

A polynomial time algorithm for determining zero euler petrie genus of an eulerian graph

Brigitte and Herman Servatius
Department of Mathematics
Worcester Polytechnic Institute
Worcester, MA 01609-2280

Abstract

A dual-eulerian graph is a plane graph which has an ordering defined on its edge set which forms simultaneously an Euler circuit in the graph and an euler circuit in the dual graph. Dual-eulerian graphs were defined and studied in the context of silicon optimization of cmos layouts. They are necessarily of low connectivity with exponentially many planar embeddings. We give a polynomial time algorithm to answer the question whether or not a planar multigraph admits an embedding which is dual eulerian and constructs such an embedding, if it exists.

1 Introduction

Let G be a connected graph and let (G, ρ) be a map with embedding $\rho : G \rightarrow S$ of G into an orientable surface S . Let $g(S)$ denote the *genus* of S , and $g(G)$ denote the *genus of G* , that is, the least genus of any surface S such that (G, ρ) is a map. A *petrie circuit* (walk) is a circuit (walk) in a map which turns alternately left and right, see Figure 1. We will often indicate a petrie walk by showing the turns at each vertex by small circular arcs. Obviously the reverse of a petrie circuit (walk) is also a petrie walk.

Petrie circuits and walks have many interesting properties and have been used in many contexts, [3]. One useful property of petrie circuits (walks) is that the same sequence of edges is also a circuit (walk) in the dual map

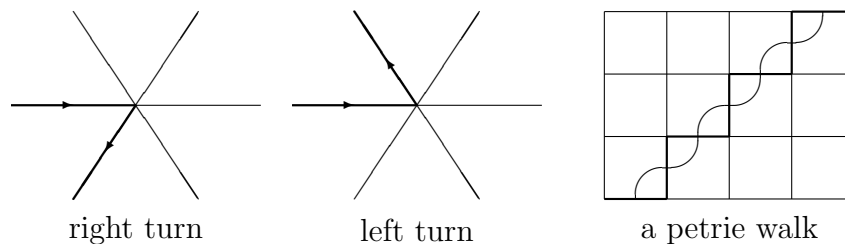


Figure 1:

$(G, \rho)^*$. A graph containing an eulerian circuit which is also a euler circuit of the dual is called *dual-eulerian*. Note that, to detect a dual-eulerian graph, it is not enough to check the graph and dual graph be eulerian, for instance the graph in Figure 2a is not dual-eulerian, while that in Figure 2b is.

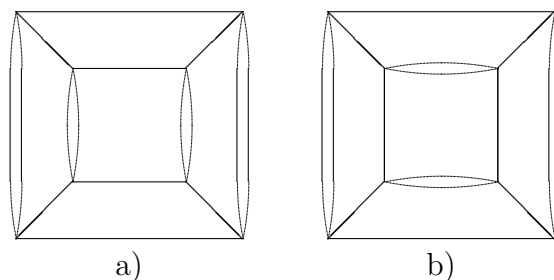


Figure 2:

If ρ is a strong embedding, that is, every closed face is homeomorphic to a closed disk, then the converse is also true. Note that a tame embedding of a graph in the plane is strong if and only if the graph is 2-connected.

A common optimization problem in VLSI design of cmos circuits has as a solution a dual-eulerian path or circuit, which, for 2-connected plane graphs, is equivalent to an euler petrie path or circuit. Although it is possible to decide in linear time whether a plane graph has an euler petrie circuit by simply starting from any edge and proceeding left-right-left-... and, if that is not successful, right-left-right-..., seeing if either of these two petrie walks generate an euler petrie walk or circuit. It is much more difficult for a non-embedded graph G to decide if it has an embedding with an euler petrie path or circuit. A more interesting and useful question is whether a planar graph has an embedding which has an euler petrie path. In [1] it is pointed out that more efficient layouts for cmos functional cells can be determined by deciding whether or not a planar multigraph admits a dual eulerian embedding.

Given an eulerian graph G and an euler circuit (path) p , it is always possible [7] to find a map (G, ρ) such that p is an euler petrie circuit (path.) It is natural to define the *euler petrie genus* of a graph G , $epg(G)$ to be the least genus of any surface in which G has an embedding with an euler petrie circuit or path.

