Finding Tutte Paths in Linear Time

Therese Biedl
David R. Cheriton School of Computer Science, University of Waterloo, Canada
https://cs.uwaterloo.ca/~biedl/
biedl@uwaterloo.ca

Philipp Kindermann
Lehrstuhl für Informatik I, Universität Würzburg, Germany
https://go.uniwue.de/pkinderm
philipp.kindermann@uni-wuerzburg.de

Abstract

It is well-known that every planar graph has a *Tutte path*, i.e., a path P such that any component of $G-P$ has at most three attachment points on P. However, it was only recently shown that such Tutte paths can be found in polynomial time. In this paper, we give a new proof that 3-connected planar graphs have Tutte paths, which leads to a linear-time algorithm to find Tutte paths. Furthermore, our Tutte path has special properties: it visits all exterior vertices, all components of $G-P$ have exactly three attachment points, and we can assign distinct representatives to them that are interior vertices. Finally, our running time bound is slightly stronger; we can bound it in terms of the degrees of the faces that are incident to P. This allows us to find some applications of Tutte paths (such as binary spanning trees and 2-walks) in linear time as well.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases planar graph, Tutte path, Hamiltonian path, 2-walk, linear time

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.23

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at [5]: http://arxiv.org/abs/1812.04543.

Funding *Therese Biedl*: Supported by NSERC.

1 Introduction

A *Tutte path* is a well-known generalization of Hamiltonian paths that allows to visit only a subset of the vertices of the graph, as long as all remaining vertices are in components with at most three attachment points. (Detailed definitions are below.) They have been studied extensively, especially for planar graphs, starting from Tutte's original result:

▶ **Theorem 1** (Tutte [19]). *Let G be a 2-connected planar graph with distinct vertices X, Y on the outer face. Let α be an edge on the outer face. Then G has a Tutte path from X to Y that uses edge α.*

We refer to the recent work by Schmid and Schmidt [15] for a detailed review of the history and applications of Tutte paths. It was long not known how to compute a Tutte path in less than exponential time. A breakthrough was achieved by Schmid and Schmidt in 2015 [13, 14], when they showed that one can find a Tutte path for 3-connected planar graphs in polynomial time. In 2018, the same authors then argued that Tutte paths can be found in polynomial time even for 2-connected planar graphs [15]. For both papers, the main insight is to prove the existence of a Tutte path by splitting the graph into non-overlapping subgraphs to recurse on; the split can be found in linear time and therefore the running time becomes quadratic.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 23; pp. 23:1–23:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we show that Tutte paths can be computed in linear time. To do so, we give an entirely different proof of the existence of a Tutte path for 3-connected planar graph. This proof is very simple if the graph is triangulated, but requires more care when faces have larger degrees. Our path (and also the one by Schmid and Schmidt [13, 14]) comes with a system of distinct representatives, i.e., an injective assignment from the components of $G \setminus P$ to vertices of P that are attachment points. Such representatives are useful for various applications of Tutte paths.

Our proof for 3-connected planar graphs is based on a Hamiltonian-path proof by Asano, Kikuchi and Saito [1] that was designed to give a linear-time algorithm; with arguments much as in their paper we can therefore find the Tutte path and its representatives in linear time. Since 3-connected planar graphs are (as we argue) the bottleneck in finding Tutte paths, this shows that the path of Theorem 1 can be found in linear time.

1.1 Preliminaries

We assume familiarity with graphs, see, e.g., Diestel [7]. Throughout this paper, $G = (V, E)$ denotes a graph with n vertices and m edges. We assume that G is *planar*, i.e., can be drawn in 2D without edge crossings. A planar drawing of G splits $\mathbb{R}^2$ into connected regions called *faces*; the unbounded region is the *outer face* while all others are called *interior faces.* A vertex/edge is called *exterior* if it is incident to the outer face and *interior* otherwise. We assume throughout that G is *plane*, i.e., one particular abstract drawing of G has been fixed (by giving the clockwise order of edges around each vertex and the edges that are on the outer face). Any subgraph of G *inherits* this planar embedding, i.e., uses the induced order of edges and as outer face the face that contained the outer face of G. The following notion will be convenient: Two vertices v and w are *interior-face-adjacent* (in a plane graph G) if there exists an interior face that is incident to both v and w. We will simply write *face-adjacent* since we never consider adjacency via the outer face.

Nooses and connectivity. For a fixed planar drawing of G, let a *noose* be a simple closed curve $\mathcal{N}$ that goes through vertices and faces and crosses no edge except at endpoints. A noose can be described as a cyclic sequence $\langle x_0, f_1, x_1, \ldots, f_s, x_s{=}x_0 \rangle$ of vertices and faces such that f_i contains x_{i-1} and x_i, and hence is independent of the chosen drawing. Frequently, the choice of faces will be clear from context or irrelevant; we then say that $\mathcal{N} = \langle x_0, \ldots, x_s{=}x_0 \rangle$ *goes through* $\{x_1, \ldots, x_s\}$. The subgraph *inside/outside* $\mathcal{N}$ is the graph induced by the vertices that are on or inside/outside $\mathcal{N}$. The subgraph *strictly inside/outside* is obtained from this by deleting the vertices on $\mathcal{N}$.

A graph G is *connected* if for any two vertices v, w there is a path from v to w in G. A *cutting k-set* in G is a set $S = \{x_1, \ldots, x_k\}$ of vertices such that $G \setminus S$ has more connected components than G. We call it a *cutting pair* for $k = 2$ and a *cutting triplet* for $k = 3$. A graph G is called *k-connected* if it has no cutting $(k-1)$-set. Since we are only studying planar graphs, it will be convenient to use a characterization of connectivity via nooses. Consider a noose $\mathcal{N}$ that goes through $\{x_1, \ldots, x_k\}$ (and no other vertices), and there are vertices both strictly inside and strictly outside $\mathcal{N}$. Then clearly $S = \{x_1, \ldots, x_k\}$ is a cutting k-set. Vice versa, in a planar graph, any cutting k-set S for $k = 1, 2, 3$ gives rise to a noose $\mathcal{N}$ through S that has vertices both strictly inside and strictly outside. A *strict cut component* C of S is a subgraph strictly inside a noose $\mathcal{N}$ through some of the vertices of S such that C contains at least one vertex not in S and is inclusion-minimal among all such nooses. In particular, a strict cut component C contains no vertices or edges of S. A *cut component* C^+ is obtained from a strict cut component C by re-inserting those vertices of S that have neighbours of C, as well as the edges from them to C.

