

Testing Mutual Duality of Planar Graphs*

Patrizio Angelini¹⁾

Thomas Bläsius²⁾

Ignaz Rutter²⁾

¹⁾ Roma Tre University
angelini@dia.uniroma3.it

²⁾ Karlsruhe Institute of Technology (KIT)
{blaesius,rutter}@kit.edu

Abstract

We introduce and study the problem MUTUAL DUALITY, asking for two planar graphs G_1 and G_2 whether G_1 can be embedded such that its dual is isomorphic to G_2 . We show NP-completeness for general planar graphs and give a linear-time algorithm for biconnected planar graphs. This algorithm implies an efficient solution to two well-known problems. In fact, it can be used to test whether two biconnected planar graphs are 2-isomorphic, namely whether their graphic matroids are isomorphic, and to test self-duality of any biconnected planar graph, which is a special case of MUTUAL DUALITY with $G_1 = G_2$. Further, we show that our NP-hardness proof extends to testing self-duality and map self-duality (which additionally requires to preserve the embedding).

In order to obtain our results, we consider the *common dual relation* \sim , where $G_1 \sim G_2$ if and only if they admit embeddings that result in the same dual graph. We show that \sim is an equivalence relation on the set of biconnected graphs and devise a compact SPQR-tree-like representation of its equivalence classes. Our algorithm for biconnected graphs is based on testing isomorphism for two such representations in linear time.

1 Introduction

Given a planar graph G with a planar embedding \mathcal{G} , the *dual* of G with respect to \mathcal{G} is the graph G^* whose vertices are in one-to-one correspondence with the faces of \mathcal{G} and for each edge e in G there is an edge e^* in G^* connecting the two faces incident to e in \mathcal{G} . Thus, G^* models the adjacencies of faces of G with respect to the embedding \mathcal{G} . Note

*Part of the research was conducted in the framework of ESF project 10-EuroGIGA-OP-003 GraDR "Graph Drawings and Representations" and of 'EU FP7 STREP Project "Leone: From Global Measurements to Local Management", grant no. 317647'. This work was supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD). This work began during a visit of Angelini at Karlsruhe Institute of Technology.

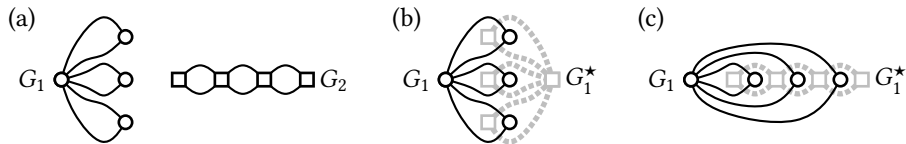


Figure 1. (a) An instance of MUTUAL DUALITY consisting of the graphs G_1 and G_2 . (b) The dual G_1^* of G_1 with respect to this embedding is not isomorphic to G_2 . (c) Embedding of G_1 such that G_1^* is isomorphic to G_2 (thus G_1 and G_2 form a yes-instance).

that this naturally induces a planar embedding \mathcal{G}^* on G^* and that the dual of G^* with respect to \mathcal{G}^* coincides with G .

We consider the following problem, that we call MUTUAL DUALITY. Given two planar graphs G_1 and G_2 , is there an embedding \mathcal{G}_1 of G_1 such that the dual G_1^* of G_1 with respect to \mathcal{G}_1 is isomorphic to G_2 ? See Figure 1 for an example.

We observe that the MUTUAL DUALITY problem is a generalization of the well-known problem GRAPH SELF-DUALITY [12], namely the problem of testing whether a graph G is *graph self-dual*, that is, whether it admits an embedding such that its dual G^* is isomorphic to G itself. Self-duality has been defined also in the stronger form of MAP SELF-DUALITY [13], in which one is required to find an embedding \mathcal{G} of the input graph G such that there exists an isomorphism from G to its dual graph G^* that preserves the embedding \mathcal{G} . Note that, since triconnected planar graphs have a unique planar embedding, the two problems are equivalent for this class of graphs.

While the computational complexity of both problems is still open, several results have been obtained from the point of view of devising techniques to construct graphs with the required properties. Servatius and Servatius [13] show the existence of biconnected planar graphs that are graph self-dual but not map self-dual. Further, Servatius and Christopher [12] show how to construct self-dual graphs from given planar graphs. Finally, Archdeacon and Richter [2] give a set of constructions for triconnected self-dual graphs and show that every such graph can be constructed in this way.

All graphs we consider are implicitly allowed to have loops and multiple edges. Graphs without loops and multiple edges are called *simple*. If G_1 is triconnected, it has a fixed planar embedding [15] and thus MUTUAL DUALITY reduces to testing graph isomorphism for planar graphs, which can be solved in linear time due to Hopcroft and Wong [9]. Note that biconnectivity and triconnectivity of a planar graph are invariant under dualization [14] (assuming that biconnected graphs have no loops and triconnected graphs are simple).

For non-triconnected planar graphs, however, MUTUAL DUALITY is more complicated, since modifying the embedding of G_1 influences the structure of its dual graph. In fact, we show in Theorem 1 that MUTUAL DUALITY is NP-complete in general. We extend this result in Theorem 2 to prove NP-completeness even for GRAPH SELF-DUALITY and MAP SELF-DUALITY. We remark that all our proofs work even for simple graphs.

On the other hand, in Theorem 7 we provide a linear-time algorithm solving MUTUAL DUALITY for biconnected planar graphs. This implies a linear-time algorithm for GRAPH

SELF-DUALITY of biconnected planar graphs (Corollary 2), as it is a special case of our problem. The algorithm is based on the definition of a new data structure, which we call *dual SPQR-tree* in analogy with the SPQR-tree data structure [5, 6]. Indeed, as SPQR-trees allow to compactly represent and efficiently handle all planar embeddings of a biconnected planar graph, the dual SPQR-trees, together with a newly-defined set of operations, allow to compactly represent and efficiently handle all dual graphs of a biconnected planar graph.

Apart from the main goal of solving the MUTUAL DUALITY problem efficiently, this new data structure gave us the possibility of deriving more general properties for the set of dual graphs of a biconnected planar graph. Namely, consider the *common dual relation* \sim , where $G_1 \sim G_2$ if and only if they have a common dual graph. We show that \sim is not transitive on the set of connected planar graphs. However we show that, due to the properties of the dual SPQR-trees, \sim is an equivalence relation on the set of biconnected planar graphs. In particular, the graphs represented by a dual SPQR-tree form an equivalence class. Thus, testing MUTUAL DUALITY reduces to testing whether two dual SPQR-trees represent the same equivalence class.

The study of the common dual relation allows us to obtain new results on another widely-studied concept in Graph Theory. The *graphic matroid* of a graph $G = (V, E)$ is the matroid (E, \mathcal{I}) where \mathcal{I} consists of the edge sets $E' \subseteq E$ that do not contain a cycle; see e.g. [16]. It is not hard to see that two biconnected planar graphs are related via the common dual relation if and only if they have the same graphic matroid (which does not hold for general planar graphs). With this insight, one can use the one-to-many reduction from graphic matroid isomorphism testing to graph isomorphism testing by Rao and Sarma [11] to solve MUTUAL DUALITY for biconnected planar graphs in polynomial time. In this paper, however, we give a one-to-one reduction leading to a linear-time algorithm. In 1933, Whitney [17] defined two graphs to be *2-isomorphic* if and only if their graphic matroids are isomorphic. Since on biconnected graphs the notions coincide, our algorithm implies a linear-time testing algorithm for 2-isomorphism of biconnected planar graphs (Corollary 3).

We believe that the new dual SPQR-tree data structure can be used to efficiently solve other related problems, especially those in which the final goal is to find an embedding of the input graph that optimizes certain criteria, such as for example minimizing the face sizes of the embedding [4, 10]. Indeed, such criteria may often be described in terms of features of the corresponding dual graph (in the example it would be the vertex degrees), and the dual SPQR-tree might be used to optimize such features. Another motivating example is given by the problem of finding an embedding that minimizes the *depth* of a planar graph, that is the maximum distance between two faces. Here, such a distance is computed by considering two faces to be adjacent if they share an edge. On the other hand, in the dual version of the problem the goal becomes to construct a dual in which the maximum distance between two vertices (namely the *diameter* of the graph) is minimized. Both the algorithms to find minimum-depth embeddings by Bienstock and Monma [3], and by Angelini et al. [1] tackle the problem in its primal version making use of SPQR-trees (or analogous data structures). However, at each step of their algorithms, they compute the dual graph of the subgraph that is currently considered, and perform most of the

computations on it. This is due to the fact that dealing with the graph-theoretic distance between vertices is easier and more natural than dealing with the edge-sharing distance between faces. Hence, tackling the problem in its dual version using dual SPQR-trees would have yielded a more direct solution.

