Computing 2-Walks in Polynomial Time

Andreas Schmid[1] and Jens M. Schmidt[2]

1 Max Planck Institute for Informatics, Saarbrücken, Germany
2 TU Ilmenau, Ilmenau, Germany

Abstract

A *2-walk* of a graph is a walk visiting every vertex at least once and at most twice. By generalizing decompositions of Tutte and Thomassen, Gao, Richter and Yu proved that every 3-connected planar graph contains a closed 2-walk such that all vertices visited twice are contained in 3-separators. This seminal result generalizes Tutte's theorem that every 4-connected planar graph is Hamiltonian as well as Barnette's theorem that every 3-connected planar graph has a spanning tree with maximum degree at most 3. The algorithmic challenge of finding such a closed 2-walk is to overcome big overlapping subgraphs in the decomposition, which are also inherent in Tutte's and Thomassen's decompositions.

We solve this problem by extending the decomposition of Gao, Richter and Yu in such a way that all pieces, in which the graph is decomposed into, are edge-disjoint. This implies the first polynomial-time algorithm that computes the closed 2-walk mentioned above.

1998 ACM Subject Classification G.2.2 Graph algorithms

Keywords and phrases Algorithms and Data Structures, 2-Walks, 3-Connected Planar Graphs, Tutte Paths, 3-Trees

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Among the most fundamental problems in graph theory is the question whether a graph is *Hamiltonian*, i.e., contains a cycle of length $n := |V|$. Whitney [17] proved that every 4-connected maximal planar graph is Hamiltonian. Tutte extended this result by showing that actually every 4-connected planar graph is Hamiltonian [16]. Thomassen [15] simplified Tutte's result and proved the generalization that every 4-connected planar graph contains a path of length $n - 1$ between any given two vertices. There are numerous examples proving that 3-connected planar graphs are not necessarily Hamiltonian; in fact, even deciding whether a 3-connected 3-regular planar graph is Hamiltonian is NP-hard [10]. However, one may ask how "close" 3-connected planar graphs are to Hamiltonicity. To this end, let a *k-walk* be a walk that visits every vertex at least once and at most k times (edges may be visited multiple times). A walk is *closed* if it has the same start- and endvertex. Thus, a closed 1-walk is a Hamiltonian cycle.

Jackson and Wormald conjectured in [13] that every 3-connected planar graph contains a closed 2-walk. In a seminal result [7], Gao and Richter proved this conjecture 1994 in the affirmative. One year later, Gao, Richter and Yu [8] published a refined decomposition that gives the existence of a very special closed 2-walk, namely one in which every vertex visited twice is contained in a 3-separator. This decomposition is involved and its presentation in [8] very densely written; in addition, it contains a flaw, which was fixed in the erratum [9]. However, as an immediate consequence, this special closed 2-walk forms a Hamiltonian cycle if the graph is 4-connected and, hence, generalizes Tutte's theorem to 3-connected planar graphs. It also generalizes Thomassen's result for 4-connected planar graphs. One of the remarkable aspects of the result from Gao, Richter and Yu is that it generalizes yet another research direction. Barnette [2] proved that every 3-connected planar graph contains a 3-*tree*, i.e., a spanning tree with maximum degree at most 3. A 3-tree can be computed in linear-time due to a Ph.D.-thesis by Strothmann [14]. Recently, Biedl showed that 3-trees (and in fact, more special variants of them) can also be computed by canonical orderings [3]. Interestingly, a 3-tree can be directly obtained from a closed 2-walk in linear time, as shown in [13, Lemma 2.2(ii)].

So far, 2-walks form the most general existence result in the above line of research. We are interested in the computational complexity of finding the above special closed 2-walk [8]. Although the existence proof is over 20 years old, it is not even known whether such a 2-walk can be computed in polynomial time.

Much more is known for its preceding variants: Inspired by Tutte's classic result, Gouyou-Beauchamps [11] showed that a Hamiltonian cycle in a 4-connected planar graph can be computed in polynomial time. The crux of this approach lies in the fact that the subgraphs arising from Tutte's decomposition may overlap in an unbounded number of vertices and edges. This made it very difficult to bound the running time spent in the recursion tree reflecting the decomposition.

Asano, Kikuchi and Saito showed that a Hamiltonian cycle can be computed in linear-time when the 4-connected planar input graph is additionally maximal planar [1]. Thomassen claimed that one could also derive a polynomial-time algorithm from his more general existence proof in [15]. In [4] it was shown that this statement was too optimistic, as the subgraphs arising from his decomposition may again overlap in big parts. Chiba and Nishizeki [5] showed that this problem can be avoided for 4-connected planar input graphs and gave a linear-time algorithm to compute a Hamiltonian cycle for these graphs. However, the general problem of overlapping subgraphs in 3-connected graphs has not been resolved. Even the decomposition in [8] bears the same obstruction that made previous algorithmic results difficult, namely big overlapping subgraphs.

As main result, we propose how to overcome this problem by extending the decomposition of Gao, Richter and Yu such that all arising subgraphs will be edge-disjoint. This leads to the first polynomial-time algorithm that computes the special closed 2-walk of [8], generalizing the previous results. The result is stated for the class of circuit graphs, which contain all 3-connected graphs. We aim for a detailed and self-contained description of this decomposition.

▶ **Theorem 1.** *Let G be a circuit graph with external face boundary C and let x, y be vertices of C. A closed 2-walk of G can be computed in polynomial time such that x and y are visited exactly once and every vertex visited twice is contained in either a 2-separator or an internal 3-separator of G.*

2 Preliminaries

We assume familiarity with standard graph theoretic notations as in [6]. A *k-separator* of a graph $G = (V, E)$ is a set of k vertices whose deletion leaves a disconnected graph. Let $n := |V|$ and $m := |E|$. A graph G is *k-connected* if $n > k$ and G contains no $(k-1)$-separator. A set of paths intersecting pairwise at most at their endvertices are called *independent*. For a path P and to vertices $x, y \in P$, let the subpath from x to y in P be xPy.

