Journal of Computer and System Sciences 77 (2011) 430–438

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Errors in graph embedding algorithms

Wendy Myrvold [a,1], William Kocay [b,*,1]

[a] Dept. of Computer Science, University of Victoria, Victoria, B.C., Canada, V8W 3P6
[b] Computer Science Department, St. Paul's College, University of Manitoba, Winnipeg, Manitoba, Canada, R3T 2N2

ARTICLE INFO

Article history:
Received 22 May 2007
Received in revised form 26 May 2010
Accepted 3 June 2010
Available online 8 June 2010

Keywords:
Graph embedding
Torus
Graph genus
Algorithm

ABSTRACT

One major area of difficulty in developing an algorithm for embedding a graph on a surface is handling bridges which have more than one possible placement. This paper addresses a number of published algorithms where this has not been handled correctly. This problem arises in certain presentations of the Demoucron, Malgrange and Pertuiset planarity testing algorithm. It also occurs in an algorithm of Filotti for embedding 3-regular graphs on the torus. The same error appears in an algorithm for embedding graphs of arbitrary genus by Filotti, Miller and Reif. It is also present in an algorithm for embedding graphs of arbitrary genus by Djidjev and Reif. The omission regarding the Demoucron, Malgrange and Pertuiset planarity testing algorithm is easily remedied. However there appears to be no way of correcting the algorithms of the other papers without making the algorithms take exponential time.

1. Introduction

A graph G embeds on a surface S (such as the plane or the torus) if G can be drawn on S with no crossing edges. An excellent reference for graph embeddings is the recent book of Mohar and Thomassen [21].

A number of published algorithms for graph embeddings contain errors, either in their presentation or content. For example, the Hopcroft and Tarjan planarity testing algorithm [14] has an associated paper [5] listing various corrections. The planarity testing algorithm of Shih and Hsu [25] requires clarification in order to provide an implementable version. Boyer [1] presented the additional planarity conditions required to program it using the PC-tree data structure described in [25]. A projective plane embedding algorithm derived by Perunicic and Duric [24] is incorrect in that it sometimes fails to find a projective plane embedding of a graph when one exists as noted by Mohar [18, p. 483] and independently observed by Williamson (private communication to Mohar [18, p. 483]).

This paper arose out of a search for torus obstructions (defined in Section 3). A fast algorithm for embedding a graph on the torus was required. In 1978 Filotti had published a paper [8] presenting an algorithm for embedding 3-regular graphs on the torus. This was followed by a much expanded version [7] in 1980, which corrected a number of minor errors. This was then followed by papers by Filotti, Miller, and Reif [10], Filotti and Mayer [9], Miller [17], and Djidjev and Reif [6]. These papers (except for [6]) all used Filotti's techniques to address embedding and isomorphism problems for graphs of bounded genus.

Filotti's algorithm is based on the planarity testing algorithm of Demoucron, Malgrange and Pertuiset [4]. We start by pointing out a misconception that Filotti [7] and also Gibbons [13, p. 89] had regarding this algorithm.

* Corresponding author.
E-mail addresses: wendym@cs.UVic.ca (W. Myrvold), bkocay@cc.umanitoba.ca (W. Kocay).
URLs: http://www.cs.uvic.ca/~wendym (W. Myrvold), http://bkocay.cs.umanitoba.ca/BillKocay.html (W. Kocay).

1 Research supported by an NSERC Discovery grant.

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2010.06.002

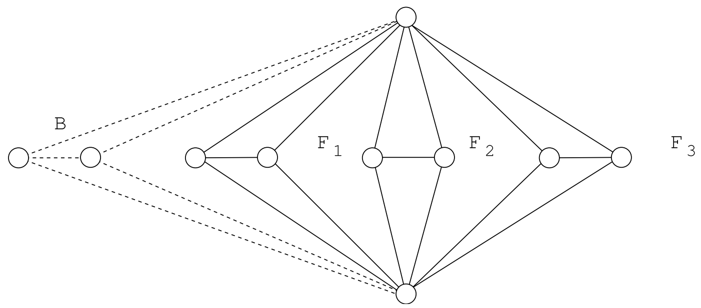

Fig. 1. A bridge B that can be placed in three faces.

A fatal flaw in Filotti's algorithm [7], which also appears in the algorithm of Filotti, Miller, and Reif [10], is then described. The algorithm of Djidjev and Reif [6] is also incorrect, and a fundamental error in it is presented. There appears to be no way to fix these problems without creating algorithms which take exponential time. The implications of this and a discussion of possible alternatives conclude the paper.

