1987

Designing Networks with Compact Routing Tables AVAILABLE IN A JOURNAL

Greg N. Frederickson
Purdue University, gnf@cs.purdue.edu

Ravi Janardan

Report Number:

87-687

DESIGNING NETWORKS WITH COMPACT ROUTING TABLES

Greg N. Frederickson
Ravi Janardan

CSD-TR-687
May 1987

DESIGNING NETWORKS WITH COMPACT ROUTING TABLES

Greg N. Frederickson *
Ravi Janardan †

Department of Computer Science
Purdue University
West Lafayette, Indiana 47907

* The research of this author was supported in part by the National Science Foundation under grant DCR–8320124, and by the Office of Naval Research on contract N 00014-86-K-0689.

† The research of this author was supported in part by the National Science Foundation under grant DCR–8320124.

0. Abstract. Classes of network topologies are identified in which shortest path information can be succinctly stored at the nodes, if they are assigned suitable names. The naming allows each edge at a node to be labeled with zero or more intervals of integers, representing all nodes reachable by a shortest path via that edge. Starting with the class of outerplanar networks, a natural hierarchy of networks is established, based on the number of intervals required. The outerplanar networks are shown to be precisely the networks requiring just one interval per edge. An optimal algorithm is given for determining the labels for edges in outerplanar networks.

1. Introduction

The routing of messages between pairs of nodes is a basic activity in a distributed network. Assuming a cost function on the edges of the network, it is desirable to route each message along a shortest path. The most straightforward approach is to maintain a complete routing table at each of the n nodes, which gives for each potential destination the name of the next node on a shortest path to the destination. This approach requires that $n-1$ items of routing information be stored at each node in the network, with each item being a node name. If the network is dense and of irregular topology, then one would not expect to be able to do appreciably better spacewise than using complete routing tables. However, for sparse networks, is it possible to maintain $o(n^2)$ items of routing information in the network and still achieve shortest path routings? We examine this question in the context of being free to assign $\log n$-bit names to the nodes. We present a node naming and message routing scheme that can handle broad classes of networks with arbitrary nonnegative costs on the edges. The scheme groups networks, starting with the class of outerplanar networks, into a natural hierarchy based on the amount of space devoted to storing routing information.

Routing schemes for tree networks and shortest path routing schemes for unit-cost ring networks have been presented in [SK], and later in [vLT1]. The nodes are assigned names from 1 to n, and the end of every edge $\{v, w\}$ incident with any node v is labeled with a subinterval of $[1, n]$, with wraparound allowed in the subinterval. The interval represents the set $R_{v,w}$ of nodes such that there is a shortest path from v to each node in $R_{v,w}$ with the first edge on this path being $\{v, w\}$. In [vLT2] interval labeling schemes are also given for certain highly regular networks with edges of unit cost, including complete graphs, complete bipartite graphs, and grids.

We first give an interval labeling scheme for outerplanar networks with arbitrary nonnegative costs on the edges. The scheme stores just d items of routing information at every node v, of degree d. Thus $\Theta(n)$ items of routing information are stored in total. For arbitrary nonnegative edge costs, we show that the outerplanar graphs are precisely the graphs for which such an interval property holds. Furthermore we establish a very nice 'reflection' property of outerplanar graphs. Using this property we are able to generate an optimal algorithm for determining the labels of all edges.

In [vLT2] a k-interval labeling scheme is proposed, in which each edge is labeled with up to k intervals encoding shortest paths. A 2-interval labeling scheme is given in [vLT2] for a unit-cost grid with row- and column-wraparound. We present a result that is stronger in that it is applicable to graphs of irregular topology and allows arbitrary nonnegative edge costs. We show that k-interval labeling schemes, for $k > 1$, can handle classes of graphs much richer than the class of outerplanar graphs. Thus, our results can be helpful in designing a network when small routing tables are desirable.

In particular, we establish that any graph that can be embedded in the plane such that all but q of the vertices are on p faces has a $\lfloor (3p+q)/2 \rfloor$-interval property. In fact, the number of items of routing information required at a node of degree d is at most $3p + q + d - 2$. If the p faces form s connected components, we show that the graph has a $\lfloor (2p + s + q)/2 \rfloor$-interval labeling, with a total of at most $2p + s + q + d - 2$ intervals at any node. Our approach can also be applied to graphs that can be embedded on a surface of positive genus g. We show that such graphs possess a $\lfloor(2p+s+q+4g)/2\rfloor$-interval labeling, and use a total of at most $2p+s+q+4g+d-2$ intervals at any node. In addition, our labeling scheme can be adapted naturally to planar graphs in which we wish to route messages to vertices on only a selected subset of the faces. This technique, together with other ideas, can then be used to generate a near-shortest paths routing scheme for general planar graphs [FJ2].

We are thus able to handle conveniently networks with more desirable properties than either tree or ring networks alone. Since our networks allow more interconnections than ring networks, distances between nodes can be made smaller than in ring networks. Since we can handle networks that are at least biconnected, a single node or edge fault will not disconnect the network, as is the case with trees. In fact, in [FJ3] we present a space-efficient and fault-tolerant routing scheme for outerplanar networks. For any combination of node and edge faults that do not disconnect the network, the scheme restores near-optimal routings by storing at each node only a constant amount of additional information per fault.

A preliminary version of this paper appeared in [FJ1].

2. Routing in outerplanar networks

We first summarize the interval routing method presented in [SK] for trees and rings. The nodes are named appropriately with the integers from 1 to n. For trees, the names are depth-first numbers. For rings, the names are assigned consecutively, going clockwise around the ring. For any vertex v of degree d, let $w_1, w_2, \ldots, w_d$ be the neighbors of v indexed in clockwise order around the exterior face starting from v. Each edge incident with v is labeled by an interval, with the intervals from all edges incident with v forming a partition of $[1,n]-v$. Wraparound is allowed in the intervals. For instance, the interval $[i, j)$, $i > j$, contains $\{i, i+1, \ldots, n, 1, \ldots, j-1\}$. Denote the intervals by $[l_i, l_{i+1})$, for $i = 1, 2, \ldots, d$, where $l_{d+1} = v$. Without loss of generality, assume that interval $[l_i, l_{i+1})$ labels edge $\{v, w_i\}$. The values l_i, $i = 1, 2, \ldots, d$, are stored in a table at node v, each with a pointer to associated edge $\{v, w_i\}$. When a message arrives at node v, if its destination u is not equal to v, then the table is searched for the entry l_i such that $l_i \leq u < l_{i+1}$. The message is then sent out on edge $\{v, w_i\}$. Since the values l_i, $i = 1, 2, \ldots, d+1$ form a rotated list [MS, F2], the table can be searched in $O(\log d)$ time using a modified binary search.

