Practical Toroidality Testing

Eugene Neufeld
University of Victoria
eneufeld@math.UVic.ca

Wendy Myrvold *
University of Victoria
wendym@csr.UVic.ca

June 26, 1996

Abstract

We describe an algorithm for embedding graphs on the torus (doughnut) which we implemented first in C, and then in C++. Although the algorithm is exponential in the worst case, it was very effective for indicating the small graphs which are torus obstructions. We have completed examination of the graphs on up to 10 vertices and the 11 vertex ones up to 24 edges, and of these 3884 are topological obstructions, and 2249 are also minor order obstructions. A cursory search of 12 and 13 vertex graphs resulted in several more. We purport that this approach has proved practical as it has permitted us to compile what we believe to be the biggest collection of torus obstructions in the world to date.

1 Introduction

A graph is said to be *embedded* on a surface if it is drawn there with no crossing edges. A graph is *planar* if it can be drawn on the sphere, and is *toroidal* if it can be drawn on the torus (a sphere with one handle). The genus of a planar graph is zero, and a non-planar graph which is toroidal has genus one. The genus of a graph and the genus of an embedding can be defined in strictly combinatorial terms as well, and in fact, this representation (cyclically ordered adjacency lists) is used in our computer program. However, we present a topological discussion here as it is easier for the reader to picture what is happening. Complete details and combinatorial proofs can be found in [16].

There have been many algorithms amenable to implementation for determining if a graph is planar including for example [4, 13, 1], and also some which operate in linear time [9, 22, 2]. Parallel algorithms have also been considered with more recent work including [12, 17]. In contrast, algorithms for toroidality [8, 7, 5, 19, 20, 15] have mainly been presented for their theoretical interest, and to date, we believe that none have been implemented. The first which could lead to a nice program is the newly announced work of Juvan, Marincek, and Mohar (JMM) [11]. Their algorithm is $O(n)$ but a $O(n^2)$ variant should prove easier

*Supported by NSERC.

to implement. Quite likely, a program arising from their work will not be available until at least one or two years down the road.

So given that there is a linear time algorithm, why should we be interested in an exponential algorithm? We started this work in 1991 and our initial goal was to enumerate torus obstructions, the "smallest" (defined rigorously in Section 4) graphs which are not torus embeddable. Their determination can provide insight to designing algorithms for torus embedding. As well, a complete database would be helpful as it could be used to verify "no" answers of torus embedding algorithms (these can be easily modified to return an obstruction when the answer is no).

It is easy to verify the genus of a given combinatorial embedding, but there currently is no safe and sure simple test for even checking that an obstruction is not toroidal. A brute force genus tester is simple, but quite slow even on these small graphs, and the results are not easily hand-checkable. We developed some quick tests which sometimes provide a quick proof that a certain graph is non-toroidal. These are explained in Section 2, and illustrated in Figure 4. These tests provide a faster indication of non-toroidality in some cases than we get by a failed embedding process.

Finally, the JMM [11] approach requires the projective planar obstructions (there are 101 of these) and in some cases, one must be embedded in the torus in all possible ways. As with various algorithms based on the work of Robertson and Seymour, this computation invokes a large constant overhead. It remains to be seen if an approach such as ours which is provably exponential performs faster on small graphs because the exponential component of our algorithm is less onerous than the "constant overhead" involved in the other.

Our code which we believe to be the only practical torus embedder in existence could be used to help test and debug newer and faster programs. As well, it may lead to the development of simpler polynomial time algorithms more amenable to a quick implementation.

Some techniques for quickly indicating non-toroidality are outlined in Section 2. An outline of the embedding process is given in Section 3. In Section 4, we define obstructions and outline the computational results. A brief word is given to explain why we think our program is reliable in Section 5. Finally, we discuss some of the many possible directions for future research in Section 6.

2 Thumbtacks and other indications of non-toroidality

The algorithm described in the next section is in general slowest on graphs which are not toroidal. This is because non-toroidality sometimes must be demonstrated by an exhaustive search for an embedding. The ideas in this section permit faster rejection of some non-toroidal graphs, in many cases avoiding the exponential work of our embedding algorithm.

A cycle C is *contractible* with respect to an embedding if it can be contracted continuously in the surface to a point. A cycle of toroidal G is *flat* if it is contractible in every torus embedding of G. The next theorem provides a tool for identifying some cycles which must be flat if the graph is toroidal. We use $G - C$ to denote the graph obtained from G by deleting the vertices of C and any incident edges.

