Computing 2-Walks in Cubic Time*

Andreas Schmid
Max Planck Institute for Informatics

Jens M. Schmidt
Technische Universität Ilmenau†

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, into which the graph is decomposed, are edge-disjoint. This implies the first polynomial-time algorithm that computes the closed 2-walk mentioned above. Its running time is $O(n^3)$.

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 [18] 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 [17]. Thomassen [16] 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 in 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,

*An extended abstract of this article was published in STACS 2015.

†This research was initiated at Max Planck Institute for Informatics, Saarbrücken.

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 [15]. 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, 9]. 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 (neither for [7] nor for [8, 9]).

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 [16]. 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 much more general problem of overlapping subgraphs in 3-connected planar graphs has not been resolved so far. Even the decomposition in [8, 9] 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, 9], generalizing the previous results. The result is stated for the class of circuit graphs, which contain all 3-connected planar 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 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 can be computed in time $O(n^3)$.*

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 two 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 it is 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 plane graph G with a (simple) cycle C as external face boundary 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.

Equivalently, a planar graph is a circuit graph if it can be obtained from a 3-connected graph by deleting a vertex. Clearly, circuit graphs are 2-connected and generalize 3-connected plane graphs. In the following, we will give several lemmas about circuit graphs that will be used throughout the paper. The next two lemmas are probably folklore.

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 $B_1, B_2, \ldots, B_k$ 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 $B_1, B_2, \ldots, B_k$ and, if $k > 1$, one of the neighbors of v in C is in $B_1 \setminus B_2$ and the other is in $B_k \setminus B_{k-1}$.*

If the external face boundary of the circuit graph is a triangle we can find an even more special structure.

Lemma 8 ([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 $B_1, B_2, \ldots, B_k$ and, if $k > 1$, z is in $B_1 \setminus B_2$ and one neighbor of w is in $B_k \setminus B_{k-1}$.*

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 neighbor of w in B_k. □

A *Tutte path* (*Tutte cycle*) of a circuit graph (G, C) is a path (*cycle*) T for which every T-bridge has exactly two attachments if it contains an edge of C and otherwise exactly three attachments. A Tutte path *from x to y through u* has startvertex x, endvertex y and contains u; we will sometimes say that u is the *intermediate vertex* of T. We end this section with a key property of the bridges of a Tutte path.

Observation 9. *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 10. *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 9, $\{c, d\}$ 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, X and Y must contain at least one vertex of C each. It follows that 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. □

3 From Tutte Paths to 2-Walks

It was shown by Gao, Richter and Yu [8, 9] that in order to find a closed 2-walk, it suffices to find a Tutte path that has a system of distinct representatives. We define these terms and briefly recall the argument of [8, 9] below. The authors of [8, 9] proved the existence of a Tutte path T with T-bridges $B_1, B_2, \ldots, B_k$, 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 11 is slightly weaker than the one in [8, 9] (in which $y \in V(G)$), but sufficient for our needs.

Theorem 11 ([8, 9]). *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.

Corollary 12 ([8, 9])**.** 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 12 is the crucial new insight of Gao, Richter and Yu's paper [8, 9]. 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 that are 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 13 ([8, 9])**.** *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.*

Outline of this paper: The computational complexity of finding the 2-walk guaranteed by Theorem 13 depends on two parts: How long it takes to find the Tutte cycle of Corollary 12, and how much time is spent on combining the Tutte paths of various components into one 2-walk. The former will be addressed in Section 4 by showing the following:

Lemma 14. *The Tutte cycle T of Corollary 12 can be computed in time $O(n^2)$.*

Proving Lemma 14 requires giving a new proof of Corollary 12, as the proof given in [8, 9] uses a decomposition into circuit graphs that may overlap in large parts. The remainder of this section is devoted to the second part, i.e., combining Tutte paths of several blocks to obtain a 2-walk. As it turns out, the approach of [8, 9] will lead to a polynomial time algorithm for this task. For this reason, we give an overview of the techniques used there and then analyze their run-time.

Computing 2-Walks from Tutte paths and Cycles. Let T be a Tutte cycle and S be an SDR as given in Corollary 12. If G is a triangle, T is itself the desired 2-walk W of Theorem 13; otherwise, we use induction on m. For every T-bridge L in G and its representative s in S, we consider a plane chain of blocks as follows.

If L has exactly two attachments (thus, L contains an edge of C), let t be the attachment different from s. Then $\{s, t\}$ is a 2-separator of G and $L \cup st$ is a circuit graph, according to Lemmas 4 and 5. According to Lemma 7, $(L \cup st) \setminus t$ and therefore also $L \setminus t$ is a plane chain of blocks $B_1, \ldots, B_l$ such that $s \in B_1$ and $t' \in B_l$ for the neighbor t' of t in $C \cap L$. Set $v_0 := v$ and $v_l := t'$.