The interval labeling method also works for outerplanar networks if the nodes are named appropriately. An *outerplanar network* is a network that can be embedded in the plane such that all nodes lie on one face [H]. Throughout this paper, we consider outerplanar networks in the context of such an embedding, called an *outerplane embedding*. We assign as names to the nodes the integers from 1 to n in consecutive order starting at an arbitrary node and proceeding clockwise around the exterior face. If any node v is visited more than once in this traversal, implying v is an articulation point of the network, then v may be named on any one of the visits. We call such a naming of the nodes a *clockwise* node naming. An outerplanar network with a clockwise node naming is shown in Figure 1a. We first show that for any assignment of costs to edges, each end of every edge can be labeled with an interval such that any message is routed along a shortest path. Such a labeling of the edges of the network of Figure 1a is shown in Figure 1b.

Theorem 2.1. Let G be an n-vertex outerplanar graph with a clockwise naming of its vertices. For any assignment of nonnegative costs to its edges, the end of every edge incident with any vertex v can be labeled with a subinterval of $[1,n]$ such that the edge is the first edge on a shortest path from v to any vertex in the subinterval.

Proof. Identify a set S of shortest paths in G such that there is a unique path in S between every pair of vertices, and any subpath of a path in S is also in S. Such a set of paths is produced, for instance, by Warshall's algorithm [W,AHU] if, for all pairs of vertices i and j, the first (i,j)-path of shortest length that is discovered in the algorithm is the shortest path chosen. If G is not biconnected, then it could be augmented with additional edges to yield G', as follows. For $i = 1, 2, \ldots, n$, if edge $\{i, (i \bmod n) + 1\}$ is not in G, include it in G' with very large cost. This forces the vertex names in G' to appear in order around the exterior face. If the costs are chosen large enough, then S will be a set of shortest paths for G' also.

Consider any neighbor w of v. If edge $\{v, w\}$ is not the first edge on the shortest path in S from v to any vertex, then clearly each end of this edge should be labeled with the empty interval. Otherwise, there is a nonempty set of vertices to each of which there is a shortest path in S from v whose first edge is $\{v, w\}$. Let u_1 and u_l be the first and last vertices in clockwise order around the exterior face from v in this set.

Suppose that for some vertex x in $[u_1, u_l]$, there were a path P_1 from v to x in S such that $\{v, w'\}$ is the first edge in P_1, for some $w' \neq w$. Since x is between u_1 and u_l on the exterior face, P_1 must cross the shortest path in S from v to one of u_1 and u_l. Call this path P_2. Let y be a vertex shared by P_1 and P_2. The subpaths of P_1 and P_2 that start at v and end at y are distinct, and thus cannot both be in S. This means that not both of P_1 and P_2 are in S, a contradiction. Thus $\{v, w\}$ is the first edge in the shortest path from v to each vertex in $[u_1, u_l]$. ∎

Thus, edge $\{v, w\}$ can be labeled at v by an interval of vertex names such that each vertex in the interval is reachable from v by a shortest path whose first edge is $\{v, w\}$. We say that $\{v, w\}$ *claims* these vertices at v.

3. Determining edge labels efficiently

We show how to determine efficiently the labels for the ends of all edges. Suppose that the graph is biconnected, and that edge costs satisfy the *generalized triangle inequality*, i.e., each edge is a shortest path between its endpoints. (At the end of this section we discuss how to handle graphs that do not satisfy these properties.) Edges $\{v, w_i\}$ and $\{v, w_{i+1}\}$ incident with vertex v claim adjacent sets of consecutive vertices around the exterior face. Let z be the farthest vertex from v in a counterclockwise direction around the exterior face that is claimed by $\{v, w_{i+1}\}$. We call z the *split vertex of* vertex v *relative to* neighbors w_i and w_{i+1}, or the *split vertex* for (v, w_i, w_{i+1}). Taking the split vertex for (v, w_0, w_1) to be w_1, and the split vertex for (v, w_d, w_{d+1}) to be v, the label on edge $\{v, w_i\}$ at v will be $[z, z')$, where z is the split vertex for (v, w_{i-1}, w_i), and z' is the split vertex for (v, w_i, w_{i+1}).

Using the above characterization, the split vertex for (v, w_i, w_{i+1}) can be found in $O(n)$ time, for any vertex v and neighbors w_i and w_{i+1}. Since there are $O(n)$ triples of the form (v, w_i, w_{i+1}) in an outerplanar graph, all split vertices can be found in $O(n^2)$ time. However, by exploiting a 'reflection' property of outerplanar graphs, an $O(n)$ time algorithm to find all split vertices can be generated.

We first generalize the discrete problem of determining all split vertices to a continuous version by viewing each vertex as a point and each edge as a continuum of points, with the distance function extended in the natural way. Let $\rho(p_1, p_2)$ denote the distance between points p_1 and p_2. If x is a point on an exterior edge

$e = \{w_1, w_2\}$, with $x \neq w_1$ and $x \neq w_2$, and w_1 following w_2 in a clockwise direction around the exterior face, we call w_1 and w_2 the neighbors of x. For any x, including points coinciding with vertices, and consecutive neighbors w_i and w_{i+1} of x, let x' be the point on an exterior edge such that $\rho(x, w_i) + \rho(w_i, x') = \rho(x, w_{i+1}) + \rho(w_{i+1}, x')$. We call x' the *split point of* point x *relative to* neighbors w_i and w_{i+1}, or the *split point* for (x, w_i, w_{i+1}).

We illustrate split points using Figure 1a. The split point for vertex 5 relative to neighboring vertices 1 and 4 is the point on edge $\{1, 2\}$ that is at distance 0.5 from vertex 2. The split point for the point on edge $\{4, 5\}$ at distance 2.5 from vertex 5 is the point on edge $\{1, 2\}$ that is at distance 3 from vertex 2.

Let u_j and u_{j+1} be consecutive neighbors of x' on the shortest paths realizing $\rho(w_{i+1}, x')$ and $\rho(w_i, x')$, respectively. Note that such neighbors can always be found, because the graph satisfies the generalized triangle inequality.

Lemma 3.1. (Reflection) Let x be any point, including an endpoint, on an exterior edge of outerplanar graph G, and let w_i and w_{i+1} be consecutive neighbors of x. Let x' be the split point for (x, w_i, w_{i+1}), and let u_j and u_{j+1} be consecutive neighbors of x' on the shortest paths realizing $\rho(w_{i+1}, x')$ and $\rho(w_i, x')$, respectively. Then x is the split point for (x', u_j, u_{j+1}).