EXAMPLE 1 K_5 is eulerian with genus 1 and euler-petrie genus 1, with euler petrie circuit $\{0, 1, 4, 0, 3, 4, 2, 3, 1, 2, 0\}$. See Figure 3.

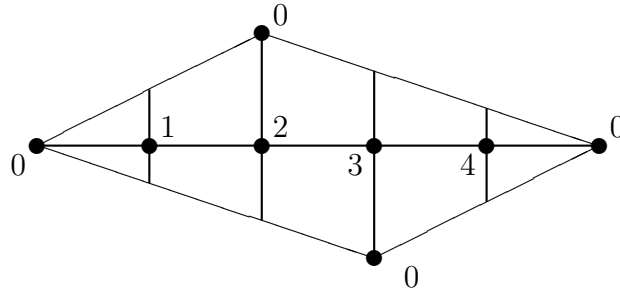


Figure 3:

EXAMPLE 2 K_7 has genus 1 but euler petrie genus at least 3, since the torus embedding is unique and any embedding of K_7 into a surface of genus 2 has at least 6 triangles.

Computing the euler petrie genus of a graph seems extremely difficult, [10, 1, 6, 11, 13] Even the euler petrie genus of complete graphs is not known. In this paper we give a polynomial time algorithm for to solve the question of whether an eulerian graph has euler petrie genus 0. This will solve the recognition problem for 2-connected dual-eulerian graphs, and in particular the case of series parallel networks. In what follows we will only consider euler petrie circuits, the case of paths being similar.

2 The 3-block tree

A planar graph G with an euler petrie circuit cannot be 3-connected (in the sense of Tutte), since all triangle free simple planar graphs must have at least 4 vertices of valence at most three. This means that every eulerian graph G with euler-petrie genus 0 must have a non-trivial 3-block tree.

The 3-block tree [9, 2] encodes the structure of a 2-connected graph in terms of the operation of edge amalgamation, in which two disjoint graphs are joined along an (oriented) edge and then that edge is erased. See Fig-

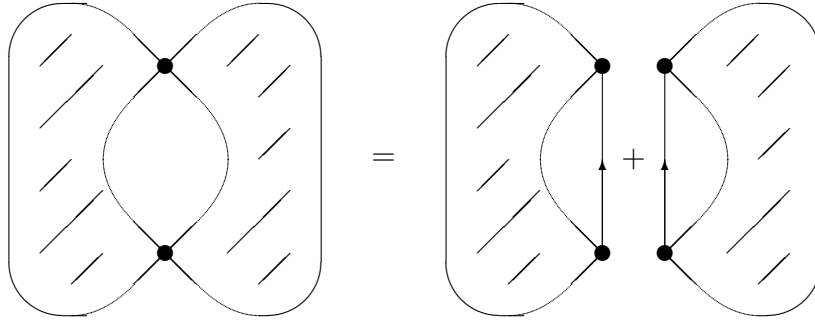


Figure 4:

ure 4. The atoms of the construction, called *3-blocks*, are simple 3-connected graphs, cycles and multilinks (a multilink is the planar dual of a cycle), each with at least three edges. Tutte [9] showed that every 2-connected graph is uniquely encoded as a tree of edge amalgams of 3-blocks, in which the nodes of the tree correspond to a disjoint collection of 3-blocks and each edge corresponds to a pair of (oriented) edges from the 3-blocks to which it is incident, and satisfying the conditions that no edge in any 3-block is amalgamated more than once, no pair of cycles is amalgamated together, and no pair of multilinks is amalgamated together. See Figure 5. Tarjan [8] showed that