2. Demoucron's planarity testing algorithm

Graph embedding algorithms often proceed by considering the embedding of a subgraph H of a graph G and trying to extend this to an embedding of the entire graph.

Definition 2.1. Let H be subgraph of G. A *bridge* of G with respect to H is a subgraph B of G such that either:

(i) B consists of one edge uv such that $u, v \in V(H)$, but $uv \notin E(H)$; or
(ii) B is the subgraph of G induced by a connected component C of $G - V(H)$, together with all edges uv of G such that $u \in V(C)$ and $v \in V(H)$.

Notice that an edge consists of a pair of vertices, and therefore always contains its endpoints. The vertices of B that are also in H are called the *vertices of attachment* of B.

The *faces* of an embedding of a graph G on a surface are the regions that remain when the points representing the vertices and edges of G are removed from the surface. The boundary of any face is a closed walk in G, called a *facial walk*, or *facial cycle*, if it is a cycle. A bridge B can be *drawn* in a face F if all the vertices of attachment lie on the boundary of F. Two bridges *hinder* each other (or *conflict* with each other) in a face F if they cannot both be embedded in F without edges crossing.

Both Gibbons and Filotti make the same mistake regarding the planarity testing algorithm of Democron, Malgrange and Pertuiset [4]. Gibbons' proof that the algorithm is correct contains an error. He states that if a graph G is 2-connected, every bridge of G with respect to a subgraph H "*has at least two points of contact and can therefore be drawn in just two faces*" [13, p. 89]. Filotti [7, p. 256] also makes the assumption that bridges can be embedded in at most two faces.

A 10-vertex counterexample is given in Fig. 1. The graph in Fig. 1 has no cut-vertices. The subgraph H embedded so far is in bold. The bridge B with respect to this embedded subgraph can be embedded in F_1, F_2 or F_3. Clearly, this example can be extended to show that there can be an arbitrary number of faces that some bridge can be drawn in, when the graph is 2-connected. This situation is not difficult to handle, but one needs to be aware of it when designing any embedding algorithm. Essentially, all bridges with the same two vertices of attachment can be grouped together as one bridge. An alternative would be to require the input graph to be 3-connected.

3. Torus obstructions

An *obstruction* for a surface is essentially a minimal graph that cannot be embedded in the surface. Kuratowski's theorem states that $K_{3,3}$ and K_5 are the only obstructions for the plane. A graph G with minimum vertex degree at least three is a *topological obstruction* for the torus if G is not embeddable on the torus but $G - e$ is embeddable, for all edges e. A *minor-order obstruction* has the additional property that contracting e results in a graph which is torus embeddable for each edge e.

The complete set of torus obstructions is not known. To date, 239,451 topological obstructions and 16,629 minor-order obstructions have been found [22,3]. One of our goals is to find all the torus obstructions and prove that the set we have is complete.

Early in this project, we wanted an idea of how large torus obstructions could be. The 3-regular graphs seemed a favorable place to search for such obstructions for two reasons. First, there were a manageable number of 3-regular graphs in the range in which we were interested (22 and 24 vertices). There are about seven million 3-regular graphs of order 22

Table 1
The numbers of 3-regular obstructions.

Number of vertices	# obstructions
12	1
14	9
16	20
18	133
20	39
22	2
24	2
Total	206

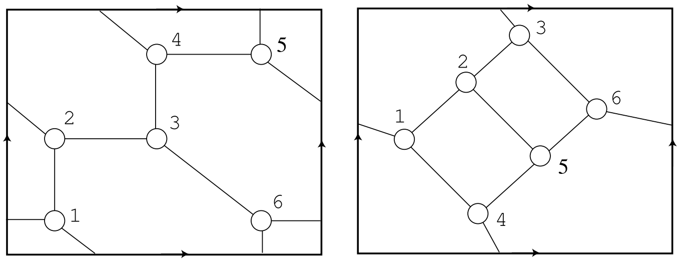

Fig. 2. Two embeddings of $K_{3,3}$ on the torus.

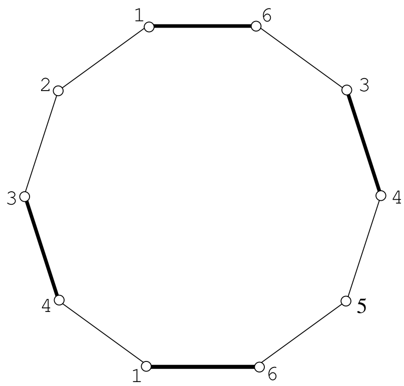

Fig. 3. A 10-gon face.

and about 118 million of order 24 [2], numbers that the exponential torus testing algorithm [23] can process in about two weeks, assuming 41 processors are used. Second, because the set of obstructions is finite, we expected that as the number of vertices increased, the maximum vertex degree would decrease, with three being the lower limit.