If L has exactly three attachments $\{s, t, z\}$, $L \cup \{st, tz, zs\}$ is a circuit graph due to the 3-Path Property. By Lemma 8, $L \cup \{st, tz, zs\} \setminus \{t, z\} = L \setminus \{t, z\}$ is a plane chain of blocks $B_1, \ldots, B_l$ such that $s \in B_1$ and $z' \in B_l$ for the neighbor z' of z on the boundary of L in direction s. Set $v_0 := s$ and $v_l := z'$.

Let v_i be the 1-separator $B_i \cap B_{i-1}$ of the constructed plane chain of blocks for every i. Each B_i is either an edge or a circuit graph. If B_i is an edge, we define an artificial walk $v_{i-1}, v_{i-1}v_i, v_i, v_iv_{i-1}, v_{i-1}$ for B_i; otherwise, there is a 2-walk in B_i by induction with $x := v_{i-1}$ and $y := v_i$. In both cases, v_i is visited exactly once, implying that the union W_L of these walks is a

2-walk of the plane chain of blocks in which v is visited exactly once. Finally we obtain the desired 2-walk W by traversing T from one representative s of a T-bridge to the next and detouring into W_L every time. Note that every s is visited exactly twice, once by T and once by W_L, as it is a representative in S.

For all steps taken in the description above, except the computation of Tutte paths and the computation of suitable circuit subgraphs (i.e., the above plane chains of blocks) for the recursion on L, the corresponding existence proofs give immediately linear-time algorithms.

We next show that a polynomial-time computation of a Tutte path implies a polynomial-time computation of a 2-walk. Assume that a Tutte cycle T of G and its SDR S can be computed in time cm^k for some integers c and k. If the 2-walks in the T-bridges have already been computed by recursion, taking the union of T and these 2-walks needs only linear time. Let $time(m)$ denote the running time of the resulting algorithm. We number all blocks of the plane chains of blocks that were constructed for T-bridges in G from 1 to j. Let m_i denote the number of edges in block i. As all these blocks are edge-disjoint and T contains at least one edge, $\sum_{i=1}^{j} m_i < m$. Thus, $time(m) = cm^k + \sum_{i=1}^{j} time(m_i) \leq cm^{k+1}$, as we always recurse on strictly smaller subgraphs and the recursion depth is at most m. Therefore, a proof of Lemma 14 implies our main result Theorem 1.

4 Finding Tutte Paths

We will prove Theorem 11 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, 9] 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, 9]. 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 11 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 be the neighbor of u in uCx. 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). The blocks of K form a tree; by minimality, K will be a plane chain of blocks. Let $B_1, \ldots, B_l$ 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 simplicity, we divide the external face boundary C_i of any block B_i of K into its *lower part*, which is $v_{i-1}C_iv_i$, and its *upper part*, which is $v_iC_iv_{i-1}$. The

lower boundary of K is then the union of the lower parts of all blocks of K, and the *upper boundary of K* is the union of the upper parts of all block of K.

4.2 Avoiding Overlapping Subgraphs

In the original proof of Theorem 11 given 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.

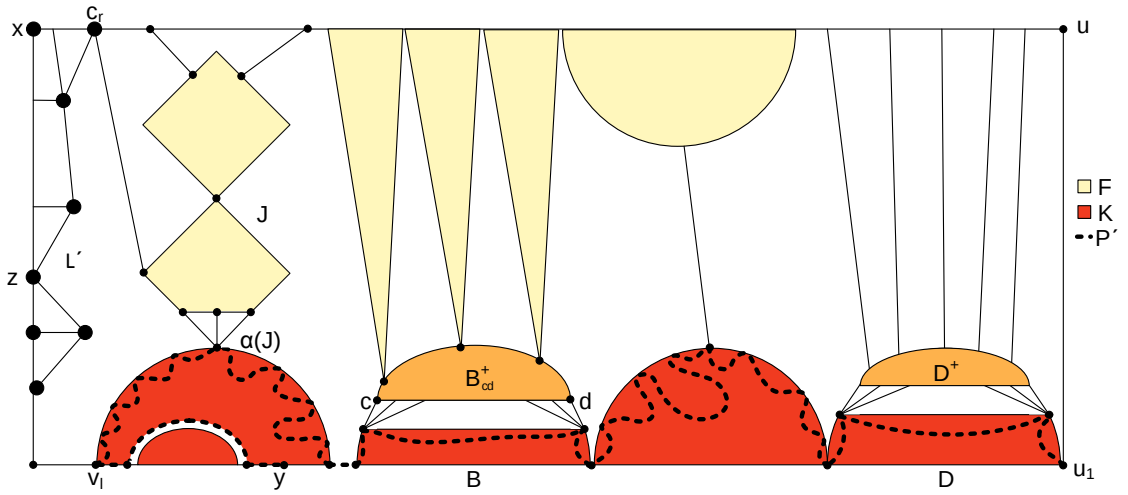


Figure 1: A circuit graph (G, C), in which the plane chain of blocks K is depicted in dark gray (red) and gray (orange), and F is the subgraph induced by xCu and the vertices of light grey (yellow) and gray subgraphs. Here, F and K overlap in the gray subgraphs B^+ and D^+. The part P' from u_1 to y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

However, the description of F in [8, 9] 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.