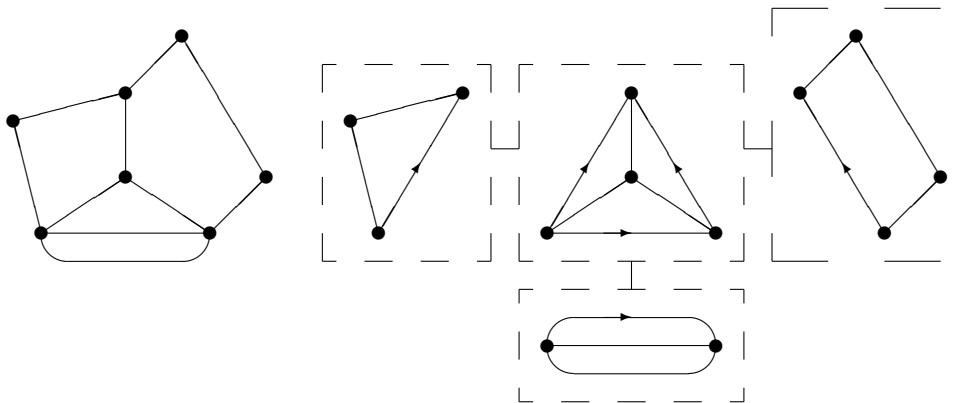


Figure 5:

Tutte's 3-block tree can be constructed in polynomial time.

Note that a 2-connected graph is planar if and only if each of its 3-blocks is planar. Moreover, a 3-connected 3-block is uniquely embeddable in the plane (in two ways counting orientation) and each multilink with m edges has $(m - 1)!$ distinguishable embeddings, so a 3-block tree all of whose 3-connected 3-blocks are planar is planar with $2^\alpha \prod (m_i - 1)!$ different embeddings, where α is the number of 3-connected 3-blocks, and m_1, m_2, \dots are the sizes of the multilinks. Thus a brute force attack on the question of whether $epg(G) = 0$ would require us to check each of these embeddings for an euler petrie circuit even if we know in advance that G is planar and eulerian. Such an algorithm would be exponential in the number of leaves in the 3-block tree, hence in the number of vertices of G .

3 The topology of euler petrie circuits.

We have noted that a planar graph with an euler petrie circuit must have a non-trivial 3-block tree, and so has a 2 vertex cutset. In this section we analyze the possible topologies of the dual eulerian paths as they pass across and back over this two vertex cutset.

Given a 2-cutset in a plane graph, it corresponds to two faces (not uniquely determined) which form a 2-cutset in $(G, \rho)^*$. The only way an euler petrie path can pass from left to right between the two lobes of the plane graph is to make one of the four turns indicated in Figure 6. The cir-

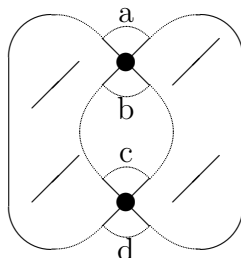


Figure 6:

cuit must use either two or four of the possible entry and exit turns. If only two are used, then we will replace the right lobe with a single edge labeled to indicate the topology of traversal as either p , (petrielike), f , (facelike) or s , spurlike. Specifically, if the entry and exit turns are a and c , then by a Whitney flip, [12] this is equivalent to b and d and the lobe is petrielike, see

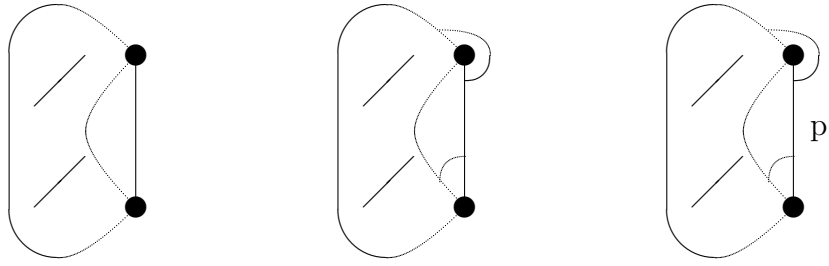


Figure 7:

Figure 7. Similarly if the entry and exit turns are a and d (b and c) then the lobe is face-like, see Figure 8. If the exit and entry turns are a and b,

