

Edge-disjoint spanning trees and the number of maximum state circles of a graph

Xiaoli Ma¹ · Baoyindureng Wu¹ · Xian'an Jin²

Published online: 27 January 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract Motivated by the connection with the genus of unoriented alternating links, Jin et al. (Acta Math Appl Sin Engl Ser, 2015) introduced the number of maximum state circles of a plane graph G , denoted by $s_{\max}(G)$, and proved that $s_{\max}(G) = \max\{e(H) + 2c(H) - v(H) \mid H \text{ is a spanning subgraph of } G\}$, where $e(H)$, $c(H)$ and $v(H)$ denote the size, the number of connected components and the order of H , respectively. In this paper, we show that for any (not necessarily planar) graph G , $s_{\max}(G)$ can be achieved by the spanning subgraph H of G whose each connected component is a maximal subgraph of G with two edge-disjoint spanning trees. Such a spanning subgraph is proved to be unique and we present a polynomial-time algorithm to find such a spanning subgraph for any graph G .

Keywords Spanning trees · Polynomial-time algorithm · State circle · Link

1 Introduction

All graphs considered in this paper are undirected, finite, may have parallel edges, but have no loops. Let $G = (V(G), E(G))$ be a graph. The *order* and *size* of G are

Research supported by NSFC (No. 11571294, 11671336).

✉ Baoyindureng Wu
baoywu@163.com

Xian'an Jin
xajin@xmu.edu.cn

¹ College of Mathematics and System Sciences, Xinjiang University, Urumqi 830046, Xinjiang, People's Republic of China

² School of Mathematical Sciences, Xiamen University, Xiamen 361005, Fujian, People's Republic of China

denoted respectively by $v(G)$ and $e(G)$, i.e. $v(G) = |V(G)|$, $e(G) = |E(G)|$. A graph is said to be *trivial* if it has only one vertex and *nontrivial* otherwise. The graph without edges are *empty graphs*. The case $V(G) = \emptyset$ is sometimes also included in the set of graphs and called the *null graph*. The *degree* $d_G(v)$ of a vertex v in G is the number of edges incident with v . A graph is called *k-regular* if all vertices have degree k . A nontrivial graph is called *k-edge connected* if at least k edges are needed to be removed in order to disconnect the graph.

A subgraph H of G is called a *spanning subgraph* of a graph G if $V(H) = V(G)$. For a graph G , we use $\tau(G)$ to denote the maximum number of edge-disjoint spanning trees in G . A subgraph obtained by vertex deletions only is called an *induced subgraph*. For $X \subseteq V(G)$, $G[X]$ denotes the subgraph induced by X , i.e. $G[X] = G - (V \setminus X)$.

The *union* of two graphs G_1 and G_2 is the graph $G_1 \cup G_2$ with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. The *intersection* $G_1 \cap G_2$ of G_1 and G_2 is defined analogously.

A graph which can be drawn in the plane \mathbb{E}^2 in such a way that edges meet only at points corresponding to their common ends is called a *planar graph*, and such a drawing is called a *planar embedding* of the graph or a *plane graph*. For undefined notations and terminologies, we refer the readers to Bondy and Murty (1976).

A *link* L is the disjoint union of embeddings of circles S^1 's (simple closed curves) into the Euclidean 3-space \mathbb{E}^3 . A *link diagram* D is a planar representation of a link L . Given a nontrivial connected plane graph G , we can construct an alternating link diagram $D(G)$ via the medial construction as shown in Fig. 1. The trivial graph corresponds to a simple circle surrounding it, i.e. the trivial knot diagram. When G is disconnected, the corresponding a link diagram will be the disjoint union of diagrams corresponding to components of G .

For each crossing of a link diagram there are two ways to smooth the crossing: the *A-splitting* (parallel to the edge of G) and the *B-splitting* (crossing the edge of G) as illustration in Fig. 2. Let D be a link diagram. A *state* S of D is a selection for each crossing to be *A-split* or *B-split*. Give a state S , making the corresponding split for each crossing gives a number of disjoint embedded closed circles, called *state circles*, for S . We call the state which possesses maximum number of state circles to be a *maximum state*. Note that the maximum state may not be unique.