Hamiltonian paths and Tutte paths. A *Hamiltonian path* is a path that visits every vertex exactly once. To generalize it to Tutte paths, we need more definitions. Fix a path P in the graph. A *P-bridge* C is a cut component of P; its *attachment points* are its vertices on P.[1] A *Tutte path* is a path P such that any P-bridge C has at most three attachment points, and if C contains exterior edges, then it has at most two attachment points. Our Tutte paths for 3-connected graphs will be such that no P-bridges contain exterior edges, so the second restriction holds automatically.

A Tutte path with a *system of distinct representatives* (SDR), also called a *T_{SDR}-path* for short, is a Tutte path P together with an injective assignment σ from the P-bridges to vertices in P such that for every P-bridge C vertex $\sigma(C)$ is an attachment point of C.

Given a path P in a plane graph, we denote by $F(P)$ the set of all interior faces that contain at least one vertex of P.

1.2 From 3-connected to 2-connected

In this section, we show that, to find the path of Theorem 1 efficiently, it suffices to consider 3-connected planar graphs.

We re-prove Theorem 1, presuming it holds for 3-connected planar graphs, by induction on the number of vertices with an inner induction on the number of exterior vertices. Say we want to find a Tutte path from X to Y that uses exterior edge $\alpha = (U, W)$, where X, Y are exterior vertices. In the base case, G is 3-connected and we are done. So assume that G has cutting pairs. If edge (X, Y) does not exist, then add it in such a way that α stays exterior, and find a Tutte path P in the resulting graph recursively (it has fewer exterior vertices). Since $\{X, Y\} \neq \{U, W\}$ (because $(U, W) \in G$ while $(X, Y) \notin G$), path P visits at least one vertex other than X, Y, and so cannot use edge (X, Y). So it is also a Tutte path of G.

Now, assume that (X, Y) exists. Repeatedly split the graph at any cutting pair $\{u, v\}$ into cut components $C_1, \ldots, C_k$, and store the *3-connected components* $C_1^+, \ldots, C_k^+$ (induced by the cut components with an additional virtual edge between the cutting pairs) in a so-called *SPQR-tree* [6, 9], which additionally creates one leaf node for every edge of G. This can be done in linear time [10].

Root the SPQR-tree at the node of edge (X, Y). For each 3-connected component C^+ other than the root, set $\{X_C, Y_C\}$ to be the cutting pair that C^+ has in common with its parent component, and observe that these two vertices are necessarily exterior in C since X, Y are exterior in G; see Figure 1.

If C^+ has only these two vertices, then let P_C be the path (X_C, Y_C). Otherwise, define an edge $\alpha_C \neq (X_C, Y_C)$ of C^+ as follows: If the node of α is a descendant of C^+, then let α_C be the virtual edge of C^+ that it shares with the child that leads to this descendant. Note that α_C is a virtual edge, and it is necessarily on the outer face of C since α is on the outer face of G. Otherwise (α is not in a descendant of C^+) choose α_C to be an arbitrary exterior edge of C other than (X_C, Y_C). Let P_C be a Tutte path that begins at X_C, ends at Y_C, and uses edge α_C; we know that this exists since C^+ is either a triangle or a 3-connected graph.

Now obtain the Tutte path P of G by repeatedly substituting paths of 3-connected components. Specifically, initiate P as the virtual copy of edge (X, Y) that was added when we created the node for (X, Y). For as long as P contains a virtual edge (u, v), let C^+ be the child component at this virtual edge and observe that $\{X_C, Y_C\} = \{u, v\}$. *Substitute P_C in place of edge (u, v) of P*, i.e., set P to be $X \overset{P}{\rightsquigarrow} u/v \overset{P_C}{\rightsquigarrow} v/u \overset{P}{\rightsquigarrow} Y$. Note that, if C^+ is not

[1] Our definition of P-bridge considers only the *proper* P-bridges [19] that contain at least one vertex.

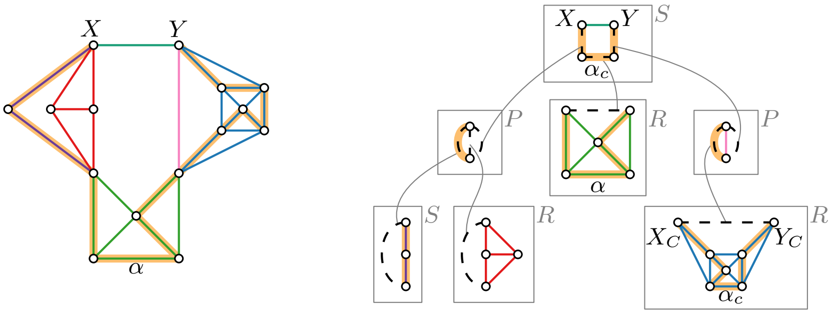


Figure 1 A 2-connected graph, its SPQR-tree (leaf nodes are omitted), and its Tutte path.

the singleton-edge (u, v), then P_C contains α_C, which is a virtual edge. This means that the process repeats until we have substituted the real edges from the leaves of the SPQR-trees. In particular (due to our choice of α_C), we will substitute the paths from all components between (X, Y) and α, which means that α is an edge of the final path P as required.

Observe that for some 3-connected components we do not substitute their paths; these become P-bridges with two attachment points. There may also be some P-bridges within each 3-connected components, but these have at most three attachment points since we used a Tutte path for each component. So the result is the desired Tutte path. Since we compute one Tutte path per 3-connected component, and this can be done in time proportional to the size of the component, the overall running time is linear.

2 Tutte paths in 3-connected planar graphs

For triangulated planar graphs, one can quite easily find a T_{SDR}-path by removing the interiors of all separating triangles, and finding for the resulting 4-connected planar graph a Hamiltonian path using the approach of Asano, Kikuchi, and Saito [1]. It is not hard to see that we can assign representatives to all separating triangles, possibly after expanding the path using the substitution trick described below. (We omit the details for space reasons.)