Contrast to the approach of [8, 9]: 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 of K. In the decomposition of [8, 9], by planarity, J can only take part in an overlapping 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 9 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 10. 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 (drawn in gray (orange) in Figure 1). Thus, if we find for some block B_i of K all 2-separators in $v_iC_iv_{i-1}$ 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.

Now we give the details of this approach. Let $\{c, d\}$ be a 2-separator of a block B_i such that c and d are in $v_iC_iv_{i-1}$ (here we denote the vertex that appears first in $v_iC_iv_{i-1}$ by c and the other by d). 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 $c'C_id'$. Note that in the special case $v_iv_{i-1} \in C_i$ two maximal 2-separators $\{v_i, c'\}$ and $\{d', v_{i-1}\}$ may occur that *interlace*, i.e. for which $v_iC_ic' \cap d'C_ic_{i-1} \neq \emptyset$. This is the only case in which two maximal 2-separators can interlace, since if otherwise $v_iv_{i-1} \notin C_i$, $\{v_i, v_{i-1}\}$ would be a 2-separator of B_i such that $v_iC_iv_{i-1}$ would contain both v_iC_ic' and $d'C_iv_{i-1}$, which contradicts their maximality. We resolve this special case of having two interlacing maximal 2-separators by always using the one of these two that contains v_i in the following description and ignoring the other. Because of this, the maximal 2-separators taken for every block B_i will 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. Possible 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 in G by a path not containing any vertex of $\{c, d\} \cup V(C)$ as inner vertex (possibly $c_l = c_r$). Figure 3 shows two examples where $c_l \neq c_r$. 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 $F_{cd} := F'_{cd} \setminus \{c, d\}$. (Continuing the above contrast to [8, 9], the graph F_{cd} contains the possibly overlapping parts of K of the original decomposition.)

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 (see Figure 2). 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 15. *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_lCc_r.*

Proof. Let b be any 1-separator of F_{cd}. We first show that $b \notin c_lCc_r$, giving the second claim. Let J be the c_lCc_r-bridge of F_{cd} containing the connected graph $B^+_{cd} \setminus \{c, d\}$. By definition of c_l and c_r, J contains c_l and $c_r \neq c_l$ as attachments. Every other c_lCc_r-bridge in F_{cd} does not touch K and therefore has at least three attachments on c_lCc_r by the 3-Paths Property. Since c_lCc_r is a path, deleting any vertex of c_lCc_r in F_{cd} leaves a connected graph.

Consider any component of $F_{cd} \setminus b$ that does not contain c_lCc_r. This component can have at most the neighbors $\{b, c, d\}$ in G. Since the component does not contain any vertex of C, its neighbor set in G must be exactly $\{b, c, d\}$, according to the 3-Paths Property. Thus, $\{b, c, d\}$ is a 3-separator of G.

Let $\{b, c, d\}$ be a 3-separator of G. Then $b \notin c_lCc_r$, as otherwise $G \setminus \{b, c, d\}$ would be connected by definition of c_l and $c_r \neq c_l$. Consider any component of $F_{cd} \setminus b$ that does not contain c_lCc_r. Since this component contains no vertex of C, its neighbor set in G is exactly $\{b, c, d\}$. Thus, b separates some vertex of that component from c_lCc_r in F_{cd} and is therefore a 1-separator of F_{cd}. □

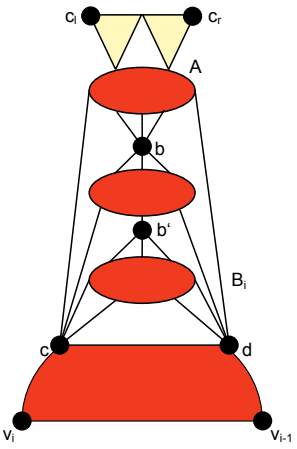


Figure 2: Two 1-separators b and b' of F_{cd}. The 1-separator b is the unique one contained in A.

Lemma 15 implies that there is a block of F_{cd} that contains c_lCc_r. We call this block A. Note that there may be many 1-separators in F_{cd} (see Figure 2). However, there is exactly one such 1-separator that is contained in A.

Lemma 16. *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 $b \in A$.*

Proof. Since F_{cd} has a 1-separator and by the maximality of the block A of F_{cd}, A contains at least one 1-separator b of F_{cd}. Assume to the contrary that A contains a 1-separator $b' \neq b$ of F_{cd}. Let H_1 and H_2 be components of $F_{cd} - b$ and $F_{cd} - b'$, respectively, that do not contain c_lCc_r. As 1-separators that are contained in the same block A separate disjoint components from A (as implied by the block-cut-tree), H_1 and H_2 are disjoint; moreover, both do not contain any vertex of C. By the 3-Paths Property, H_1 and H_2 are neighbored exactly to $\{b, c, d\}$ and $\{b', c, d\}$ in G, respectively. Then the union of C and the set of three paths from H_1 and from H_2 to C due to the 3-Paths Property form a $K_{3,3}$, which contradicts the planarity of G. □

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

Definition 17. 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.