An exhaustive computer search of the 3-regular graphs having at most 24 vertices has been completed and rechecked as part of the Master's thesis research of three students [22,3,26] (all supervised by W. Myrvold). Brinkmann's program [2] was used to generate the 3-regular graphs. The results are summarized in Table 1. Only about 20 million of the roughly two trillion 3-regular graphs of order 26 have been tested, none of which are obstructions. To finish testing the 26-vertex 3-regular graphs, a faster torus embedding program would be very desirable. This motivated us to consider implementing Filotti's algorithm for embedding 3-regular graphs on the torus [7].

4. Repeated vertices on facial walks

If a 2-connected graph G is embedded in the plane, then the facial boundaries are cycles in G with no repeated vertices or edges. On the torus, it is often the case that a facial walk may have several repeated vertices and edges. An embedding on a surface is said to be *quasi-planar* if there are no repeated vertices or edges on any facial boundary.

Fig. 2 shows the two distinct embeddings of $K_{3,3}$ on the torus. Here the torus is represented as a rectangle in which opposite sides have been identified. The left embedding is quasi-planar. The one on the right has a 10-gon face (illustrated in Fig. 3) which has repeated vertices and edges on its boundary.

In a planar embedding, two bridges B_1 and B_2 of a graph H which conflict in a face F, will also conflict in every face of H in which they can both be embedded. This is not so for the torus, even for quasi-planar embeddings of H. For example,

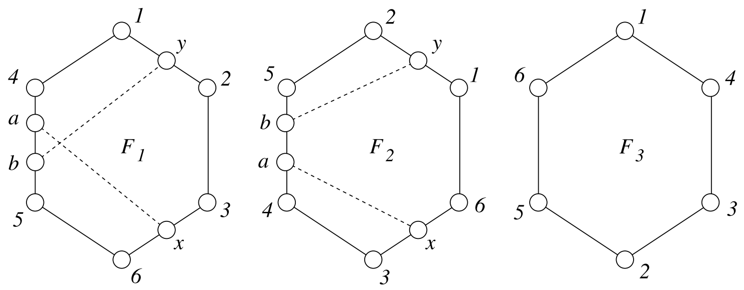

Fig. 4. Bridges which conflict in F_1 but not in F_2.

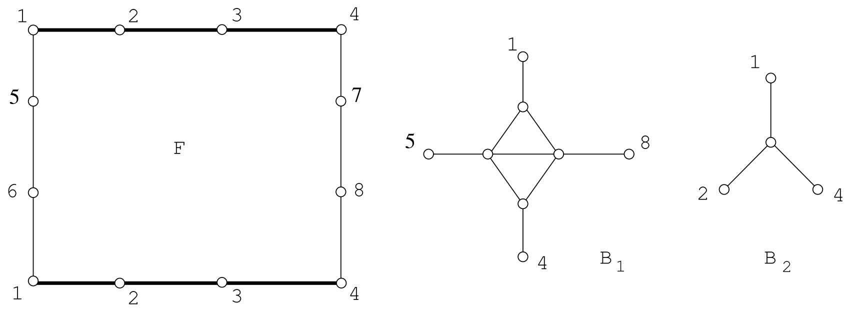

Fig. 5. Bridges B_1 and B_2 to be embedded in face F.

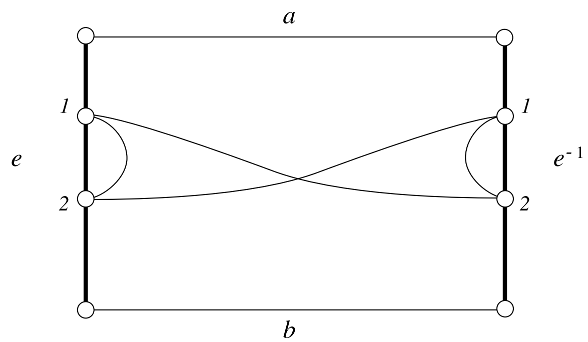

Fig. 6. A cylindrical face, with 4 possible embeddings of a $(1, 2)$-path.

let H be a graph which is a subdivision of $K_{3,3}$ which is embedded on the torus as in the quasi-planar embedding of $K_{3,3}$ on the left of Fig. 2. The three faces of H corresponding to the hexagonal faces of $K_{3,3}$ are drawn in Fig. 4. There are two bridges which are the single edges $[a, x]$ and $[b, y]$. They conflict in F_1 but not in F_2. Note also that any bridges with all attachments selected from $\{1, 2, 3, 4, 5, 6\}$ could be embeddable in all three faces.