A central concept for the decomposition is the notion of H-bridges: For a subgraph H of G, an *H-bridge* of G is a component K of $G - V(H)$ together with all edges joining vertices of K with vertices of H and the endvertices of these edges. Although standard notation allows an H-bridge to be a single edge, we excluded this case from the definition, as such bridges will not play any role for 2-walks. A vertex in an H-bridge L is an *attachment* of L if it is also in H and an *internal* vertex of L otherwise.

A *plane* graph is a planar embedding of a graph. For two vertices x, y of a cycle C in a plane graph, let xCy be the clockwise path from x to y in C. For a cycle C in a plane graph G, let the subgraph of G *inside* C be the subgraph induced by $E(C)$ and all edges intersecting the open disc-homeomorph of the plane interior of C. A subgraph inside a cycle of a 3-connected plane graph G is not necessarily 3-connected; however, its only 2-separators must have both vertices on the external face. Since we will often use induction on such subgraphs when describing the decomposition, we will deal with circuit graphs instead of 3-connected plane graphs. A *circuit graph* (G, C) is a 2-connected plane graph G with external face boundary C such that the following property is satisfied:

▶ **Definition 2** (3-Paths Property)**.** For every vertex v in $G \setminus C$, G contains three independent paths from v to distinct vertices in C.

Clearly, circuit graphs generalize 3-connected plane graphs. In the following, we will give several lemmas about circuit graphs that will be used throughout the paper.

▶ **Lemma 3.** *Let $\{u, v\}$ be a 2-separator of a circuit graph (G, C). Every component of $G \setminus \{u, v\}$ contains a vertex of C.*

Proof. Assume to the contrary that $G \setminus C$ has a component K with $V(K) \cap V(C) = \emptyset$. Since K does not contain a vertex of C, each path from a vertex $w \in V(K)$ to C contains u or v. Thus, there are no three independent paths from w to C, contradicting the 3-Paths Property. ◀

▶ **Lemma 4.** *Let $\{u, v\}$ be a 2-separator of a circuit graph (G, C). Then u and v are contained in C and $G \setminus \{u, v\}$ has exactly two components.*

Proof. First assume that u or v, say u, is not contained in C. As $\{u, v\}$ is a 2-separator of G, $G \setminus \{u, v\}$ has at least two components. Since $u \notin V(C)$, one component of $G \setminus \{u, v\}$ must contain all remaining vertices of C. This contradicts Lemma 3. For the second claim, observe that $G \setminus \{u, v\}$ has at most two components that contain vertices of C, as $C \setminus \{u, v\}$ is the union of at most two paths. Thus, a third component would contradict Lemma 3. ◀

Next, we state several lemmas how a circuit graph can be decomposed into smaller circuit graphs.

▶ **Lemma 5** ([7])**.** *Let $\{u, v\}$ be a 2-separator of a circuit graph (G, C). For each $\{u, v\}$-bridge H of G (recall that $H \neq uv$), $H \cup uv$ is a circuit graph.*

▶ **Lemma 6** ([7])**.** *Let C' be any cycle in a circuit graph (G, C) and let H be the subgraph inside C'. Then (H, C') is a circuit graph.*

A *block* is a maximal connected subgraph that does not contain a 1-separator. Every block is either 2-connected or has at most two vertices. It is well-known that the blocks of a graph partition its edge-set. A graph G is called a *chain of blocks* if it consists of blocks $B_1, B_2, ..., B_k$ such that $V(B_i) \cap V(B_{i+1})$, $1 \leq i < k$, are pairwise distinct 1-separators of G and G contains no other 1-separator. Thus, a chain of blocks is a graph, whose block-cut tree [12] is a path. A key idea in the decomposition is that deleting a vertex of the external face boundary of a circuit graph results in a plane chain of blocks. Every such block will again be a circuit graph due to Lemma 6.

▶ **Lemma 7** ([7])**.** *Let (G, C) be a circuit graph and let $v \in V(C)$. Then $G \setminus v$ is a plane chain of blocks $B_1, B_2, ..., B_k$ such that one of the neighbours of v in C is in $B_1 \setminus B_2$ and the other is in $B_k \setminus B_{k-1}$.*

The lemma above can be further strengthened if the external face boundary of the circuit graph is a triangle.

▶ **Lemma 8.** *Let (G, C) be a circuit graph such that $C = \{v, w, z\}$ is a triangle and $G \neq C$. Then $G \setminus v$ is a circuit graph and $G \setminus \{v, w\}$ is a plane chain of blocks $B_1, ..., B_k$ with $z \in B_1$ and one neighbour of w in B_k.*

Proof. Due to Lemma 7, $G \setminus v$ is a plane chain of blocks with $z \in B_1$ and $w \in B_k$. According to the 3-Paths Property, G contains independent paths from every vertex in $G \setminus V(C)$ to v, w and z. Thus, $G' := G \setminus v$ is a block and therefore forms a circuit graph (G', C'). Applying Lemma 7 to (G', C') gives that $G' \setminus w$ is a plane chain of blocks with $z \in B_1$ and a neighbour of w in B_k. ◀

3 From Tutte Paths to 2-Walks

We recapitulate the fundamental steps of Gao, Richter and Yu [8] for proving the existence of a closed 2-walk. A crucial role plays the notion of a Tutte path. A *Tutte path* (*Tutte cycle*) of a circuit graph (G, C) is a path (*cycle*) T for which every T-bridge has exactly 2 attachments if it contains an edge of C and otherwise exactly 3 attachments. A Tutte path *from x to y through u* has start vertex x, end vertex y and contains u; we will sometimes say that u is the *intermediate vertex* of T. Tutte paths can be used to construct a closed 2-walk if the attachments of its T-bridges are sufficiently disjoint. In [8], the existence of a Tutte path T with T-bridges $B_1, B_2, \ldots, B_k$ was proven for which a set $S = \{s_1, s_2, \ldots, s_k\}$ of vertices exists such that s_i is an attachment of B_i for each i. The set S is called *system of distinct representatives (SDR)* of the T-bridges. The next results give the existence of such Tutte paths and cycles; Theorem 9 is slightly weaker than the one in [8] (in which $y \in V(G)$), but sufficient for our needs.