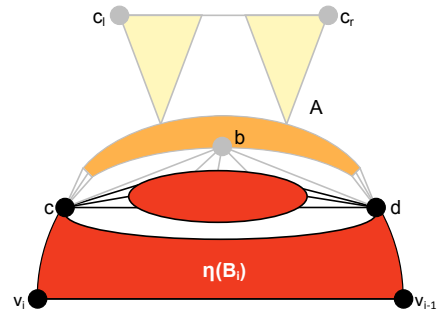
Case 1: $c_l = c_r$
Do nothing.

Case 2: $c_l \neq c_r$ and F_{cd} contains a 1-separator (see Figure 3(a))
Replace B^+_{cd} with $B^+_{cd} \setminus A$.

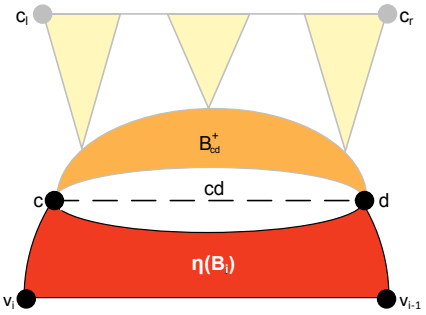
Case 3: $c_l \neq c_r$ and F_{cd} contains no 1-separator (see Figure 3(b))
Delete all inner vertices of B^+_{cd} and add the edge cd if cd does not already exist.

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 no longer maximal in G. However, every $\eta(B_i)$ that is not just an edge is still a circuit graph, as shown next.

Lemma 18. *Every $\eta(B_i)$ that is not an edge is a circuit graph.*



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

Proof. Clearly the claim is true when $\eta(B_i) = B_i$, thus assume the contrary. We consider a B_i after one Case 2 or Case 3 modification of Definition 17; the arguments extend readily to multiple such modifications.

Consider a Case 2 modification. Let b be the unique 1-separator of Lemma 16. According to Lemma 15, $\{b, c, d\}$ is a 3-separator of G. Let H denote the (unique) $\{b, c, d\}$-bridge of G that does not contain a vertex of C. By the definition of bridges, $H \setminus b$ is connected. We show that the boundary part of $H \setminus b$ from c to d that contains all former neighbors of b is a path. Otherwise, an clockwise boundary traversal from c to d would visit some vertex z twice, which gives a 2-separator $\{z, b\}$ that contradicts Lemma 4. Thus, the claim follows directly from extending this path by dC_ic (which is internally disjoint) and applying Lemma 6.

Consider a Case 3 modification. Then $B^+_{cd} \cup cd$ is a circuit graph by Lemma 5. □

4.3 Extending the Decomposition

We extend the decomposition described so far. From now on, we will name the input graph (G', C') instead of (G, C), but keep all other notation such as K, B_i, x, y, u, u_1.

For every $(K \cup xC'u)$-bridge L in G', 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 $(K \cup xC'u)$-bridge by definition. It is however possible that there is a $(K \cup xC'u)$-bridge that contains $v_lC'x$. If so, we denote this special bridge by L' (otherwise, $v_lC'x$ is just an edge). The bridge L' is special among the $(K \cup xC'u)$-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.

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

A $(K \cup xC'u)$-bridge $L \neq L'$ in G' is *isolated* if $\alpha(L)$ does not exist (i.e., $L \cap K = \emptyset$), and its 2-separator $\{c_l, c_r\}$ of $xC'u$ is maximal in $xC'u$ with respect to the 2-separators of all other such bridges. Thus, L is different from L' and has at least three attachments on $xC'u$.

We now transform the input graph (G', C') into a graph (G, C) that does not contain L' anymore. If L' does not exist in G', then we simply set $G := G'$. Otherwise, we apply the following

modification on G' that depends on the number of attachments of L'. If L' has exactly two attachments (namely, v_l and x), obtain G from G' by replacing L' with the edge $v_l x$. Otherwise, L' has at least the three attachments v_l, x, c_r (as is the case in Figure 1). Let $L^* := (L' \cup C'(L')) \setminus v_l$. Note that L^* may not be 2-connected. We obtain G from G' by contracting L^* to one vertex c_r (which will be x in G) and subsequently deleting multiedges.

Note that K and $\eta(K)$ are the same for G and G'. In Section 4.3.1, we will find a Tutte path of $\eta(K)$ and an SDR S of its bridges. In Section 4.3.2, this Tutte path of $\eta(K)$ will be modified to a Tutte path of G. Eventually, we show in Section 4.3.3 how to deal with the special bridge L' in G' and thereby extend the Tutte path and its SDR found in G to a Tutte path of G'.