The idea of studying links via graphs has been proven to be successful in many ways, for instance, the proof (Thistlethwaite 1987) of a near 100-year conjecture posed

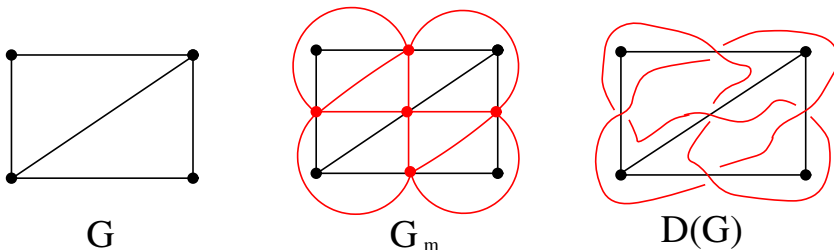


Fig. 1 A plane graph G and its medial graph G_m (in red); turning vertices of G_m to crossings of link diagrams: the alternating link diagram $D(G)$ (in red) constructed from G (Color figure online)

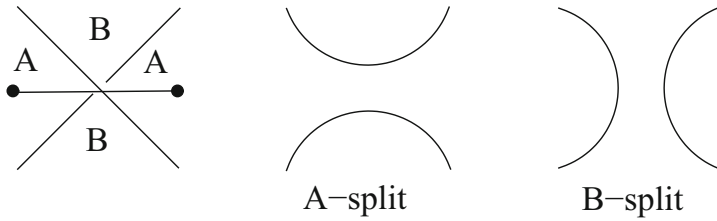


Fig. 2 A crossing and its A-split and B-split

by Tait in the late 1900s. Let G be a plane graph. Jin et al. (2015) defined $s_{\max}(G)$ to be the number of state circles of a maximum state of $D(G)$, call it *the number of maximum state circles* of G . In other words, $s_{\max}(G)$ is the maximum number of cycles over all bent Eulerian partitions of G_m .

This number is an upper bound of the maximum number of circles of Seifert states of the corresponding alternating link diagram, while the maximum number of circles of Seifert states of an alternating link diagram is related to the genus of corresponding unoriented alternating link, a classical geometric invariant of links (Cheng et al. 2010; Liu and Zhang 2012). In addition, Nakamura et al. (2014) introduced the state polynomial $P(D; x)$ for a virtual link diagram D which includes a classical link diagram as a special case. So, $s_{\max}(G)$ is exactly the degree of $P(D(G); x)$ plus one.

Jin et al. (2015) observed that

$$s_{\max}(G) = \max_{H \subseteq G} \{e(H) + 2c(H) - v(H)\}, \tag{1}$$

where the summation is over all spanning subgraphs H of G and $c(H)$ is number of connected components of H . Thus, the notion of the number of maximum state circles can be extended from planar graphs to any abstract graphs. Later we shall use $s(H)$ to denote $e(H) + 2c(H) - v(H)$.

From (1), $s_{\max}(G)$ is independent of the particular embedding of a planar graph G . A weak point of (1) is that computing $s_{\max}(G)$ for a plane graph G is still time consuming. In Sect. 2, we show that

Theorem 1.1 *For any graph G ,*

$$s_{\max}(G) = s(H) = e(H) + 2c(H) - v(H),$$

where H is the spanning subgraph of G whose each component C is a maximal subgraph of G with $\tau(C) \geq 2$.

Note that “ C is a maximal subgraph of G with $\tau(C) \geq 2$ ” in Theorem 1.1 means that there is no other subgraph F of G with $\tau(F) \geq 2$ properly contains C , and thus C is an induced subgraph of G . The uniqueness of H will be proved in Theorem 2.4. In Sect. 3, we provide a polynomial-time algorithm to find the spanning subgraph H of a graph G , as described in the above theorem. In the last Sect. 4, we pose a problem for further study.

2 The proof and some consequences

A fundamental structural theorem on edge-disjoint spanning trees in a graph, found independently by Nash-Williams (1961) and Tutte (1961), will play an important role in our investigation.

Theorem 2.1 (The Nash-Williams–Tutte Theorem) *A graph G has k edge-disjoint spanning trees if and only if, for every partition \mathcal{P} of $V(G)$ into nonempty parts,*

$$e(G/\mathcal{P}) \geq k(|\mathcal{P}| - 1),$$

where $e(G/\mathcal{P})$ is the graph obtained from G by contracting each part of \mathcal{P} into a single vertex.