▶ **Theorem 9** ([8])**.** *Let (G, C) be a circuit graph, let $x, u, y \in V(C)$ with $x \neq y$ and let $a \in \{x, u\}$. Then there is a Tutte path P of G from x to y through u and an SDR S of the P-bridges such that $a \notin S$.*

According to Lemma 7, $G \setminus x$ is a plane chain of blocks. By computing a Tutte path for every such block and extending the union of these Tutte paths to x (using the two incident edges in C), we immediately obtain a Tutte cycle of G. Note that the time for computing this Tutte cycle is dominated by the computation of the Tutte paths (see Lines 2–4 of Algorithm 1).

▶ **Corollary 10** ([8])**.** *Let (G, C) be a circuit graph and let $x, y \in V(C)$. Then there is a Tutte cycle T of G and an SDR S of the T-bridges in G with $x, y \in V(T)$ and $x, y \notin S$.*

Proving the existence of an SDR as in Corollary 10 is the crucial new insight of Gao, Richter and Yu's paper [8]. It implies the existence of a closed 2-walk. The idea is to use the vertices of the SDR S as branch vertices, at which the walk deviates from T into 2-walks of the T-bridges, which exist by induction. The constructed closed 2-walk will therefore have special properties for the vertices visited twice. Let an *internal 3-separator* S of a circuit graph (G, C) be a 3-separator such that $G - S$ contains a component disjoint from C.

▶ **Theorem 11** ([8]). *Let (G,C) be a circuit graph and let $x, y \in V(C)$. Then there is a closed 2-walk W in G visiting x and y exactly once such that every vertex visited twice is contained in either a 2-separator or an internal 3-separator of G.*

We are interested in the computational complexity of finding the 2-walk of Theorem 11 when an efficient subroutine for computing Tutte paths is known.

Algorithm 1

```
1: procedure 2-WALK($(G,C)$, $x, y \in V(C)$)
2:     for every block $B$ of the plane chain of blocks $G \setminus x$ do
3:         Compute a Tutte path $P_B$ of $B$                                ▷ crucial
4:     Join the union of all computed Tutte paths to $x$ and obtain a Tutte cycle $T$ of $G$
5:     for every $T$-bridge $L$ do
6:         Recurse on $L$ to compute a 2-walk $W_L$         ▷ polynomially dependent on Line 3
7:     Output the union of $T$ and all $W_L$
```

Algorithm 1 gives a high-level description of the steps taken for the proof of Theorem 11. For all steps except the computation of Tutte paths in Line 3 and the computation of suitable circuit subgraphs for the recursion on L in Line 6, the corresponding existence proofs give immediately linear-time algorithms. It can be readily shown that the computation of Line 6 exceeds the time spent for computing a Tutte path by at most a factor m (we refer to the appendix for details); hence, we can reduce to computing Tutte paths.

However, it is not even clear whether a Tutte path itself can be computed in polynomial time, as its existence proof uses a decomposition into circuit subgraphs that may overlap in large parts. We will show that a Tutte path can be computed in polynomial time; this implies our main Theorem 1.

4 Finding Tutte Paths

We will prove Theorem 9 by extending the decomposition of Gao, Richter and Yu. The extended decomposition will only branch into edge-disjoint circuit graphs and thus turn out to be algorithmically accessible. In the following sections, we will first review some steps given in [8] needed to set up the decomposition, then explain how we can avoid overlapping subgraphs, and finally give the details of the extended decomposition.

4.1 Setting up the Decomposition

We review the initial steps taken for the original decomposition in [8]. Let (G,C) be a circuit graph, let $x, u, y \in V(C)$ with $x \neq y$ and let $a \in \{x, u\}$. We want to find a Tutte path from x to y through u. The vertex a acts as a place-holder that allows us to prevent x or u to be in the SDR S; this will be useful for the induction.

We first eliminate some symmetric cases. If $u = x$, we can choose any other vertex $v \in V(C) \setminus x$ and assign $u = v$. The same holds if $u = y$ and $a \neq u$. If $a = u = y$, we interchange the roles of x and y and proceed as above. Thus we can assume that $u \notin \{x, y\}$. We will need y to be in uCx in a later step. Therefore if $y \in xCu$, we flip the current embedding of G such that in the new embedding $y \in uCx$.

The proof of Theorem 9 proceeds by induction on the number of edges in G. If $|E(G)| = 3$, G is a triangle. In that case, the Tutte path we are looking for is xuy, the corresponding SDR S is empty and there are clearly no overlapping subgraphs. For the induction step, let

u_1 the neighbour of u that is not in xCu. In the special case that $u_1 = y$, we define $K := u_1$. Otherwise, we define K as the *minimal connected union* of blocks of $G \setminus xCu$ that contains u_1 and y, where minimality is with respect to the number of blocks (see Figure 1). As argued before, the blocks of $G \setminus xCu$ form a tree; by minimality, K will be a plane chain of blocks. Let $B_1, \dots, B_l$ be the blocks of K such that $u_1 \in B_1$ and $y \in B_l$ and let C_i be the external face boundary of B_i. We number the 1-separators in K from v_1 to v_{l-1}, i.e., the blocks B_i and B_{i+1} intersect exactly in v_i. In addition, we set $v_0 := u_1$ and define v_l as the vertex in B_l nearest to x in u_1Cx.

