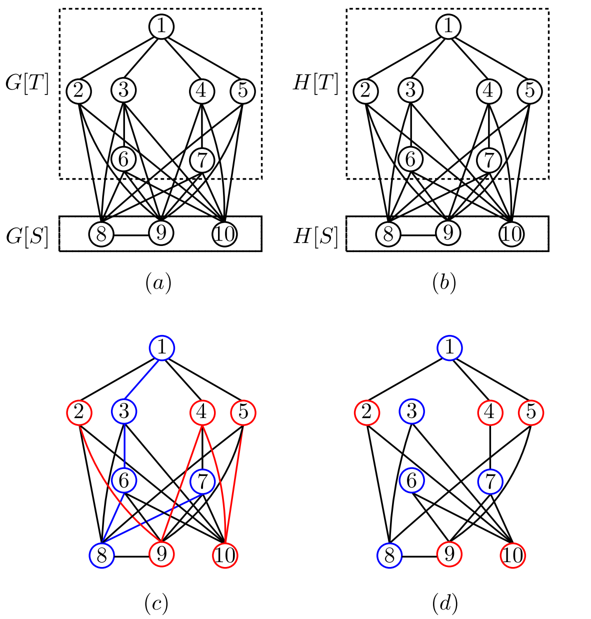

Fig. 7. The local illustration of (a) G; (b) $H = G - \{(3,9)\}$; (c) $H[V_1]$ and $H[V_2]$; and (d) $B(V_1, V_2, H)$.

In the following, we give examples to support Note 1. For example, let G be a graph which has a bipartition $\{S, T\}$ satisfying the conditions (1), (2)* and (3) in Theorem 6. Moreover, assume that $G[T]$ is a tree rooted at the vertex v with t leaves, $d_{G[T]}(v) = t$ and S is an independent set of G. Let the number of vertices in T with degree two in $G[T]$ be x. Then $|T| = x + t + 1$ and $|S| = t - 1$, which implies that $|V(G)| = x + 2t$. Let vertices in T with degree two in $G[T]$ be $v_1, v_2, \ldots, v_x$, leaves in $G[T]$ be $y_1, y_2, \ldots, y_t$ and $S = \{s_1, s_2, \ldots, s_{t-1}\}$.

By Table 2, the minimum degree of G is t. Then $t < \frac{|V(G)|}{2} = \frac{x+2t}{2}$ if $x > 0$, failing to meet the condition in Theorem 3. Every two leaves y_i and y_j with $i, j \in \{1, \ldots, t\}$ in $G[T]$ are not adjacent in G, and by Table 2, we have that $d_G(y_i) + d_G(y_j) = 2t < x + 2t$ if $x > 0$ which fails to meet the condition in Lemma 4. By Table 3, every two leaves y_i and y_j in $G[T]$ with $i, j \in \{1, \ldots, t\}$ have that $|N_G(y_i) \cup N_G(y_j)| = t + 1 < \frac{x+2t}{2}$ for every $x > 2$, which fails to meet the condition in Lemma 5. By the arbitrariness of x satisfying $x > 0$ or $x > 2$, there exists an infinite number of graphs in which the existence of a dual-CISTs can be obtained by Theorem 6, while each of Theorems 3, 4 and 5 does not work.

Note 2. *The bound $t \geq 4$ for the graph G in Theorem 6 is sharp.*

In the following, we give examples to support Note 2. For example, let G be a graph with a vertex bipartition $\{S, T\}$ satisfying the following conditions: (i) $G[T]$ is a tree rooted at the vertex v with three leaves, where $d_{G[T]}(v) = 3$; (ii) each vertex in $G[T]$ with degree i has exactly $|S| + 1 - i$ neighbors in S; and (iii) $G[S]$ is an independent set with two vertices. Here $t = 3$. Assume that the number of vertices in T with degree two in $G[T]$ be x. Then $|T| = x + 4$ and $|S| = 2$, which implies that $|V(G)| = x + 6$ and $|E(G[T])| = |T| - 1 = x + 3$. We have that $|E(G)| = |E(T, S)| + |E(G(T))| = x + 6 + x + 3 = 2x + 9$. By Theorem 1, two CISTs are edge-disjoint. Since each spanning tree of G has $x + 5$ edges and $2x + 9 < 2(x + 5)$ if $x \geq 0$, the bound $t \geq 4$ is sharp.