Note that the trivial graph K_1 is a graph which contains k edge-disjoint spanning trees for any $k \geq 1$. The following result is an easy consequence of the above theorem.

Corollary 2.2 (Gusfield 1983) *If a graph G is $2k$ -edge-connected, then $\tau(G) \geq k$.*

To prove Theorem 1.1, we need the following lemma.

Lemma 2.3 *Let G_1 and G_2 be two graphs with $V(G_1) \cap V(G_2) \neq \emptyset$. If $\tau(G_i) \geq 2$ for each $i \in \{1, 2\}$, then $\tau(G_1 \cup G_2) \geq 2$.*

Proof Assume that T_1 and T_2 are two edge-disjoint spanning trees of G_1 ; T'_1 and T'_2 are two edge-disjoint spanning trees of G_2 . It is easy to see that $T_1 \cup (T'_1 \setminus E(G_1))$ and $T_2 \cup (T'_2 \setminus E(G_1))$ are two edge-disjoint connected spanning subgraphs of $G_1 \cup G_2$. Each connected spanning subgraph of $G_1 \cup G_2$ contains a spanning tree, thus $\tau(G_1 \cup G_2) \geq 2$. \square

Theorem 2.4 *For a graph G , there is unique spanning subgraph H of G such that each component C of H is a maximal subgraph of G with $\tau(C) \geq 2$.*

Proof By Lemma 2.3, for a vertex $v \in V(G)$, there is unique maximal subgraph G_v of G with $v \in V(G_v)$ and $\tau(G_v) \geq 2$. So, the result follows. \square

Now we are ready to prove our main theorem.

Proof of Theorem 1.1 Suppose that H is a spanning subgraph of G such that $s_{\max}(G) = s(H) = e(H) + 2c(H) - v(H)$ with $c(H)$ as small as possible. Let H_1, H_2, \dots, H_k be all components of H , where $k = c(H)$.

Claim 1 Each H_i is an induced subgraph of G for $i = 1, 2, \dots, k$.

Suppose that H_j is not an induced subgraph of G for some j . Then $e(G[V(H_j)]) > e(H_j)$. Let H' be the spanning subgraph of G obtained from H by replacing H_j with $G[V(H_j)]$. It is clear that $s(H') - s(H) > 0$. This proves the claim.

It remains to show the following claim.

Claim 2 Each component H_i ($1 \leq i \leq k$) is a maximal subgraph of G with two edge-disjoint spanning trees.

Without loss of generality, let H_1 has no two edge-disjoint spanning trees. By Theorem 2.1, there is a partition \mathcal{P} of $V(H_1) : V_{1_1}, V_{1_2}, \dots, V_{1_p}$ such that

$$e(H_1/\mathcal{P}) < 2(|\mathcal{P}| - 1),$$

where $p = |\mathcal{P}|$. Let H' be the spanning subgraph obtained from H by replacing H_1 with $G[V_{1_1}], G[V_{1_2}], \dots, G[V_{1_p}]$. So,

$$\begin{aligned} s(H') - s(H) &= [e(H') + 2c(H') - v(H')] - [e(H) + 2c(H) - v(H)] \\ &= -e(H_1/\mathcal{P}) + 2(|\mathcal{P}| - 1) > 0, \end{aligned}$$

contradicting the choice of H .

Next we show that each H_i is a maximal subgraph of G with two edge-disjoint spanning trees. Suppose that H_1 is not maximal with respect to the above property. By Claim 1, there exists an induced subgraph H'_1 of G with two edge-disjoint spanning trees such that $H_1 \subseteq H'_1$ and $V(H_1) \subset V(H'_1)$.

Let $\mathcal{A} = \{j \mid V(H_j) \cap V(H'_1) \neq \emptyset \text{ with } 1 \leq j \leq k\}$ and let $H_1^* = G[\cup_{j \in \mathcal{A}} V(H_j)]$. By Lemma 2.3, H_1^* has two edge-disjoint spanning trees. By Theorem 2.1, for any vertex partition \mathcal{P}_1 of H_1^* , $e(H_1^*/\mathcal{P}_1) - 2(|\mathcal{P}_1| - 1) \geq 0$.