For each $(K \cup xCu)$-bridge L, L intersects K in at most one vertex, as otherwise a block of K would not be maximal. We call this vertex, if it exists, $\alpha(L)$. Note that the edge uu_1 is not a $(K \cup xCu)$-bridge by definition. It is however possible that there is a $(K \cup xCu)$-bridge that contains v_lCx. If so, we denote this special bridge by L' (otherwise, v_lCx is just an edge). The bridge L' is special among the $(K \cup xCu)$-bridges, as it is the only one that may have exactly two attachments; all other bridges have at least three attachments by the 3-Path Property.

4.2 Avoiding Overlapping Subgraphs

In the proof of Theorem 9 in [8, 9], the authors define a second connected subgraph F that overlaps with K and then recurse on both subgraphs separately by constructing Tutte paths of every block of these subgraphs (see Figure 1). The recursively constructed Tutte paths of F (giving a path from x to u) and in K (giving a path from u_1 to y) are then concatenated with uu_1 to get the desired Tutte path of G. The overlapping parts of F and K may therefore receive multiple recursive calls, which prevents to bound the running time of this decomposition.

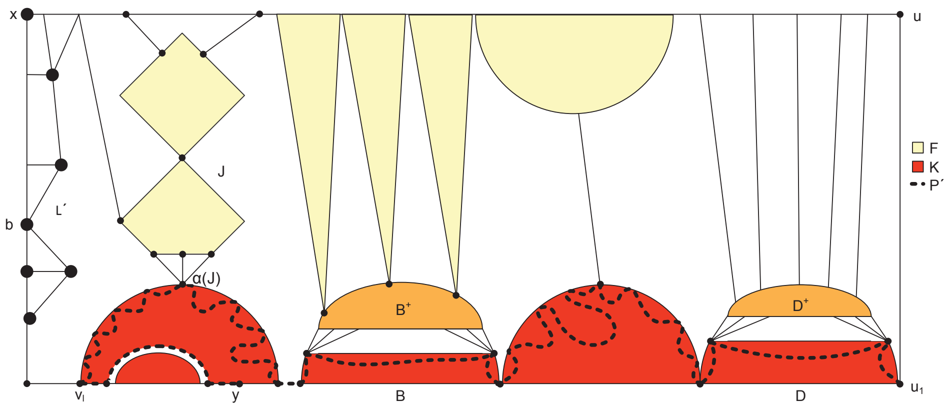


Figure 1 A circuit graph (G, C), in which the plane chain of blocks K is depicted in dark grey (red) and F is the subgraph induced by xCu and the vertices of light grey (yellow) subgraphs. Here, F and K overlap in the grey (orange) subgraphs B^+ and D^+. The part P' from u_1 to y of the desired Tutte path of G can be computed by induction on the blocks of K.

However, the description of F in [8] suggests that an overlapping subgraph in this decomposition consists always of the inner vertex set of some bridge of the Tutte path computed for K. In the following, we will compute a Tutte path from u_1 to y, but instead of doing this in K, we will do this in a slightly modified subgraph $\eta(K)$. This augmentation will allow us to identify and exclude possible overlapping subgraphs in advance. We first state some results about bridges of Tutte paths T. For the next observation, recall that

T-bridges are not single edges.

▶ **Observation 12.** *Let (G, C) be a circuit graph and let T be a Tutte path of G. Then the attachments of any T-Bridge with two attachments form a 2-separator in G.*

According to Lemma 3, both vertices of a 2-separator in a circuit graph lie on the external face boundary. The following lemma strengthens this statement for the 2-separators that are attachments of T-bridges.

▶ **Lemma 13.** *Let (G, C) be a circuit graph with a Tutte path T from $x \in V(C)$ to $y \in V(C)$. Then every T-Bridge with two attachments has either both attachments on xCy or both on yCx.*

Proof. Assume otherwise. Let J be a T-bridge with two attachments $\{c, d\}$, $c \in xCy \setminus \{x, y\}$ and $d \in yCx \setminus \{x, y\}$. By Observation 12, $\{c, d\}$ is a 2-separator in G. Thus, $G \setminus \{c, d\}$ contains exactly two components X and Y with $x \in X$ and $y \in Y$ that cover $C \setminus \{c, d\}$, according to Lemma 4. Due to Lemma 3, the inner vertex set of J is either X or Y. In both cases, J contains an edge of T, which contradicts that J is a T-bridge. ◀

We explain the idea for our decomposition; the precise decomposition will be given in the next section. Let T be a Tutte path from u_1 to y of K and consider any T-bridge J. In the decomposition of [8], by planarity, J can only intersect an overlapping part if it intersects the upper external face boundary of K. Then J has exactly two attachments c and d, according to the definition of a Tutte path and the fact that J contains a boundary edge of some block of K. By Observation 12 and Lemma 4, c and d must be as well on the boundary of K. In fact, c and d are on the upper boundary of K by Lemma 13. In summary, the only parts of K that would have possibly overlapped in the original decomposition are the T-bridges with exactly two attachments on the upper boundary of K (see also Figure 1).

Thus, if we find all 2-separators in $v_iC_iv_{i-1}$ for a block B_i of K before we actually compute a Tutte path of this block, we have identified all subgraphs of this block which would have possibly overlapped in the original decomposition. Let $\{c, d\}$ be a 2-separator of a block B_i such that c and d is in $v_iC_iv_{i-1}$. Let further B^+_{cd} be the $\{c, d\}$-bridge in B_i that contains the path cC_id (see Figure 1). We call a 2-separator $\{c, d\}$ in $v_iC_iv_{i-1}$ *maximal* in $v_iC_iv_{i-1}$ if there is no other 2-separator $\{c', d'\}$ in $v_iC_iv_{i-1}$ with c and d in $c'C_id'$. A block B_i may contain several maximal 2-separators; however, they must be consecutive on $v_iC_iv_{i-1}$. For the computation of a Tutte path of B_i, we will first find all maximal 2-separators in C_i. The next smaller 2-separators inside them will only be computed if necessary.