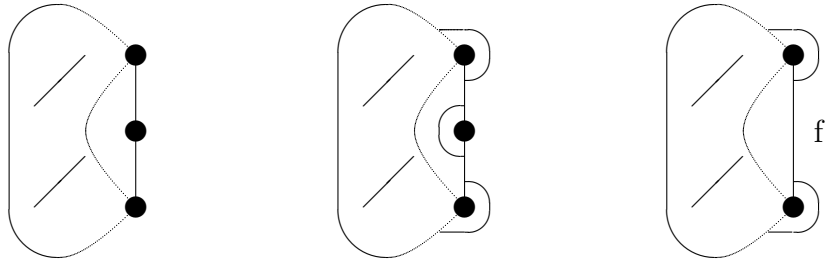


Figure 8:

so a whitney flip only replaces the tour with its reverse, then we say it is spur-like and we indicate the topology by an oriented edge, see Figure 9 It

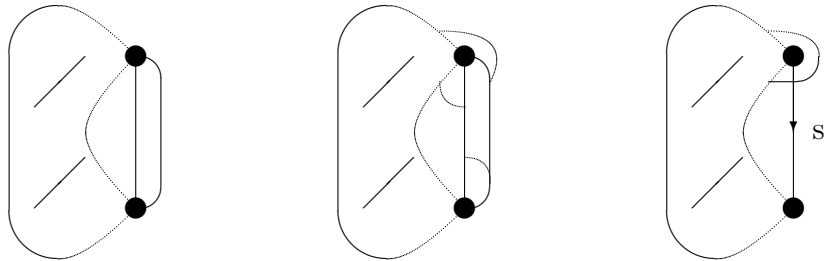


Figure 9:

may happen that there are two petrie tours possible, starting and ending with the complementary pair of turns, in which case, both are petrie-like, both are face-like, or both are spur-like. If both are spurlike we do not orient the edge labeled s.

The other possibility is that the euler circuit enters the right lobe twice, each segment of the circuit covering a disjoint set of edges in the lobe. In the case that segments have bounding turns (a,d), and (b,c), then the topology of the circuit is like that of Figure 10 and we label it $2f$. If the two segments use

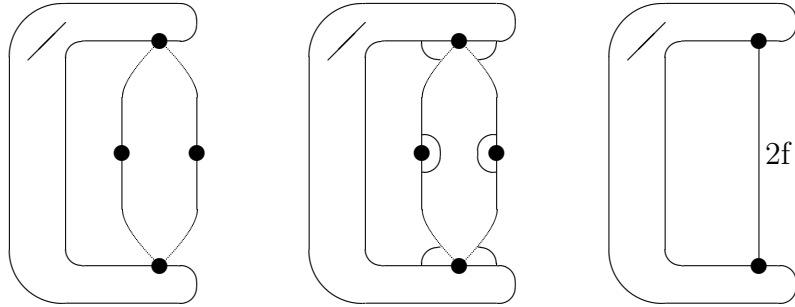


Figure 10:

turns (a,b) and (c,d), then the topology is that of two spurs, see Figure 11, and we label it with $2s$. The case $2p$, with segments bounded by (a,c) and

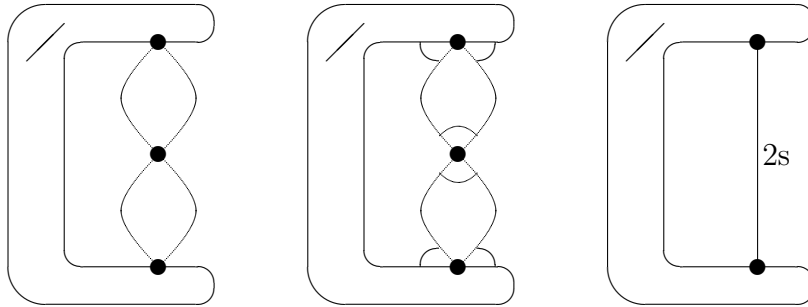


Figure 11:

(b,d) cannot occur in the case of euler petrie circuits, since two petrie paths cannot cross one another transversely without sharing at least one edge in common.

DEFINITION 1 Let (G, ρ) be a map whose edges are labeled from the set $\{p, f, s, 2f, 2s\}$ with the s edges perhaps oriented. A pfs -euler circuit is a closed tour which traverses all the edges of G labeled p and f once, those labeled s , and $2f$ twice, and those labeled $2s$ four times, each according to the turning instructions indicated by its label.