Proof. Since x' is the split point for (x, w_i, w_{i+1}) and u_j and u_{j+1} are on shortest paths realizing $\rho(w_{i+1}, x')$ and $\rho(w_i, x')$, respectively, we have $\rho(x, w_i) + \rho(w_i, u_{j+1}) + \rho(u_{j+1}, x') = \rho(x, w_{i+1}) + \rho(w_{i+1}, u_j) + \rho(u_j, x')$. Note that w_i and w_{i+1} are on shortest paths realizing $\rho(u_{j+1}, x)$ and $\rho(u_j, x)$, respectively, since otherwise the generalized triangle inequality would not hold. Thus, $\rho(u_{j+1}, x) = \rho(u_{j+1}, w_i) + \rho(w_i, x)$ and $\rho(u_j, x) = \rho(u_j, w_{i+1}) + \rho(w_{i+1}, x)$. It follows that

$\rho(x', u_{j+1}) + \rho(u_{j+1}, x) = \rho(x', u_j) + \rho(u_j, x)$, i.e., x is the split point for (x', u_j, u_{j+1}). ∎

The idea behind the algorithm is to match up triples (x, w_i, w_{i+1}) and (x', u_j, u_{j+1}). Since there are infinitely many points, the matching is done collectively. A maximal sequence of exterior edges, with partial edges at either end of the sequence, is matched up with a similar sequence such that for any pair of points on one sequence, their split points are the same distance apart on the other sequence. If G contains just one interior face, then one sequence of edges suffices, with each point mapped to a point halfway around the face. Otherwise choose an interior face f with just one interior edge. Denote the vertices of f in clockwise order by $v_1, v_2, \ldots, v_r$, where $e_i = \{v_i, v_{i+1}\}$, $i = 1, 2, \ldots, r-1$, are exterior edges and $e_r = \{v_r, v_1\}$ is the interior edge. Let $\| f \|$ be the length of the boundary of f.

Let $y = \frac{1}{2} \| f \| - \| e_r \|$. Let the path of points from v_1 to v_r around the exterior face be split into three consecutive subpaths, P_1, P_2, and P_3, where P_1 and P_3 consist of closed sets of points, and are both of length y. First, match P_1 with P_3, relative to neighbors on f. Second, contract paths P_1 and P_3 to vertices v_1 and v_r, respectively. Note that the beginning and end of path P_2 may be in the middle of edges, say e_j and e_k, so that the contraction induces new edges e'_j and e'_k. Finally, delete e_r from G. Call the resulting graph G'. Recursively match up edge sequences in G'. Denote this algorithm as *CONTMATCH*.

Lemma 3.2. For any biconnected outerplanar graph G, algorithm *CONTMATCH* correctly matches up edge sequences such that corresponding points on two edge sequences are the split points of each other relative to appropriate neighbors.

Proof. The proof is by induction on i, the number of interior faces of G. Clearly, for $i = 1$, the matching is done correctly. For $i > 1$, consider some face f with precisely one interior edge. The points in P_1 have their split points in P_3, and vice versa. Thus these points are correctly handled.

For any point x on P_2, its split point x' relative to w_i and w_{i+1} in G will be the same in G'. This follows since the distance from x to any point not in $P_1 \cup P_2 \cup P_3$ will decrease by exactly y in the transformation from G to G'. For any point x such that neither x nor x' are on f in G, $\rho(w_i, x')$ and $\rho(w_{i+1}, x')$ will remain the same in G', since neither shortest path will use P_1 or P_3 in G. Furthermore, if one of these shortest paths does use e_r in G, then the corresponding shortest path in G' would have the same length, since the shortest path P_2 between v_1 and v_r has length $\| e_r \|$.

Thus the transformation from G to G' preserves the matching of points with their split points for all points not in P_1 or P_3. Since G' has one fewer interior face, *CONTMATCH* matches up edge sequences in G' correctly, by the induction hypothesis. Thus all points in G are correctly matched. ∎

The matching of edge sequences for the graph in Figure 1a is shown in Figure 2. Matching sequences are labeled with the same letter. Each edge sequence is also labeled with its length. Note that a sequence may either be open or closed at either end. A sequence is shown to be closed at one end if there is an up mark at that end. Note that sequences f are open at both ends, sequences b are open at one end and closed at the other, and sequences a and c are closed at both ends.

We now discuss the changes necessary to transform *CONTMATCH* into an algorithm that determines split vertices for every vertex in G. Consider a point x representing vertex v in G, with neighbors w_i and w_{i+1}. If the split point for

(x, w_i, w_{i+1}) falls on vertex z, then z is the split vertex for (v, w_i, w_{i+1}). Otherwise the split point falls on some edge $e = \{u_1, u_2\}$, where u_1 directly follows u_2 clockwise around the exterior face. By definition, the split vertex for (v, w_i, w_{i+1}) is u_1. Thus for each exterior edge $e = \{u_1, u_2\}$ in such orientation, we set $cwvertex(e)$ to u_1.

Algorithm *DISCMATCH* is similar to *CONTMATCH*. Values of $cwvertex(e)$ are supplied for every exterior edge e in G, along with the value $\| f \|$ for each interior face f. When *CONTMATCH* matches edge sequences in paths, *DISCMATCH* also processes the paths in the following way. It moves down P_1 and P_3, finding for each point x representing vertex v with neighbors w_i and w_{i+1} in one path, the split point x' for x, and the edge e that x' is on, in the other path. The algorithm then outputs the 4-tuple $(v, w_i, w_{i+1}, cwvertex(e))$.

When G' is formed, then $cwvertex(e'_j)$ should be set to $cwvertex(e_j)$, and similarly for edge e'_k.

Theorem 3.1. For any biconnected outerplanar graph G, algorithm *DISCMATCH* correctly determines the split vertex for each vertex v relative to neighbors w_i and w_{i+1}. Furthermore *DISCMATCH* uses $O(n)$ time.

Proof. Correctness follows from the correctness of *CONTMATCH* along with the correctness of the handling of the $cwvertex(e)$ values. We analyze the time complexity as follows. If G contains just one interior face, then linear time suffices to determine the split vertices. Otherwise consider the handling of an interior face f with exactly one interior edge. The time for this is proportional to $c_1 + c_2 n'$, where c_1 and c_2 are constants and n' is the number of vertices eliminated in the transformation from G to G'. We charge c_1 to f and c_2 to each vertex eliminated. Since there are $O(n)$ interior faces, the result then follows. ∎

If G is not biconnected, then apply *DISCMATCH* to each nontrivial biconnected component H of G and infer the split vertices as before. However, if some split vertex of H is an articulation point of G, then a straightforward application of the method for labeling edges from the split vertices will not yield the correct edge labels. For instance, suppose that edge $\{v, w_i\}$ is labeled with the interval $[z, z')$, where z is the split vertex for (v, w_{i-1}, w_i) and z' is the split vertex for (v, w_i, w_{i+1}). If z is an articulation point, then $\{v, w_i\}$ claims not only z, but also the vertices in the biconnected components attached to H at z. However, it is easy construct examples where the names of these vertices do not get included in $[z, z')$. On the other hand, if z' is an articulation point then the opposite problem can arise, where too many vertices are included in $[z, z')$.