Let $\{c, d\}$ be a 2-separator of B_i with c and d in $v_iC_iv_{i-1}$ and let v be an inner vertex of B^+_{cd}. Then c_l and c_r are defined as the vertices in xCu closest to x and u, respectively, that are reachable from v by a path not containing any vertex of $\{c, d\} \cup V(C)$ as inner vertex (possibly $c_l = c_r$). For a 2-separator $\{c, d\}$ of B_i with c and d in $v_iC_iv_{i-1}$, let F'_{cd} be the $\{c, d, c_l, c_r\}$-bridge that contains B^+_{cd} and let $F_{cd} := F'_{cd} \setminus \{c, d\}$. The subgraph F_{cd} contains the overlapping parts of K of the original decomposition as discussed above.

In order to modify K to $\eta(K)$, we iterate through all maximal 2-separators $\{c, d\}$ of every block of K and "cut off" some B^+_{cd} in a predefined way. This will allow us to compute Tutte paths for every block of $\eta(K)$ and iteratively detour these Tutte paths to the subgraphs B^+_{cd} if necessary. For some B^+_{cd}, we will add a special edge to $\eta(K)$ whose containment in the previously computed Tutte path will decide whether such a detour is needed. The exact definition of $\eta(K)$ is dependent on the existence of a 1-separator in F_{cd}. For the relevant case $c_l \neq c_r$, we will prove that a vertex b is a 1-separator of F_{cd} if and only if $\{b, c, d\}$ is a 3-separator of G. If such a 1-separator b exists, we will show that b can actually be chosen in such a way that the subgraph of F_{cd} "above" b is a block; such a vertex will additionally be unique.

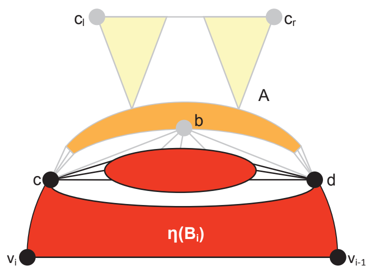
▶ **Lemma 14.** *Let $c_l \neq c_r$. A vertex $b \in F_{cd}$ is a 1-separator of F_{cd} if and only if $\{b, c, d\}$ is a 3-separator of G. No 1-separator of F_{cd} is contained in $c_l C c_r$.*

For proofs, we refer to the appendix. Lemma 14 implies that there is a block of F_{cd} that contains $c_l C c_r$. We call this block A. Note that there may be many 1-separators in F_{cd}. However, there is exactly one such 1-separator that is contained in A.

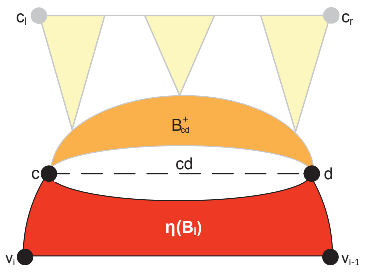
▶ **Lemma 15.** *Let $c_l \neq c_r$ and let F_{cd} contain a 1-separator. Then F_{cd} contains a unique 1-separator b such that $b \in A$.*

We are now ready to define $\eta(K)$.

▶ **Definition 16.** Let $\eta(K)$ be the graph obtained from K by performing the following for every maximal 2-separator $\{c, d\} \neq \{v_i, v_{i-1}\}$ of every block B_i of K.
Case 1: $c_l = c_r$
Do nothing.
Case 2: $c_l \neq c_r$ and F_{cd} contains a 1-separator (see Figure 2(a))
Replace B^+_{cd} with $B^+_{cd} \setminus A$.
Case 3: $c_l \neq c_r$ and F_{cd} contains no 1-separator (see Figure 2(b))
Delete all inner vertices of B^+_{cd} and add the edge cd if cd does not already exist.



(a) Case 2: $c_l \neq c_r$ and F_{cd} contains a 1-separator b. We replace B^+_{cd} with $B^+_{cd} \setminus A$.



(b) Case 3: $c_l \neq c_r$ and F_{cd} does not contain a 1-separator. We delete all inner vertices of B^+_{cd} and add the edge cd if it does not already exist.

Figure 2 The two cases of modifying K to $\eta(K)$. In both cases, the remaining part of B^+_{cd} is the dark grey (red) subgraph, i.e., the grey (orange) part of B^+_{cd} is deleted.

For a block B_i of K, let $\eta(B_i)$ be the corresponding block of $\eta(K)$. Let $\eta(C_i)$ be the external boundary of $\eta(B_i)$. Note that $\eta(K)$ is no longer a plane chain of blocks of $G \setminus xCu$, as the modified blocks $\eta(B_i)$ are not maximal any more in G. However, every $\eta(B_i)$ that is not just an edge is still a circuit graph, as shown next (see the appendix for a proof).

▶ **Lemma 17.** *Every $\eta(B_i)$ that is not an edge is a circuit graph.*

In the following, whenever dealing with a maximal 2-separator $\{c, d\}$ of K, the variables $F_{cd}, F'_{cd}, c_l, c_r, B_i, A$ will always refer to the previously defined objects and b will refer to the unique 1-separator of F_{cd} defined in Lemma 15.

4.3 Extending the Decomposition

We extend the decomposition described so far. First, we find a preliminary Tutte path P of $\eta(K)$, which will eventually be modified to a Tutte path of G in Section 4.3.2. As a speciality in advance, there are two kinds of $(K \cup xCu)$-bridges, for which the extension of P into these bridges is not hard to show; these are the *isolated* $(K \cup xCu)$-bridges, which have all their attachments, on xCu and the special bridge L'. For a thorough treatment of these bridges, we refer to the Appendix. Here, we will assume that G contains neither isolated bridges nor L'.

For a $(K \cup xCu)$-bridge L, let $C(L)$ be the shortest path in v_lCu that contains all attachments of L in v_lCu. When considering such L, the endpoints of $C(L)$ closest to v_l and u in v_lCu are called c_l and c_r, respectively ($c_l = c_r$ is possible).