Let H'' be the spanning subgraph of G obtained from H by replacing those components $H_j, j \in \mathcal{A}$ with H_1^* . Let \mathcal{P}_1 be the partition of $V(H_1^*)$ composed of $V(H_j), j \in \mathcal{A}$. Therefore,

$$\begin{aligned} s(H'') - s(H) &= [e(H'') + 2c(H'') - v(H'')] - [e(H) + 2c(H) - v(H)] \\ &= e(H_1^*/\mathcal{P}_1) - 2(|\mathcal{P}_1| - 1) \geq 0. \end{aligned}$$

By the maximality of $s(H), s(H'') = s(H)$. However, $c(H'') < c(H)$, contradicting the choice of H .

This completes the proof of the claim and thus the proof of the theorem. □

Corollary 2.5 *If G is a graph with $\tau(G) \geq 2$, then $s_{\max}(G) = e(G) - v(G) + 2$.*

Proof By the assumption, G is itself a maximal subgraph of G with $\tau(G) \geq 2$. So, by Theorem 1.1,

$$s_{\max}(G) = s(G) = e(G) - v(G) + 2c(G) = e(G) - v(G) + 2. \tag*{□}$$

Corollary 2.6 *If G is a 4-edge-connected graph, then $s_{\max}(G) = e(G) - v(G) + 2$.*

Proof It is an immediate consequence of Corollaries 2.2 and 2.5. □

For a plane graph $G, f(G)$ denotes the number of faces in G .

Theorem 2.7 [Euler’s Formula (Bondy and Murty 1976)] *For a connected plane graph $G, v(G) - e(G) + f(G) = 2$.*

By Theorem 2.7 and Corollary 2.6, we can immediately obtain the following result in Jin et al. (2015).

Corollary 2.8 [(Theorem 4.4 in Jin et al. (2015))] *If G is a 4-edge-connected plane graph, then $s_{\max}(G) = f(G)$.*

Corollary 2.9 *Let G be a graph. If $\tau(F) < 2$ for any nontrivial subgraph F of G , then*

$$s_{\max}(G) = v(G).$$

Proof Let H be spanning subgraph of G whose each component is a maximal subgraph of G with two edge-disjoint spanning trees. By Theorem 1.1, $s_{\max}(G) = e(H) + 2c(H) - v(H)$. By the assumption, each component of H must be trivial, implying that H is empty. Therefore,

$$s_{\max}(G) = s(H) = e(H) - v(H) + 2c(H) = 0 - v(G) + 2v(G) = v(G).$$

□

Let I_k denotes the graphs with exactly two vertices joined by k parallel edges. Note that there are just two connected cubic graph of order 4, i.e. K_4 and G_4 as shown in Fig. 3 (the left one).

Corollary 2.10 *If G is a connected cubic graph then*

$$s_{\max}(G) = \begin{cases} 3 = v(G) + 1, & \text{if } G \cong I_3 \\ v(G), & \text{otherwise} \end{cases}$$

Proof Since G is cubic, $v(G)$ is even. If $v(G) = 2$, the $G \cong I_3$. It is clear that $\tau(G) \geq 2$, and thus by Corollary 2.5, $s_{\max}(G) = e(G) - v(G) + 2 = 3 = v(G) + 1$.

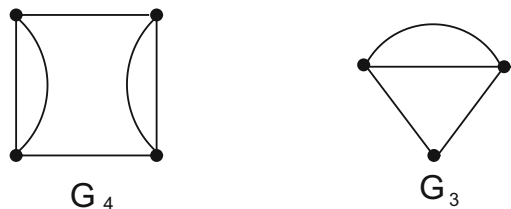
If $v(G) = 4$, then $G \cong K_4$ or $G \cong G_4$. It is clear that $\tau(G) \geq 2$. By Corollary 2.5, $s_{\max}(G) = e(G) - v(G) + 2 = 4 = v(G)$.

Next assume that $v(G) \geq 6$.

Claim If F is a nontrivial subgraph of G with $\tau(F) \geq 2$, then $F \cong I_2$ or $F \cong G_3$.

Since $\tau(F) \geq 2$, we have $e(F) \geq 2(v(F) - 1)$. On the other hand, since $\Delta(F) \leq 3$, $e(F) \leq \frac{3}{2}v(F)$. Combining the above two inequalities with $v(G) \geq 6$, it follows that

Fig. 3 Two small graphs G_4 and G_3 with parallel edges



$v(F) \leq 4$ and $F \neq G$. So, we conclude that $F \cong I_2$ or $F \cong G_3$ (the right one as shown in Fig. 3).