Theorem 2.1 *If cycle C of toroidal G is such that $G-C$ is nonplanar, then C is flat in G. Further, if flat C is chordless and $G-C$ is connected, then C is a flat face in any torus embedding of G.*

This theorem is fairly obvious and can be seen as a corollary of Kuratowski's theorem. The flat faces (as indicated by Theorem 2.1) introduce some constraints to the embedding. We use the notation $\{a,b\}_v$ to indicate that when the neighbours of v are traversed in a circular order with respect to an embedding, a and b occur consecutively. Certain sets of constraints are incompatible with any torus embedding. For example, $\{\{x,a\}_v, \{x,b\}_v, \{x,c\}_v\}$ is impossible. We use n, m, and f to denote respectively the numbers of vertices, edges, and faces of a graph and its embedding.

Some other properties incompatible with toroidality that we check for are:

1. **Too many edges** ($m > 3n$). We can add edges while preserving toroidality by triangulating flat faces.
2. **Too many faces**. ($f > m - n$). Flat faces are determined as remarked above.
3. **Edge overuse**. Any edge occurs in exactly two faces of a torus embedding. See for example Figure 1 (a).
4. **Dwarf rim**. There cannot be a cyclic set of constraints of size less than the degree of a vertex v. See for example Figure 1 (b).

Certain constraints can be forced. For example, if v has $degree(v)-1$ compatible constraints, we can add one more. Even when G has a vertex v with $degree(v)-2$ constraints we can sometimes add two more (if they are like Figure 2 (a), but not if they are like Figure 2 (b)).

As a heuristic for fast detection of non-toroidality, we take a collection of short cycles and look for flat faces as per Theorem 2.1. The above mentioned conditions are checked in the hopes of proving non-toroidality.

Sometimes this process results in a complete set of constraints. In this case, a procedure called *reification* is used to unify these to get a toroidal embedding of the original graph or to conclude that none exists in $O(n)$ further time.

The information gathered at this stage of the algorithm could easily be used to cut down on the exponential step at the next phase. Due to time pressures however, our code does not include this enhancement. This would be an interesting avenue of future development.

An alternative way to exploit Theorem 2.1 is to observe that if $G-C$ has one non-planar component F (and other planar components), then the problem can be reduced to torus embedding C, F, and the edges between them, and planar embedding the graph obtained by contracting F to a single vertex. We suspect that this could prove to be a very effective reduction for larger graphs, but it is not currently included as part of our system.

Figure 1: Combinatorial thumbtacks

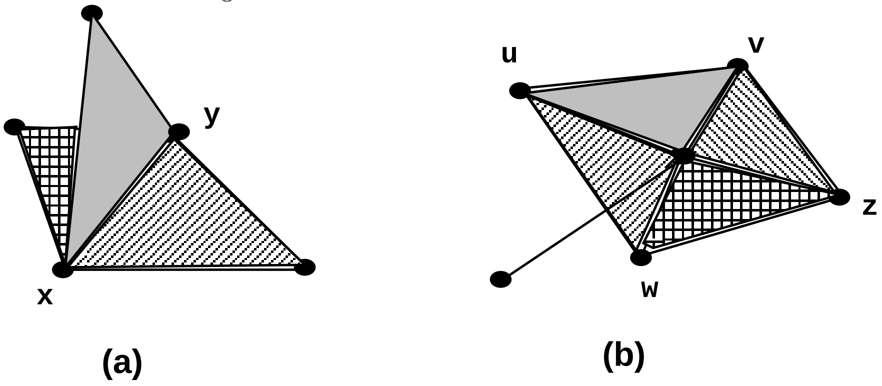
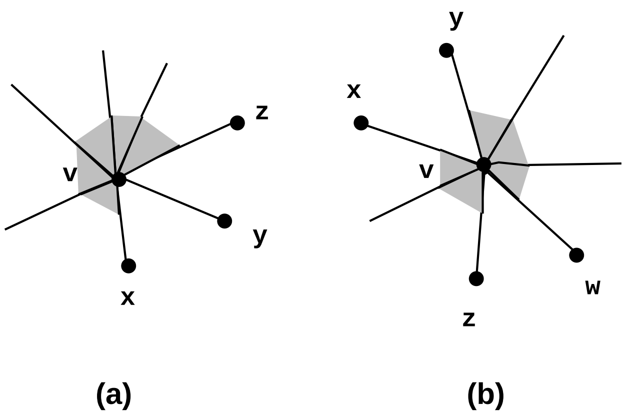


Figure 2: Forcing two constraints

Constraints are shaded



3 The embedding

Here we provide an intuitive description of the algorithm implemented. A mathematically rigorous description will be found in the upcoming journal paper, or is available in [16]. Our strategy is to reduce the problem to that of determining planarity: something we can easily solve.