The problem can be overcome as follows. Note that for each articulation point a in H, the names of the nodes in $G - (H - a)$ form an interval $[l(a), r(a)]$. Label the edges of H as before, except that for each split vertex that is an articulation point, use the corresponding $l(\cdot)$ instead of the name of the split vertex itself. All the intervals $[l(\cdot), r(\cdot)]$ can be identified in $O(n)$ time.

Any edges of G that do not satisfy the generalized triangle inequality can be removed by applying the following recursive algorithm to each biconnected component of G. Precompute for each interior face the cost of its boundary and identify the interior faces adjacent to it.

If the graph contains just one interior face, then find the maximum cost edge on its boundary. If the cost of this edge exceeds the cost of the remainder of the boundary, then delete this edge before returning. Otherwise let f be an interior face with exactly one interior edge e. If $\| e \| > \| f - e \|$, then delete e, coalesce the remaining portions of the two interior faces that shared e into a single face and recurse on the resulting graph. Otherwise, delete $f - e$, recurse on the resulting graph, and re-introduce $f - e$ into the graph G' remaining after this call. Let f' be the interior face defined by this re-introduction and let e' be a maximum cost edge of $f - e$. If $\| e' \| > \| f' - e' \|$, then delete e' before returning.

Theorem 3.2. Let G be an n-vertex outerplanar graph with nonnegative edge costs. The above algorithm enforces the generalized triangle inequality on G, and does so in $O(n)$ time.

Proof. We prove correctness by induction on the number i of interior faces of G. The basis case of $i = 1$ is immediate. The inductive hypothesis is that the algorithm works correctly if G has fewer than i interior faces, $i > 1$.

Suppose that G has i interior faces. If $\| e \| > \| f - e \|$, then the algorithm is called recursively on $G - e$, which has $i - 1$ interior faces. The theorem then follows from the inductive hypothesis. Otherwise, the algorithm is invoked recursively on $G - (f - e)$, which has $i - 1$ interior faces. By the inductive hypothesis, the graph G' returned from this call satisfies the triangle inequality. Let f'' be the portion of f' contained in G', and let e'' be any edge of f''. We claim that the addition of $f - e$ to G' cannot decrease the distance between the endpoints of e''. If f'' is e then the claim is immediate, since $\| e'' \| = \| e \| \leq \| f - e \|$. Otherwise, since e was deleted in the recursive call, we have $\| f'' \| \leq \| e \|$, and $\| e'' \| \leq \| f'' \| \leq \| f - e \|$ follows. Thus the claim holds, and e'' satisfies the triangle inequality in $G' \cup (f - e)$. This is also true of any other edge of G', since any path between its endpoints that uses $f - e$ must contain the endpoints of some edge of f''. Thus only edges of $f - e$ need be checked, and this is handled correctly.

The running time analysis is as follows. Information about interior faces can be precomputed in $O(n)$ time from a standard representation of an embedding [LT]. Any edge is deleted at most once, added back at most once, and examined once in finding a maximum cost edge. Constant work is done in updating information about faces when an edge or a portion of an interior face is deleted, since at most two faces are lost and at most one gained in the process. Thus the algorithm takes $O(n)$ time. ∎

4. The 1-interval property for graphs

In this section we characterize the class of graphs for which an interval labeling of edge ends exists. We consider two criteria for identifying these graphs. In the first, the vertices must be given names that will be appropriate for any assignment of edge costs. In the second, the edge costs are assumed fixed before vertices are named. The first case corresponds to an application in which the costs of network edges may change over time to reflect the state of network. When this happens, it is necessary to recompute the interval routing information. However, it is not reasonable to rename nodes.

An n-vertex graph has the *1-interval property* if there is a naming of its vertices with integers from 1 to n such that for every assignment of nonnegative costs to the edges the following holds. At each vertex, the end of every edge can be labeled by a subinterval of $[1, n]$, such that the edge is the first edge on a shortest path from the vertex to any vertex in the subinterval. The following theorem characterizes the graphs with this property.

Theorem 4.1. A graph has the 1-interval property if and only if it is outerplanar.

Proof. If a graph is outerplanar, then, by Theorem 2.1, it has the 1-interval property. To show the converse, we use a forbidden subgraph characterization of outerplanar graphs [CH] (see also [H] for a statement without proof), which states that a graph is outerplanar if and only if it does not contain a subdivision of K_4 or $K_{2,3}$. Here K_4 is the complete graph on four vertices and $K_{2,3}$ is the complete bipartite graph on sets of size two and three, and a *subdivision* of K_4 (resp. $K_{2,3}$) is a graph obtained by inserting zero or more vertices into the edges of K_4 (resp. $K_{2,3}$).

Let G be a graph that is not outerplanar and consider any naming of G. If G contains a subdivision H of K_4, then let $l_1 < l_2 < l_3 < l_4$ be the names of the vertices of K_4 in H. Assign costs to the edges of G such that each path in H obtained by subdividing one of the edges $\{l_1, l_2\}$, $\{l_1, l_3\}$, and $\{l_2, l_4\}$ of K_4 receives a total cost of 1, while each path obtained by subdividing one of the edges $\{l_1, l_4\}$, $\{l_2, l_3\}$, and $\{l_3, l_4\}$ of K_4 receives a total of cost 3. Then the shortest paths from l_1 to l_2, l_3 and l_4 are all unique, with the first edge on the paths to l_2 and l_4 being the same, but different from the first edge on the path to l_3. Thus no 1-interval labeling is possible at l_1.