For 3-connected planar graphs that are not triangulated, we use the same approach, but must generalize many definitions from Asano, Kikuchi, and Saito [1] and add quite a few cases because now face-adjacent vertices are not necessarily adjacent. To keep the proof self-contained, we re-phrase everything from scratch.[2]

We need a few definitions. The *outer stellation* of a planar graph G is the graph obtained by adding a vertex in the outer face and connecting it to all exterior vertices. A planar graph G is called *internally 3-connected* if its outer stellation is 3-connected. Note that this implies that G is 2-connected, any cutting pair is *exterior* (i.e., has both vertices on the outer face) and has only two cut components. In the following, we endow G with k *corners*, which are k vertices $X_1, \ldots, X_k$ that appear in this order on the outer face. Usually, $k = 3$ or 4, but occasionally we allow larger k. A *side* of such a graph is the outer face path between two consecutive corners that does not contain any other corners. The *corner stellation* G^s is obtained by adding a vertex in the outer face and connecting it to the corners. We say

[2] Indeed, due to attempts to simplify the notations similar as done in [4], the reader familiar with [1] may barely see the correspondence between the proof and [1]. Roughly, their Condition (W) corresponds to $c3c(X, W, Y)$, their Case 1 is our Case 3a, and their Case 3 combines our Case 2 with our Case 4a (but resolves it in a symmetric fashion).

that G is *corner-3-connected with respect to corners* $X_1, \dots, X_k$ (abbreviated to "*G satisfies* $c3c(X_1, \dots, X_k)$") if G^s is 3-connected. Figure 2a illustrates this condition. It is easy to show that G satisfies $c3c(X_1, \dots, X_k)$ if and only if $k \geq 3$, G is internally 3-connected, and no cutting pair $\{v, w\}$ of G has both v and w on one side of G.

For ease of proof, we make the induction hypothesis stronger than just having a T_{SDR}-path, by restricting which vertices *must* be visited and which vertices *must not* be representatives. A T_{int}*-path* is a T_{SDR}-path P that visits all exterior vertices, and where representative $\sigma(C)$ is interior, for all P-bridges C. The goal of the remainder of this section is to prove the following result (which immediately implies Theorem 1 for 3-connected graphs[3]):

▶ **Lemma 2.** *Let G be a plane graph with distinct vertices X, Y on the outer face. Let $(U, W) \neq (X, Y)$ be an edge on the outer face. If G satisfies $c3c(X, U, W, Y)$, then it has a T_{int}-path that begins at X, ends at Y, and contains (U, W).*

We need a second result for the induction. Let a T_{end}*-path* be a T_{SDR}-path P that visits all exterior vertices, and where representative $\sigma(C)$ is interior or the last vertex of P, for all P-bridges C.

▶ **Lemma 3.** *Let G be a plane graph with distinct vertices X, Y on the outer face. Let $(U, W) \neq (X, Y)$ be an edge on the outer face. If G satisfies $c3c(X, U, W, Y)$ and*

$(\sqsupset)$ *(W, Y) and (Y, X) are edges,*

then G has a T_{end}-path P that begins at X, ends at Y, and uses (U, W) and (W, Y).[4] *Further, if Y is the representative of a P-bridge C, then C has W and Y as attachment points.*

See Figure 5c for a graph that satisfies $(\sqsupset)$.

We assume throughout that X, U, W, Y are enumerated in ccw order along the outer face, the other case can be resolved by reversing the planar embedding.

The following trick will help shorten the proof: If graph G satisfies $(\sqsupset)$, then Lemma 3 implies Lemma 2. Namely, if Lemma 3 holds, then we have a T_{end}-path P from X to Y through (U, W) and (W, Y). If this is not a T_{int}-path, then some P-bridge C has Y as representative, and by assumption also has W as attachment point. It must have a third attachment point u, otherwise $\{W, Y\}$ would be a cutting pair within one side of G, contradicting corner-3-connectivity. It has no more attachment points since P is a Tutte path, so $\{W, Y, u\}$ is a cutting triplet. We apply the *substitution trick* described below (and useful in other situations as well), which replaces (W, Y) with a path through C that does not use u. Thus, C no longer needs a representative and we obtain a T_{int}-path.

The substitution trick. This trick can be applied whenever we have an edge $e = (w, y)$ used by some T_{SDR}-path P, and a P-bridge C that resides inside a noose through some cutting triplet $\{u, w, y\}$ for some vertex u. Define $C^+ = G[C] \cup \{(u, w), (w, y)\} \setminus \{(u, y)\}$, where edges are added only if they did not exist in $G[C]$.[5]

[3] Theorem 1 allows $(U, W) = (X, Y)$, but then holds trivially since using edge (X, Y) as path satisfies all conditions. We require $(U, W) \neq (X, Y)$ since we want not just a Tutte path but a T_{int}-path, and the single-edge path (X, Y) would allow only exterior vertices as representatives.

[4] This lemma is a special case of the "Three Edge Lemma" [17], which states that for any three edges on the outer face there exists a Tutte cycle containing them all. However, it cannot simply be obtained from it since we require restrictions on the location of representatives.

[5] We apply the substitution trick even when $V(C^+) = V(G)$ and G has a triangular outer face; not adding edge (w, y) will ensure that C^+ has fewer interior vertices and induction can be applied.

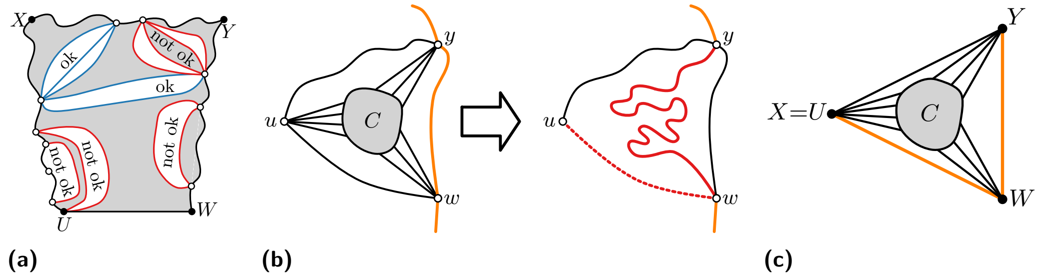


Figure 2 (a) Corner-3-connectivity $c3c(X, U, W, Y)$, (b) the substitution trick, and (c) Case 1.

C^+ satisfies $c3c(u, v, w)$, else there would have been a cutting pair in G that was interior or within one side. Hence, by induction, C^+ has a T_{int}-path P_{C^+} from u to y that uses edge (u, w). It does not use the edge (u, y) since P_{C^+} begins at u with edge (u, w). So $P_{C^+} \setminus (u, w)$ is a path in C from w to y that does not visit u. Substitute this in place of edge (w, y) of P; see Figure 2b. We claim that the resulting path P' is a T_{int}-path. We prove a more general statement in the full version [5], but roughly speaking, combining paths preserves T_{int}-paths because every P'-bridge can inherit its representative from P or P_{C^+}, and no vertex is used twice as representative since P_{C^+} does not use $\{u, v, w\}$ as representatives.

2.1 Proof of Lemma 2 and Lemma 3

We prove the two lemmas simultaneous by induction on the number of vertices of G, with an inner induction on the number of interior vertices. The base case is $n = 3$ where G is a triangle, but the same construction works whenever the outer face is a triangle (see below). For the induction step, we need the notation S_{xy}, which is the outer face path from x to y in ccw direction. In particular, the four sides are S_{XU}, S_{UW}, S_{WY}, and S_{YX}. We sometimes name sides as suggested by Figure 2a, so S_{XU}, S_{UW}, S_{WY}, and S_{YX} are the *left/bottom/right/top side*, respectively.