Our first step is to find a cycle that can be embedded around the handle assuming the graph is toroidal. Actually, we do not know how to determine such a cycle effectively. What we can do however is to determine a small number of candidates (either $O(n)$ with Theorem 3.1 or $O(1)$ with Theorem 3.2), and be guaranteed that at least one must work.

Theorem 3.1 *If G is genus one, at least two cycles of a cycle basis are not contractible in any particular toroidal embedding (which two can depend on the torus embedding considered).*

Theorem 3.2 *If G is genus one, then it contains a Kuratowski obstruction ($K_{3,3}$ or K_5), and at least two cycles in the cycle basis for the obstruction are noncontractible in in any particular torus embedding.*

To see if the graph can be embedded with a particular candidate cycle around the handle, we embed it around the handle and then cut along the cycle, duplicate it, and mark the interiors of the cycles which mapped on the torus to the interior of the surface as forbidden regions for embedding. Now, if we could embed the graph in the plane attaching edges which were incident to the cycle either to one copy or the other, gluing the two copies of the cycle back together gives a torus embedding.

There are two problems with this. First, we do not know which copy of the cycle an edge should be incident to. Our solution is to try all bipartitions of the edges incident to the cycle and it is this that makes our algorithm exponential.

The other problem is that embedding "in between" the two copies of the cycle is still not a planar embedding problem. However, if we could find two edge disjoint paths between the two copies of the cycle, we can then apply Demoucron's planarity algorithm [4] with the initial face list containing the two faces separated by the paths (but not the interiors of the two copies of the cycle) to get what we want.

To find the paths, we add two new vertices s and t, each incident to all vertices of a copy of the cycle. Standard flow techniques are applied to look for two edge disjoint s, t-paths. If we find them, we are done. Otherwise, by Menger's theorem there is a cut vertex. In this case, the graph is split apart into two pieces at the cut vertex, each part is embedded in the the plane separately, and then the pieces are put back together again.

Because it was most important to select cycles with few incident edges, we applied Theorem 3.1 instead of Theorem 3.2. Horton's algorithm [10] with carefully chosen edge weights was used to select a cycle basis guaranteed to minimize the exponential component of the algorithm.

4 Computational Results

A graph G is a *topological obstruction* if it has no degree two vertices and removing any edge from the graph results in a graph with smaller genus. If a topological obstruction G also satisfies the property that the contraction of any edge results in a graph with lower genus, then we say that G is also a *minor order obstruction.*

It was conjectured by Wagner in [21], and proved by Robertson and Seymour in [18], that the number of minor order obstructions to toroidality is finite. So far, we have succeeded in enumerating only a partial list of these.

We applied Brendan McKay's **makeg** program based on his graph isomorphism tester **nauty** [14] to generate isomorphically distinct small graphs. The graphs tested were blocks with no degree two vertices since all obstructions which are not a block are composed of blocks which are one of the two planarity obstructions ($K_{3,3}$ or K_5). The results are summarized in Figures 3 to 7. The obstructions of order 8 are all both topological and minor order obstructions. They are pictured in Figure 4 and they provide nice examples of the indications for non-toroidality discussed in Section 2.

Figure 3: 2-connected obstructions to toroidality—8 vertices

number of edges	19	20	21	22	23	24	25		
number of obstructions	0	0	0	1	0	1	1	TOTAL	3

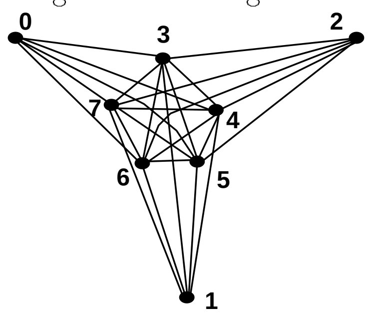
Time: several minutes on a Sparc station SLC to process 11,001 graphs.

Cursory searches for obstructions with more than 11 vertices were made. A twenty hour run on graphs with 12 vertices found 37 minor order obstructions, and the same amount of time on graphs with 13 vertices found a single minor order obstruction. (Eight minor order obstructions on 13 vertices were found in all. We suspect there are more.) Several hundred hours revealed no 14 vertex obstructions.