TABLE 2
The Vertex Degree in G

The vertex in G	Degree of the vertex
v	t
v_i $(1 \leq i \leq x)$	t
y_j $(1 \leq j \leq t)$	t
s_ℓ $(1 \leq \ell \leq t-1)$	$\geq t$ and $\leq x + t$

TABLE 3
Number of Neighbors of Two Non-Adjacent Vertices in G

Two non-adjacent vertices in G	No. of neighbors
v and v_i $(i \in \{1, \ldots, x\})$	$\leq 2t$
v and y_j $(j \in \{1, \ldots, t\})$	$\leq 2t$
v and s_ℓ $(\ell \in \{1, \ldots, t-1\})$	$\leq x + t$
v_i and v_j $(i, j \in \{1, \ldots, x\})$	$\leq t + 3$
v_i and y_j $(i \in \{1, \ldots, x\}, j \in \{1, \ldots, t\})$	$\leq t + 2$
v_i and s_ℓ $(i \in \{1, \ldots, x\}, \ell \in \{1, \ldots, t-1\})$	$\leq x + 2t - 2$
y_i and y_j $(i, j \in \{1, \ldots, t\})$	$\leq t + 1$
y_j and s_ℓ $(j \in \{1, \ldots, t\}, \ell \in \{1, \ldots, t-1\})$	$\leq x + 2t - 1$
s_ℓ and s_h $(\ell, h \in \{1, \ldots, t-1\})$	$\leq x + t$

5 Conclusion

It is interesting that several well-known conditions for Hamiltonian graphs are also sufficient conditions for completely independent spanning trees. In this article, we show that a new Hamiltonian sufficient condition implies the existence of dual-CISTs in an infinite class of networks, including irregular network topology. The results obtained here inspire the researchers to study whether they are some other sufficient conditions of dual-CISTs. In addition, the conditions in our results provide more novel insights and guideline views on the construction of fault-tolerant networks. Appropriate protection routing can be configurated by the dual-CISTs in these networks at the same time. By exploring CISTs in this way will it be possible to move towards a better understanding of how to design more reliable protection routing.

Acknowledgments

The authors express their sincere thanks to the editor and the anonymous referees for their valuable suggestions which improved the original manuscript.

References

[1] J. Tapolcai, “Sufficient conditions for protection routing in IP networks,” *Optim. Lett.*, vol. 7, no. 4, pp. 723–730, 2013.
[2] K.-J. Pai and J.-M. Chang, “Dual-CISTs: Configuring a protection routing on some Cayley networks,” *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1112–1123, Jun. 2019.
[3] S. Sundaram and C. N. Hadjicostis, “Distributed function calculation via linear iterative strategies in the presence ofmalicious agents,” *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1495–1508, Jul. 2011.
[4] A. Itai and M. Rodeh, “The multi-tree approach to reliability in distributed networks,” *Inf. Comput.*, vol. 79, pp. 43–59, 1988.
[5] J.-S. Yang, J.-M. Chang, S.-M. Tang, and Y.-L. Wang, “Reducing the height of independent spanning trees in chordal rings,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 5, pp. 644–657, May 2007.
[6] S.-M. Tang, J.-S. Yang, Y.-L. Wang, and J.-M. Chang, “Independent spanning trees on multidimensional torus networks,” *IEEE Trans. Comput.*, vol. 59, no. 1, pp. 93–102, Jan. 2010.

[7] C.-F. Lin, J.-F. Huang, and S.-Y. Hsieh, "Constructing independent spanning trees on transposition networks," *IEEE Access*, vol. 8, pp. 147 122–147 132, 2020.

[8] J.-F. Huang, S.-S. Kao, S.-Y. Hsieh, and R. Klasing, "Top-down construction of independent spanning trees in alternating group networks," *IEEE Access*, vol. 8, pp. 112 333–112 347, 2020.

[9] D.-W. Cheng, C.-T. Chan, and S.-Y. Hsieh, "Constructing independent spanning trees on pancake networks," *IEEE Access*, vol. 8, pp. 3427–3433, 2020.