Contribution and Outline. We start with some preliminary definitions in Section 2. Then, in Section 3 we show that MUTUAL DUALITY is NP-complete, even if both input graphs are required to be simple. The proof can be extended to show that MAP SELF-DUALITY and GRAPH SELF-DUALITY are NP-complete in general.

To solve MUTUAL DUALITY efficiently for biconnected graphs, we first describe decomposition trees as a generalization of SPQR-trees in Section 4. In Section 5 we describe the dual SPQR-tree and show that it compactly represents all dual graphs of a biconnected planar graph. We consider the common dual relation in Section 6 and give a counterexample showing that \sim is not transitive on the set of connected planar graphs. On the other hand, we show that the properties of the dual SPQR-tree data structure imply that \sim is an equivalence relation on the set of biconnected planar graphs. Hence, solving MUTUAL DUALITY is equivalent to testing whether two dual SPQR-trees represent the same equivalence class. In Section 7 we show that this can be further reduced to testing graph isomorphism of two planar graphs, which leads to a linear-time algorithm for MUTUAL DUALITY of biconnected planar graphs; from this result, we derive an efficient solution to GRAPH SELF-DUALITY and 2-isomorphism for the same class of graphs.

2 Preliminaries

A *drawing* of a graph is a mapping of each vertex to a distinct point of the plane and of each edge to a curve between its endpoints. A *planar drawing* is such that no two edges intersect except, possibly, at common endpoints. A planar drawing of a graph determines a circular ordering of the edges incident to each vertex. Two drawings of the same graph are *equivalent* if they determine the same circular ordering around each vertex. A *planar embedding* is an equivalence class of planar drawings. A planar drawing partitions the plane into topologically connected regions, called *faces*. The unbounded face is the *outer face*.

A graph is *connected* if there exists a path between any pair of vertices. A *separating k -set* is a set of k vertices whose removal disconnects the graph. Separating 1-sets and 2-sets are *cutvertices* and *separation pairs*, respectively. A connected graph is *biconnected* if it does not have a cutvertex or a loop. A biconnected graph is *triconnected* if it does not have a separation pair or multiple edges. The maximal biconnected components of a graph are called *blocks*.

Let G be a planar graph with planar embedding \mathcal{G} and let F be the set of faces of \mathcal{G} . The *dual* of G with respect to \mathcal{G} is the graph $G^* = (F, E^*)$ with $E^* = \{e^* \mid e \in E\}$, where the *dual edge* e^* of e connects the two faces incident to e in \mathcal{G} .

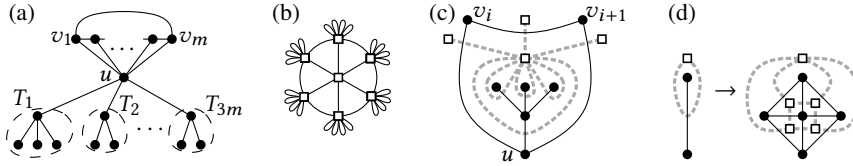


Figure 2. The graphs G_1 (a) and G_2 (b) of the reduction from 3-PARTITION. (c) Embedding a tree T_i inside a face f creates a_i loops at the corresponding dual vertex. (d) Bridges and corresponding loops can be replaced by small graphs.

3 Complexity

In this section we first show that MUTUAL DUALITY is NP-complete, by means of a reduction from 3-PARTITION. Then we show that the resulting instances of MUTUAL DUALITY can be further reduced to equivalent instances of MAP and GRAPH SELF-DUALITY.

An instance (A, B) of 3-PARTITION consists of a positive integer B and a set $A = \{a_1, \dots, a_{3m}\}$ of $3m$ integers, with $B/4 < a_i < B/2$ for $i = 1, \dots, 3m$. The question is whether A admits a partition into a set \mathcal{A} of triples such that for each triple $X \in \mathcal{A}$ we have $\sum_{x \in X} x = B$. The problem 3-PARTITION is strongly NP-hard [7], i.e., it remains NP-hard even if B is bounded by a polynomial in m .

Theorem 1. MUTUAL DUALITY is NP-complete, even for simple graphs.

Proof. Clearly, MUTUAL DUALITY is in NP, as we can guess an embedding for G_1 and then check in polynomial time whether the corresponding dual is isomorphic to G_2 .

To show NP-hardness we give a reduction from 3-PARTITION. We first give a construction containing loops, afterwards we show how to get rid of them. Let (A, B) be an instance of 3-PARTITION with $|A| = 3m$. The graph G_1 contains an m -wheel, i.e., a cycle v_1, \dots, v_m of m vertices together with a center u connected to each v_i . Additionally, for each element $a_i \in A$ we create a star T_i with $a_i - 1$ leaves and connect its center to u ; see Figure 2a. The graph G_2 is an m -wheel along with B loops at every vertex, except for the center; see Figure 2b. We claim that G_1 and G_2 form a YES-instance of MUTUAL DUALITY if and only if (A, B) is a YES-instance of 3-PARTITION.

Suppose that there exists a partition \mathcal{A} of A . The embedding of the m -wheel in G_1 is fixed and it has exactly m faces incident to its center u . Hence, the only degree of freedom is that of deciding the embedding of the trees T_i into these m faces. We perform this operation as follows. For each triple $X = \{a_i, a_j, a_k\} \in \mathcal{A}$ we pick a distinct face and embed T_i, T_j and T_k into it. Call the resulting embedding \mathcal{G}_1 and consider the dual G_1^* with respect to \mathcal{G}_1 . The m -wheel of G_1 determines an m -wheel in G_1^* . Consider a tree T_i that is embedded in a face f . Since T_i contains a_i bridges, which are all embedded in f , the corresponding vertex of G_1^* has a_i loops; see Figure 2c. Due to the construction, each face contains exactly three trees with a total of B bridges. Thus, G_1^* is isomorphic to G_2 .

Conversely, assume that we have an embedding \mathcal{G}_1 such that the dual G_1^* of G_1 with respect to \mathcal{G}_1 is isomorphic to G_2 . Again, the m -wheel in G_1 forms m faces incident

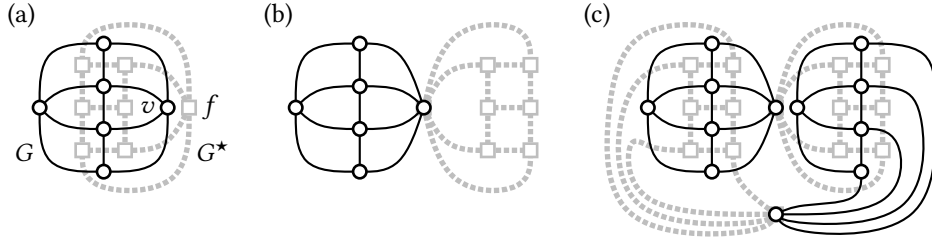


Figure 3. (a) A graph G and its dual G^* (gray) with vertex v and incident face f . (b) The adhesion of G and G^* with respect to v and f . (c) Illustration of the self-duality of the adhesion.

to u , and since G_1^* is isomorphic to G_2 , the trees must be embedded such that each face contains exactly B bridges. Since embedding T_i inside a face f places a_i bridges into f and since $B/4 < a_i < B/2$, each face contains exactly three trees. Thus, the set of triples determined by trees that are embedded in the same face form a solution to 3-PARTITION.

Clearly, the transformation can be computed in polynomial time if B is bounded by a polynomial in m , and thus MUTUAL DUALITY is NP-hard. Moreover, graph G_2 can be made simple (G_1 is already simple) by replacing each bridge in G_1 and each loop in G_2 with a 4-wheel as shown in Figure 2d. The resulting graphs G'_1 and G'_2 are obviously dual to each other if G_1 and G_2 are dual to each other. Conversely, assuming that $m \geq 5$, we can undo the transformation by replacing every 4-wheel in G'_1 with a bridge and every 4-wheel in G'_2 with a loop to obtain mutually dual embeddings of G_1 and G_2 . Moreover, G'_1 and G'_2 are simple, which concludes the proof. \square

In the following we show how the above construction can be used to show NP-completeness for MAP and GRAPH SELF-DUALITY. To this end, we use the *adhesion* operation introduced by Servatius and Christopher [12]. Let v be a vertex of G incident to a face f . Then the adhesion of G and its dual G^* (with respect to v and f) is obtained by identifying v in G and f^* in G^* with each other; see Figure 3a and 3b. Servatius and Christopher [12] show that the adhesion of an embedded planar graph and its dual is graph self-dual; see Figure 3c. Moreover, although not explicitly mentioned, they show that this adhesion is even map self-dual.