To apply Theorem 1.1, let H be the spanning subgraph of G whose each component C is a maximal subgraph of G with $\tau(C) \geq 2$. By the above claim, each component of H is isomorphic to K_1 , or I_2 , or G_3 . Let c_1, c_2 and c_3 be the number of components of H , which are isomorphic to K_1 , or I_2 , or G_3 .

$$\begin{cases} c_1 + 2c_2 + 3c_3 = v(H) \\ c_1 + c_2 + c_3 = c(H) \\ 2c_2 + 4c_3 = e(H) \end{cases}$$

By Theorem 1.1,

$$\begin{aligned} s_{max}(G) &= s(H) \\ &= e(H) - v(H) + 2c(H) \\ &= (2c_2 + 4c_3) - (c_1 + 2c_2 + 3c_3) + 2(c_1 + c_2 + c_3) \\ &= c_1 + 2c_2 + 3c_3 \\ &= v(G). \end{aligned}$$

So, the proof is completed. □

In next section, we give a polynomial-time algorithm for finding the spanning subgraph H of G such that each component C of H is a maximal subgraph of G with $\tau(C) \geq 2$. Thereby, the computing the $s_{max}(G)$ for any graph is polynomial.

3 An algorithm

The following optimization problem closely related to the above is widely studied, see Clausen and Hansen (1980), Hobbs (1989), Held and Karp (1971), Kundu (1974), Roskind and Tarjan (1985), Tarjan (1974), Tarjan (1976) and Yao (1975).

The edge-disjoint spanning tree problem

Input a graph with $w(e)$ as the multiplicity for each edge e .

Output determine $\tau(G)$ and find $\tau(G)$ edge-disjoint spanning trees of G .

Zhang and Ou (2008) introduced the following operation when they were investigating the dynamic density of a graph. For a connected graph G , let T be a spanning tree of G , and F be a spanning forest of $G \setminus T$.

Zhang–Ou operation

If $E(G) \setminus (T \cup F) \neq \emptyset$, then we choose an edge $e_0 \in E(G) \setminus (T \cup F)$.

First, the edge e_0 is colored red. Then, for the spanning tree T , the subgraph $T + e_0$ of G contains a unique cycle C . Color all these edges in C with red. Second, for each red $e' \in T$, color all edges of the unique cycle contained in $F + e'$ (if this cycle exists). Take all newly colored edge e'' , color all edges in the unique cycle of $T + e''$ and so on. This will be repeated until one of the following two cases occurs:

1. A red edge $e^{(1)}$ joins two distinct components of F ;

2. No more edge can be further colored.

If (1) happens, $F + e^{(1)}$ is larger than F . However, the edge $e^{(1)}$ may not be that extra edge e_0 . That is, it is possible that the edge $e^{(1)}$ came from the spanning tree T . Now, let us trace back our coloring process. The edge $e^{(1)}$ was colored red because it is contained in a cycle of $T + e^{(2)}$, where $e^{(2)}$ was colored red before the edge $e^{(1)}$ was colored. Continue this trace-back procedure, we find a sequence of edges $e^{(1)}, \dots, e^{(s)}$ such that $e^{(i)} \in T$ if i is odd and $e^{(i)} \in F$ if i is even for any $i \geq 1$. Note that s is even and $e^{(s)}$ is the extra edge e_0 that the process started with. In addition, $e^{(i)}$ was colored red because it is contained in the cycle of $T + e^{(i+1)}$ if i is odd ($F + e^{(i+1)}$ if i is even). Now, the forest F is ready to be expanded by the following adjustment,

$$F := F + e^{(1)} - e^{(2)} + \dots + e^{(s-3)} - e^{(s-2)} + e^{(s-1)}$$

and

$$T := T - e^{(1)} + e^{(2)} + \dots - e^{(s-1)} + e^{(s)}.$$

Lemma 3.1 (Page 6 in Zhang and Ou 2008) *If (2) happens, then $\tau(H - e) \geq 2$ for any $e \in E(H)$, where H is the subgraph of G induced by all red edges. In particular, $T \cap H$ and $F \cap H$ are two edge-disjoint spanning trees of H .*