THEOREM 1 *Let G be a graph whose edge labels are taken from the set $\{p, f, s, 2f, 2s\}$. Then we can decide if G has a planar embedding with a pfs -euler tour in polynomial time.*

PROOF: We may assume that G is planar, eulerian, and, using Tarjan's algorithm, that we have constructed the 3-block tree. Consider a pendant 3-block P .

Case 1: P is 3-connected as in Figure 12. If G has a pfs -euler circuit in

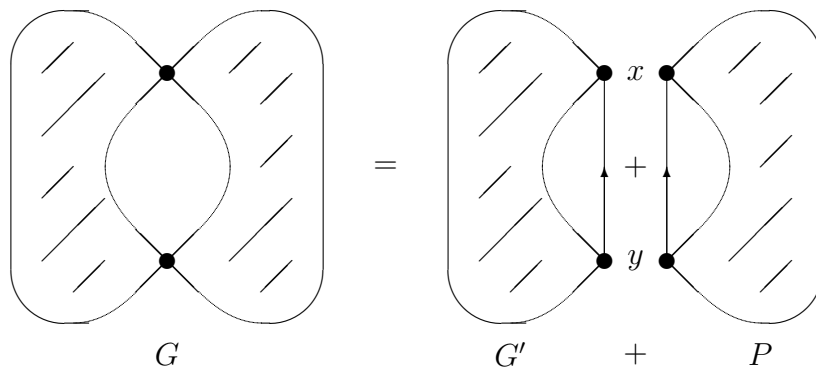


Figure 12:

some planar embedding, then in the unique embedding of P there is a pfs -path from x to y covering all the the edges of P except the edge connecting x and y , which is unlabeled. In polynomial time we may embed P and simply check the two possible pfs -paths starting at the edges incident to x and y and lying closest to (x, y) . If neither of these work, then the answer is no, and we may stop. If the answer is yes, then, as we have seen the topology of the path corresponds to just one of the labels in the set $\{p, f, s, 2f, 2s\}$. We label the edge (x, y) in G' with that label and proceed to check G' .

Case 2: The pendant node corresponds to a cycle, as in Figure 13. Here we may proceed as in the previous case except that we need expend no time either looking for an embedding or choosing between two paths. We pause here to note which label configurations are consistent with pfs -euler circuits and which topologies they correspond to. For overall topologies p and f for the circuit, we can have only p and f labels on the individual edges of P , with an odd number of p 's if the circuit has topology p and an even number for f . For the circuit to have topology s we need all edges of P labeled $2f$ except one labeled s , which will be incident with x or y and oriented toward

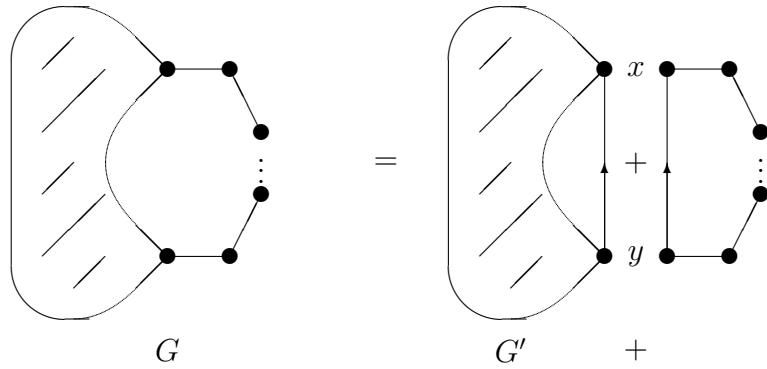


Figure 13:

it. For the circuit to have topology $2f$, we must have all P labels $2f$. Lastly, for a $2s$ topology we need all P labels $2f$ except for either a $2s$ in the middle, or two s 's oriented toward one another.