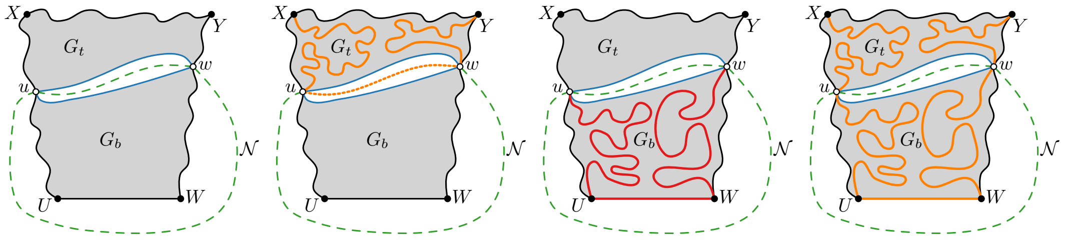
2.1.1 Case 1: The outer face is a triangle

Figure 2c illustrates this case. We know that $X \neq Y$ and $U \neq W$, so we must have $X = U$ or $W = Y$. For Lemma 3, we know that $(\sqsupset)$ holds, which forces $W \neq Y$, hence $X = U$. For Lemma 2, we may assume $X = U$ by symmetry, for otherwise we reverse the planar embedding, find a path from Y to X that uses (W, U) (with this, we have $X' = U'$) and then reverse the result.

So $X = U$. Define P to be $\langle X{=}U, W, Y\rangle$ and observe that this is a T_{end}-path, because the unique P-bridge C (if any) has attachment points $\{U, W, Y\}$, and we can assign Y to be its representative. So Lemma 3 holds. Since condition $(\sqsupset)$ is satisfied, this implies Lemma 2.

2.1.2 Case 2: G has a cutting pair $\{u, w\}$ with u and w on the left and right side

Figure 3 illustrates this case. Let $\mathcal{N}$ be a noose through u and w along a common interior face f^* and then going through the outer face. Let G_t and G_b be the subgraphs inside and outside $\mathcal{N}$, named such that G_b contains the bottom side. Let G_t^+/G_b^+ be the graphs obtained from G_t/G_b by adding (u, w) if not in the graph yet. We add (u, v) even if it did not exist in G (we will ensure that the final path does not use it).



(a) Case 2 **(b)** The path P_t **(c)** The path P_b **(d)** The path P

Figure 3 (a) Case 2, (b)–(d) proof of Lemma 2 for Case 2.

We first show Lemma 2. One can easily verify that G_t satisfies $c3c(X, u, w, Y)$ since its outer face is a simple cycle; see the full version [5]. Apply induction and find a T_{int}-path P_t of G_t^+ from X to Y that uses edge (u, w). Now apply a modified substitution trick to (u, w). Namely, by induction, there is a T_{int}-path P_b of G_b^+ from u to w that uses edge (U, W). Substitute P_b into P_t in place of (u, w) to get P. Path P uses (U, W) since P_b does. It does not use (u, w) since we removed this from P_t, and P_b starts at u, ends at w, and visits (U, W) in between. So after inheriting representatives from P_b and P_t we obtain a T_{int}-path P in G.

To prove Lemma 3, note that exactly one of G_t^+ and G_b^+ contains (W, Y); use a T_{end}-path for this subgraph and create P as above. Only one graph uses Y as representative, and one easily shows that P is a T_{end}-path.

2.1.3 Case 3: G has a cutting pair $\{y, w\}$ with y and w on the top and right side, respectively. Furthermore, there is an interior face f^* containing y and w that does not contain Y.

For later applications, we first want to point out that if G has a cutting pair $\{y, w\}$ on the top and right side for which (y, w) is an edge, then such a face f^* always exists, because there are two interior faces containing y and w, and not both can contain Y.

Figure 4 illustrates this case. We know that $w \neq Y \neq y$, else $\{y, w\}$ would be a cutting pair within one side. We may assume $y \neq X$; else we can use Case 2. Hence, the top side contains at least three vertices X, y, Y, so ($\sqsupset$) does not hold and we have to prove only Lemma 2.

We choose $\{y, w\}$ such that w is as close to W as possible (along the right side). The face f^* containing y, w may have multiple edges on the top side; let y be the one that is as close to Y as possible. Define G_b, G_b^+, G_t, G_t^+ to be as in Case 2. Since the outer face of G_b^+ is a simple cycle, it satisfies $c3c(X, U, W, w, y)$. But since we chose w to be as close to W as possible, it also satisfies $c3c(X, U, W, y)$. Namely, assume for contradiction that some cutting pair $\{y', w'\}$ exists along the side $S_{Ww} \cup (w, y)$ of G_b^+; see Figure 4a. Since there is no cutting pair within S_{Ww}, it must have the form $\{y, w'\}$ for some $w' \neq w$ on S_{Ww}. As f^* does not contain Y, neither can any face containing $\{y, w'\}$, so $\{y, w'\}$ could have been used for Case 3, contradicting our choice of w.

By induction, we can find a T_{int}-path P_b of G_b^+ from X to y that includes the edge (U, W). The plan is to combine P_b with a path through G_t, but we must distinguish some cases.

Case 3a: P_b does not contain (y, w) or $(y, w) \in G$. Observe that G_t^+ satisfies $c3c(y, w, Y)$. By induction, find a T_{int}-path P_t in G_t^+ from Y to w that uses edge (y, w). Append the reverse of $P_t \setminus (y, w)$ to P_b to obtain a T_{int}-path; see Figure 4b.

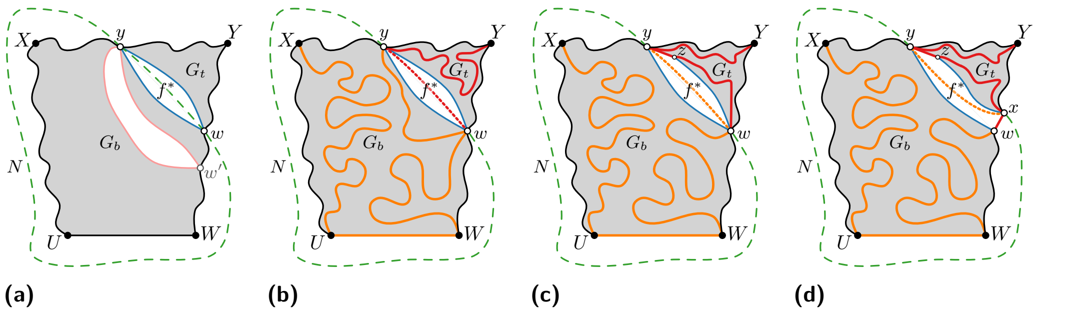


Figure 4 Case 3: (a) G_b^+ satisfies $c3c(X, U, W, y)$, (b) Case 3a, (c) Case 3b-1, (d) Case 3b-2.