To show the following theorem we essentially transform the instance of MUTUAL DUALITY consisting of the two graphs G_1 and G_2 described in the proof of Theorem 1 into an equivalent instance of MAP and GRAPH SELF-DUALITY by performing the adhesion of G_1 and G_2 .

Theorem 2. GRAPH SELF-DUALITY and MAP SELF-DUALITY are NP-complete, even for simple graphs.

Proof. Clearly, GRAPH SELF-DUALITY (MAP SELF-DUALITY) is in NP as we can guess an embedding for G together with a bijection between the vertices of G and the vertices of G^* and then check in polynomial time whether this bijection is an isomorphism (that preserves the embedding).

Let G_1 and G_2 be two simple graphs that form an instance of MUTUAL DUALITY obtained from an instance of 3-PARTITION as described in the proof of Theorem 1. Let G be the graph obtained from G_1 (Figure 2a) and G_2 (Figure 2b) by identifying a vertex that is not the center of the m -wheel in G_2 with the vertex u in G_1 . By construction G is simple. In the following we consider G as an instance of GRAPH SELF-DUALITY and MAP SELF-DUALITY. We claim the following.

Claim 1 If G_1 and G_2 form a YES-instance of MUTUAL DUALITY, then G is a YES-instance of MAP SELF-DUALITY.

Claim 2 If G is a YES-instance of MAP SELF-DUALITY, then it is a YES-instance of GRAPH SELF-DUALITY.

Claim 3 If G is a YES-instance of GRAPH SELF-DUALITY, then G_1 and G_2 form a YES-instance of MUTUAL DUALITY.

The three claims together show that the instance G_1 and G_2 of MUTUAL DUALITY, the instance G of GRAPH SELF-DUALITY and the instance G of MAP SELF-DUALITY are equivalent.

To prove Claim 1, assume that G_1 and G_2 form a YES-instance of MUTUAL DUALITY, that is G_1 and G_2 admit embeddings such that they are dual to each other. As the vertex u is incident to all faces in G_1 except for the face forming the center of the m -wheel in G_1^* , it is in particular incident to the face dual to the vertex in G_2 chosen for the adhesion. Thus, it follows from the results by Servatius and Christopher [12] that the adhesion G of G_1 and G_2 is map self-dual.

Claim 2 is trivial, since being map self-dual is a stricter requirement than being graph self-dual.

It remains to show Claim 3. Let G^* be the dual graph of G with respect to a fixed embedding and let $\varphi: V(G) \rightarrow V(G^*)$ be a graph isomorphism between G and G^* . As G is the adhesion of G_1 and G_2 , there is a unique vertex v in G belonging to G_1 and G_2 , and a unique face f incident to both graphs G_1 and G_2 . Since v was chosen to be u in G_1 , it is the only vertex in G that is a cutvertex and the center of an m -wheel. Moreover, f is the only cutvertex in G^* that can be the center of an m -wheel. Thus, φ has to map v to f . The blocks incident to v are a block with degree 3 at v stemming from G_2 , B loops also stemming from G_2 , a block consisting of an m -wheel with center v stemming from G_1 , and $3m$ attached trees stemming from G_1 . Similarly, the vertex f in G^* is incident to a block having degree 3 at f contained in G_1^* , a set of loops stemming from the trees in G_1 (the number of loops depends on the embedding), an m -wheel with center f contained in G_2^* , and a set of bridges stemming from the loops at G_2 . Thus, φ has to map all vertices in G_1 to vertices in G_2^* and all vertices in G_2 to vertices in G_1^* . This directly shows that G_1 and G_2 form a YES-instance of MUTUAL DUALITY, which concludes the proof. \square

4 Decomposition Trees and the SPQR-Tree Data Structure

In the following we consider *decomposition trees* of biconnected planar graphs, containing the *SPQR-trees* introduced by Di Battista and Tamassia [5,6] as a special case. Let G be a planar biconnected graph and let $\{s, t\}$ be a *split pair*, that is either a separation pair

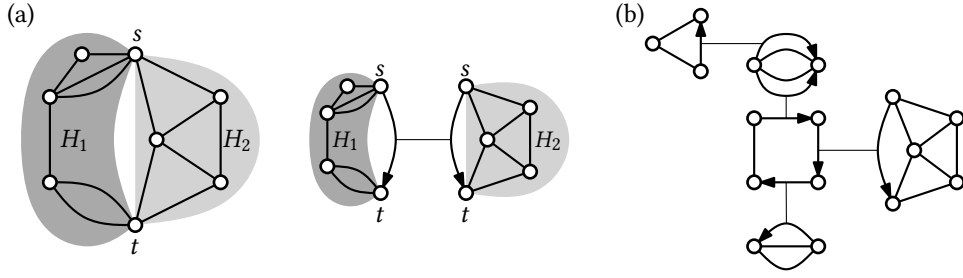


Figure 4. (a) Illustration of a decomposition step. (b) The SPQR-tree of the graph in a (to improve readability, the Q-nodes are omitted).

or a pair of adjacent vertices. Let further H_1 and H_2 be two subgraphs of G such that $H_1 \cup H_2 = G$ and $H_1 \cap H_2 = \{s, t\}$; see Figure 4a. Consider the tree \mathcal{T} consisting of two nodes μ_1 and μ_2 associated with the graphs $H_1 + (s, t)$ and $H_2 + (s, t)$, respectively. For each node μ_i , the graph $H_i + (s, t)$ associated with it is the *skeleton* of μ_i , denoted by $\text{skel}(\mu_i)$, and the special directed edge (s, t) is called *virtual edge*. The edge connecting the nodes μ_1 and μ_2 in \mathcal{T} associates the virtual edge $\varepsilon_1 = (s, t)$ in $\text{skel}(\mu_1)$ with the virtual edge $\varepsilon_2 = (s, t)$ in $\text{skel}(\mu_2)$; we say that ε_1 is the *twin* of ε_2 and vice versa. Moreover, we say that ε_1 in $\text{skel}(\mu_1)$ *corresponds* to the neighbor μ_2 of μ_1 . This can be expressed as a bijection corr_μ mapping the virtual edges of $\text{skel}(\mu)$ to the neighboring vertices of μ in \mathcal{T} . In the example above we have $\text{corr}(\varepsilon_1) = \mu_2$ and $\text{corr}(\varepsilon_2) = \mu_1$ (the subscript of corr is omitted as it is clear from the context).

The above-described procedure is called *decomposition* and can be applied further to the skeletons of the nodes of \mathcal{T} , leading to a larger tree with smaller skeletons. The decomposition can be undone by *contracting* an edge in \mathcal{T} . Let $\{\mu, \mu'\}$ be an edge in \mathcal{T} and let ε be the virtual edge in $\text{skel}(\mu)$ with $\text{corr}(\varepsilon) = \mu'$ having ε' in $\text{skel}(\mu')$ as twin. The contraction of $\{\mu, \mu'\}$ collapses μ and μ' into a single node whose skeleton is as follows. The skeletons of μ and μ' are *glued* together at the twins ε and ε' according to their orientation, that is the source and target of ε is identified with the source and target of ε' , respectively. The resulting virtual edge is removed. Iteratively applying the contraction in \mathcal{T} leads to a tree consisting of a single node μ , whose skeleton is independent of the contraction order. The graph *represented* by \mathcal{T} is $\text{skel}(\mu)$.

A *reversed decomposition tree* is defined as a decomposition tree with the only difference that in the decomposition step one of the two twin edges is reversed and in the contraction step they are glued together such that they point in opposite directions. Note that a reversed decomposition tree can be transformed into an equivalent normal decomposition tree representing the same graph by reversing one virtual edge in each pair of twin edges.

A decomposition tree is an SPQR-tree (Figure 4b) if each inner node is either an S-, a P-, or an R-node whose skeletons contain only virtual edges forming a cycle, a bunch of at least three parallel edges, or a triconnected planar graph, respectively, such that no two S-nodes and no two P-nodes are adjacent. Moreover, each leaf is a Q-node whose skeleton consists of two vertices connected by one virtual and one normal edge. The *reversed SPQR-tree* is defined analogously as a special case of the reversed decomposition

tree. The SPQR-tree of a biconnected planar graph is unique up to reversals of pairs of virtual edges that are twins. We assume without loss of generality that all virtual edges in the skeleton of each P-node are oriented in the same direction and those in the skeleton of each S-node form a directed cycle.