4.3.1 Finding a Tutte Path of eta(K)

We continue the decomposition of a circuit graph (G, C) of Section 4.1 by computing a Tutte path $P_{\eta(K)}$ of $\eta(K)$ from u_1 to y and an SDR of the $P_{\eta(K)}$-bridges. For each block $\eta(B_i)$ of $\eta(K)$, we compute $P_{\eta(B_i)}$ and an SDR $S_{\eta(B_i)}$ of the $P_{\eta(B_i)}$-bridges as follows.

If $\eta(B_i)$ is just an edge $v_{i-1}v_i$, set $P_{\eta(B_i)} := v_{i-1}v_i$ and $S_{\eta(B_i)} := \emptyset$. Otherwise, if $i < l$, compute by induction a Tutte path $P_{\eta(B_i)}$ of $\eta(B_i)$ from v_{i-1} to v_i and a SDR $S_{\eta(B_i)}$ of all $P_{\eta(B_i)}$-bridges such that $v_i \notin S_{\eta(B_i)}$ (as intermediate vertex, an arbitrary vertex in $V(C_i) \setminus \{v_{i-1}, v_i\}$ can be chosen). If $i = l$, compute a Tutte path $P_{\eta(B_l)}$ of $\eta(B_l)$ from v_{l-1} to y through v_l and an SDR $S_{\eta(B_l)}$ of all $P_{\eta(B_l)}$-bridges. We apply the induction on $\eta(B_l)$ such that $v_l \notin S_{\eta(B_l)}$. Then $P_{\eta(K)} = \cup_{i=1}^{l} P_{\eta(B_i)}$ is the desired Tutte path of $\eta(K)$ from u_1 to y.

Every $P_{\eta(B_i)}$-bridge with three attachments in $\eta(B_i)$ is also a $P_{\eta(B_i)}$-bridge with three attachments in G. Every internal vertex of such a $P_{\eta(B_i)}$-bridge has the same neighbourhood in $\eta(B_i)$ as in G. Therefore, each such bridge preserves its number of attachments in G. The same argument holds for the $P_{\eta(B_i)}$-bridges in $\eta(B_i)$ that have exactly two attachments and contain an edge of C. In fact, these two observations do not only hold for $P_{\eta(B_i)}$, but for any Tutte path P_H of some circuit graph $H \subset G$. We will therefore only discuss P_H-bridges in the remainder of the paper that have exactly two attachments in H and do not contain any edge of C. We will show that these bridges have exactly three attachments in G.

In order to find the desired Tutte path P of (G, C), we initially set $P := xCu_1 \cup P_{\eta(K)}$ and then modify P step by step such that the final path P is a Tutte path, does not contain any edge cd that was added in Case 3, and admits an SDR S of all P-bridges. We will decompose G into smaller circuit graphs on which we apply induction. These graphs will pairwise intersect in at most one vertex, i.e., they are *edge-disjoint*. By carefully choosing a when applying the induction, we will avoid that the vertex in the intersection is a representative in both graphs. The modification of P starts by handling the $(K \cup xCu)$-bridges that have an attachment on K, but are not contained in any F_{cd}. We next show useful details of these bridges.

▶ **Lemma 18.** *Let L be any $(K \cup xCu)$-bridge for which $\alpha(L)$ exists and which is not contained in some F_{cd}. Then $\alpha(L) \in \eta(K)$ and $\alpha(L) \in P_{\eta(B_i)}$.*

4.3.2 Finding a Tutte Path of G

Algorithm 2: *FindTuttePath*$((G, C), x, u, y, P, S)$

Input: $(G, C), x, u, y, P, S$, where P is the preliminary Tutte path from x to y of Section 4.3.1 and S the corresponding SDR
Output: A Tutte path of (G, C) stored in P and an SDR S of the P-bridges in G stored in S

1. For every $(K \cup xCu)$-bridge L in G with $\alpha(L) \in \eta(K)$:
- According to Lemma 18, $\alpha(L) \in P_{\eta(B_i)}$ for some B_i.
- Let $J = (L \cup C(L)) \setminus \alpha(L)$.
- J is 2-connected: L has an inner vertex by definition of bridge and thus at least two attachments on C by the 3-Paths Property. Hence, $|V(J)| \geq 3$. Starting with $C(L)$ and adding the two paths to $C(L)$ from every remaining vertex in J due to the 3-Paths Property gives an *open ear decomposition* [18]. Thus, J is 2-connected.
- It follows that the boundary of J is a cycle and J is a circuit graph.

a. Compute a Tutte path P_J from c_l to c_r and an SDR S_J of all P_J-bridges by induction such that depending on a, either c_l or c_r is not in S_J. If $a = x$, apply the induction such that $c_l \notin S_J$. Otherwise, if $a = u$, apply the induction such that $c_r \notin S_J$.

b. Set $P := P \setminus c_lCc_r \cup P_J$ and $S := S \cup S_J$.
- By the 3-Paths Property, every P_J-bridge in J that has exactly two attachments and does not contain an edge of C must contain a vertex that in G is a neighbour of $\alpha(L)$. Each such P_J-bridge will therefore become a P-bridge with exactly three attachments in G.

2. For every maximal 2-separator $\{c, d\}$ of K satisfying Case 1 of Definition 16:
- Let J be any $P_{\eta(B_i)}$-bridge in $\eta(B_i)$ that contains an edge of $c\eta(C_i)d$. We show that every such J becomes a P-bridge in G with exactly three attachments. By the 3-Path Property, there is a path from every inner vertex of J to some vertex in C that does neither contain c nor d. In this case the only possible such vertex is $c_l = c_r$. Thus, J is a P-bridge in G with exactly three attachments, one of which is c_l.

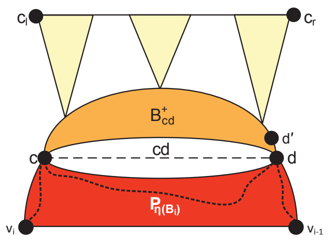
3. For every maximal 2-separator $\{c, d\}$ of K satisfying Case 2 of Definition 16:

a. Compute a Tutte path P_A of the block A of F_{cd} from c_l to c_r through b and an SDR S_A of all P_A-bridges. Apply the induction such that $a \notin S_A$, analogously to Step 1(a).