Theorem 3.2 *Let G be a connected graph and let $\mathcal{C}(G) = \{(T, F) \mid T \text{ is a spanning tree of } G, F \text{ is a spanning forest of } G \setminus T\}$. If for a pair $(T', F') \in \mathcal{C}(G)$, the case (2) occurs for any edge $e \in E(G) \setminus (T' \cup F')$, then $|F'| \geq |F|$ for any $(T, F) \in \mathcal{C}(G)$.*

Proof Assume that the result is not true and let (T'', F'') be a pair in $\mathcal{C}(G)$ such that $|F''|$ is maximum, and subject to this, $|(T'' \cup F'') \cap (T' \cup F')|$ is maximum.

Claim 1 There is an (red) edge $e_0 \in (T' \cup F') \setminus (T'' \cup F'')$.

Since $|T'' \cup F''| > |T' \cup F'|$, there exists an edge $e'' \in (T'' \cup F'') \setminus (T' \cup F')$. First color e'' in red and start Zhang–Ou operation for the pair (T', F') . Let H' be the subgraph induced by all red edges. by Lemma 3.1, $\tau(H' - e'') \geq 2$. Moreover, $T' \cap H'$ and $F' \cap H'$ are two edge-disjoint spanning trees of H' . This proves the claim.

Now we start another coloring process. First color e_0 in red and start Zhang–Ou operation for the pair (T'', F'') . By the maximality of F'' , (2) happens. By the argument similar to the proof of Claim 1, there is a red edge $e^* \in (T'' \cup F'') \setminus (T' \cup F')$ and a red edge sequence $e^{(1)}, e^{(2)}, \dots, e^{(r)}$ such that $e^{(1)} = e^*, e^{(r)} = e_0$. We consider two possible cases.

Case 1 $e^* \in T''$

Then r must be even. Let

$$\begin{aligned} T^* &= T'' - e^{(1)} + e^{(2)} - \dots - e^{(r-1)} + e^{(r)} \\ F^* &= F'' - e^{(2)} + e^{(3)} + \dots - e^{(r-2)} + e^{(r-1)}. \end{aligned}$$

Case 2 $e^* \in F''$

Then r must be odd. Let

$$T^* = T'' - e^{(2)} + e^{(3)} - \dots - e^{(r-1)} + e^{(r)}$$

$$F^* = F'' - e^{(1)} + e^{(2)} - \dots - e^{(r-2)} + e^{(r-1)}.$$

In the above two cases, it is clear that $(T^*, F^*) \in \mathcal{C}(G)$ and

$$|(T^* \cup F^*) \cap (T' \cup F')| > |(T'' \cup F'') \cap (T' \cup F')|,$$

contradicting that $|(T'' \cup F'') \cap (T' \cup F')|$ is maximum. □

Theorem 3.3 *Let G be a connected graph and let $\mathcal{C}(G)=\{(T, F) \mid T \text{ is a spanning tree of } G, F \text{ is a spanning forest of } G \setminus T\}$. Assume that $(T', F') \in \mathcal{C}(G)$ with the property that $|F'| \geq |F|$ for any $(T, F) \in \mathcal{C}(G)$ and let F'_1, \dots, F'_t be all components of F' . Let $\Gamma_i = G[V(F'_i)]$ for each i . If $\tau(G') \geq 2$ for a subgraph G' of G , then $G' \subseteq \Gamma_i$ for some $i \in \{1, \dots, t\}$.*

Proof Suppose that G' is a subgraph of G with $\tau(G') \geq 2$ such that $V(G') \cap V(\Gamma_i) \neq \emptyset$ for each $i \in \{1, \dots, s\}$, where $2 \leq s \leq t$. Let T'_1 and T'_2 be two edge-disjoint spanning trees of G' . Note that $T'_1 \cup (T \setminus T'_1)$ is a connected spanning subgraph of G and $T'_2 \cup (F \setminus T'_2)$ is spanning subgraph of G with $c(T'_2 \cup (F \setminus T'_2)) < c(F)$. Moreover, they are edge-disjoint. Take a spanning tree T^* of $T'_1 \cup (T \setminus T'_1)$ and a spanning forest F^* of $T'_2 \cup (F \setminus T'_2)$ with $c(F^*) = c(T'_2 \cup (F \setminus T'_2))$. It is clear that $(T^*, F^*) \in \mathcal{C}(G)$. However, $|F^*| > |F'|$, a contradiction. □

Algorithm 1.