The SPQR-tree \mathcal{T} of a biconnected planar graph G represents all planar embeddings of G , as there is a bijection between these embeddings and the set of all combinations of embeddings of the skeletons in \mathcal{T} . The embedding choices for the skeletons consist of reordering the parallel edges in a P-node and choosing one of the two possible planar embeddings of the skeleton in an R-node. Since for an R-node skeleton one planar embedding can be obtained from the other by reversing all edge orders around vertices, changing the embedding of an R-node skeleton is also referred to as *flipping*. The SPQR-tree of a biconnected planar graph can be computed in linear time [8]. Fixing the embeddings of the skeletons in an arbitrary decomposition tree \mathcal{T} also determines a planar embedding of the represented graph G . However, there may be planar embeddings that are not represented by \mathcal{T} .

We assume the skeletons of the SPQR-tree of a graph to be embedded graphs if and only if the graph itself is embedded.

5 Compact Representation of all Duals of a Biconnected Graph

Let G be a biconnected graph with SPQR-tree \mathcal{T} and planar embedding \mathcal{G} . In the following we study how a change in the embedding of G reflects in a change of the dual graph G^* . To this end, we do not consider the graphs themselves but their SPQR-trees. We first show how the SPQR-tree of the dual graph G^* can be directly obtained from the SPQR-tree of the primal graph G . This can then be used to understand the effects on G^* caused by changing the embedding of a skeleton in \mathcal{T} .

We first define the *dual decomposition tree* \mathcal{T}^* of a decomposition tree \mathcal{T} representing G (with respect to a fixed embedding \mathcal{G} of G that can be represented by \mathcal{T}). We will later show that the dual decomposition tree represents the dual graph G^* of G . Essentially, \mathcal{T}^* is obtained from \mathcal{T} by replacing each skeleton with its directed dual and interpreting the resulting tree as a reversed decomposition tree. More precisely, for each node μ in \mathcal{T} , the dual decomposition tree \mathcal{T}^* contains a *dual node* μ^* having the dual of $\text{skel}(\mu)$ as skeleton. An edge ε^* in $\text{skel}(\mu^*)$ dual to a virtual edge ε in $\text{skel}(\mu)$ is again virtual and oriented from right to left with respect to the orientation of ε . Similarly, an edge dual to a normal edge is also a normal edge in the dual skeleton. Two virtual edges in \mathcal{T}^* are twins if and only if their primal edges are twins. This has the effect that $\text{corr}(\varepsilon)^* = \text{corr}(\varepsilon^*)$ holds. Obviously, the tree structures of \mathcal{T}^* and \mathcal{T} are isomorphic with respect to the map assigning each node in \mathcal{T} to its dual node in \mathcal{T}^* . In case \mathcal{T} is the SPQR-tree of G , we obtain the following. The dual of a triconnected skeleton is triconnected, the dual of a (directed) cycle is a bunch of parallel edges (all directed in the same direction), and the dual of a normal edge with a parallel virtual edge is a normal edge with a parallel virtual edge. Thus, if a node μ in \mathcal{T} is an S-, P-, Q-, or R-node, its dual node μ^* in \mathcal{T}^* is

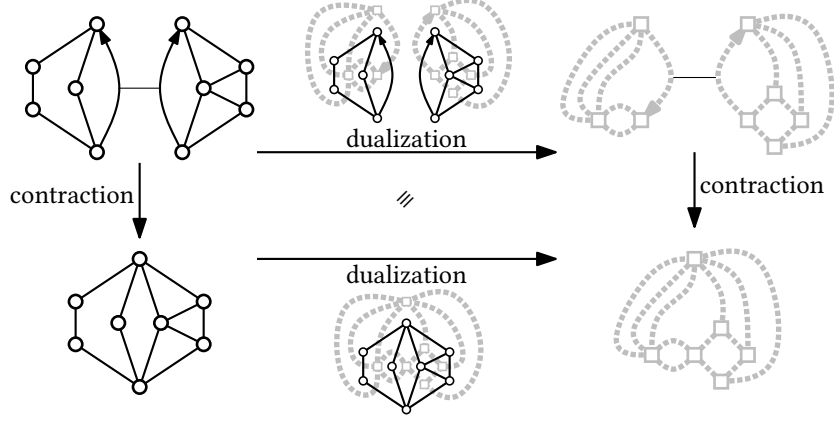


Figure 5. An example illustrating that contracting edges in a decomposition tree and taking the dual decomposition tree commute.

a P-, S-, Q-, or R-node, respectively. This in particular shows that the dual SPQR-tree is again an SPQR-tree and not just an arbitrary decomposition tree.

Lemma 1. *Let G be a biconnected planar graph with SPQR-tree \mathcal{T} and planar embedding \mathcal{G} . The dual SPQR-tree \mathcal{T}^* with respect to \mathcal{G} is the reversed SPQR-tree of the dual G^* .*

Proof. We show a slightly more general result by dealing with arbitrary decomposition trees instead of SPQR-trees. We show that first contracting an edge $\{\mu, \mu'\}$ in a decomposition tree \mathcal{T} into a node $\mu\mu'$ and then taking the dual decomposition tree is equivalent to first taking the dual decomposition tree \mathcal{T}^* and then contracting the edge $\{\mu^*, \mu'^*\}$ into $\mu^*\mu'^*$ (recall that \mathcal{T}^* is interpreted as reversed decomposition tree, thus the gluing operation contained in the contraction of $\{\mu^*, \mu'^*\}$ is reversed); see Figure 5 for an example. Applying this operation iteratively until the trees \mathcal{T} and \mathcal{T}^* consist of single nodes directly shows that the reversed decomposition tree \mathcal{T}^* represents the graph G^* dual to the graph G represented by \mathcal{T} .

Let ε and ε' be the virtual edges in $\text{skel}(\mu)$ and $\text{skel}(\mu')$ corresponding to the edge $\{\mu, \mu'\}$ in \mathcal{T} . Let further f_ℓ and f_r , and f'_ℓ and f'_r be the faces left and right of ε and of ε' with respect to the orientation of ε and ε' , respectively; see Figure 6a and b. We denote the graph $\text{skel}(\mu) - \varepsilon$ by H and the graph $\text{skel}(\mu^*) - \varepsilon^*$ by H^* (note that H^* is not really the dual graph of H). The graphs H' and H'^* are defined analogously. When contracting $\{\mu, \mu'\}$, the virtual edges ε and ε' are glued together, that is u and v are identified with u' and v' , respectively; see Figure 6c. Consider now the dual of the resulting graph. This dual graph contains the graphs H^* and H'^* . Since H and H' share a pair of faces, say f_r and f'_ℓ (the outer face in Figure 6c), these two faces are identified into a single vertex of the dual. The other two faces incident to ε and ε' are connected by the edge dual to (u, v) . Finally, removing the edge (u, v) contracts f_ℓ and f'_r into a single vertex of the dual; see Figure 6d. Thus the dual graph $\text{skel}(\mu\mu')^*$ of the resulting skeleton $\text{skel}(\mu\mu')$ can be obtained from $\text{skel}(\mu)^*$ and $\text{skel}(\mu')^*$ by removing the virtual edges ε^* and ε'^* and identifying their endpoints with each other, reversing their orientation. As this is

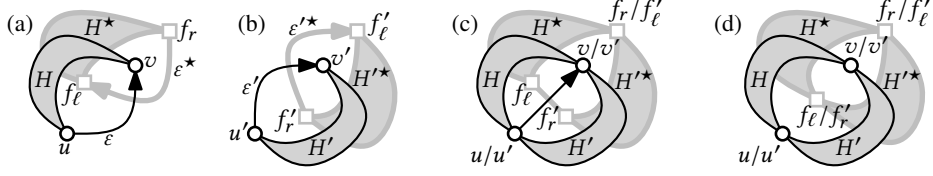


Figure 6. (a)–(c) Gluing together the virtual edges ε and ε' . (d) Removing the resulting edge.

equal to contracting $\{\mu^*, \mu'^*\}$ in \mathcal{T}^* , we have $\text{skel}(\mu\mu')^* = \text{skel}(\mu^*\mu'^*)$, which concludes the proof. \square

In the following we consider how the dual SPQR-tree changes when the embedding of the skeletons in the SPQR-tree change. Flipping the skeleton of an R-node and reordering the virtual edges in a P-node give rise to the following two operations: *reversal* of R-nodes and *restacking* of S-nodes.

A reversal applied to an R-node μ reverses the direction of all virtual edges in $\text{skel}(\mu)$. As no other skeleton is changed by this operation, this only affects how $\text{skel}(\mu)$ is glued to the skeletons of its adjacent nodes.