Case 3b: P_b contains (y, w) and $(y, w) \not\in G$. In this case, we must remove (y, w) from the path and hence use a subpath in G_t to reach vertex y. This requires further subcases. Let π_f be the path along f^* from y to w that becomes part of the the outer face of G_t. Let (y, z) be the edge incident to y on π_f.

Case 3b-1: π_f contains no vertex on the outer face of G other than y and w. See Figure 4c. The outer face of G_t is then a simple cycle and G_t satisfies $c3c(w, y, Y)$. By induction, we can find a T_{int}-path P_t in G_t that begins at Y, ends at w, and uses (y, z).

Case 3b-2: π_f contains a vertex $x \neq y, w$ on the outer face of G. See Figure 4d. Since x is on f^*, it cannot be on the top side by choice of y. So $x \in S_{wY} \setminus Y$. In fact, x must be the neighbor of w on both S_{wY} and π_f, else there would be a cutting pair within the right side. Set G'_t to be the graph inside a noose through y and x that has Y inside. Since π_f has no vertices other than y, x, w on the outer face of G, graph G'_t has a simple cycle as outer face, so it satisfies $c3c(Y, y, z, x)$. By induction, we can find a T_{int}-path P'_t of G_t that begins at Y, ends at x, and uses (y, z). We append (w, x) to obtain P_t.

In both cases, we obtain a path P_t that begins at Y, ends at w, and visits all of G_t. Appending the reverse of this to $P_b \setminus (y, w)$ gives the T_{int}-path.

2.1.4 Case 3′: G has a cutting pair $\{y, w\}$ with y and w on the top and left side, respectively. Furthermore, there is an interior face f^* containing y and w that does not contain X.

This is handled symmetrically to Case 3.

2.1.5 Case 4: None of the above

In this case, we split G into one big graph G_0 and (possibly many) smaller graphs $G_1, \dots, G_s$, recurse in G_0, and then substitute T_{int}-paths of $G_1, \dots, G_s$ or use them as P-bridges.

We need two subcases, but first give some steps that are common to both. Let Y_X be the neighbor of Y on the top side. Define a *B-necklace* (for $B \in \{U, W\}$) to be a noose $\mathcal{N}_0 : \langle Y_X{=}x_0, f_1, x_1, \dots, x_{s-1}, f_s, x_s{=}B, f_o \rangle$, (where f_o is the outer face) for which x_i is face-adjacent to at least one vertex on $S_{WY} \setminus \{B\}$ for $1 \leq i \leq s-1$. See also Figure 5. We say that the necklace is *simple* if it contains no vertex twice, and *interior* if every x_i (for $0 < i < s$) is an interior vertex. One can argue that if none of the previous cases applies, then there always exists a simple interior B-necklace (see the full version [5]).

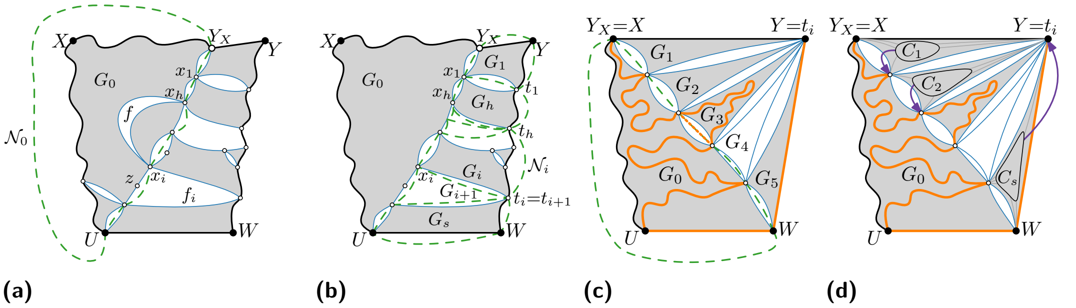


Figure 5 Case 4. (a) A simple interior U-necklace that is not leftmost due to face f (which yields a cutting pair $\{x_h, x_i\}$), and since it could include vertex z. (b) The graphs $G_1, \dots, G_s$. (c–d) Case 4a. The path P^+ after using the substitution trick and (d) assignment of the representatives.

Route $\mathcal{N}_0$ through the outer face such that the left side is in its interior, and let G_0 (the "left graph") be the graph inside $\mathcal{N}_0$. We say that $\mathcal{N}_0$ is *leftmost* if (among all simple interior B-necklaces) its left graph G_0 is smallest, and (among all simple interior B-necklaces whose left graph is G_0) it contains the most vertices of G_0. Fix a leftmost B-necklace $\langle x_0, \dots, x_s \rangle$.

▷ Claim 4. If (x_i, x_{i+1}) is not an edge for some $0 \le i < s$, then the face f_i of $\mathcal{N}_0$ contains no vertex of $S_{WY} \setminus \{B\}$.

Proof. If (x_i, x_{i+1}) is not an edge of G, then both paths from x_i to x_{i+1} on f_i contain at least one other vertex. One of them, say z, is inside $\mathcal{N}_0$. If f_i contains a vertex of $S_{WY} \setminus \{B\}$, then x_i and z are face-adjacent, z and x_{i+1} are face-adjacent, and z has a neighbor on $S_{WY} \setminus \{B\}$, so $x_0, \dots, x_i, z, x_{i+1}, \dots, x_s$ is a simple interior B-necklace with the same left graph but containing more vertices of G_0. Hence $\mathcal{N}_0$ is not leftmost, a contradiction. ◁

For $i = 0, \dots, s-1$, let t_i be the vertex on $S_{WY} \setminus \{B\}$ that is face-adjacent to x_i and closest to Y (along the right side) among all such vertices. Set $t_s = W$ if $x_s = U$, and $t_s = Y$ otherwise. For $0 < i \le s$, define $\mathcal{N}_i$ to be the noose through $\langle x_{i-1}, x_i, t_i, t_{i-1} \rangle$ such that the left side is outside $\mathcal{N}_i$. For $0 < i \le s$ let G_i be the graph inside $\mathcal{N}_i$ (i.e., a cut component of $\{x_{i-1}, x_i, t_{i-1}, t_i\}$); see Figure 5b.