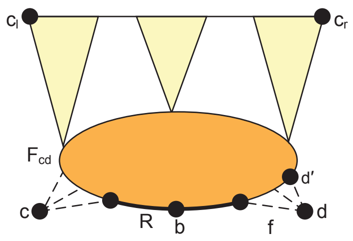
b. Set $P := P \setminus c_lCc_r \cup P_A$ and $S := S \cup S_A$.
- Let H be the $\{b, c, d\}$-bridge in G that does not contain c_lCc_r, according to Lemma 14.
- Consider any P_A-bridge J with exactly two attachments in A that does not contain an edge of C. By the 3-Paths Property, J must contain an inner vertex that has a neighbour in $G \setminus A$. Since b is a 1-separator of F_{cd} in A and $b \in P_A$, the set of all such neighbours is either $\{c\}$, $\{d\}$ or $\{c, d\}$. We will show that the last case is not possible; hence, every such P_A-bridge will become a P-bridge with exactly three attachments in G. As P_A is a Tutte path and J has only two attachments, J contains an edge of the external boundary of A. By planarity and the existence of (the connected) $\{b, c, d\}$-bridge H in G, J cannot be adjacent to both, c and d.
- In the case that $P_{\eta(B_i)}$ contains an edge of H, there may exist $P_{\eta(B_i)}$-bridges $J \subseteq H \setminus b$ with two attachments having both attachments in $c\eta(C_i)d$. We show that every such J becomes a P-bridge in G with exactly three attachments. By the 3-Path Property, there is a path from every inner vertex of J to some vertex in C that does neither contain c nor d. As $J \subset H$, this path contains b. Thus, J is a P-bridge in G with exactly three attachments, one of which is b.

4. For every maximal 2-separator $\{c, d\}$ of K satisfying Case 3 of Definition 16:

a. If $cd \notin P_{\eta(B_i)}$ (see Figure 3):



(a) A maximal 2-separator $\{c, d\}$ of B_i such that $c_l \neq c_r$ and F_{cd} contains no 1-separator. In this case, cd is not contained in $P_{\eta(B_i)}$.



(b) The subgraph F_{cd} (not containing dashed edges). We compute a Tutte path $P_{F_{cd}}$ of F_{cd} from c_l to c_r through $b \in R$ (the fat line depicts the path R).

Figure 3 Step 4(a) of *FindTuttePath*

- Let f be the face in B_i that contains cd and an inner vertex of B^+_{cd}.
- Let R be the path obtained from the boundary of B^+_{cd} in f by deleting c and d.

i. Choose an arbitrary vertex b in R.

ii. Compute a Tutte path $P_{F_{cd}}$ of F_{cd} from c_l to c_r through b by induction on F_{cd} and an SDR $S_{F_{cd}}$ of all $P_{F_{cd}}$-bridges. Apply the induction such that $a \notin S_{F_{cd}}$, analogously to Step 1(a).

iii. Set $P := P \setminus c_lCc_r \cup P_{F_{cd}}$ and $S := S \cup S_{F_{cd}}$.

- Consider any $P_{F_{cd}}$-bridge J with exactly two attachments in F_{cd} that does not contain an edge of C. By the 3-Paths Property, the inner vertex set of J is neighboured to either $\{c\}$, $\{d\}$ or $\{c, d\}$. We show that the last case is not possible, which proves that every such $P_{F_{cd}}$-bridge becomes a P-bridge in G with exactly three attachments. By the choice of R, the only vertex that may be adjacent to c and d is b (in that case, $R = \{b\}$). However, b is not a neighbour of an inner vertex of J, as $b \in P_{F_{cd}}$. This proves the claim. Ninja.

b. If $cd \in P_{\eta(B_i)}$:

- Recall that cd was possibly added during the construction of $\eta(K)$ and may therefore not be in G. We aim to replace cd in $P_{\eta(B_i)}$ with a Tutte path of B^+_{cd} from c to d.
- According to Lemma 5, $B^+_{cd} \cup cd$ is a circuit graph.
- Let d' be the neighbour of d on the boundary of $B^+_{cd} \cup cd$ that is different from c.
- Let $K' := (B^+_{cd} \cup cd) \setminus d$. According to Lemma 7, K' is a plane chain of blocks $B'_1, B'_2, \ldots, B'_{l'}$ such that $d' \in B'_1$ and $c \in B'_{l'}$. Note that K' is a subgraph of G, as it does not contain cd.
- By planarity, every $K \cup xCu$-bridge L in G that is contained in F_{cd} has its attachment $\alpha(L)$ (if exists) on the upper boundary of K', while every neighbour of d is on the lower boundary of K'.
- We will replace $cd \in P_{\eta(B_i)}$ with the union of the edge dd' and a Tutte path of $\eta(K')$ from d' to c; the Tutte path is constructed in the very same way as we did for K, i.e., by first computing $\eta(K')$, then Tutte paths of the blocks of $\eta(K')$ and then branching into the different steps of *FindTuttePath*. This will iterate on the maximal 2-separators of K', which are the sets of next smaller 2-separators of K. Note that $\eta(K)$ and $\eta(K')$ are edge-disjoint.
- Technically, $\eta()$ is defined on a given circuit graph. We face this problem by constructing the following artificial circuit graph G', which allows for a proper definition of $\eta(K')$.