Algorithms for embedding often start with an embedded subgraph, and then augment the embedded subgraph by selecting a path, also called a *chain*, which is embedded across one of the faces (for example, [4,7]). The choice of which path to use cannot be made arbitrarily. Fig. 5 shows a situation where selecting a chain from B_2 which goes across the face (say from 1 on the left upper corner of the rectangle to 4 on the bottom right) makes it impossible to embed B_1. However, there is an embedding of the chain which permits both bridges to be embedded in the face.

5. The 2-chains theorem

Consider a facial walk W that has a single repeated path, denoted e. In traversing W, the path e will be traversed in both directions. The portions of W between the two traversals of e can be denoted a and b. Thus we can write $W = eae^{-1}b$, as per Fig. 6. As the closure of such a face is a cylinder, we call such faces *cylinders*. Filotti calls a repeated path on a facial boundary an *internal chain*. A chain with vertices of attachment on a cylindrical face can have four different embeddings within that face, as shown in Fig. 6. The key to deriving a polynomial time algorithm is to determine which embedding to use for each chain without involving an exponential backtrack to try all possibilities.

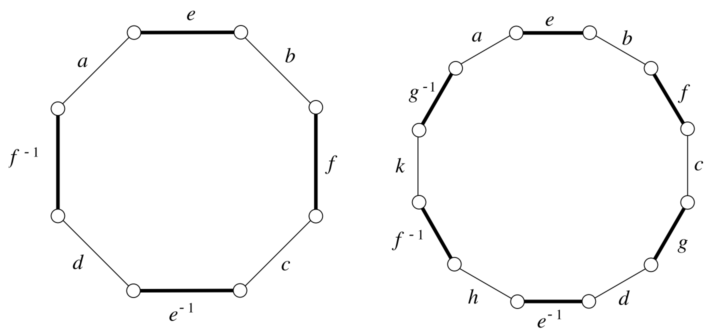

Fig. 7. Punctured tori faces.

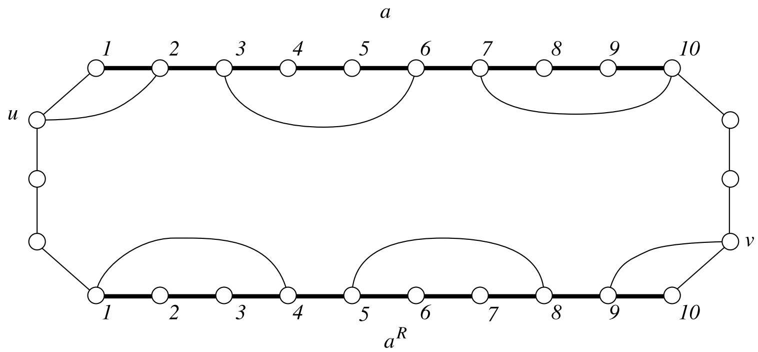

Fig. 8. Multiple chains could be required.

During the course of Filotti's algorithm, faces which have three or four repeated chains on their boundary might also be encountered (as per Fig. 7). These are called "special" faces. The flaw in the algorithm is the method by which Filotti claims that these can be handled.

The following assumption (taken verbatim from Filotti's paper [7, p. 272]) is where the mistake in reasoning occurs. The labeling of the face is the same as the face on the left-hand side of Fig. 7. The notation g refers to an embedding of a subgraph of the graph.

We shall say that internal chains e and e^{-1} are separated *if no two corresponding points on e and e^{-1} are on the same face of g. It is easily seen that e and e^{-1} can be separated in one of three ways*:

(i) *one chain C from x to y where x is a point of bfc and y is a point of $df^{-1}a$*;

(ii) *two chains C_1 from x_1 to y_1 and C_2 from x_2 to y_2 where x_1 is a point of bfc, y_1 is a point of e, x_2 is a point of $df^{-1}a$, and y_2 is a point of e^{-1}*;

(iii) *two chains C_1 from x_1 to y_1 and C_2 from x_2 to y_2 where x_1 is a point of $df^{-1}a$, y_1 is a point of e, x_2 is a point of bfc, and y_2 is a point of e^{-1}*.

The problem with this assumption is that there are situations when more than two chains could be required to separate e from e^{-1}. The picture in Fig. 8 shows a situation where six chains separate e from e^{-1} (denoted a and a^R in the diagram) but no subset containing five or fewer chains works.