Otherwise, G contains a subdivision H of $K_{2,3}$. Let $l_1 < l_2 < l_3 < l_4 < l_5$ be the names of the vertices of $K_{2,3}$ in H. Without loss of generality, let l_1 and l_u, $u \neq 1$, be the vertices from the size 2 bipartition of $K_{2,3}$. Let I_u be the set of even indices less than u and odd indices greater than u of vertices from the size 3 bipartition. Note that I_u has cardinality 2. Assign edge costs in G similarly as above, such that the shortest paths from l_1 to the other vertices of $K_{2,3}$ are unique, with the first edge on the paths to the vertices with indices in $I_u \cup \{u\}$ being the same, but different from the first edge on the path to the vertex whose index is not in $I_u \cup \{u\}$. Since the latter vertex and l_1 cannot be consecutive, even with wraparound, a 1-interval labeling at l_1 is not possible. ∎

If vertex names can be chosen *after* the assignment of edge costs is known, then the class of graphs for which each edge end at any vertex can be labeled by a single interval is slightly larger than the class of outerplanar graphs. We say that a graph has the *weak 1-interval property* if for each assignment of nonnegative edge costs, there is a naming of its vertices such that each edge end at any vertex can be labeled by an interval encoding shortest paths.

Theorem 4.2. A graph has the weak 1-interval property if and only if its biconnected components are either outerplanar or K_4.

Proof. Consider any graph G whose biconnected components are either outerplanar or K_4. Let nonnegative weights be assigned to its edges. If any edge of G does not satisfy the generalized triangle inequality, then label it at each end with the empty interval and delete it. Let G' be the resulting graph. We show that there is a naming of the nodes of G' such that a 1-interval labeling of its edges is possible. This labeling along with the empty labels on the deleted edges yields a 1-interval labeling of the edges of G. For the purpose of naming the vertices of G', remove an edge from each K_4 and generate an outerplane embedding of the resulting graph. Assign a clockwise naming with respect to this embedding. For any K_4 in G', each edge claims at one of its endpoints the nodes in the subgraph attached to the K_4 at the other endpoint. As the names of these nodes are the consecutive integers assigned between the first and last visits of the latter endpoint, they form an interval. For any outerplanar component of G', rename each articulation point of the component by an interval of integers which represents the nodes in the subgraph attached to the component at the articulation point. It then follows from Theorem 2.1 that each edge end of the component can be labeled by an interval.

For the converse, consider any graph G which has at least one biconnected component that is neither outerplanar nor K_4. This component must contain a subdivision of $K_{2,3}$. (We may ignore subdivisions of K_4 for the following reason. If the biconnected component contains a proper subdivision of K_4, then a subdivision of $K_{2,3}$ can be inferred from it. If the biconnected component contains K_4 itself, then the latter must necessarily be a proper subgraph of the component. A subdivision of $K_{2,3}$ can be inferred from the K_4 and some of the nodes and edges not in K_4.) Suppose that the subdivision is $K_{2,3}$ itself. Assign costs 1, 2 and 2 to the edges incident with one of the vertices v in the size 2 bipartition, and costs 1, 3 and 3 to the edges incident with the other vertex u, such that both edges of cost 1 are incident with the same vertex w in the size 3 bipartition. Assign cost 5 to the remaining edges of G. We claim that for no naming of G does there exist a 1-interval labeling of edge ends in G. Suppose to the contrary that there is such a naming. Let l_i, $1 \leq i \leq 5$ be the names assigned to the vertices of $K_{2,3}$, where, without loss of generality, v is named l_1, u is named l_2, and w is named l_3. Since $\{l_1, l_3\}$ claims l_3 and l_2 at l_1, and $\{l_2, l_3\}$ claims l_3 and l_1 at l_2, either l_1, l_3, l_2 or l_2, l_3, l_1 forms an interval. As $\{l_4, l_1\}$ claims l_1, l_3 and l_5 at l_4, it follows that either l_5, l_1, l_3, l_2 or l_2, l_3, l_1, l_5 forms an interval. Now, $\{l_5, l_1\}$ claims l_1, l_3 and l_4 at l_5, implying that some permutation of l_1, l_3, l_4 forms an interval, which is impossible.

The preceding argument can be extended to any proper subdivision of $K_{2,3}$ by assigning cost 0 to each edge of the subdivision not incident with either u or v. ∎

5. Multi-interval edge labelings

The property of any outerplanar graph that makes it amenable to interval routing is that there is an embedding such that all vertices are on the boundary of the same face. Any edge $\{v, w\}$ incident on v claims a set of consecutive vertices around this face. If a planar graph has an embedding in which two faces together contain all the vertices, then any edge will claim a subset of vertices from each face, consecutive around each face. Name the vertices around one face in turn, and then around the other face. Each edge will claim at most one interval on each face, except that for each face there can be one edge that claims some of the lowest and highest numbered vertices on the face. Since more than one face is needed to contain all the vertices, the lowest and highest numbered vertices will not be consecutive, even with wraparound, and so require two intervals. An example is shown in Figure 3, in which edges $\{1,6\}$, $\{4,5\}$, $\{5,12\}$, $\{12,13\}$, $\{13,7\}$, $\{7,8\}$, $\{3,9\}$, $\{9,10\}$, and $\{10,11\}$ have cost 1, and all other edges have cost 6. The edge labels are given for edges incident on vertex 2, where for example edge $\{2,8\}$ incident on vertex 2 is labeled with three intervals.

Thus it is possible to generate reasonable multi-interval labelings for edges in graphs in which there are just a few faces that contain all the vertices. Interval routing can be used as before in a corresponding network, with a short table containing the routing information stored at each node, except that now an edge $\{v, w\}$ incident on v may be associated with several different values of l_i in the table. In this section we explore the potential for such graphs, when the few special faces have no vertices in common.

Let $G = (V, E)$ be a connected graph with positive edge weights. Let $R_{v,w}$ be the set of vertices to which there is a shortest path from v containing edge $\{v, w\}$. Suppose that G is planar. Consider a plane embedding $G_e = (V, E, F)$ of G, where F is the set of faces. Let $C_{v,w}$ be a simple closed curve in the plane that separates $R_{v,w}$ from $G - R_{v,w}$, and crosses no edge twice. Let a *segment* of $C_{v,w}$ be a maximal continuous portion of $C_{v,w}$ that is contained in a face.

Lemma 5.1. At most one segment of $C_{v,w}$ is contained in any one face of G_e.

Proof. Since G is connected and v is in $G - R_{v,w}$, it follows that $G - R_{v,w}$ is connected. Since $C_{v,w}$ crosses no edge twice, any edge that it crosses has one endpoint in $R_{v,w}$ and the other not in $R_{v,w}$. Suppose there were a face such that two continuous segments S_1 and S_2 of $C_{v,w}$ are contained in it. Then a continuous curve S_3 can be drawn from an interior point of S_1 to an interior point of S_2, such that S_3 remains in the face, i.e, crosses no edge. But then $R_{v,w}$ is not connected, since there is a nonempty subset of it within each of the regions created by separating $C_{v,w}$ by S_3. ∎

We introduce the notion of a *trivial face*, whose boundary comprises a single vertex. Thus we include the possibility of having zero or more vertices, each of which creates an interval by itself. A *face covering* F' of the vertices of G_e is a set of faces, such that any vertex in G is on one of the faces in F'. A face covering F' is *disjoint* if no two faces in F' share a vertex. Assign the vertices names in order around each face in turn.