5 Correctness

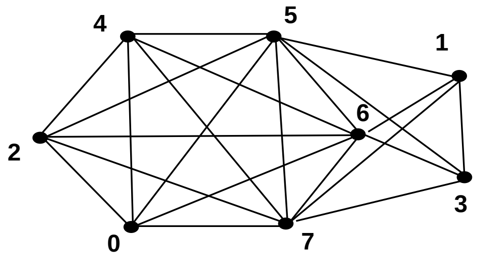
A claim of toroidality was checked by a very simple procedure which counts the faces of the combinatorial embedding, and we also checked that the graph indicated by the embedding was identical to the input graph. Claims of non-toroidality are harder to verify, but we have some evidence the program is behaving correctly. Enquiries about such lists generated by others have been fruitless (although a few examples from Decker's thesis, [3], have been checked, and a list of cubic irreducible graphs of genus one is rumoured to exist). Uncertainty about the computed results however, is inevitable, and perhaps the greatest value of the lists is precisely as a basis for comparison with lists produced by other algorithms, when they are forthcoming.

Figure 4: The eight vertex obstructions



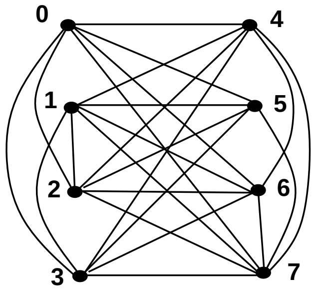
0: 3,4,5,6,7
1: 3,4,5,6,7
2: 3,4,5,6,7
3: 0,1,2,4,5,6,7
4: 0,1,2,3,5,6,7
5: 0,1,2,3,4,6,7
6: 0,1,2,3,4,5,7
7: 0,1,2,3,4,5,6

(Edge:vertex ratio of 25:8 > 3 means edge overflow)



0: 2,4,5,6,7
1: 3,5,6,7
2: 0,4,5,6,7
3: 1,5,6,7
4: 0,2,5,6,7
5: 0,1,2,3,4,6,7
6: 0,1,2,3,4,5,7
7: 0,1,2,3,4,5,6

(1,3 is an overused edge being in flat faces 1,3,7, 1,3,5, and 1,3,6)



0: 2,3,4,5,6,7
1: 2,3,4,5,6,7
2: 0,1,4,5,6,7
3: 0,1,4,5,6,7
4: 0,1,2,3,6,7
5: 0,1,2,3,7
6: 0,1,2,3,4,7
7: 0,1,2,3,4,5,6

(flat faces 1,3,5, 1,2,5, 0,3,5, and 0,2,5 create a dwarf rim about 5)

Figure 5: 2-connected obstructions to toroidality—9 vertices

# edges	19	20	21	22	23	24	25	26		
# topol. obstruc.s	2	4	2	9	17	6	2	5	TOTAL	47
# minor o. obstruc.s	2	4	2	9	13	6	2	4	TOTAL	42

Time: ten hours on a Sparc station IPC to process 260,536 graphs.

Figure 6: 2-connected obstructions to toroidality—10 vertices

# edges	19	20	21	22	23	24	25	26	27	28	29	30		
# top. obs.	14	8	34	40	190	170	102	75	21	1	0	1	TOTAL	656
# m. o. obs.	14	2	18	31	117	90	92	72	17	1	0	1	TOTAL	455

Time: about 4000 hours on a mixture of SUN 4's for about 15 million graphs.

The lists of toroidal obstructions on eight, nine, and ten vertices computed by an early version (August '92) of the program have been confirmed by a sequence of later versions, *even though* each later version uncovered bugs in some of the oldest routines. Similarly for the lists of minor order obstructions as of September '92.

6 Future Research

Our work is only a first attempt at developing a practical program for torus embedding. We feel that we have been very successful as it has proven a very effective tool for enumerating torus obstructions (our initial goal). But obviously there is much room for future extensions to this work.

Now that code for torus embedding exists, this opens the door to the development of algorithms which are fast for toroidal graphs in spite of being intractible for graphs of arbitrary genus. Further, we now have a tool for network designers desiring to consider toroidal architectures.

It remains to be seen whether this program or an improved version is faster than the programs resulting from the newly announced linear time algorithms. We suspect that due to large constant overhead, our approach may be competitive, at least for small graphs.

We found thousands of torus obstructions, but it seems likely that there are many that we have not found. A bound on their maximum order is needed, or perhaps an alternate approach to enumerating them such as the novel ideas of Fellows and Langston [6]. Further, since any coding process is subject to error, an independent verification of our obstruction list is in order.