4.3.1 Finding a Tutte Path of $\eta(K)$

We continue the decomposition of the circuit graph (G, C) (as described in Section 4.1) by computing a Tutte path $P_{\eta(K)}$ of $\eta(K)$ from u_1 to y and an SDR $S_{\eta(K)}$ 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 an 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. Since we may need $v_l \in B_l$ as representative for L' in Section 4.3.3, we have to ensure that v_l does not become a representative for any $P_{\eta(B_l)}$-bridge in $\eta(B_l)$. Thus, 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 and $S_{\eta(K)} = \cup_{i=1}^{l} S_{\eta(B_i)}$ is an SDR of $P_{\eta(K)}$s bridges in $\eta(K)$.

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 neighborhood 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.

4.3.2 Finding a Tutte Path of G

In order to find the desired Tutte path P of (G, C) and an SDR S for its bridges, we initially set $P := xCu_1 \cup P_{\eta(K)}$ and $S := S_{\eta(K)}$, and then modify P and S step by step such that the final path P is a Tutte path of (G, C), does not contain any edge cd that was added in Case 3 of the definition of η, and S is an SDR 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 ensure that the intersection vertex is a representative in at most one intersecting graph. 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.

Let L be any $(K \cup xCu)$-bridge for which $\alpha(L)$ exists and which is not contained in some F_{cd}.

Lemma 19. *$\alpha(L) \in \eta(K)$ and $\alpha(L) \in P_{\eta(B_i)}$.*

Proof. For the first claim, assume to the contrary that $\alpha(L) \in K$ is not in $\eta(K)$. Then $\alpha(L)$ lies on the boundary of K on a path between the vertices of a maximal 2-separator $\{c, d\}$ and thus must be part of F_{cd}, contradicting the assumption.

Next assume $\alpha(L) \notin P_{\eta(B_i)}$. As $\alpha(L)$ is on the boundary of $\eta(B_i)$, $\alpha(L)$ must be contained in a $P_{\eta(B_i)}$-bridge in $\eta(B_i)$ with two attachments $\{c', d'\}$. By Observation 9, $\{c', d'\}$ is a 2-separator of $\eta(B_i)$. As L is not contained in some F_{cd}, $\{c', d'\}$ is not a maximal 2-separator of B_i. Thus, there exists a maximal 2-separator $\{c, d\}$ with $c'C_id' \subseteq cC_id$. This gives a contradiction, as then by construction $\alpha(L) \notin \eta(K)$. □

Let J' be the union of $L, C(L)$ and all $C(L)$-bridges of G which have all their attachments in $C(L)$. Let $J = J' \setminus \alpha(L)$.

Lemma 20. *J is a circuit graph.*

Proof. We first prove that J is 2-connected: L has an inner vertex by the definition of a 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* [19]. Thus, J is 2-connected. It follows that the boundary of J is a cycle and J is a circuit graph. □

We are now ready to describe an algorithm that, given the circuit graph (G, C), vertices $x, u, y \in V(C)$ and the preliminary Tutte path P as defined above, outputs a Tutte path of G and an SDR of its bridges in G.

Algorithm 1: *FindTuttePath*$((G, C), x, u, y, P, S)$
Input: $(G, C), x, u, y, P, S$, where P is the preliminary Tutte path $xCu_1 \cup P_{\eta(K)}$ from x to y and $S = S_{\eta(K)}$ the corresponding SDR.
Output: A Tutte path of (G, C) and an SDR of its bridges in G stored in P and S respectively.

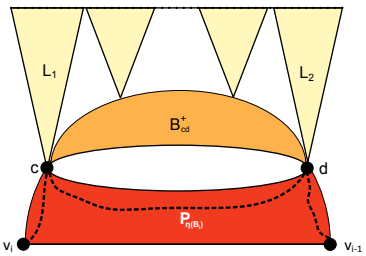
1. For every $(K \cup xCu)$-bridge L in G with $\alpha(L) \in \eta(K)$ (see Figure 4):
   - According to Lemma 19, $\alpha(L) \in P_{\eta(B_i)}$ for some B_i.
   - Let J' be the union of $L, C(L)$ and all $C(L)$-bridges of G which have all their attachments in $C(L)$. Let $J = J' \setminus \alpha(L)$. By Lemma 20, 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 $c_r \notin S_J$.