Theorem 5.1. Let G_e be a planar embedding with a disjoint face covering of p nontrivial faces and q trivial faces. The number of intervals needed to label any edge is at most $\lfloor (3p + q)/2 \rfloor$.

Proof. Let nontrivial face f have vertices $i_f, i_f + 1, \ldots, j_f$. It follows from Lemma 5.1 that $R_{v,w}$ either contains no vertices from f, contains all vertices in some interval $[i', j']$, or contains vertices in two intervals $[i_f, i']$ and $[j', j_f]$, where $i_f \leq i'$ and $i' + 1 < j' \leq j_f$.

Represent $R_{v,w}$, defined over the indices 1 to n, symbolically as $R'_{v,w}$ defined over the indices 1 to $3p + q$, in the following way. Represent face f, $f = 1, 2, \ldots, p$ by indices $3f - 2$, $3f - 1$, and $3f$. Let $R'_{v,w}$ contain $3f - 2$ if $R_{v,w}$ contains i_f, $3f$ if $R_{v,w}$ contains j_f, and $3f - 1$ if $R_{v,w} \cap [i_f + 1, j_f - 1] \neq \emptyset$. Also, let the vertex in any trivial face u, $u = 1, 2, \ldots, q$, be represented by index $3p + u$. The number of intervals in $R'_{v,w}$ and $R_{v,w}$ will be the same. But $R'_{v,w}$ can have at most $\lfloor (3p+q)/2 \rfloor$ nonadjacent intervals. ∎

We next bound the number of intervals labeling all edge ends at a vertex v of degree d, which gives a bound on the total number of items of routing information stored at a node of degree d in a corresponding network. Let F' be a face covering of the vertices of G_e, consisting of disjoint nontrivial faces. Let C_v be a collection of simple curves in the plane, with the following properties. The simple curves partition the plane into regions, each region containing one of the sets $R_{v,w_1}, R_{v,w_2}, \ldots, R_{v,w_d}$, where $w_1, w_2, \ldots, w_d$ are the neighbors of v, and with the boundary of each region containing v. Thus the only vertex contained in any of the curves is v. No curve is allowed to cross any edge of G_e more than once, and no two curves share more than their endpoints. Let F_v be those faces in F', each of which is crossed by a curve in C_v. We first derive an upper bound $r(d)$ on the number of crossings r_v that C_v makes with the boundaries of faces in F'.

To derive the bound $r(d)$, we generate from G_e, F_v, and C_v an embedded graph $\hat{G}_e$, called a *mimicking graph*, that describes the relevant features. Let a *crossing point* be a point where a simple curve of C_v crosses a boundary edge of some nontrivial face f in F_v. Also call v a crossing point if v is on the boundary of some face in F_v. Coinciding with the position of each crossing point x in G_e, include vertex u_x in $\hat{G}_e$. If v comprises a trivial face in G_e, then include v in $\hat{G}_e$ and let $\delta_v = 1$. Otherwise, let $\delta_v = 0$.

For each face f in F_v, include the following edges in $\hat{G}_e$. Include edges that connect, in order of x, the sequence of vertices u_x for face f into a simple cycle, thus tracing out the boundary of face f. For each portion of C_v outside of any face of F_v, going from crossing point x of face f to crossing point x' of face f', where f and f' are possibly the same, include an edge from u_x to $u_{x'}$, coinciding with that portion of C_v. If $\delta_v = 1$, then for any portion of C_v going from crossing point x to v, include an edge from u_x to v.

We illustrate the construction of the mimicking graph using Figures 4a and 4b. Figure 4a shows the graph of Figure 3 together with the simple curves separating the sets of vertices claimed at vertex 2 by the incident edges. The mimicking graph is shown in Figure 4b.

Lemma 5.2. Let v be a vertex of degree d. The number of crossings that the curves in C_v make with a set F' of p disjoint nontrivial faces in G_e is at most $r(d) = 2p + d - 2$.

Proof. We count vertices, edges, and faces in $\hat{G}_e$. Let p_v be the number of faces in F_v. The number of vertices is $r_v + \delta_v$. This follows since for each face f with r_f crossings, there are r_f vertices of type u_x. The number of edges is $3r_v/2+d/2+\delta_v-1$, as is seen by the following. For each face f, there is a cycle of r_f edges. In addition, there are $(r_v + d)/2 + \delta_v - 1$ edges representing portions of C_v outside of the faces in F_v. We get the latter by counting endpoints of curves in C_v and then dividing by two. There are $r_v + \delta_v - 1$ endpoints of curves in C_v, excluding v, and $d + \delta_v - 1$ curves of C_v not in F_v which have an endpoint at v. The number of faces in $\hat{G}_e$ is $p_v + d$. This follows since there is one face for each face in F_v, and d other faces incident on v.

Recall Euler's formula for planar graphs [H, p. 103]:

$$|V| - |E| + |F| = 2$$

Substituting the values from $\hat{G}_e$ yields

$$(r_v + \delta_v) - (3r_v/2 + d/2 + \delta_v - 1) + (p_v + d) = 2$$

Solving for r_v gives $r_v = 2p_v + d - 2$. Since $p_v \leq p$, the bound can be chosen as $r(d) = 2p + d - 2$. ∎

Theorem 5.2. Let G_e be a planar embedding with a disjoint face covering of p nontrivial and q trivial faces. The number of intervals labeling all edge ends at a vertex v of degree d is at most $3p + q + d - 2$.

Proof. From Lemma 5.2, the number of crossings of nontrivial faces is at most $r = 2p + d - 2$. Each consecutive pair of crossings on a face induces an interval, except for at most one pair per face, which induces two intervals. Thus there are at most $3p + d - 2$ intervals induced by nontrivial faces. There are at most q intervals induced by the trivial faces. ∎

Suppose that only a subset of vertices in the graph are allowed to be the destination of messages. The following corollary is used in an approximate routing scheme presented in [FJ2].

Corollary 5.1. Let G_e be a planar embedding of a graph G with destination set D. If all but q vertices in D are on p faces, then the number of intervals labeling all edge ends at a vertex of degree d is at most $3p + q + d - 2$. ∎

6. Edge labelings from adjacent faces

It is possible to get a better bound on the number of intervals labeling an edge when faces in the face covering share vertices. Let F' be a face covering consisting of nontrivial faces. Define a relation $\equiv$ on nontrivial faces in F' such that $f_1 \equiv f_2$ if and only if either f_1 and f_2 share a vertex, or there is a face f_3 such that $f_1 \equiv f_3$ and $f_3 \equiv f_2$. A *component* F'' of a face covering F' is a maximal subset of nontrivial faces of F' such that for any f_1 and f_2 in F'', $f_1 \equiv f_2$.