Filotti, Miller and Reif (FMR) have a similar theorem [10, p. 33, Theorem 2] although their statement of it is much harder to comprehend because they reinvent the standard graph theory terminology (e.g. the terms "graph" and "cut-point" have been given unorthodox definitions. But still they use Menger's theorem, which would appear not to apply to their unorthodox definitions).

The "two chains theorem" is stated in FMR as follows:

If (a, a^R) is an internal pair of region E then one and only one of the following conditions is satisfied:

(1) E has a cut-point on (a, a^R) (they define a *cut-point* to be a vertex which appears more than once on the boundary of E).

(2) There exist two vertex-disjoint chains in E from distinct corners of $[aEa^R]$ to distinct corners of $[a^REa]$.

This is illustrated in Fig. 9.

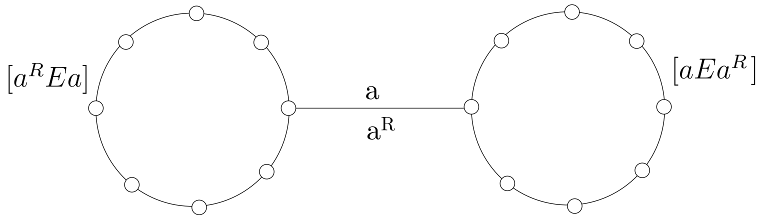

Fig. 9. The two chains theorem.

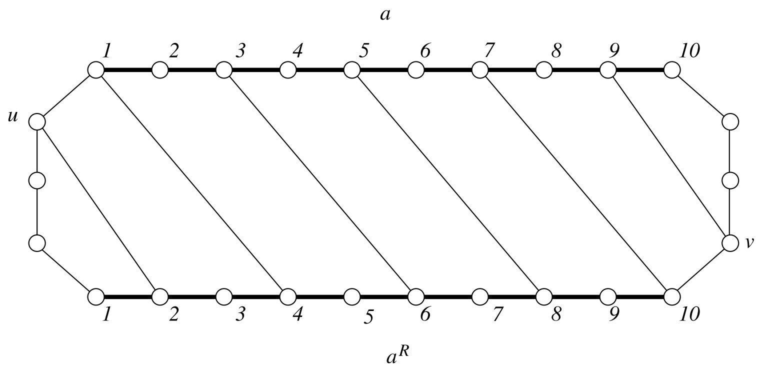

Fig. 10. An alternate embedding.

The FMR paper identifies a face or region with its facial walk. Given a closed walk E, bounding a region also called E, an internal pair (a, a^R) refers to a repeated path a on E, where a^R is equivalent to a^{-1}. Then the portion of E that follows a until a^R is reached is denoted $[aEa^R]$. Similarly, $[a^R Ea]$ refers to the portion of E following a^R until a is reached. Notice that $[aEa^R]$ and $[a^R Ea]$ are cycles in the graph, connected by the path a.

The FMR paper allows the possibility that the two chains connecting $[aEa^R]$ and $[a^R Ea]$ could intersect the repeated path a. Thus in the example of Fig. 8 the two chains could be $[u, 2, 3, 6, 7, 10]$ and $[1, 4, 5, 8, 9, v]$. But then there is a different problem since the authors assume that once the two chains have been selected, there is only one way to embed them in the face E. For example, [10, p. 33, Step (5)] states that one should "*embed them* (*in the unique way*)". Also, it is clear that in step c.aa.b of the Extension Algorithm (Case 3) [7, p. 273] there has been no consideration that the two chains could have multiple embeddings. An alternate embedding of the above two chains is shown in Fig. 10.

There is another fundamental problem with the FMR algorithm when there are facial walks with more than one pair of repeated paths on the closed walk E. In order to prove the "2-chains theorem", they remove the problem of having more than one repeated path, as follows. The focus of the theorem is restricted to the part of G embedded inside E. Repeated vertices on the boundary, other than those of the path a, are then treated as distinct vertices, by doubling them, as necessary. Once this has been done, Menger's theorem is used. If two vertex-disjoint paths connecting $[aEa^R]$ and $[a^R Ea]$ do not exist, then there must be a cut-vertex on the path a. Such a cut-vertex will also be a "cut-point", that is, a repeated vertex on E. The problem with this approach is that doubling the vertices cannot be used during the course of the algorithm, as it changes the graph being embedded. Furthermore, Menger's theorem is concerned with connectivity, and not with repeated vertices on a facial boundary, or with the way in which the two paths found are embedded. It is quite possible for two paths to exist, and to be embedded so that a "cut-point" simultaneously exists on a; and the graph remains 2-connected. This results in an erroneous application of the "2-chains theorem" to their algorithm, as described below.