- Let G' be the union of $K' \cup c_l C c_r$, all $K \cup xCu$-bridges that are contained in F_{cd}, and the new edges cc_l and $c_r d'$. Clearly, G' is a circuit graph (G', C'). Let $x' := c_l$, $u' := c_r$, $u'_1 := d'$ and $y' := c$.
- Then K' is consistent to our previous definition, i.e., the *minimal connected union* of blocks of $G' \setminus x'C'u'$ that contains y' and u'_1, and $\eta(K')$ is well-defined in dependence of G' and $\{x', u', y'\}$.

i. Compute $\eta(K')$ from K'.
ii. For each block $\eta(B'_i)$ of $\eta(K')$, compute a Tutte path $P_{\eta(B'_i)}$ and an SDR $S_{\eta(B'_i)}$ of the $P_{\eta(B'_i)}$-bridges in $\eta(B'_i)$ by induction, as described in Section 4.3.1.
iii. Set $P' := c_l P c_r \cup P_{\eta(B'_1)} \cup \dots \cup P_{\eta(B'_{l'})} \cup c_r d'$.
iv. Set $S' := S_{\eta(B'_1)} \cup \dots \cup S_{\eta(B'_{l'})}$.
v. Apply $\mathit{FindTuttePath}((G', C')), x', u', y', P', S')$.
vi. Set $P := P \setminus c_l C c_r \setminus cd \cup xPc_l \cup c_l P' c_r \cup c_r P d \cup dd' \cup d'P'c \cup cPy$.
vii. Set $S := S \cup S'$.

- By construction, (G', C') does neither contain an L'-bridge nor an isolated bridge; moreover, P' is exactly the preliminary Tutte path of (G', C') computed in Section 4.3.1. Thus, $\mathit{FindTuttePath}((G', C')), x', u', y', P')$ outputs a Tutte path of (G', C') and stores it in P'. The above construction of P then forwards the changes that were made for P' to P.
- Since P' is a Tutte path of (G', C') and by the 3-Paths Property, the only P'-bridges with two attachments that do not contain an edge of C must have an inner vertex that is a neighbour of d. As $d \in P$, such P'-bridges will become P-bridges with exactly three attachments in G.

4.4 Polynomial Time Bound for Computing Tutte Paths

It remains to show that Algorithm 2 runs in polynomial time. Clearly, all recursive calls are made on pairwise edge-disjoint circuit subgraphs; it is also easy to see that every single step of Algorithm 2 can be computed in polynomial time $O(m^k)$. It thus suffices to show that the number of recursion calls is polynomial in m and that we did not add too many new edges for the recursive calls.

Let $T(m)$ be the running time of Algorithm 2 on G having m edges. If there are j recursive calls made for the circuit graph G, let G_i be the circuit graph of the ith such call and let $m_i := |E(G_i)|$ for all $1 \le i \le j$. If we would not add any new edge during Algorithm 2, $T(m) = O(m^k) + \sum_{i=1}^{j} T(m_i)$. Let w be the neighbor of v_l in $v_l C x$. As all G_i are edge-disjoint and do not contain the edges uu_1 and $v_l w$, we have $\sum_{i=1}^{j} m_i \le m - 2$. Solving the recurrence gives then $T(m) \in O(m^{k+1})$.

However, we may have added an edge cd during the construction of $\eta(K)$ whenever we were in Case 3 of Definition 16. In each such case, the only recursive call made for G in which cd takes part is the one, say G_1, that computes the Tutte path of $\eta(B_i)$ (see Section 4.3.1). In G_1 and for each such cd, the edge dd' (see Figure 3(a)) is not contained, which restores validity of the above argument.

The most crucial open question that we want to investigate in the future is how the given polynomial running time for computing a special closed 2-walk can be improved to a low order polynomial.

References

1 T. Asano, S. Kikuchi, and N. Saito. A linear algorithm for finding Hamiltonian cycles in 4-connected maximal planar graphs. *Discrete Applied Mathematics*, 7(1):1–15, 1984.

2 D. Barnette. Trees in polyhedral graphs. *Canadian Journal of Mathematics*, 18:731–736, 1966.

3 T. Biedl. Trees and co-trees with bounded degrees in planar 3-connected graphs. In *14th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT'14)*, pages 62–73, 2014.

4 N. Chiba and T. Nishizeki. A theorem on paths in planar graphs. *Journal of graph theory*, 10(4):449–450, 1986.

5 N. Chiba and T. Nishizeki. The Hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs. *Journal of Algorithms*, 10(2):187–211, 1989.

6 R. Diestel. *Graph Theory*. Springer, fourth edition, 2010.

7 Z. Gao and R. B. Richter. 2-Walks in circuit graphs. *Journal of Combinatorial Theory, Series B*, 62(2):259–267, 1994.

8 Z. Gao, R. B. Richter, and X. Yu. 2-Walks in 3-connected planar graphs. *Australasian Journal of Combinatorics*, 11:117–122, 1995.

9 Z. Gao, R. B. Richter, and X. Yu. Erratum to: 2-Walks in 3-connected planar graphs. *Australasian Journal of Combinatorics*, 36:315–316, 2006.

10 M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5(4):704–714, 1976.

11 D. Gouyou-Beauchamps. The Hamiltonian circuit problem is polynomial for 4-connected planar graphs. *SIAM Journal on Computing*, 11(3):529–539, 1982.

12 F. Harary and G. Prins. The block-cutpoint-tree of a graph. *Publ. Math. Debrecen*, 13:103–107, 1966.

13 B. Jackson and N. C. Wormald. k-Walks of graphs. *Australasian Journal of Combinatorics*, 2:135–146, 1990.

14 W.-B. Strothmann. *Bounded degree spanning trees*. PhD thesis, FB Mathematik/Informatik und Heinz Nixdorf Institut, Universität-Gesamthochschule Paderborn, 1997.

15 C. Thomassen. A theorem on paths in planar graphs. *Journal of Graph Theory*, 7(2):169–176, 1983.

16 W. T. Tutte. A theorem on planar graphs. *Transactions of the American Mathematical Society*, 82:99–116, 1956.

17 H. Whitney. A theorem on graphs. *Annals of Mathematics*, 32(2):378–390, 1931.

18 H. Whitney. Non-separable and planar graphs. *Transactions of the American Mathematical Society*, 34(1):339–362, 1932.