Let μ be an S-node with virtual edges $\varepsilon_1, \dots, \varepsilon_k$. A restacking of μ picks an arbitrary ordering of $\varepsilon_1, \dots, \varepsilon_k$ and glues their end-points such that they create a directed cycle C in that order. Then, $\text{skel}(\mu)$ is replaced by C .

Lemma 2. *Let \mathcal{T} and \mathcal{T}^* be the SPQR-trees of an embedded biconnected planar graph and of its dual, respectively. Flipping an R-node and reordering a P-node in \mathcal{T} corresponds to reversing its dual R-node and restacking its dual S-node, respectively, in \mathcal{T}^* .*

Proof. Due to Lemma 1 we can work with the dual SPQR-tree instead of the SPQR-tree of the dual. Obviously, flipping an R-node μ in \mathcal{T} exchanges left and right in $\text{skel}(\mu)$ and thus reverses the orientation of each virtual edge in $\text{skel}(\mu^*)$, where μ^* is the node in \mathcal{T}^* dual to μ . Thus, flipping μ corresponds to a reversal of μ^* . Similarly, reordering the virtual edges in the skeleton of a P-node μ has the effect that the virtual edges in its dual S-node μ^* are restacked, yielding a different cycle. Note that this cycle is again directed, since the virtual edges in μ are still all oriented in the same direction. \square

This lemma shows that the SPQR-tree of the dual graph with respect to a fixed embedding can be used to represent the dual graphs with respect to all possible planar embeddings by allowing reversal and restacking operations. We say that an SPQR-tree *represents a dual graph* if it can be obtained by applying reversal and restacking operations. The following theorem directly follows.

Theorem 3. *The dual SPQR-tree of a biconnected planar graph G represents all and only the dual graphs of G .*

When we are not interested in the embedding of the dual graph but only in its structure (as a graph), we may also allow the usual SPQR-tree operations, that is flipping R-nodes

and reordering the virtual edges in P-nodes. Note that the reversal operation applied to P-nodes only changes the embedding of the graph and not its structure. Moreover, reversing a Q-node does not change anything, while the reversal of an S-node can be seen as a special way of restacking it. This observation can be used to show the following lemma.

Lemma 3. *Let G be a biconnected planar graph with embedding \mathcal{G} and let G^* be its dual graph with SPQR-tree \mathcal{T}^* . Let $\mathcal{T}_\varepsilon^*$ be the SPQR-tree obtained from \mathcal{T}^* by reversing the orientation of the virtual edge ε in \mathcal{T}^* , and let G_ε^* be the graph it represents. Then there exists an embedding \mathcal{G}_ε of G such that G_ε^* is the dual graph of G with respect to \mathcal{G}_ε .*

Proof. Let μ be the node in \mathcal{T}^* containing the virtual edge ε and let $\text{corr}(\varepsilon) = \mu'$ be the neighbor of μ corresponding to ε . Removing the edge $\{\mu, \mu'\}$ splits \mathcal{T}^* into two subtrees \mathcal{T}_μ^* and $\mathcal{T}_{\mu'}^*$. We claim that the reversal of all nodes in one of these subtrees (no matter which one) yields an SPQR-tree $\mathcal{T}_{\mu\mu'}^*$ representing G_ε^* . Then, it follows from Lemma 2 and from the observation above, that G_ε^* is a dual graph of G .

It remains to show the claim. As it does not matter whether the orientation of ε or of its twin in μ' is changed, we can assume without loss of generality that all nodes in \mathcal{T}_μ^* are reversed in $\mathcal{T}_{\mu\mu'}^*$. The graph represented by $\mathcal{T}_{\mu\mu'}^*$ can be obtained by contracting the edges in an arbitrary order. Contracting edges in the subtree $\mathcal{T}_{\mu'}^*$ has the same effect as in the original graph, since $\mathcal{T}_{\mu'}^*$ was not changed. Similarly, contracting an edge in \mathcal{T}_μ^* also has the same effect as the orientation of both corresponding virtual edges is reversed. Finally, when contracting the edge $\{\mu, \mu'\}$ the skeletons are glued together oppositely, as ε is reversed whereas its twin remains the same. Thus, reversing all nodes in \mathcal{T}_μ^* is equivalent to reversing the orientation of ε , which concludes the proof. \square

Lemma 2 and Lemma 3 together yield the following theorem.

Theorem 4. *For two SPQR-trees \mathcal{T}_1 and \mathcal{T}_2 , the following three statements are equivalent.*

1. \mathcal{T}_1 and \mathcal{T}_2 represent the same set of dual graphs.
2. \mathcal{T}_1 and \mathcal{T}_2 can be transformed into each other by reversal and restacking operations.
3. \mathcal{T}_1 and \mathcal{T}_2 can be transformed into each other by choosing orientations for the virtual edges and by restacking S-node skeletons.

6 Equivalence Relation

In this section we study the relation \sim on the set of planar graphs, defined as follows. Two graphs G_1 and G_2 are related, i.e., $G_1 \sim G_2$, if and only if G_1 and G_2 can be embedded such that they have the same dual graph $G_1^* = G_2^*$. We call \sim the *common dual relation*.

Theorem 5. *The common dual relation \sim is an equivalence relation on the set of biconnected planar graphs. For a biconnected planar graph G , the set of dual graphs of G is an equivalence class with respect to \sim .*

Proof. Clearly, \sim is symmetric and reflexive. For the transitivity, let G_1, G_2 , and G_3 be three biconnected planar graphs such that $G_1 \sim G_2$ and $G_2 \sim G_3$. Let further $\mathcal{T}_1^*, \mathcal{T}_2^*$,

and \mathcal{T}_3^* be the dual SPQR-trees representing all duals of G_1 , G_2 , and G_3 , respectively. Due to $G_1 \sim G_2$, there exists a graph G that is represented by \mathcal{T}_1^* and \mathcal{T}_2^* . Thus, \mathcal{T}_1^* and \mathcal{T}_2^* can both be transformed into the SPQR-tree representing G using reversal and restacking operations, which shows that they represent the same set of duals (Theorem 4). The same argument shows that G_2 and G_3 have the same set of dual graphs, due to $G_2 \sim G_3$. Thus, also G_1 and G_3 have exactly the same set of dual graphs, which yields $G_1 \sim G_3$.

For the second statement, let C^* be the set of dual graphs of G . Clearly, for $G_1^*, G_2^* \in C^*$ the graph G is a common dual, thus $G_1^* \sim G_2^*$. On the other hand, let $G_1^* \in C^*$ and $G_1^* \sim G_2^*$. By the above argument, G_1^* and G_2^* have the same set of dual graphs. Thus G is a dual graph of G_2^* , yielding $G_2^* \in C^*$. \square

Theorem 5 shows that the equivalence class C of a biconnected planar graph G with respect to the common dual relation is exactly the set of dual graphs that is represented by the SPQR-tree \mathcal{T} of G . The dual SPQR-tree \mathcal{T}^* of G also represents a set of dual graphs forming the equivalence class C^* . We say that C^* is the *dual equivalence class* of C . Given an arbitrary graph $G \in C$ and an arbitrary graph $G^* \in C^*$, graphs G and G^* can be embedded such that they are dual to each other, since C^* contains exactly the graphs that are dual to G . The problems MUTUAL DUALITY and GRAPH SELF-DUALITY can be reformulated in terms of the equivalence classes of the common dual relation. Two biconnected planar graphs are a YES-instance of MUTUAL DUALITY if and only if their equivalence classes are dual to each other. A biconnected planar graph is graph self-dual if and only if its equivalence class is dual to itself. This in particular means that either every or no graph in an equivalence class is graph self-dual.

Although it might seem quite natural that the common dual relation is an equivalence relation, this is not true for general planar graphs. This fact is stated in the following theorem.