Suppose that we are embedding a graph G, and that a subgraph has already been embedded. Consider their procedure "Remove Internal Edges" (p. 33 of [10]). Here we have a repeated path e on a facial walk E, as per Fig. 11. The purpose of the procedure is to find two chains across the face E, when they exist, and to embed them so that e will no longer be repeated on any facial boundary. (Note: in their algorithm, e is called an "edge", but it is clear from steps (5) and (7) of the algorithm, that it is really a path.) Let e have endpoints x and y. Denote the facial walk by $[y, e, x, a, x, e^R, y, b]$. Here a and b are cycles on the facial boundary containing x and y. Suppose that a and b intersect in a path (as is the case in Fig. 7). In step (4) of the procedure, they use an augmenting path algorithm to search for two vertex-disjoint chains from x to some vertex of a, and from y to some vertex of b. When the chains are found to exist, let them be called Q_x and Q_y. They make a number of incorrect, related assumptions. We focus on the following two:

1) Q_x and Q_y have a unique embedding in E – see step 5, p. 33 of [10];
   – As we have seen in Figs. 8 and 10 above, this assumption is not valid. Examples with many possible distinct embeddings can be constructed. If the algorithm is changed to recursively consider all possible embeddings of Q_x and Q_y, then the polynomial bound of the running time no longer holds.

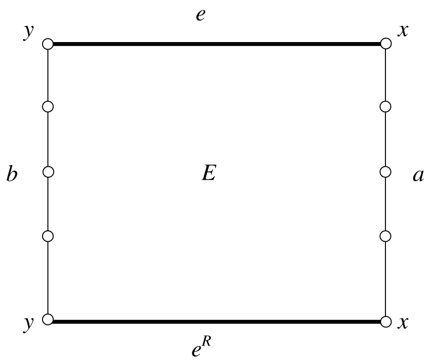

Fig. 11. Procedure "Remove Internal Edges".

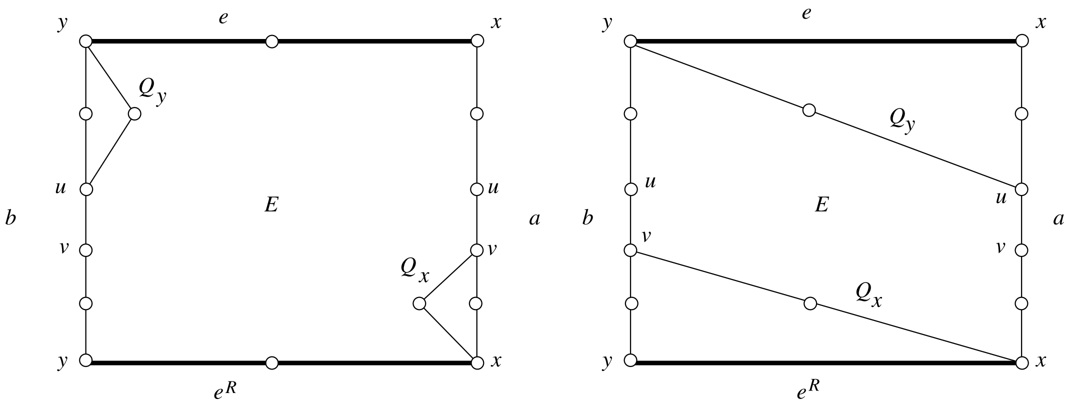

Fig. 12. Two chains, with and without "cut-points".

2) If Q_x and Q_y exist, there is no "cut-point" on e (i.e., a repeated vertex on the boundary) – see steps 4a and 4b;
   – Here they are using the "2-chains theorem" (p. 33 of [10]), which is proved via Menger's theorem. A "cut-point" is defined as a repeated vertex on a facial walk. However, the existence of a "cut-point" depends on the embedding, and not on the existence of two vertex-disjoint paths. In Fig. 12 we have a situation where two vertex-disjoint paths exist, but they are embedded in the diagram on the left so as to create a "cut-point". In the diagram on the right, the same two paths are embedded so that there is no "cut-point". In both cases the graph is 2-connected.