Figure 7: Some 2-connected obstructions to toroidality—11 vertices

# edges	17	18	19	20	21	22	23	24	...		
# top. obs.	0	4	1	47	87	269	892	1878	...	(SUB)TOTAL	3178
# m. o. obs.	0	4	1	0	46	131	569	998	...	(SUB)TOTAL	1749

Time: about 5000 hours on a mixture of SUN 4's.

The most interesting extension to this work would be a simple algorithm guaranteed to finish in polynomial time. A more modest step is to improve the efficiency of various components of the algorithm. Our goal at the time was to keep the code simple and to get something working, so efficiency has often been compromised to correctness and speed of development. Some other suggestions for improvements likely to make a big difference have been suggested in this paper.

More rigorous testing on larger graphs is definitely needed. Although testing on the small graphs has been comprehensive to the point indicated, we have not examined in detail the performance on larger graphs. One open question is how best to generate graphs for testing purposes that should prove a challenge to the program and to other torus embedders. The n-vertex graphs having $3n-1$ edges were a sore spot for our program.

References

[1] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ–tree algorithms. *J. Comp. Sys. Sci.*, 13:335–379, 1976.

[2] N. Chiba, T. Nishizeki, A. Abe, and T. Ozawa. A linear algorithm for embedding planar graphs using PQ–trees. *J. Comp. Sys. Sci.*, 7:354–375, 1985.

[3] R. W. Decker. *The Genus of Certain Graphs.* PhD thesis, The Ohio State University, 1978.

[4] G. Demoucron, Y. Malgrange, and R. Pertuiset. Graphes planaires. *Rev. Française Recherche Opérationnelle*, 8:33–47, 1964.

[5] H. Djidjev and J. Reif. An efficient algorithm for the genus problem with explicit construction of forbidden subgraphs. *STOC*, pages 337–347, 1991.

[6] M. R. Fellows and M. A. Langston. An analogue of the Myhill-Nerode theorem and its use in computing finite-basis characterizations. *FOCS*, pages 520–525, 1989.

[7] I.S. Filotti. An algorithm for embedding cubic graphs in the torus. *J. Comp. Sys. Sci.*, 2:255–276, 1980.

[8] I.S. Filotti, G.L. Miller, and J. Reif. On determining the genus of a graph in $o(v^{O(g)})$ steps. *STOC*, pages 27–37, 1979.

[9] J. Hopcroft and R. Tarjan. Efficient planarity testing. *J. ACM*, 21:549–568, 1974.

[10] J. D. Horton. A polynomial time algorithm to find the shortest cycle basis of a graph. *SIAM J. Comp.*, 16:358–366, 1987.

[11] M. Juvan, J. Marincek, and B. Mohar. Embedding a graph into the torus in linear time. preprint, 1994.

[12] P. N. Klein and J. H. Reif. An efficient algorithm for planarity. *FOCS*, pages 465–477, 1986.

[13] A. Lempel, S. Even, and I. Cedarbaum. An algorithm for planarity testing of graphs. In P. Rosenstiehl, editor, *Theory of Graphs*, pages 215–232, New York, 1967. (Proc. Int. Symp. Rome, July 1966), Gordon and Breach.

[14] B. D. McKay. Practical graph isomorphism. *Proc. Tenth Manitoba Conf. Numerical Math. and Computing, Cong. Num.*, 30:45–87, 1981.

[15] B. Mohar. Embedding graphs in an arbitrary surface in linear time. *STOC*, pages 392–397, 1996.

[16] E. Neufeld. Practical toroidality testing. Master's thesis, Department of Computer Science, University of Victoria, Victoria, B.C., 1993.

[17] V. Ramachandran and J. Reif. An optimal parallel algorithm for graph planarity. *FOCS*, pages 282–287, 1989.

[18] N. Robertson and P. Seymour. Graph minors XVI. Wagner's conjecture. To appear.

[19] N. Robertson and P. Seymour. Generalising Kuratowski's theorem. *Cong. Num.*, 45:129–138, 1984.

[20] N. Robertson and P. Seymour. Graph minors - a survey. In *Surveys in Combinatorics*. Cambridge University Press, 1985.

[21] K. Wagner. Über einer eigenschaft der ebener complexe. *Math. Ann.*, 14:570–590, 1937.

[22] S. G. Williamson. Embedding graphs in the plane- algorithmic aspects. *Ann. Disc. Math.*, 6:349–384, 1980.