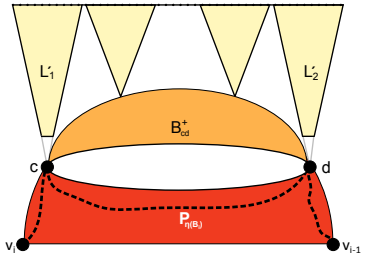
   (b) Set $P := P \setminus C(L) \cup P_J$ and $S := S \cup S_J$.
   - By the 3-Paths Property, every P_J-bridge in J that has exactly two attachments and does not contain an edge of C must contain a vertex that in G is a neighbor of $\alpha(L)$. Each such P_J-bridge will therefore become a P-bridge with exactly three attachments in G.
2. For every maximal 2-separator $\{c, d\}$ of K satisfying Case 1 of Definition 17:
   - Let J be any $P_{\eta(B_i)}$-bridge in $\eta(B_i)$ that contains an edge of $c\eta(C_i)d$ (recall that $\eta(C_i)$ denotes the external boundary of $\eta(B_i)$). 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 contains neither 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 and its representative in S will be as chosen in $S_{\eta(B_i)}$.
3. For every maximal 2-separator $\{c, d\}$ of K satisfying Case 2 of Definition 17 (see Figure 5):

   (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. 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 $P := P \setminus c_lCc_r \cup P_A$ and $S := S \cup S_A$.



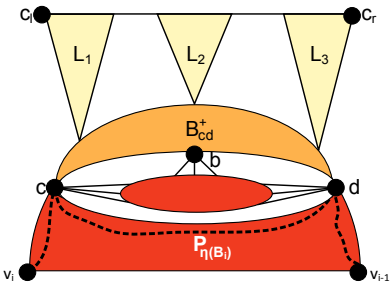
(a) Only the $(K \cup xCu)$-bridges L_1 and L_2 have an attachment in $\eta(K)$.



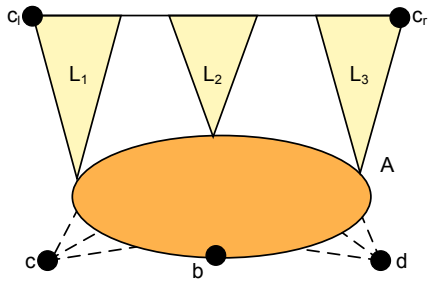
(b) From L_1 and L_2, circuit graphs L'_1 and L'_2 are constructed. We compute a Tutte path in each of them.

Figure 4: Step 1 of *FindTuttePath*. We consider the $(K \cup xCu)$-bridges that have an attachment in $P_{\eta(B_i)}$ (dashed line).

- Let H be the $\{b, c, d\}$-bridge in G that does not contain $c_l C c_r$, according to Lemma 15.
- 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 neighbor 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 neighbors is either $\{c\}$, $\{d\}$ or $\{c, d\}$. We will show that the last case is not possible. Namely, 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. Hence, every such P_A-bridge will become a P-bridge with exactly three attachments in G.
- In the case that $P_{\eta(B_i)}$ contains an edge of H, there may exist a $P_{\eta(B_i)}$-bridge $J \subseteq H \setminus b$ with two attachments having both attachments in $c\eta(C_i)d$. By the 3-Path Property, there is a path from every inner vertex of J to some vertex in C that contains neither 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.
- By applying the induction depending on the value of a, we ensure that a is not a representative of any bridge in the final SDR S. Furthermore it ensures that the vertex in the intersection of two subgraphs F_{cd} and $F_{c'd'}$ is used as a representative in the result of at most one induction call made by the algorithm.



(a) A maximal 2-separator $\{c, d\}$ of B_i such that $c_l \neq c_r$ and F_{cd} contains a 1-separator. The unique 1-separator of F_{cd} in $A \subset F_{cd}$ is b.

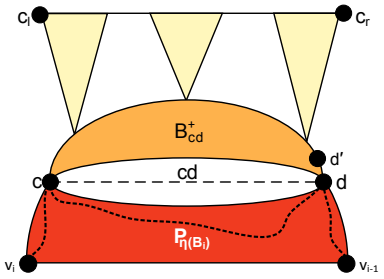


(b) The block A of F_{cd} that contains $c_l C c_r$ (dashed edges are not part of A). We compute a Tutte path P_A of A from c_l to c_r through b.

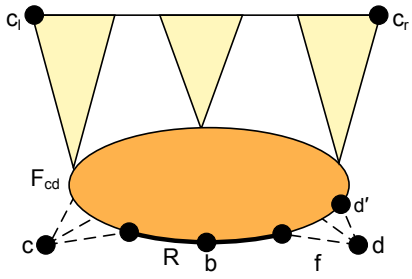
Figure 5: Step 3 of *FindTuttePath*

4. For every maximal 2-separator $\{c, d\}$ of K satisfying Case 3 of Definition 17:
   (a) If $cd \notin P_{\eta(B_i)}$ (see Figure 6):
   - 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. If $a = x$, apply the induction such that $c_l \notin S_J$. Otherwise, if $a = u$, apply the induction such that $c_r \notin S_J$.
   iii. Set $P := P \setminus c_l C c_r \cup P_{F_{cd}}$ 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 neighbored 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 neighbor of an inner vertex of J, as $b \in P_{F_{cd}}$. This proves the claim.