Let G^+ be the graph obtained from G by adding *virtual edges* (x_i, x_{i-1}) and (t_i, t_{i-1}) (for $i = 1, \dots, s$) whenever these two vertices are distinct and the edge did not exist in G. Let G_0^+ be the graph obtained from G_0 by likewise adding virtual edges $(x_0, x_1), \dots, (x_{s-1}, x_s)$. This makes the outer face of G_0 a simple cycle, so G_0^+ satisfies $c3c(X, U, B{=}x_s, \dots, x_0{=}Y_X)$. We distinguish two cases.

Case 4a: ($\sqsupset$) holds, i.e., (X, Y) and (W, Y) are edges. We only have to prove Lemma 3 since this implies Lemma 2. Consider Figs. 5c and d. Let $\langle x_0{=}Y_X{=}X, x_1, \dots, x_s{=}W \rangle$ be a leftmost W-necklace. By $S_{WY} = (W, Y)$, we have $t_i = Y$ for all i. Since $x_0 = Y_X = X$, we have that G_0^+ satisfies $c3c(X{=}x_0, x_1, \dots, x_s{=}W, U)$. But observe that G_0^+ has no cutting pair $\{x_h, x_i\}$ with $0 \le h < i \le s$, for otherwise the face f containing x_h and x_i could be used as a shortcut and $\mathcal{N}_0$ was not leftmost (see Figure 5a). So G_0^+ actually satisfies $c3c(X, W, U)$. Use induction to obtain a T_{int}-path P_0 from X to W in G_0^+ that uses edge (U, W). Then $P^+ = P_0 \cup (W, Y)$ is a path in G^+ that contains (U, W), and (W, Y).

Fix some $i = 1, \dots, s$. If P^+ used edge (x_{i-1}, x_i) and it was virtual, then by Claim 4 f_i contains no vertex of S_{WY}, which means that the interior of G_i is non-empty. Apply the substitution trick to remove (x_{i-1}, x_i) from P^+, replacing it with a path through G_i.

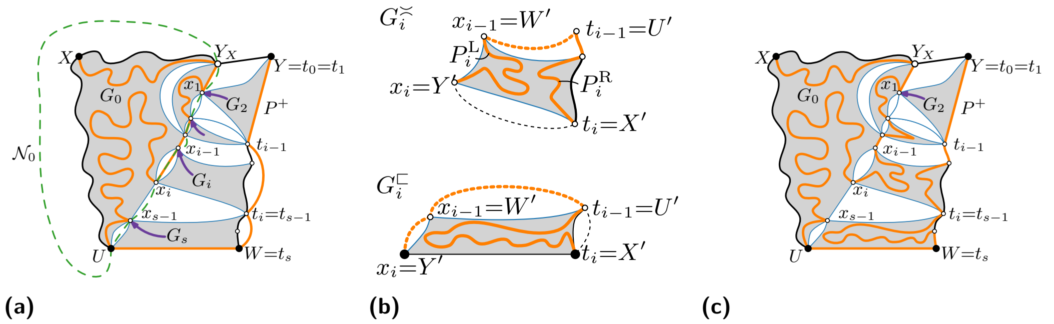


Figure 6 Case 4b. (a) Construction of P^+ with representatives; (b) $G_i^{\asymp}$ (Case 4b-2) and $G_i^{\sqsubset}$ (Case 4b-3) used to substitute virtual edges; (c) P^+ with representatives after all substitutions.

Otherwise, we keep G_i as a P^+-bridge. We let its representative be x_i if $1 \le i < s$, and Y if $i = s$. Observe that this representative is interior or Y, and was not used by P_0 since P_0 was a T_{int}-path. So we obtain a T_{end}-path with the desired properties.

Case 4b: ($\sqsupset$) does not hold. We must prove only Lemma 2 and may therefore by symmetry assume that $X \neq U$. We claim that this implies that $\deg(Y_X) \ge 3$. For if $\deg(Y_X) = 2$, then its neighbors form a cutting pair, which by corner-3-connectivity means that Y_X is a corner, hence $Y_X = X$. Since $X \neq U$, the two neighbors of Y_X are then Y and a vertex on the left side, and we could have applied Case 2. So $\deg(Y_X) \ge 3$. Let (Y_X, x_1) be the edge at Y_X that comes after (Y_X, Y) in clockwise order (see Figure 6a). Note that x_1 is face-adjacent to Y. It must be an interior vertex, for otherwise by $\deg(Y_X) \ge 3$ edge (Y_X, x_1) is a cutting pair that we could have used for Case 3 or 3′.

Let $\mathcal{N}_0 = \langle x_0{=}Y_X, x_1, \ldots, x_s{=}U\rangle$ be a simple interior U-necklace; see Figure 6a. We use a U-necklace that is leftmost among all U-necklaces that contain x_1. Note that Claim 4 holds for $\mathcal{N}_0$ even with this restriction, since (x_0, x_1) is an edge. We know that G_0^+ satisfies $c3c(X, Y_X, x_1, \ldots, x_s{=}U)$. But observe that G_0^+ has no cutting pair $\{x_h, x_i\}$ for $1 \le h < i \le s$, for otherwise (as in Figure 5a) $\mathcal{N}_0$ would not be the leftmost necklace that uses x_1. So G_0^+ actually satisfies $c3c(X, Y_X, x_1, U)$.

Use induction to obtain a T_{int}-path P_0 in G_0 from U to X through edge (x_1, x_0). Append the path $\langle U, W, t_s, \ldots, t_0{=}Y\rangle$ to the reverse of P_0 to obtain path P^+. This path begins at X, ends at Y, and contains (U, W). Any P^+-bridge is either a P_0-bridge (and receives a representative there) or is G_i for some $1 \le i \le s$. For $i > 1$, assign x_{i-1} as representative to G_i. Graph G_1 has an empty interior by choice of x_1 and needs no representative.

There are two reasons why we cannot always use P^+ for the result. First, it may use virtual edges and hence not be a path in G. Second, some P^+-bridge G_i may have four attachment points. Both are resolved by expanding P^+ via paths through the G_i's. Fix one i with $1 \le i \le s$ and consider the following cases:

Case 4b-1: (x_{i-1}, x_i) is virtual and used by P^+, and $t_{i-1} = t_i$. By Claim 4, the interior of graph G_i is non-empty and inside the separating triplet $\{x_{i-1}, x_i, t_i\}$. Replace (x_{i-1}, x_i) by a path through G_i with the substitution trick; see graph G_3 in Figure 6.