Input : a connected graph G of order n

Output : the spanning subgraph H of G whose each component of H is a maximal subgraph of G with two edge-disjoint spanning trees.

Step 0. $\Gamma := G$ and H be the null graph.

Step 1. Find a spanning tree T of Γ and a spanning forest F of $\Gamma \setminus T$ with $c(F) = c(\Gamma \setminus T)$.

If $c(F) = 1$, go to step 3.1.

Otherwise, go to Step 2.

Step 2. Is $E(\Gamma) \setminus (T \cup F) = \emptyset$? If yes, go to substep 2.1; if no, go to Substep 2.2.

Substep 2.1. Let $F' := F$, go to Step 3.2.

Substep 2.2. Take an edge $e_0 \in E(\Gamma) \setminus (T \cup F)$, do the Zhang–Ou operation until (2) happens for any edge $e \in E(\Gamma) \setminus (T \cup F)$. Denote the resulting pair by (T', F') . Go to Step 3.

Step 3. Is F' connected or not? If yes, go to substep 3.1; if no, go to Substep 3.2.

Substep 3.1. $H := H \cup \Gamma$.

Substep 3.2. Let F'_1, \dots, F'_t be all components of F' . Let $\Gamma := G[V(F'_i)]$ for each i , and go to Step 1.

Step 4. Return H .

Theorem 3.4 *Algorithm 1 is correct. That is, Algorithm 1 finds the spanning subgraph H of G whose each component C is a maximal subgraph of G with $\tau(C) \geq 2$.*

Proof By induction on the order n of G . The result is trivially true for $n = 1$. Now assume that $n \geq 2$. If $\tau(G) = 1$, then after Step 2.2, we will find a pair $(T', F') \in \mathcal{C}(G)$ with maximum $|F'|$. Let F'_1, \dots, F'_t be all components of F' . By Theorem 3.3, any maximal subgraph G' of G with $\tau(G') \geq 2$, $G' \subseteq H_i$ for an integer $i \in \{1, \dots, t\}$, where $\Gamma_i = G[V(F'_i)]$ for each i . By Substep 3.2, we will repeat the algorithm for each Γ_i . Thus, by induction hypothesis, our algorithm will find the spanning subgraph H_i of F_i whose each component C is a maximal subgraph of F_i with $\tau(C) \geq 2$. It is clear that $H = \cup_i H_i$. \square

4 Concluding remark

The genus of an oriented link is the minimum genus of any connected orientable surface spanning it. The genus of an unoriented link L , denoted by $g(L)$, is the minimum taken over all possible orientations. Genus is an important invariant in knot theory. However, it is hard to compute in general.

Let D be a link diagram of a link L . Let \vec{D} be an orientation (going straight ahead when meeting a crossing) of D . Splitting each crossing of \vec{D} in the only way compatible with the orientation, we obtain a set of mutually disjoint simple circles, called *Seifert circles* (Seifert 1935) (see Fig. 4). We use $\vec{s}_{\max}(D)$ to denote the maximum number of Seifert circles of \vec{D} taken over all possible orientations of D . It is clear that for a plane graph G ,

$$\vec{s}_{\max}(D(G)) \leq s_{\max}(D(G)).$$

In some special cases, the equality holds. For example, G is 3-edge connected even plane graph (i.e. the degree of each vertex is even). Let G be a plane graph. In fact, the genus $g(L(G))$ of the link $L(G)$ the link diagram $D(G)$ represents is related to $\vec{s}_{\max}(D(G))$ by the following equation (Crowell 1959; Gabai 1986; Murasugi 1958).