[10] J.-S. Yang, J.-M. Chang, K.-J. Pai, and H.-C. Chan, "Parallel construction of independent spanning trees on enhanced hypercubes," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 11, pp. 3090–3098, Nov. 2015.

[11] Y. Wang, J. Fan, G. Zhou, and X. Jia, "Independent spanning trees on twisted cubes," *J. Parallel Distrib. Comput.*, vol. 72, pp. 58–69, 2012.

[12] B. Cheng, J. Fan, J. W. Yang, Y. Wang, and S. K. Zhang, "Independent spanning trees on special BC networks," *Appl. Mech. Mater.*, vol. 263–266, pp. 3301–3305, 2012.

[13] B. Cheng, J. Fan, X. Jia, and J. Wang, "Dimension-adjacent trees and parallel construction of independent spanning trees on crossed cubes," *J. Parallel Distrib. Comput.*, vol. 73, pp. 641–652, 2013.

[14] B. Cheng, J. Fan, X. Jia, and S. Zhang, "Independent spanning trees in crossed cubes," *Inf. Sci.*, vol. 233, pp. 276–289, 2013.

[15] J.-S. Yang, H.-C. Chan, and J.-M. Chang, "Broadcasting secure messages via optimal independent spanning trees in folded hypercubes," *Discrete Appl. Math.*, vol. 159, pp. 1254–1263, 2011.

[16] T. Hasunuma, "Completely independent spanning trees in the underlying graph of a line digraph," *Discrete Math.*, vol. 234, no. 1–3, pp. 149–157, 2001.

[17] T. Hasunuma, "Completely independent spanning trees in maximal planar graphs," in *Proc. 28th Int. Workshop Graph-Theoretic Concepts Comput. Sci.*, 2002, pp. 235–245.

[18] F. Péterfalvi, "Two counterexamples on completely independent spanning trees," *Discrete Math.*, vol. 312, no. 4, pp. 808–810, 2012.

[19] B. Darties, N. Gastineau, and O. Togni, "Completely independent spanning trees in some regular graphs," *Discrete Appl. Math.*, vol. 217, no. 2, pp. 163–174, 2017.

[20] T. Hasunuma and C. Morisaka, "Completely independent spanning trees in torus networks," *Networks*, vol. 60, no. 1, pp. 59–69, 2012.

[21] M. Matsushita, Y. Otachi, and T. Araki, "Completely independent spanning trees in (partial) k-trees," *Discuss. Math. Graph Theory*, vol. 35, no. 3, pp. 427–437, 2015.

[22] K.-J. Pai, S.-M. Tang, J.-M. Chang, and J.-S. Yang, "Completely independent spanning trees on complete graphs, complete bipartite graphs and complete tripartite graphs," in *Proc. Advances Intell. Sys. Appl.*, 2013, pp. 107–113.

[23] B. Cheng, D. Wang, and J. Fan, "Constructing completely independent spanning trees in crossed cubes," *Discrete Appl. Math.*, vol. 219, pp. 100–109, 2017.

[24] K.-J. Pai and J.-M. Chang, "Constructing two completely independent spanning trees in hypercube-variant networks," *Theor. Comput. Sci.*, vol. 652, pp. 28–37, 2016.

[25] K.-J. Pai, R.-S. Chang, R.-Y. Wu, and J.-M. Chang, "A two-stages tree-searching algorithm for finding three completely independent spanning trees," *Theor. Comput. Sci.*, vol. 784, no. 13, pp. 65–74, 2019.

[26] K.-J. Pai, R.-S. Chang, R.-Y. Wu, and J.-M. Chang, "Three completely independent spanning trees of crossed cubes with application to secure-protection routing," *Inf. Sci.*, vol. 541, pp. 516–530, 2020.

[27] K.-J. Pai, R.-S. Chang, and J.-M. Chang, "A protection routing with secure mechanism in Möbius cubes," *J. Parallel Distrib. Comput.*, vol. 140, pp. 1–12, 2020.

[28] X.-W. Qin and R.-X. Hao, "Reliability analysis based on the dual-CIST in shuffle-cubes," *Appl. Math. Comput.*, vol. 397, 2021, Art. no. 125900.