Case 4b-2: (x_{i-1}, x_i) is virtual and used by P^+, and $t_{i-1} \neq t_i$. See Figure 6b(top). We want to replace both (x_{i-1}, x_i) and (t_{i-1}, t_i) (which is always used by P^+) with a path through G_i. Let $G_i^{\asymp}$ be the graph G_i with (t_i, x_i) and (t_{i-1}, x_{i-1}) added. The outer face of $G_i^{\asymp}$ is a simple cycle since f_i contains no vertex of the right side by Claim 4, so $G_i^{\asymp}$ satisfies $c3c(t_i, t_{i-1}, x_{i-1}, x_i)$. By induction, find a T_{int}-path P_i in $G_i^{\asymp}$ from t_i to x_i

that uses the edge (t_{i-1}, x_{i-1}). So removing (t_{i-1}, x_{i-1}) from P_i splits it into two paths: path P_i^{R} connects t_i to t_{i-1}, and path P_i^{L} connects x_{i-1} to x_i. (No other split is possible by planarity.) Neither path uses the added edge (t_i, x_i) since it connects the ends of P_i. Use P_i^{R} to replace (t_{i-1}, t_i) and P_i^{L} to replace (x_{i-1}, x_i) in P^+.

Case 4b-3: Subgraph G_i has a non-empty interior and $t_i \neq t_{i-1}$. See Figure 6b(bottom). In this case, G_i is a P^+-bridge with four attachment points, a violation of Tutte path properties. If Case 4b-2 applied to G_i, then G_i is no longer a bridge of the resulting path and we are done. Otherwise, we do a substitution that uses a different supergraph of G_i. Let $G_i^{\sqsubset}$ be G_i with edges from path $\langle t_{i-1}, x_{i-1}, x_i, t_i \rangle$ added if not already in G_i. This graph satisfies $c3c(t_i, t_{i-1}, x_{i-1}, x_i)$ and satisfies condition ($\sqsupset$) if we set $X' = t_i$, $U' = t_{i-1}, W' = x_{i-1}$, and $Y' = x_i$. So we can find a T_{end}-path P_i' of $G_i^{\sqsubset}$ from t_i to x_i that uses (t_{i-1}, x_{i-1}) and (x_{i-1}, x_i). Thus, P_i' ends with $\langle t_{i-1}, x_{i-1}, x_i \rangle$ and $P_i' \setminus \{(t_{i-1}, x_{i-1}), (x_{i-1}, x_i)\}$ is a path from t_{i-1} to t_i in G_i that does not visit x_{i-1} or x_i. Substitute this path in place of edge (t_{i-1}, t_i) in P^+. Note that one P_i'-bridge C may use x_i as its representative, but if so, then it also has x_{i-1} as attachment point. We set x_{i-1} (which was G_i's representative and is no longer needed as such) to be the representative of C.

Case 4b-4: $t_{i-1} \neq t_i$ and (t_{i-1}, t_i) is virtual. Since P^+ always uses edge (t_{i-1}, t_i), we must replace this edge with a path through G_i. This is done automatically because Case 4b-3 applies. Namely, if (t_{i-1}, t_i) is virtual, then there is at least one vertex between t_{i-1} and t_i on the right side. This vertex is exterior in G and hence neither x_i nor x_{i-1}. So it is strictly inside $\mathcal{N}_i$, hence G_i has a non-empty interior and (by $t_{i-1} \neq t_i$) Case 4b-3 applies.

After doing these substitutions, there are no virtual edges in the path, no bridges have four attachment points, every bridge has an interior vertex as representative, and no vertex was used twice as representative; see Figure 6c. This ends the proof of Lemma 2 and 3.

2.2 Linear time complexity

It should be clear that our proof is algorithmic. The main bottlenecks for its running time are to determine which case to apply (i.e., whether there is a cutting pair) and to find the B-necklace. Both can be done in linear time, by computing all cutting pairs [6, 9] and by finding a leftmost path in the subgraph induced by vertices that are face-adjacent to $S_{WY} \setminus B$. This would yield quadratic running time overall. For triangulated planar graphs, this is easily reduced to linear: cutting pairs correspond to interior edges where both ends are exterior, and the necklace can be found, as in [1], with a left-first search that only advances neighbors of $S_{WY} \setminus B$. But for graphs that are not triangulated we need a few extra data structures. We sketch only some ideas for this here; details are in the full version [5].

Globally, we keep track of the corners X, U, W, and Y. For each *interior vertex* w and every *side* S_{ab}, we keep a list $\mathcal{V}(w, S_{ab})$ of faces that contain w as well as a vertex on S_{ab}. In these lists, we can look up quickly whether an interior vertex is face-adjacent to a side. Also, each face knows for each side which vertices it has on it. Finally, for each *pair of sides* S_{ab} and S_{cd}, we store a list $\mathcal{P}(S_{ab}; S_{cd})$ of faces that are incident to a vertex on S_{ab} and a (different) vertex on S_{cd}, i.e., faces that connect cutting pairs.

This allows to test for Case 2 and Case 3 easily ("is $\mathcal{P}(S_{XU}, S_{WY})$ resp. $\mathcal{P}(S_{WY}, S_{YX})$ non-empty?"), and Case 1 and Case 4 are easily determined from the planar embedding. We keep $\mathcal{P}(S_{WY}, S_{YX})$ in an order such that its first entry is the appropriate cutting pair in Case 3. To find a necklace, we *scan* the faces incident to $x_1, \ldots, x_s$. More precisely, we consider (for vertex x_i, presuming we know face f_i already) each face f in ccw order after f_i, and along face f each vertex w in ccw order after x_i, until we find vertex B (then we are done) or a vertex that is face-adjacent to a vertex in $S_{WY} \setminus B$ (then this is x_{i+1} and $f_{i+1} = f$ and we repeat). The running time for this is proportional to the degrees of vertices and faces that were scanned. We also need to update the data structures when recursing into a subgraph; here, we scan along all vertices (and their incident faces) that were in some necklace along which we cut the graph, or that became newly exterior.

A few crucial insights are needed to bound the running time. First, by corner-3-connectivity every face has at most two vertices on each side. In particular, the above data structures have linear size. Second, we need to scan vertices and faces only if they become incident to a side that they were not previously incident to. Finally, once a vertex or face is incident to a side, it remains incident to it forever (though the side may change role, e.g. from “left” to “top”). This means that every vertex and face is scanned only a constant number of times, because there are only four sides to have incidences with. In fact, we only scan vertices and faces that are incident to the outer face in some subgraph, which means that they will be incident to the path P that we compute, and we have the following:

▶ **Theorem 5.** *The Tutte path P for Theorem 1, Lemma 2 or Lemma 3 can be found in linear time. More specifically, the running time is $O(\sum_{f \in F(P)} \deg(f))$.*

3 Applications

A number of interesting properties of planar 3-connected graphs can be derived easily from the existence of T_{SDR}-paths. In particular, every planar 3-connected graph has a spanning tree of maximum degree 3 [2] (a concept known in the literature as a *3-tree*, but we prefer to use the term *binary spanning tree* to avoid confusion with maximal graphs of treewidth 3). Secondly, every planar 3-connected graph has a *2-walk*, i.e., a walk that visits every vertex at least once and at most twice [8]. In the full version [5], we show that, using Lemma 2, these can be found in linear time; this was known for binary spanning trees [16, 3], but for 2-walks the previous best running time was $O(n^3)$ [15].