Theorem 6. *The common dual relation \sim is not transitive on the set of planar graphs.*

Proof. Consider the graph G_1 depicted in Figure 7a, consisting of a triconnected planar graph with an additional loop attached to a vertex. Hence, its dual graph is a triconnected component with a bridge attached to it. Then, consider the graph G_2 in Figure 7b, consisting of the same triconnected planar graph with an additional loop attached to a different vertex. Note that, in both graphs G_1 and G_2 the loop can be embedded into the same face of the triconnected component, yielding the same dual graph (with a different embedding). Thus, G_1 and G_2 have a common dual, i.e., $G_1 \sim G_2$. Using the same argument it is possible to observe that G_2 (with respect to the embedding in Figure 7c) and G_3 (Figure 7d) have a common dual graph, i.e., $G_2 \sim G_3$. However, G_1 and G_3 do not have a common dual, for the following reason. Let v_1 and v_3 be the vertices in G_1 and G_3 incident to the loop, respectively. The only embedding choice in G_1 and G_3 is to embed the loop into one of the faces incident to v_1 and v_3 , respectively. In the dual graphs this has the effect that the bridge is attached to the corresponding vertex. Since all faces incident to v_1 have degree 3 and all faces incident to v_3 have degree 4, the resulting dual graphs cannot be isomorphic. Thus, $G_1 \not\sim G_3$, even though $G_1 \sim G_2$ and $G_2 \sim G_3$. \square

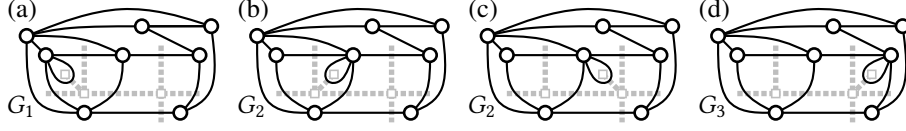


Figure 7. Illustration of Theorem 6

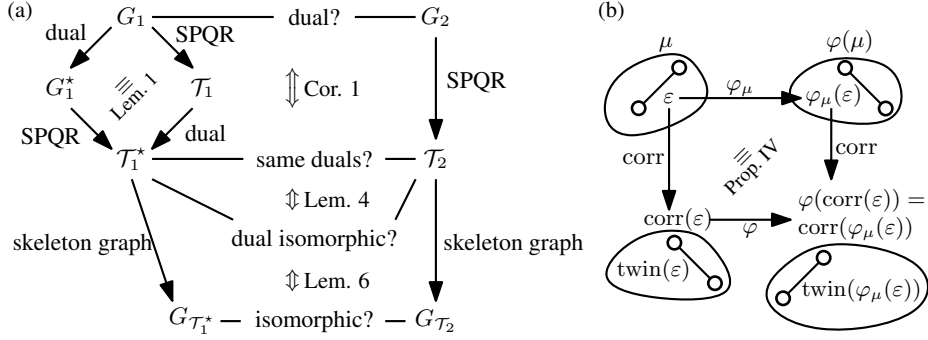


Figure 8. (a) Overview of our strategy. (b) Commutative diagram illustrating Property IV.

The reversal and restacking operations we defined for biconnected graphs in the previous section are equivalent to the operations that do not change the graphic matroid of a biconnected graph [17]. Thus, the equivalence classes of the common dual relation on biconnected planar graphs are in one-to-one correspondence with the matroids of these graphs. General planar graphs that are related with respect to the common dual relation also have the same graphic matroid. However, there are graphs with the same graphic matroid that are not related with respect to the common dual relation, like for example the graph G_1 and G_3 from Figure 7.

7 Algorithm for Biconnected Planar Graphs

Due to Theorem 3, problem MUTUAL DUALITY can be rephrased as follows.

Corollary 1. *Let G_1 and G_2 be two biconnected planar graphs with SPQR-trees \mathcal{T}_1 and \mathcal{T}_2 , respectively. There is an embedding \mathcal{G}_1 of G_1 such that G_2 is dual to G_1 with respect to \mathcal{G}_1 if and only if \mathcal{T}_2 and the dual SPQR-tree \mathcal{T}_1^* represent the same set of dual graphs.*

In the following we show that two SPQR-trees represent the same set of dual graphs if and only if they are *dual isomorphic* (we define this in a moment). Then we show that testing the existence of such an isomorphism reduces to testing graph isomorphism for planar graphs, and hence can be done in linear time. Figure 8a sketches this strategy.

For two simple graphs G and G' with vertices $V(G)$ and $V(G')$, and edges $E(G)$ and $E(G')$, respectively, a map $\varphi: V(G) \rightarrow V(G')$ is a *graph isomorphism* if it is bijective and $\{u, v\} \in E(G)$ if and only if $\{\varphi(u), \varphi(v)\} \in E(G')$ (for directed graphs, the direction

of the edges is disregarded). For non-simple graphs, we also require that for all pairs of vertices $u, v \in V$ the number of edges between u and v is the same as the number of edges between $\varphi(u)$ and $\varphi(v)$.

For simple graphs a graph isomorphism φ induces a bijection between $E(G)$ and $E(G')$, and we use $\varphi(e)$ for $e \in E(G)$ to express this bijection. As we consider the edges to be undirected, fixing $\varphi(\cdot)$ only for the edges does not determine a map for the vertices. As the SPQR-tree has more structure than a normal tree, we require some additional properties. A *dual SPQR-tree isomorphism* between two SPQR-trees \mathcal{T} and \mathcal{T}' consists of several maps. First, a map $\varphi: V(\mathcal{T}) \rightarrow V(\mathcal{T}')$ such that:

- (I) φ is a graph isomorphism between \mathcal{T} and \mathcal{T}' ; and
- (II) for each node $\mu \in V(\mathcal{T})$, the node $\varphi(\mu) \in V(\mathcal{T}')$ is of the same type.

Second, a map $\varphi_\mu: V(\text{skel}(\mu)) \rightarrow V(\text{skel}(\varphi(\mu)))$ for every R-node μ in \mathcal{T} such that:

- (III) φ_μ is a graph isomorphism between $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$; and
- (IV) $\text{corr}(\varphi_\mu(\varepsilon)) = \varphi(\text{corr}(\varepsilon))$ holds for every virtual edge ε in $\text{skel}(\mu)$.

If there is a dual SPQR-tree isomorphism between \mathcal{T} and \mathcal{T}' , then we say that \mathcal{T} and \mathcal{T}' are *dual isomorphic*. Note that by Property IV the map φ_μ is uniquely determined by φ . Observe further that Property IV (illustrated in Figure 8b) is a natural requirement and one would usually require it also for S-nodes and P-nodes. However, for P-nodes the mapping φ_μ (uniquely determined by Property IV) is always an isomorphism (Property III) as it simply permutes edges. Moreover, not requiring the existence of φ_μ for S-nodes implicitly allows restacking their skeletons. As the graph isomorphisms $\varphi_\mu(\cdot)$ do not care about the orientation of virtual edges, it is also implicitly allows to reverse them. These observations lead to the following lemma showing that this definition of dual SPQR-tree isomorphism is well suited for our purposes.

Lemma 4. *Two SPQR-trees represent the same set of dual graphs if and only if they are dual isomorphic.*

Proof. Let \mathcal{T} and \mathcal{T}' be two SPQR-trees representing the same set of dual graphs. By Theorem 4 this implies that they can be transformed into each other using reversal and restacking operations. Clearly, the identity map, mapping \mathcal{T} and each of its skeletons to itself, is a dual SPQR-tree isomorphism. It remains a dual SPQR-tree isomorphism when restacking an S-node, since Properties I and II are independent of the skeletons, and Properties III and IV are only required for R-nodes. Moreover, the reversal of an R-node preserves Properties I–IV since our definition of graph isomorphism considers edges to be undirected. It follows that \mathcal{T} and \mathcal{T}' are dual isomorphic.

For the opposite direction assume that φ together with $\varphi_{\mu_1}, \dots, \varphi_{\mu_k}$ is a dual SPQR-tree isomorphism from \mathcal{T} to \mathcal{T}' . In particular φ maps every node of \mathcal{T} to a node of \mathcal{T}' that has the same type and the same degree. For every virtual edge ε in an R-node μ , the map φ_μ determines whether the orientation of ε has to be reversed to match the orientation of $\varphi_\mu(\varepsilon)$. Moreover, how φ maps the neighbors of an S-node μ to the neighbors of $\varphi(\mu)$ determines a restacking operation transforming $\text{skel}(\mu)$ into $\text{skel}(\varphi(\mu))$. It follows that \mathcal{T} can be transformed into \mathcal{T}' by applying restacking and reversal operations. Hence, \mathcal{T} and \mathcal{T}' represent the same set of dual graphs, which concludes the proof. \square

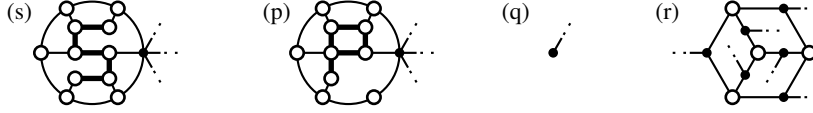


Figure 9. The subgraphs H_μ of the skeleton graph depending on the type of node μ . The small black vertices are the attachment vertices.