In step 4b of their algorithm, they speak of "cut-points from (e_1, e_3) to (e_2, e_4)". It is not clear what this means, but it would appear to have something to do with actual cut-vertices. But since the paths a and b can intersect, there need be no cut-vertices, even when the desired paths do not exist. For example, consider the embedding of $K_{3,3}$ on the right in Fig. 2, which contains a face which is a 10-gon, redrawn in Fig. 3. The edge $[1, 6]$ is a repeated path on the facial boundary. The endpoints of this path are connected by paths $a = [1, 4, 3, 2, 1]$ and $b = [6, 3, 4, 5, 6]$ which are part of the facial boundary, and which are cycles in the graph. There is only one other edge in the graph, namely $[2, 5]$, not contained on the facial boundary. Therefore there do not exist two chains to embed inside the face. Furthermore, it is not necessary to embed $[2, 5]$ inside the face at all, as shown by the embedding of Fig. 2.

6. The Djidjev–Reif paper

Djidjev and Reif [6] present an algorithm to embed a graph G into an oriented surface of minimal genus g, and also to find a Kuratowski subgraph which cannot be embedded in an oriented surface of genus $g - 1$. This paper refers heavily to the previous two papers [7,10]. But it also has at least one major error of its own.

Like other algorithms, this algorithm uses an algorithm for 2-Satisfiability (2-SAT) to embed bridges of G with respect to an already embedded subgraph H. A bridge B of G with respect to an embedding of H is said to be 2-constrained if there are at most two ways of extending the embedding of H to an embedding of $H \cup B$. An embedding of H is said to be *weakly quasi-planar* (*WQP*) if every bridge is 2-constrained. If every bridge is 2-constrained, then it can be converted to an instance of the 2-SAT problem, and solved by a polynomial algorithm [16].

Let F be a facial walk of an embedding of graph H. Given a vertex $v \in F$, $\mathrm{rep}_F(v)$ is the repetition number of v on F, that is, the number of times that v occurs on F. Djidjev and Reif write $SF = \sum(\mathrm{rep}_F(v) - 2)$, where the sum is over all vertices of F with $\mathrm{rep}_F(v) \geqslant 3$. Then $S(H) = \sum_F SF$, where the sum is over all facial walks of the embedding of H. They

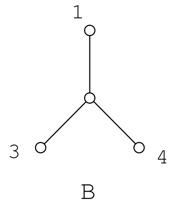

Fig. 13. A bridge B with 8 embeddings in the 10-gon.

make the statement "*Note that an embedding of H is WQP iff* $S(H) = 0$" [6, p. 340, after Corollary 3.1]. This statement is false. A counterexample can be constructed as follows. Let H be a subgraph isomorphic to $K_{3,3}$, embedded as in the right diagram of Fig. 2. Let F be the facial walk bounding the 10-gon face, as illustrated in Fig. 3. Notice that $\text{rep}_F(v) \leqslant 2$, for each $v \in F$, so that $S(H) = 0$. Let B be the bridge illustrated in Fig. 13. As $\text{rep}_F(1) = \text{rep}_F(3) = \text{rep}_F(4) = 2$, it is easy to see that there are 8 ways to embed B in the face bounded by F, so that the embedding of H is not WQP. But $S(H) = 0$, contradicting the above statement.

The algorithm of Djidjev and Reif depends on the above statement in order to conclude that an embedding of H is WQP, so that 2-SAT can be used. The algorithm depends crucially on 2-SAT, because 2-SAT can be solved in polynomial time, whereas 3-SAT is NP-complete. Thus it cannot be concluded that the algorithm of [6] runs in polynomial time.

7. Implications

Several other papers are also affected by the error in Filotti's technique. Filotti and Mayer [9] claim to have a polynomial time algorithm for determining isomorphism of graphs with fixed genus. However, on p. 241 of Section 5 they make the same error as Filotti and so this algorithm is not correct. Miller [17] also claims to have a polynomial time algorithm for isomorphism testing of graphs of fixed genus. On p. 229 there is again a dependence on the flawed portion of the Filotti paper, so it is not correct either.

8. Conclusions

A fast torus embedding program would be a valuable tool for searching for a complete set of torus obstructions. But so far, the only reasonably fast algorithms implemented run in exponential time [23,26,12]. The special case where a graph has no subgraph homeomorphic to $K_{3,3}$ can be recognized and these can be embedded efficiently using [11].

Juvan, Marinček and Mohar have created a linear time torus embedding algorithm [15]. These ideas have been extended to give a linear time algorithm for surfaces of arbitrary genus by Mohar [19,20]. But these approaches are very complex, and very difficult to implement and to ensure the resulting code is correct. However, programming these might either point out further flaws in the reasoning or provide a better understanding of these results.

An attempt is being made by Mohar, Orbanic and Bonnington to create an implementation of [15], but as of July 2006, the program still has bugs (there is a consensus that the two 24-vertex 3-regular obstructions mentioned earlier are torus obstructions, but the code as of July 2006 failed to recognize this).