Let G_e be a planar embedding with a face covering F' of p nontrivial faces, all in the same face component. Construct an Eulerian multigraph $G' = (V, E')$ as follows. Insert each edge into E' with multiplicity equal to the number of faces on whose boundary the edge appears. Consider a plane embedding G'_e of G' consistent with G_e. A *restricted walk* W of G'_e is an Eulerian walk of G'_e such that if W enters a vertex v on one edge, it leaves v on the next edge around v in either a clockwise or counterclockwise direction. Such a walk always exists, by extending arguments presented in [F1]. Name the vertices in G in order as they first appear in the restricted walk of G'_e. An example of a planar graph with a face covering of two adjacent faces is shown in Figure 5a. The corresponding Eulerian multigraph is shown in Figure 5b, along with a restricted walk, which generates the vertex names.

Recall that $C_{v,w}$ is a simple curve in the plane separating $R_{v,w}$ from $G - R_{v,w}$, and crossing no edge twice.

Lemma 6.1. For any vertex v and neighbor w, closed curve $C_{v,w}$ intersects the restricted walk W of G'_e no more than $2p$ times.

Proof. $C_{v,w}$ will cross each face of G_e at most once. It will intersect an edge in G'_e at each end of the crossing. Thus at most p faces in G_e are crossed, each of which results in two intersections of $C_{v,w}$ with W. ∎

Lemma 6.2. The number of intervals in $R_{v,w}$ is at most p.

Proof. Traverse the planar Eulerian walk W of G'_e in order starting with a vertex inside $C_{v,w}$. When W crosses outside of $C_{v,w}$ hop ahead to the next vertex on W inside $C_{v,w}$. Since a vertex is inside $C_{v,w}$ if and only if its first occurrence is visited, ignore all visits except the first. Thus all vertices are visited in order, with gaps in the consecutive numbering occurring when W goes outside $C_{v,w}$ and then comes back in. Since there are $2p$ intersections of $C_{v,w}$ with W, there are at most p such occurrences. Thus the number of intervals in $R_{v,w}$ is at most p. ∎

For a planar graph G with embedding G_e that has several face components, find a restricted walk of each face component of G_e. Assign names to the vertices in order as they first appear in an Eulerian walk around each face component in turn, and then the trivial faces.

Theorem 6.1. Let G_e be a planar embedding with a face covering of p nontrivial faces, forming s face components, and q trivial faces. The number of intervals needed to label any edge is at most $\lfloor (2p+s+q)/2 \rfloor$.

Proof. Let nontrivial face component c consist of p_c faces and have vertices $i_c, i_c + 1, \ldots, j_c$, for $c = 1, 2, \ldots, s$. From Lemma 6.2, $R_{v,w}$ contains either at most p_c intervals from c, or $p_c + 1$ intervals from c, where one interval contains i_c and another contains j_c. As in the proof of Theorem 5.2, represent $R_{v,w}$ symbolically as $R'_{v,w}$ over the indices 1 to $2p+s+q$. A face component c will be represented by $2p_c+1$ indices, and a trivial face will be represented by 1 index. The number of intervals in $R'_{v,w}$ will exactly equal the number of intervals in $R_{v,w}$. But then $R'_{v,w}$ can have at most $\lfloor (2p+s+q)/2 \rfloor$ nonadjacent intervals. ∎

Theorem 6.2. Let G_e be a planar embedding with a face covering of p nontrivial faces, forming s face components, and q trivial faces. The number of intervals labeling all edge ends at a vertex v of degree d is at most $2p+s+q+d-2$.

Proof. By Lemma 5.2, the number of times that the curves enter faces in F' is at most $r(d)=2p+d-2$. In a restricted walk of the Eulerian multigraph of any face component, if all visits to a vertex but the first are ignored, then there is one interval per entrance, plus one interval for not having wraparound. In addition, there is one interval per trivial face. ∎

7. Edge labelings for nonplanar graphs

Our results can also be extended to yield reasonable edge labelings for graphs that can be embedded on a surface of small genus. We shall show that the number of intervals needed to label an edge increases by 2 for each increase of 1 in the genus. An example of a nonplanar graph that can be embedded on a torus is shown in Figure 6, with the torus represented as a rectangle in which both pairs of opposite sides are identified. (See [H, p. 116-117] for terminology relating to embedding graphs on surfaces.) The embedded graph has a face covering of two faces, whose boundary edges are shown in bold. Let edges $\{1,2\}$, $\{2,3\}$, $\{5,6\}$, $\{7,8\}$, $\{9,10\}$, $\{11,12\}$ and $\{13,14\}$ have cost 2, and all other edges have unit cost. At vertex

15, edge $\{15, 16\}$ will be labeled with 5 intervals, [2,3), [6,8), [10,12), [13,14), and [16,1).

Let g be the smallest integer such that G can be embedded on a surface of genus g and let $G_e = (V, E, F)$ be such an embedding. Let F' be a set of disjoint faces in this embedding. This case is essentially the same as if there are just 2 edges incident on v. Let $F_{v,w}$ be those faces in F', each of which is crossed by closed curve $C_{v,w}$. We first derive an upper bound $r(2, g)$ on the number of times $r_{v,w}$ that $C_{v,w}$ crosses the boundaries of faces in F'.

As before, we generate a mimicking graph $\hat{G}_e$ from G_e, $F_{v,w}$, and $C_{v,w}$. The construction is essentially the same, except for the following. Note that v will not be a crossing point, or otherwise included in $\hat{G}_e$. Remove those handles on the surface that are not used by any edge. Let g' be the genus of the resulting surface.

Lemma 7.1. The number of times that closed curve $C_{v,w}$ crosses a set F' of p disjoint nontrivial faces in an embedding of G on a surface of genus g is at most $r(2, g) = 2p + 4g$.

Proof. We count vertices, edges, and faces in $\hat{G}_e$. Let $p_{v,w}$ be the number of faces in $F_{v,w}$. The number of vertices is $r_{v,w}$. This follows since for each face f with r_f crossings, there are r_f vertices of type u_x. The number of edges is $3r_{v,w}/2$ as is seen by the following. For each face f, there is a cycle of r_f edges. In addition, there are $r_{v,w}/2$ edges representing portions of $C_{v,w}$ outside of the faces in F_v, since no point of $C_{v,w}$ will coincide with v in G_e. The number of faces in $\hat{G}_e$ is $p_{v,w} + 2$. This follows since there is one face for each face in F_v, and two faces split by $C_{v,w}$.