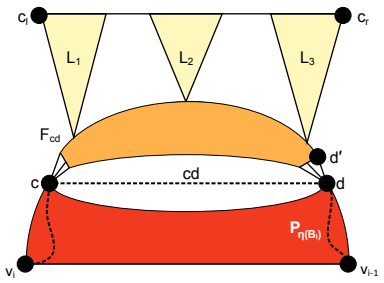


(a) A maximal 2-separator $\{c, d\}$ of B_i such that $c_l \neq c_r$ and F_{cd} contains no 1-separator. In this case, cd is not contained in $P_{\eta(B_i)}$.

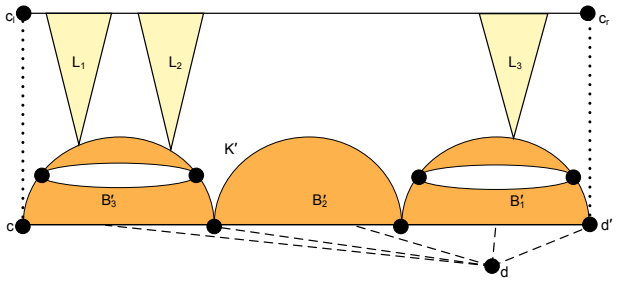


(b) The subgraph F_{cd} (not containing dashed edges). We compute a Tutte path $P_{F_{cd}}$ of F_{cd} from c_l to c_r through $b \in R$ (the fat line depicts the path R).

Figure 6: Step 4(a) of *FindTuttePath*



(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 contained in $P_{\eta(B_i)}$.



(b) The circuit graph (G', C') (not containing dashed edges), which contains the plane chain of blocks K'. We iterate the computation of a Tutte path on $\eta(K')$ in (G', C'), which corresponds to iterating on the next smaller maximal 2-separators of K.

Figure 7: Step 4(b) of *FindTuttePath*

(b) If $cd \in P_{\eta(B_i)}$ (see Figure 7):

- 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.
- This case is more complicated than the previous ones due to the fact that both c and d could already be representatives in S. The induction hypothesis allows us to protect only one vertex by choosing the parameter a. In the following, we will therefore apply induction on a modification of the graph $B^+_{cd} \cup F_{cd}$ such that d is not contained in this graph and $c \notin S$ in the end.
- According to Lemma 5, $B^+_{cd} \cup cd$ is a circuit graph.
- Let d' be the neighbor 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 $B'_1, B'_2, \dots, B'_{l'}$ 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' (see Figure 7b), while every neighbor 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 $\overline{G}$, which allows for a proper definition of $\eta(K')$.
  - Let $\overline{G}$ 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, $\overline{G}$ is a circuit graph $(\overline{G}, \overline{C})$. 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 $\overline{G} \setminus x'\overline{C}u'$ that contains y' and u'_1, and $\eta(K')$ is well-defined in dependence of $\overline{G}$ 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 $\mathit{FindTuttePath}((\overline{G}, \overline{C})), x', u', y', P', S')$.

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

with exactly three attachments in G.

5. For every isolated $(K \cup xCu)$-bridge L in G:
   - Any isolated bridge L that is contained in some F_{cd} for a maximal 2-separator $\{c, d\}$ of K has already been part of a recursive call that computed the Tutte-path $P_{F_{cd}}$. Therefore, it does not have to be considered again and we restrict ourselves to isolated bridges that are not contained in some F_{cd}.
   - Any path from x to u in $G \setminus K$ must pass through $J(L)$ and, in particular, through the vertices $\{c_l, c_r\}$ of L. In G, P contains $c_l C c_r$. Recall that $J(L)$ is a circuit graph. We aim for replacing the subpath $c_l C c_r$ in P with a Tutte path of $J(L)$ from c_l to c_r.
   - As c_l or c_r may already be in S, we have to be careful about how we apply the induction. In Steps 1(a), 3(a) and 4(a), we applied the induction on all F_{cd} subgraphs depending on the vertex a; hence, we know that not both c_l and c_r are already in S. In order to ensure that we do not add a to S in the case that $a \in J(L)$ (for example if $a = x$) and neither reuse c_l nor c_r as a representative, we will apply the induction in the same fashion depending on a.

   (a) Compute a Tutte path $P_{J(L)}$ of $J(L)$ from c_l to c_r by induction on $J(L)$ and an SDR $S_{J(L)}$ of all $P_{J(L)}$-bridges. If $a = x$, apply the induction such that $c_l \notin S_{J(L)}$. Otherwise, if $a = u$, apply the induction such that $c_r \notin S_{J(L)}$.

   (b) Set $P := P \setminus c_l C c_r \cup P_{J(L)}$ and $S := S \cup S_{J(L)}$.
   - Since L is an isolated bridge, every $P_{J(L)}$-bridge has no neighbor in $G \setminus J(L)$, and therefore does not change its number of attachments as a bridge of P in G.