$$g(L(G)) = \frac{2 - \vec{s}_{\max}(D(G)) + e(G) - \mu(L(G))}{2}, \quad (2)$$

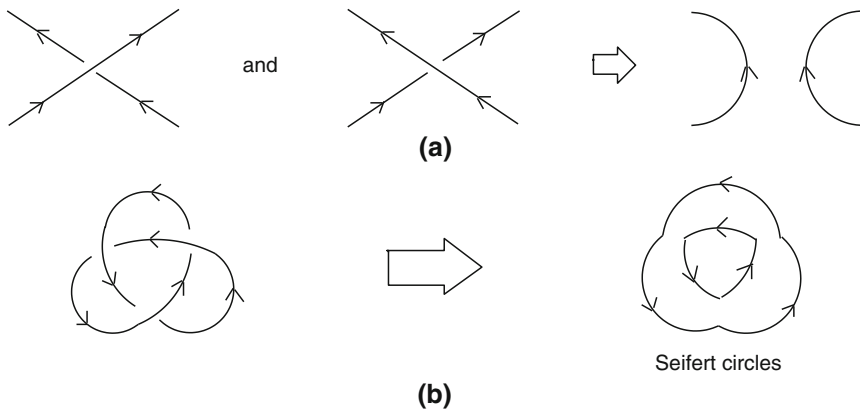


Fig. 4 **a** Splitting a crossing in the way compatible with the orientation; **b** Seifert circles of trefoil knot

where $\mu(L(G))$ is the number of components of the link $L(G)$, i.e. the number of straight-ahead walks of G_m or the number of left-right paths of G .

It is interesting to formulate a graph-theoretical problem and find an efficient algorithm for computing $\vec{s}_{\max}(D(G))$.

Acknowledgements We are grateful to the referees for their constructive comments which helped to improve the presentation of the paper. The first two authors are grateful to Professor C.Q. Zhang for sending the appendix of the paper (Zhang and Ou 2008). The third author thanks Professors F.M. Dong, Yuqing Lin, and Eri Matsudo for some helpful discussions. This work is also supported by President’s Funds of Xiamen University (No. 20720160011).

References

Bondy JA, Murty USR (1976) Graph theory with applications. The Macmillan Press Ltd, London

Cheng X, Liu S, Zhang H, Qiu W (2010) Fabrication of a family of pyramidal links and their genus. MATCH Commun Math Comput Chem 63:623–636

Clausen J, Hansen LA (1980) Finding k edge disjoint spanning trees of minimum total weight network: an application of matroid theory. Math Program Stud 13:88–101

Crowell RH (1959) Genus of alternating link types. Ann Math 69:258–275

Gabai D (1986) Genera of the alternating links. Duke Math J 53:677–681

Gusfield D (1983) Connectivity and edge-disjoint spanning trees. Inform Process Lett 16:87–89

Held M, Karp RM (1971) The traveling salesman and minimum spanning trees. Math Program 1:6–25

Hobbs M (1989) Computing edge-toughness and fractional arboricity. Contemp Math 89:89–106

Jin X, Ge J, Cheng X, Lin Y (2015) The number of circles of a maximum state of a plane graph with applications. Acta Math Appl Sin Engl Ser

Kundu S (1974) Bounds on the number of edge-disjoint spanning trees. J Comb Theory Ser B 17:199–203

Liu S, Zhang H (2012) Genera of the links derived from 2-connected plane graphs. J Knot Theory Ramif. 21:1250129

Murasugi K (1958) On the genus of the alternating knot, I, II. J Math Soc Jpn 10(94–105):235–248

Nakamura T, Nakanishi Y, Satoh S, Tomiyama Y (2014) The state numbers of virtual knot. J Knot Theory Ramif 23:1450016

Nash-Williams CSJA (1961) Edge-disjoint spanning trees of finite graphs. J Lond. Math Soc 36:445–450

Roskind J, Tarjan R (1985) A note on finding minimum-cost edge-disjoint spanning trees. Math Oper Res 10:701–708

Seifert H (1935) Über das Geschlecht von Knoten. Math Ann 110:571–592

- Tarjan RE (1974) A good algorithm for edge-disjoint branching. *Inform Process Lett* 3:51–53
- Tarjan RE (1976) Edge-disjoint spanning trees and depth-first search. *Acta Inform* 6:171–185
- Thistlethwaite MB (1987) A spanning tree expansion of the Jones polynomial. *Topology* 26:297–309
- Tutte WT (1961) On the problem of decomposing a graph into n connected factors. *J Lond Math Soc* 36:221–230
- Yao AC (1975) An $O(|E| \log \log |V|)$ algorithm for finding minimum spanning trees. *Inform Process Lett* 4:21–23
- Zhang C, Ou Y (2008) Clustering, community partition and disjoint spanning trees. *ACM Trans Algorithms* 4:35