In the following we reduce dual SPQR-tree isomorphism testing to graph isomorphism testing for planar graphs, which can be solved in linear time [9]. We define the *skeleton graph* $G_{\mathcal{T}}$ of an SPQR-tree \mathcal{T} as follows; see the lower right part of Figure 10 for an example. For each node μ in \mathcal{T} there is a subgraph H_μ in $G_{\mathcal{T}}$, and for each edge $\{\mu, \mu'\}$ in \mathcal{T} , the skeleton graph contains an edge connecting H_μ and $H_{\mu'}$. In the following we describe the subgraph H_μ for the cases in which μ is an S-, P-, Q-, or R-node, and define the *attachment vertices* that are incident to the edges connecting H_μ to other subgraphs.

If μ is an S- or P-node, then subgraph H_μ contains only one attachment vertex v_μ , and all subgraphs representing neighbors of μ are attached to v_μ . To distinguish between S- and P-nodes, small non-isomorphic subgraphs called *tags* are attached to v_μ , see Figure 9s and p, respectively. If μ is a Q-node, then H_μ is a single attachment vertex, see Figure 9q. Note that μ is a leaf in \mathcal{T} and thus H_μ is also a leaf in $G_{\mathcal{T}}$. If μ is an R-node, then H_μ is the skeleton $\text{skel}(\mu)$, where additionally every virtual edge ε is subdivided by an attachment vertex v_ε ; see Figure 9r for an illustration of the case $\text{skel}(\mu) = K_4$. The subgraph $H_{\text{corr}(\varepsilon)}$ stemming from the neighbor $\text{corr}(\varepsilon)$ of μ is attached to H_μ via the attachment vertex v_ε .

Lemma 5. *The skeleton graph of an SPQR-tree is planar and can be computed in linear time.*

Proof. Clearly, the skeleton graph $G_{\mathcal{T}}$ of an SPQR-tree \mathcal{T} can be computed in linear time by processing each node μ separately to compute the subgraph H_μ consuming time linear in the number of vertices of $\text{skel}(\mu)$. Note that this implicitly shows that the number of vertices of $G_{\mathcal{T}}$ is linear.

Let \mathcal{T} be an SPQR-tree rooted at an arbitrary node. The skeleton graph $G_{\mathcal{T}}$ can be embedded in a planar way by embedding the subgraphs corresponding to the nodes in \mathcal{T} top-down with respect to the chosen root. Obviously, every subgraph in $G_{\mathcal{T}}$ corresponding to a node in \mathcal{T} is planar, thus we can start by embedding the subgraph corresponding to the root arbitrarily. Let μ be a non-root node in \mathcal{T} and let μ' be its parent. If μ is not an R-node, H_μ can be embedded with its only attachment vertex on the outer face. If μ is an R-node, H_μ can be embedded with the attachment vertex corresponding to the parent μ' of μ in \mathcal{T} on the outer face. Thus, in any case, H_μ can be placed inside a face incident to the attachment vertex stemming from μ' corresponding to μ , yielding a planar drawing. \square

Lemma 6. *Two SPQR-trees are dual isomorphic if and only if their skeleton graphs are isomorphic.*

Proof. Let φ together with $\varphi_{\mu_1}, \dots, \varphi_{\mu_k}$ be a dual SPQR-tree isomorphism between the SPQR-trees \mathcal{T} and \mathcal{T}' . We show how this induces a graph isomorphism φ_G between the skeleton graphs $G_{\mathcal{T}}$ and $G_{\mathcal{T}'}$. If μ is an S-, P-, or Q-node, then its corresponding subgraph in H_{μ} only contains a single attachment vertex v_{μ} . Since $\varphi(\mu)$ is of the same type (due to Property II of dual SPQR-tree isomorphisms), the subgraph $H_{\varphi(\mu)}$ also contains a single attachment vertex $v_{\varphi(\mu)}$ and we set $\varphi_G(v_{\mu}) = v_{\varphi(\mu)}$. For S- and P-nodes we additionally map their tags isomorphically to each other. If μ is an R-node, the map φ_{μ} is a graph isomorphism between $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$ (Property III). Thus, it induces a graph isomorphism between H_{μ} and $H_{\varphi(\mu)}$, since these subgraphs are obtained from $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$, respectively, by subdividing each virtual edge. It remains to show that φ_G respects the edges between attachment vertices of different subgraphs. Since φ is a graph isomorphism (Property I), attachment vertices of two subgraphs of $G_{\mathcal{T}}$ are connected if and only if the corresponding subgraphs in $G_{\mathcal{T}'}$ are connected. Moreover, Property IV ensures that for a subgraph stemming from an R-node the right attachment vertices are chosen (for other nodes this is clear since their subgraphs have unique attachment vertices).

For the opposite direction, assume φ_G is a graph isomorphism between $G_{\mathcal{T}}$ and $G_{\mathcal{T}'}$. Let H_{μ} be the subgraph stemming from a node μ in \mathcal{T} . As H_{μ} is a block (maximal biconnected component) in $G_{\mathcal{T}}$ (or a leaf, if μ is a Q-node), it has to be mapped to a block in $G_{\mathcal{T}'}$. As all edges in $G_{\mathcal{T}'}$ connecting attachment vertices of subgraphs stemming from different nodes are bridges, all vertices in H_{μ} have to be mapped to vertices in $H_{\mu'}$ for some node μ' in \mathcal{T}' . This defines the map φ by setting $\varphi(\mu) = \mu'$. Clearly, φ is a graph isomorphism between \mathcal{T} and \mathcal{T}' , since two subgraphs in a skeleton graph are connected by an edge if and only if the corresponding nodes in its SPQR-tree are adjacent; thus, φ satisfies Property I. Since the only leaves in a skeleton graph stem from Q-nodes, $\varphi(\mu)$ is a Q-node if and only if μ is a Q-node. Let v be an attachment vertex stemming from an inner node μ in \mathcal{T} . Then v is a cutvertex and, since every cutvertex in a skeleton graph is an attachment vertex, $\varphi_G(v)$ is also an attachment vertex in $G_{\mathcal{T}'}$. The vertex v has degree 3 if and only if μ is an R-node, thus φ maps R-nodes to R-nodes. Moreover, if μ is an S-node, v cannot be mapped to an attachment vertex stemming from a P-node, since the tags attached to S- and P-nodes are not isomorphic. Hence, φ maps S- and P-nodes to S- and P-nodes, respectively, and thus satisfies Property II.

To obtain a dual SPQR-tree isomorphism, it remains to define a map φ_{μ} for each R-node μ in \mathcal{T} that satisfies Properties III and IV. As observed before, φ_G defines a bijection between the vertices in the subgraph H_{μ} stemming from μ and the vertices in $H_{\varphi(\mu)}$ stemming from $\varphi(\mu)$. As H_{μ} and $H_{\varphi(\mu)}$ are the skeletons $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$ (with a subdivision vertex on each virtual edge), φ_G defines a graph isomorphism φ_{μ} between $\text{skel}(\mu)$ and $\text{skel}(\varphi(\mu))$ (satisfying Property III). To show that Property IV holds, consider a virtual edge ε in $\text{skel}(\mu)$ and the attachment vertex v_{ε} in H_{μ} stemming from it. Further, denote by $v_{\text{corr}(\varepsilon)}$ the attachment vertex in $H_{\text{corr}(\varepsilon)}$ such that $G_{\mathcal{T}}$ contains the edge $\{v_{\varepsilon}, v_{\text{corr}(\varepsilon)}\}$. Then, φ_G maps $\{v_{\varepsilon}, v_{\text{corr}(\varepsilon)}\}$ to an edge $\{\varphi_G(v_{\varepsilon}), \varphi_G(v_{\text{corr}(\varepsilon)})\}$ in $G_{\mathcal{T}'}$. Since $\varphi_G(v_{\varepsilon}) = v_{\varphi_{\mu}(\varepsilon)}$ holds by the definition of φ_{μ} , and $\varphi_G(v_{\text{corr}(\varepsilon)})$ stems from the node $\varphi(\text{corr}(\varepsilon))$ by the definition of φ , we have that $\text{corr}(\varphi_{\mu}(\varepsilon)) = \varphi(\text{corr}(\varepsilon))$ by the definition of the skeleton graph $G_{\mathcal{T}'}$. As this establishes Property IV, it concludes the proof. \square