4.3.3 Dealing with L'

We show how to deal with the bridge L' that we removed in advance. In the graph G, let P be a Tutte path from x to y through u with SDR S of all P-bridges in G, as computed by Algorithm 1. Assume that the bridge L' exists in G', as otherwise there is nothing to do. If L' has exactly two attachments in G' (namely x and v_l), then $v_l \in P$ and $v_l \notin S$ by the construction of P. In that case, P is a Tutte path of G', L' is a P-bridge of G' with two attachments, and we simply add v_l to S as the representative of L'.

Otherwise, L' has at least the three attachments v_l, x, c_r in G' (see Figure 1). Let $L^* := (L' \cup C(L')) \setminus v_l$. Note that L^* may not be 2-connected. The following steps will extend the subpath $c_r P y$ of P and the SDR S computed by Algorithm 1 to a Tutte path from x to y and SDR of G'.

1. If L^* is not 2-connected:
   - Every 1-separator z of L^* is contained in $v_l C c_r \setminus v_l$ (note that $v_l \notin L^*$), as otherwise $\{z, v_l\}$ would be a 2-separator with $z \notin C$, contradicting Lemma 4. Furthermore, $z \in v_l C x \setminus v_l$, as otherwise $z \in xCc_r \setminus x$ would imply that $\{z, v_l\}$ is a 2-separator that violates the choice of L' (e.g., c_r would not be an attachment of L').

   (a) Let z be the 1-separator of L^* in $v_l C x \setminus v_l$ closest to x (possibly $z = x$).

   (b) Let L^*_B be the $\{z\}$-bridge of L^* containing c_r.
2. If L^* is 2-connected:

   (a) Let z be the neighbor of v_l in $C \cap L^*$.

   (b) Let $L^*_B := L^*$.
3. Compute a Tutte path $P_{L^*_B}$ of L^*_B from x to c_r through z and an SDR $S_{L^*_B}$ of all $P_{L^*_B}$-bridges in L^*_B by induction.
   - In both cases, L^*_B is 2-connected and thus a circuit graph due to Lemma 6.
   - We apply the induction such that, depending on the value of a, either x or c_r is not in

S_{L^*}.

4. Obtain the Tutte path $P := P_{L_B^*} \cup c_r P y$ of G' with SDR $S := S_{L_B^*} \cup S$.
   - If L^* is not 2-connected, then the part of L^* that is not contained in L_B^* becomes a P-bridge with attachments v_l and z. As Algorithm 1 ensures $v_l \notin S$, we can add v_l to S as the representative of that P-bridge.

4.4 A Quadratic Time Bound for Computing Tutte Paths

We consider the overall algorithm A on the circuit graph G' with m edges, which modifies G' to a graph G having no special bridge L' (see beginning of Section 4.3), then computes a preliminary Tutte path P of G (see Section 4.3.1), and eventually invokes Algorithm 1 to extend P. Let $time(m)$ be the running time of Algorithm A on G'. We need to show that $time(m) = O(m^2) = O(n^2)$.

Clearly, all recursive calls of Algorithm A are made on pairwise edge-disjoint circuit graphs. For detecting blocks and chains of blocks, we use any algorithm such as [14] that is able to compute the 2-connected components of a graph in linear time. Every single step of Algorithm A that is not a recursive call uses elementary graph operations or computes maximal 2-separators and can therefore be done in time $O(m)$.

It thus suffices to show that the number of recursive calls of A is linear in m and that we did not add too many new edges for every recursive call. If j recursive calls were invoked on G', let G_i be the circuit graph of the ith such call and let $m_i := |E(G_i)|$ for all $1 \leq i \leq j$.

If we would not add any new edge during the computation of A, every G_i would be a subgraph of G' and we would have the recurrence $time(m) = O(m) + \sum_{i=1}^{j} time(m_i)$. 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 $\sum_{i=1}^{j} m_i \leq m - 2$. Solving the recurrence above gives then $time(m) = O(m^{1+1}) = O(n^2)$.

However, we may have added a new edge cd in Algorithm A only when constructing $\eta(K)$ (in Case 3 of Definition 17), either during the computation of the preliminary Tutte path P or before the recursive call of Algorithm 1 in Case 4(b)i. Every such new edge cd is part of exactly one recursive call made for G' on a graph G_i that is a block of ηK (see Section 4.3.1). However, for every such G_i and cd, the unique edge dd' (as shown in Figure 6(a)) is not contained in any recursive call made for G'. Since this edge dd' compensates the additional edge cd, this restores the validity of the above argument.

This proves Lemma 14 and hence Theorem 1. The most crucial open question is how the given cubic running time for computing a special closed 2-walk can be improved to a polynomial of lower order.

Acknowledgments. We wish to thank the anonymous reviewers for their highly detailed and valuable feedback.

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] Jens M Schmidt. A simple test on 2-vertex-and 2-edge-connectivity. *Information Processing Letters*, 113(7):241–244, 2013.

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

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

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

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

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