▶ **Theorem 6.** *Let G be a 3-connected plane graph with exterior vertex X. Then G has a binary spanning tree T that can be found in linear time. Moreover, when rooting T at X, a vertex v has two children only if it is an interior vertex and part of a cutting triplet $\{v, w, x\}$ of G; one of the subtrees of v contains exactly the vertices interior to $\{v, w, x\}$.*

▶ **Theorem 7.** *Let G be a 3-connected plane graph with exterior vertex X. Then G has a 2-walk P that can be found in linear time. Moreover, P visits X exactly once, and it visits a vertex v twice only if v is part of a separating triplet.*

4 Outlook

In this paper, we improved on a very recent result that shows that Tutte paths in planar graphs can be found in quadratic time. We gave a different existence proof which leads to a linear-time algorithm. For 3-connected planar graphs, we obtain not only a Tutte path, but furthermore endow it with a system of distinct representatives, none of which is on the outer face. With this, we can also find 2-walks and binary spanning trees in 3-connected planar graphs in linear time.

The main remaining questions concern how to find Tutte path in other situations or with further restriction. For example, Thomassen [18] and later Sanders [12] improved Tutte's result and showed that we need not restrict the ends of the Tutte path to lie on the outer face. These paths can be found in quadratic time [15]. But our proof does not seem to carry over to the result by Sanders, because the ends of the path crucially must coincide with corners of the graph. Can we find such a path in linear time?

Furthermore, the existence of Tutte paths has been studied for other types of surfaces (see, e.g., Kawarabayashi and Ozeki [11] and the references therein). Can these Tutte paths be found in polynomial time, and preferably, linear time?

References

1 Takao Asano, Shunji Kikuchi, and Nobuji Saito. A Linear Algorithm for Finding Hamiltonian Cycles in 4-Connected Maximal Planar Graphs. *Discrete Applied Mathematics*, 7:1–15, 1985. doi:10.1016/0166-218X(84)90109-4.

2 David W. Barnette. Trees in Polyhedral Graphs. *Canadian Journal of Mathematics*, 18:731–736, 1966. doi:10.4153/CJM-1966-073-4.

3 Therese Biedl. Trees and Co-trees with Bounded Degrees in Planar 3-Connected Graphs. In R. Ravi and Inge Li Gørtz, editors, *Scandinavian Symposium and Workshops on Algorithms Theory (SWAT'14)*, volume 8503 of *Lecture Notes in Computer Science*, pages 62–73. Springer-Verlag, 2014. doi:10.1007/978-3-319-08404-6_6.

4 Therese Biedl and Martin Derka. 1-String B_2-VPG-Representations of Planar Graphs. *Journal on Computational Geometry*, 7(2):191–215, 2016. doi:10.20382/jocg.v7i2a8.

5 Therese Biedl and Philipp Kindermann. Finding Tutte Paths in Linear Time. *Arxiv report*, abs/1812.04543, 2018. arXiv:1812.04543.

6 Giuseppe Di Battista and Roberto Tamassia. Incremental Planarity Testing. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS'89)*, pages 436–441. IEEE Computer Society, 1989. doi:10.1109/SFCS.1989.63515.

7 Reinhard Diestel. *Graph Theory, 5th Edition*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2016. URL: http://diestel-graph-theory.com/.

8 Zhicheng Gao and R. Bruce Richter. 2-Walks in Circuit Graphs. *Journal of Combinatorial Theory, Series B*, 62(2):259–267, 1994. doi:10.1006/jctb.1994.1068.

9 Carsten Gutwenger and Petra Mutzel. A Linear Time Implementation of SPQR-Trees. In Joe Marks, editor, *Proceedings of the 8th International Symposium on Graph Drawing (GD'00)*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2000. doi:10.1007/3-540-44541-2_8.

10 John E. Hopcroft and Robert E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3):135–158, 1973. doi:10.1137/0202012.

11 Ken-ichi Kawarabayashi and Kenta Ozeki. 4-Connected Projective-Planar Graphs Are Hamiltonian-Connected. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'13)*, pages 378–395. Society for Industrial and Applied Mathematics, 2013. doi:10.1016/j.jctb.2012.11.004.

12 Daniel P. Sanders. On Paths in Planar Graphs. *Journal of Graph Theory*, 24(4):341–345, 1997. doi:10.1002/(SICI)1097-0118(199704)24:4<341::AID-JGT6>3.0.CO;2-O.

13 Andreas Schmid and Jens M. Schmidt. Computing 2-Walks in Polynomial Time. In Ernst W. Mayr and Nicolas Ollinger, editors, *Proceedings of the 32nd International Symposium on Theoretical Aspects of Computer Science (STACS'15)*, volume 30 of *LIPIcs*, pages 676–688. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.STACS.2015.676.

14 Andreas Schmid and Jens M. Schmidt. Computing 2-Walks in Polynomial Time. *ACM Transactions on Algorithms*, 14(2):22:1–22:18, 2018. doi:10.1145/3183368.

15 Andreas Schmid and Jens M. Schmidt. Computing Tutte Paths. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *Proceedings of the 45th International Colloquium on Algorithms, Languages and Programming (ICALP'18)*, volume 107 of *LIPIcs*, pages 98:1–98:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.98.

16 Willy-Bernhard Strothmann. *Bounded-Degree Spanning Trees.* Ph.d. thesis, Universität Paderborn, Heinz Nixdorf Institut, Theoretische Informatik, Paderborn, 1997. ISBN 3-931466-34-5. URL: https://www.hni.uni-paderborn.de/pub/468.

17 Robin Thomas and Xingxing Yu. 4-Connected Projective-Planar Graphs Are Hamiltonian. *Journal of Combinatorial Theory, Series B*, 62:114–132, 1994. doi:10.1006/jctb.1994.1058.

18 Carsten Thomassen. A Theorem on Paths in Planar Graphs. *Journal of Graph Theory*, 7(2):169–176, 1983. doi:10.1002/jgt.3190070205.

19 William T. Tutte. Bridges and Hamiltonian Circuits in Planar Graphs. *Aequationes Mathematicae*, 15(1):1–33, 1977. doi:10.1007/BF01837870.