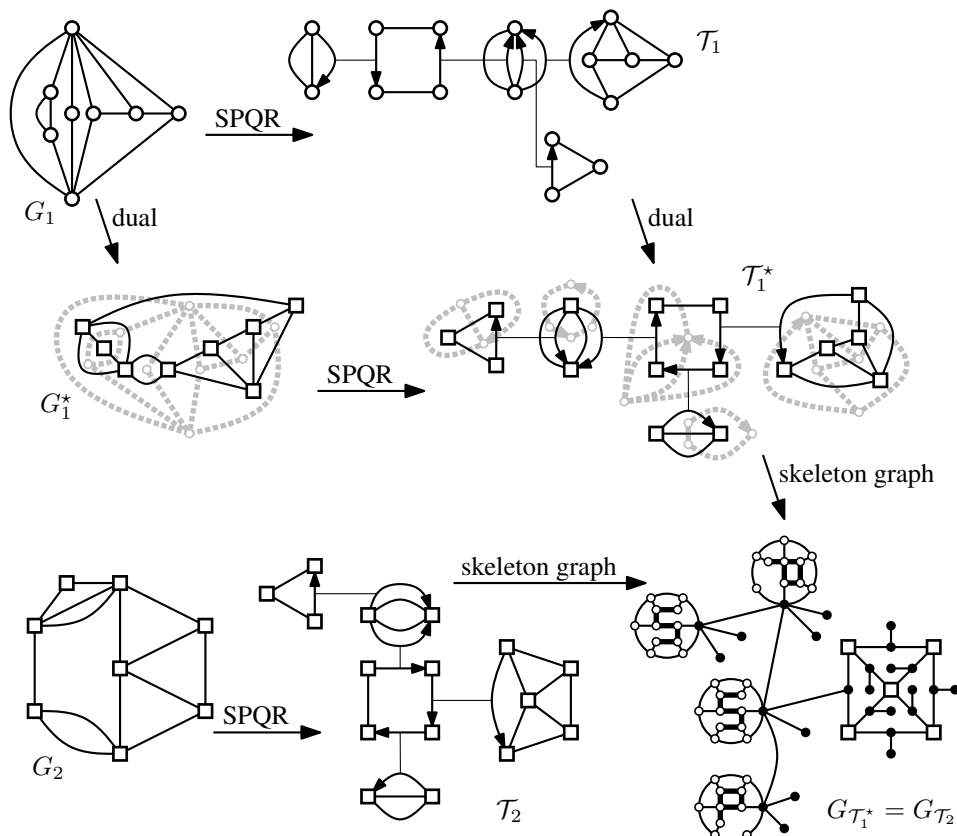


Figure 10. First building the dual graph G_1^* of G_1 (with respect to a fixed embedding) and then building its SPQR-tree, or first building its SPQR-tree \mathcal{T}_1 and then its dual SPQR-tree yields the same tree \mathcal{T}_1^* (Lemma 1). The graphs G_1 and G_2 are dual to each other (with respect to at least one pair of embeddings) if and only if \mathcal{T}_1^* and \mathcal{T}_2 represent the same set of duals (Corollary 1), which is the case if and only if their skeleton graphs $G_{\mathcal{T}_1^*}$ and $G_{\mathcal{T}_2}$ are isomorphic (Lemma 4 and Lemma 6).

Following the outline given in Figure 8a, problem MUTUAL DUALITY for biconnected planar graphs can be reduced to isomorphism testing for planar graphs, which is linear-time solvable [9]. We formalize this result in the following theorem and give an example in Figure 10 to illustrate the steps of the algorithm.

Theorem 7. MUTUAL DUALITY can be solved in linear time for biconnected planar graphs.

Proof. Corollary 1 states that MUTUAL DUALITY can be solved by testing whether two SPQR-trees (that can be computed in time linear in the number of vertices of the input graphs [8]) represent the same set of dual graphs. By Lemma 4, this is equivalent to testing whether these two SPQR-trees are dual isomorphic, which can be done by testing whether their skeleton graphs are isomorphic, due to Lemma 6. The skeleton graph of an SPQR-tree is planar and has a linear number of vertices, see Lemma 5. Hence, we can

use the linear-time algorithm for testing whether two planar graphs are isomorphic by Hopcroft and Wong [9] yielding a linear-time algorithm solving MUTUAL DUALITY. \square

As GRAPH SELF-DUALITY is a special case of MUTUAL DUALITY, we have the following corollary.

Corollary 2. GRAPH SELF-DUALITY *can be solved in linear time for biconnected planar graphs.*

As observed above, two biconnected planar graphs are related with respect to the common dual relation if and only if their graphic matroids are isomorphic. As testing whether two biconnected graphs lie in the same equivalence class of the common dual relation is exactly what we did in this section, we get the following.

Corollary 3. *Testing whether two biconnected planar graphs are 2-isomorphic (i.e., whether their graphic matroids are isomorphic) can be done in linear time.*

8 Conclusions

In this paper we defined and studied the problem MUTUAL DUALITY of testing for two graphs G_1 and G_2 whether there exists an embedding of G_1 such that the corresponding dual graph is isomorphic to G_2 . We proved that MUTUAL DUALITY is NP-complete in general, while it is solvable in polynomial (actually linear) time for biconnected planar graphs.

The interest on this problem is twofold. On one hand, it represents a new step in the fundamental theory of planar graph isomorphism; this is also testified by the fact that, as a side effect, it provides the same results for the well-known problem of testing GRAPH SELF-DUALITY [2, 12]. On the other hand, it could be seen as a single example among a plethora of problems whose goal is to find a dual graph of G_1 satisfying certain properties. In this direction, we believe that the definition of the new data-structure *dual SPQR-tree*, which allows to efficiently handle all the duals of a biconnected planar graph, could be considered as a main result of this paper, independently of its application to solve MUTUAL DUALITY; indeed, we strongly believe that it may successfully be used to tackle many other problems of the same type.

As remarked above, the results we obtained on MUTUAL DUALITY can be used for testing GRAPH SELF-DUALITY, asking whether a given graph G can be embedded in such a way that the corresponding dual is isomorphic to G . The constrained version MAP SELF-DUALITY [13] of GRAPH SELF-DUALITY requires the embedding of G to be preserved in the isomorphism with the corresponding dual. We proved that the NP-completeness result for MUTUAL DUALITY extends to MAP SELF-DUALITY, but we could not prove the same for the polynomial-time testing algorithm. Hence, we leave as an open problem the question whether MAP SELF-DUALITY can be solved efficiently for biconnected planar graphs.

References

- [1] P. Angelini, G. Di Battista, and M. Patrignani. Finding a minimum-depth embedding of a planar graph in $O(n^4)$ time. *Algorithmica*, 60:890–937, 2011.
- [2] D. Archdeacon and R. B. Richter. The construction and classification of self-dual spherical polyhedra. *Journal of Combinatorial Theory, Series B*, 54(1):37–63, 1992.
- [3] D. Bienstock and C. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. *Algorithmica*, 5:93–109, 1990.
- [4] G. Da Lozzo, V. Jelínek, J. Kratochvíl, and I. Rutter. Planar embeddings with small and uniform faces. In *Proceedings of the 25th International Symposium on Algorithms and Computation*, LNCS, 2014. Full version available at <http://arxiv.org/abs/1409.4299>.
- [5] G. Di Battista and R. Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15(4):302–318, 1996.
- [6] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM Journal on Computing*, 25:956–997, 1996.
- [7] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [8] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In J. Marks, editor, *Proceedings of the 8th International Symposium on Graph Drawing (GD'00)*, volume 1984 of LNCS, pages 77–90. Springer, 2001.
- [9] J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing (STOC'74)*, pages 172–184. ACM, 1974.
- [10] P. Mutzel and R. Weiskircher. Optimizing over all combinatorial embeddings of a planar graph (extended abstract). In G. Cornuéjols, R. E. Burkard, and G. J. Woeginger, editors, *Integer Programming and Combinatorial Optimization (IPCO'99)*, volume 1610 of LNCS, pages 361–376. Springer, 1999.
- [11] B.V. Raghavendra Rao and M.N. Jayalal Sarma. On the complexity of matroid isomorphism problem. *Theory of Computing Systems*, 49(2):246–272, 2011.
- [12] B. Servatius and P. R. Christopher. Construction of self-dual graphs. *The American Mathematical Monthly*, 99(2):153–158, 1992.
- [13] B. Servatius and H. Servatius. Self-dual graphs. *Discrete Mathematics*, 149(1-3):223–232, 1996.
- [14] W. T. Tutte. Connectivity in matroids. *Canadian Journal of Mathematics*, 18:1301–1324, 1966.

- [15] H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1):150–168, 1932.
- [16] H. Whitney. Non-separable and planar graphs. *Transactions of the American Mathematical Society*, 34(2):339–362, 1932.
- [17] H. Whitney. 2-isomorphic graphs. *American Journal of Mathematics*, 55:245–254, 1933.