Errors in reasoning can be subtle and it is a testament to the difficulties of the embedding problem that an algorithm like Filotti's [7] can stand for 25 years before the errors are pointed out. One major obstacle with this direction of research, is that the other algorithms seem to be much more difficult to comprehend than Filotti's approach.

References

[1] J. Boyer, Additional PC-tree planarity conditions, in: J. Pach (Ed.), Proceedings of the 12th International Conference on Graph Drawing 2004, Springer-Verlag, 2004, pp. 82–88.
[2] G. Brinkmann, Fast generation of cubic graphs, J. Graph Theory 2 (1996) 139–149.
[3] J. Chambers, Hunting for torus obstructions, MSc thesis, University of Victoria, 2002.
[4] G. Demoucron, Y. Malgrange, R. Pertuiset, Graphes planaires, Rev. Française Recherche Opérationnelle 8 (1964) 33–47.
[5] N. Deo, Note on Hopcroft and Tarjan planarity algorithm, J. ACM 23 (1976) 74–75.
[6] H. Djidjev, J. Reif, An efficient algorithm for the genus problem with explicit construction of forbidden subgraphs, in: Proc. of the 23rd ACM Symp. Theory of Comp., 1991, pp. 337–347.
[7] I.S. Filotti, An algorithm for embedding cubic graphs in the torus, J. Comput. System Sci. 2 (1980) 255–276.
[8] I.S. Filotti, An efficient algorithm for determining whether a cubic graph is toroidal, in: Proc. of the 10th ACM Symp. Theory of Comp., 1978, pp. 133–142.
[9] I.S. Filotti, Jack Mayer, A polynomial algorithm for determining the isomorphism of graphs of fixed genus, in: Proc. of the 12th ACM Symp. Theory of Comp., 1980, pp. 236–243.
[10] I.S. Filotti, G.L. Miller, J. Reif, On determining the genus of a graph in $O(v^{O(g)})$ steps, in: Proc. of the 11th ACM Symp. Theory of Comp., 1979, pp. 27–37.
[11] A. Gagarin, W. Kocay, Embedding graphs containing K_5 subdivisions, Ars Combin. 68 (2002) 33–45.
[12] A. Gagarin, W. Kocay, D. Neilson, Embeddings of small graphs on the torus, Cubo 5 (2003) 351–371.
[13] A. Gibbons, Algorithmic Graph Theory, Cambridge University Press, Cambridge, 1985.

[14] J. Hopcroft, R. Tarjan, Efficient planarity testing, J. ACM 21 (1974) 549–568.
[15] M. Juvan, J. Marinček, B. Mohar, Embedding a graph in the torus in linear time, in: Integer Programming and Combinatorial Optimization, in: Lecture Notes in Comput. Sci., vol. 920, Springer, 1995, pp. 360–363.
[16] W. Kocay, D. Kreher, Graphs, Algorithms, and Optimization, Chapman and Hall, CRC Press, Boca Raton, FL, 2005.
[17] Gary Miller, Isomorphism testing for graphs of bounded genus, in: Proc. of the 12th ACM Symp. Theory of Comp., 1978, pp. 225–235.
[18] B. Mohar, Projective planarity in linear time, J. Algorithms 15 (1993) 482–502.
[19] B. Mohar, Embedding graphs in an arbitrary surface in linear time, in: Proc. of the 28th ACM Symp. Theory of Comp., 1996, pp. 392–397.
[20] B. Mohar, A linear time algorithm for embedding graphs in an arbitrary surface, SIAM J. Discrete Math. 12 (1999) 6–26.
[21] B. Mohar, C. Thomassen, Graphs on Surfaces, Johns Hopkins University Press, Baltimore, MD, 2001.
[22] E. Neufeld, Practical toroidality testing, MSc thesis, University of Victoria, 1993.
[23] E. Neufeld, W. Myrvold, Practical toroidality testing, in: Proc. of the 8th ACM-SIAM Symp. on Disc. Alg., 1997, pp. 574–580.
[24] B. Perunicic, Z. Duric, An efficient algorithm for embedding graphs in the projective plane, in: The Fifth Quadrennial International Conference on the Theory and Applications of Graphs with Special Emphasis on Algorithms and Computer Science Applications, Western Michigan University, John Wiley and Sons, Inc., 1984, pp. 637–650.
[25] W.-K. Shih, W.-L. Hsu, A new planarity test, Theoret. Comput. Sci. 223 (1999) 179–191.
[26] J. Woodcock, A faster algorithm for torus embedding, MSc thesis, University of Victoria, 2006.