Case 3: The pendant node corresponds to a multilink, see Figure 14. In this case, if the multilink has m edges, there are $(m - 1)!$ embeddings

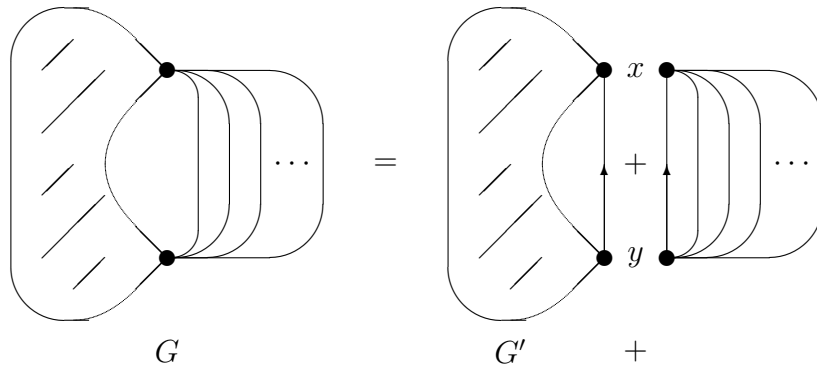


Figure 14:

and we have to be careful that no re-embedding of P can give a pfs -path with a different topology, since otherwise, as the algorithm proceeds we will have to check and recheck these possibilities and our algorithm will become exponential.

Fortunately, the possibilities are easily seen to be dual to those of the previous case: For topology p and s the multilink can only have p 's and s 's as labels, with an odd number of p 's for topology p and an even number for s . For topology f we need all edges labeled $2s$ except one f on the far right or

far left of the multilink. For topology $2s$ we must have all labels $2s$. Lastly, for topology $2f$ we need all edge labels to be $2s$ except for either a $2f$ in the middle, or two f 's.

Since the label sets of these topologies are disjoint, we immediately see not only which topologies are possible, but the only possible embeddings which will realize it.

The algorithm now proceeds by pruning the 3-block tree one pendant node at a time. \square

Now, to decide whether a graph has an euler petrie circuit, we simply label all the edges with p and start the algorithm described above.

THEOREM 2 *There is a polynomial time algorithm to decide whether a graph has euler-petrie genus zero.*

References

- [1] B. Carlson, C. Chen, and C. Meliksetian, *Dual eulerian properties of plane multigraphs*, SIAM J. Disc. Math., vol. 8, No. 1, pp. 33–50, 1995.
- [2] C. Droms, B. Servatius, and H. Servatius, *The structure of locally finite 2-connected graphs*, Electronic Journal of Combinatorics # R17, (1995).
- [3] H. Fleischner, “Eulerian graphs and related topics”, North Holland, Amsterdam, 1990-1991.
- [4] Y. Huang and M. Sarrefzadeh, *A parallel algorithm for minimum dual cover with application to cmos layout*, J. Circuits, Systems Comput., 1, pp. 177-204, 1991.
- [5] R. Maziasz and J. Hayes, *Layout optimization of static cmos functional cells*, IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, **9**, pp. 708-719, 1990.
- [6] R. Nair, A. Bruss, and J. Reif, *Linear time algorithms for optimal cmos layout*, in VLSI: Algorithms and Architectures, P. Bertolazzi and F. Luccio (eds.), Elsevier, North Holland, Amsterdam, pp. 327-338, 1985.
- [7] B. Servatius and H. Servatius, *Dual Eulerian Graphs*, Preprint.

- [8] J. E. Hopcroft and R. E. Tarjan, *Dividing a graph into triconnected components*, SIAM J. Comput. **3**, 2, 135–158, 1973.
- [9] W. T. Tutte, *Connectivity in Graphs*, University of Toronto Press (1966).
- [10] T. Uehara and W. Van Cleemput, *Optimal layout of cmos functional arrays*, IEEE Trans Comput., C-30, pp. 305-314, 1081.
- [11] S. Ueno, K. Tsuji, and Y. Kajitani, *On dual eulerian paths and circuits in plane graphs*, in Proc. of International Symposium on Circuits and Systems, pp. 1835-1838, 1988.
- [12] H. Whitney, *2-isomorphic graphs*, Amer. J. Math. **55**, 1933, 245–254.
- [13] S. Wimer, R. Pinter, and J. Feldman, *Optimal chaining of cmos transistors in a functional cell*, IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, CAD-6, pp. 795-801, 1987.