[29] X.-W. Qin, J.-M. Chang, and R.-X. Hao, "Constructing dual-CISTs of DCell data center networks," *Appl. Math. Comput.*, vol. 362, 2019, Art. no. 124546.

[30] X.-W. Qin, R.-X. Hao, and J.-M. Chang, "The existence of completely independent spanning trees for some compound graphs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 1, pp. 201–210, Jan. 2020.

[31] G. Chen, B. Cheng, and D. Wang, "Constructing completely independent spanning trees in data center network based on augmented cube," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 3, pp. 665–673, Mar. 2021.

[32] K.-J. Pai, R.-S. Chang, and J.-M. Chang, "A well-equalized 3-CIST partition of alternating group graphs," *Inf. Process. Lett.*, vol. 155, 2020, Art. no. 105874.

[33] T. Araki, "Dirac's condition for completely independent spanning trees," *J. Graph Theory*, vol. 77, no. 3, pp. 171–179, 2014.

[34] G. Fan, Y. Hong, and Q. Liu, "Ore's condition for completely independent spanning trees," *Discrete Appl. Math.*, vol. 177, pp. 95–100, 2014.

[35] X. Hong and H. Zhang, "A Hamilton sufficient condition for completely independent spanning tree," *Discrete Appl. Math.*, vol. 279, pp. 183–187, 2020.

[36] X.-W. Qin, R.-X. Hao, K.-J. Pai, and J.-M Chang, "Comments on 'A Hamilton sufficient condition for completely independent spanning tree'," *Discrete Appl. Math.*, vol. 283, pp. 730–733, 2020.

[37] X. Hong and Q. Liu, "Degree condition for completely independent spanning trees," *Inf. Process. Lett.*, vol. 116, no. 10, pp. 644–648, 2016.

[38] H.-Y. Chang, H.-L. Wang, J.-S. Yang, and J.-M. Chang, "A note on the degree condition of completely independent spanning trees," *IEICE Trans. Fundam.*, vol. E98-A, no. 10, pp. 2191–2193, 2015.

[39] T. Hasunuma, "Minimum degree conditions and optimal graphs for completely independent spanning trees," in *Proc. 26th Int. Workshop Combinatorial Algorithms*, 2016, pp. 260–273.

[40] F. Cao, "Decycling number and nonseparating independence number of graphs," Ph.D. Thesis, East China Normal Univ., 2020, doi: 10.27149/d.cnki.ghdsu.2020.000086.

Xiao-Wen Qin received the master's degree from the Department of Mathematics, Beijing Jiaotong University, Beijing, China. She is currently working toward the PhD degree in the Department of Mathematics, Beijing Jiaotong University, Beijing, China from September 2018. Her research interests include graph theory, fault diagnosis, network reliability, data center networks, and high-performance algorithm.

Rong-Xia Hao received the PhD degree from Beijing Jiaotong University, Beijing, China, in 2002. From 1998 to 2006, she was an associate professor. Since 2006, she was a professor with the Department of Mathematic, Beijing Jiaotong University. She received Beijing Jiaotong University Zhi Jin Foundation Outstanding Youth Teaching Award in 2007 and the First Prize of 2008 Excellent Paper Awarded by Beijing Operations Research Society. Her research interests include graph theory, interconnection network, and fault tolerant computing.

Jie Wu (Fellow, IEEE) is currently the director of Center for Networked Computing and a Laura H. Carnell professor with Temple University. He also serves as the director of International Affairs with the College of Science and Technology. He served as the chair for the Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and the associate vice provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was the program director of the National Science Foundation. He was a distinguished professor with Florida Atlantic University. He regularly publishes in scholarly journals, conference proceedings, and books. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He serves on several editorial boards including the *IEEE Transactions on Service Computing* and the *Journal of Parallel and Distributed Computing*. He was an IEEE Computer Society distinguished visitor, ACM distinguished speaker, and the chair of the IEEE Technical Committee on Distributed Processing (TCDP). He is a CCF distinguished speaker. He was the general co-chair of IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, and the program co-chair of the IEEE INFOCOM 2011 and CCF CNCC 2013. He was a recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.