Recall Euler's formula generalized to polyhedra of genus g [H, p. 117]:

$$|V| - |E| + |F| = 2 - 2g$$

Substituting our values from $\hat{G}_e$, we get:

$$r_{v,w} - 3r_{v,w}/2 + (p_{v,w} + 2) = 2 - 2g'$$

Solving for $r_{v,w}$, we get

$$r_{v,w} = 2p_{v,w} + 4g'$$

Since $p_{v,w} \leq p$ and $g' \leq g$, we can choose as our bound

$$r(2, g) = 2p + 4g. \blacksquare$$

Assign names to the vertices in G in order around each face of F' in turn.

Theorem 7.1. Let G_e be an embedding of G on a surface of genus g, and let G_e have a face covering of p nontrivial faces, forming s face components, and q trivial faces. The number of intervals needed to label any edge is at most $\lfloor (2p + s + q + 4g)/2 \rfloor$.

Proof. Similar to that of Theorem 6.1. ∎

Lemma 7.2. The number of crossings that the curves in C_v make with a set F' of p disjoint nontrivial faces in G_e is at most $r(d, g) = 2p + 4g + d - 2$.

Proof. A mimicking graph $\hat{G}$ is generated and Euler's formula generalized to polyhedra of genus g is used, as in Lemma 5.2. ∎

Theorem 7.2. Let G_e be an embedding of G on a surface of genus g, and let G_e have a face covering of p nontrivial faces, forming s face components, and q trivial faces. The number of intervals labeling all edge ends at a vertex v of degree d is at most $2p + s + q + 4g + d - 2$.

Proof. Similar to that of Theorems 5.2 and 6.2. ∎

8. A proper hierarchy of graphs

Analogous to the 1-interval property, for $k > 1$ we say that a graph has the *k-interval property*, if there is a naming of its vertices with integers from 1 to n such that for every assignment of nonnegative edge costs, each edge end can be labeled with at most k intervals encoding shortest paths.

The k-interval properties, for $k > 0$, lead to a natural hierarchy of graphs. Let $\mathcal{G}(k)$ be the class of graphs having the k-interval property, $k > 0$. We show that the classes $\mathcal{G}(k)$ form a proper hierarchy. Thus, by using an increasingly larger number of intervals to label edges, it is possible to handle progressively richer classes of graphs.

The following lemma provides a means for separating the classes.

Lemma 8.1. For $k > 0$, $K_{2,2k+1}$ has the $k+1$ interval property, but not the k-interval property.

Proof. $K_{2,2k+1}$ has a planar embedding with a face covering of 1 regular and $2k-1$ trivial faces. Thus by Theorem 5.1 it has the $(k+1)$-interval property.

We show that $K_{2,2k+1}$ does not have the k-interval property as follows. Without loss of generality, let one of the nodes in the size 2 bipartition be named 1, and the other be named u. Define set S_u as $\{i \mid i \text{ is even and } i < u\} \cup \{i \mid i \text{ is odd and } i > u\}$. Let $w = \min \{S_u\}$. Then there is an assignment of costs to edges such that edge $\{1, w\}$ is on the shortest path from 1 to each of the vertices in $S_u \cup \{u\}$. There are $k+1$ nodes not in $S_u \cup \{u\}$, no two of which are consecutive, even with wraparound. Thus $k+1$ intervals are necessary to label edge $\{1, w\}$. ∎

Theorem 8.1. For $k > 0$,

$$\mathcal{G}(k) \subsetneq \mathcal{G}(k+1).$$

Proof. Clearly $\mathcal{G}(k) \subseteq \mathcal{G}(k+1)$. By the preceding theorem, the inclusion is proper, since for each k, there is a graph that is not in $\mathcal{G}(k)$ but is in $\mathcal{G}(k+1)$. ∎

Recall from Section 4 that $\mathcal{G}(1)$ is precisely the class of outerplanar graphs. For $k > 1$, the problem of characterizing the classes $\mathcal{G}(k)$ remains open.

Acknowledgements. The authors would like to thank the referees for a careful reading of the paper and for numerous suggestions.

References

[AHU] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Reading MA, 1974.

[CH] G. Chartrand and F. Harary, "Planar permutation graphs", *Annales de l'Institut Henri Poincaré*, Section B3, 1967, pp. 433–438.

[F1] G.N. Frederickson, "Approximation algorithms for some postman problems", *Journal of the ACM*, Vol. 26, No. 3, July 1979, pp. 538–554.

[F2] G.N. Frederickson, "Implicit data structures for the dictionary Problem", *Journal of the ACM*, Vol. 30, No. 1, January 1983, pp. 80–94.

[FJ1] G.N. Frederickson and R. Janardan, "Optimal message routing without complete routing tables", *Proceedings of the 5^{th} Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, Calgary, August 1986, pp. 88–97.

[FJ2] G.N. Frederickson and R. Janardan, "Separator-based strategies for efficient message routing", *Proceedings of the 27^{th} Annual IEEE Symposium on Foundations of Computer Science*, Toronto, October 1986, pp. 428–437.

[FJ3] G.N. Frederickson and R. Janardan, "Efficient fault-tolerant routings in outerplanar networks", manuscript, 1987.

[H] F. Harary, *Graph theory*, Addison-Wesley, Reading MA, 1969.

[LT] R.J. Lipton and R.E. Tarjan, "A separator theorem for planar graphs", *SIAM Journal on Applied Mathematics*, Vol. 36, No. 2, April 1979, pp. 177–189.

[MS] J.I. Munro and H. Suwanda, "Implicit data structures for fast search and update", *Journal of Computer and System Sciences*, Vol. 21, No. 2, October 1980, pp. 236–250.

[SK] N. Santoro and R. Khatib, "Labelling and implicit routing in networks", *The Computer Journal*, Vol. 28, No. 1, February 1985, pp. 5–8.

[vLT1] J. van Leeuwen and R.B. Tan, "Computer networks with compact routing tables", in *The Book of L*, G. Rozenberg and A. Salomaa (eds.), Springer-Verlag, 1986, pp. 259–273.

[vLT2] J. van Leeuwen and R.B. Tan, "Interval routing", Technical Report RUU-CS-85-16, Department of Computer Science, University of Utrecht, The Netherlands.

[W] S. Warshall, "A theorem on boolean matrices", *Journal of the ACM*, Vol. 9, No. 1, January 1962, pp. 11-12.

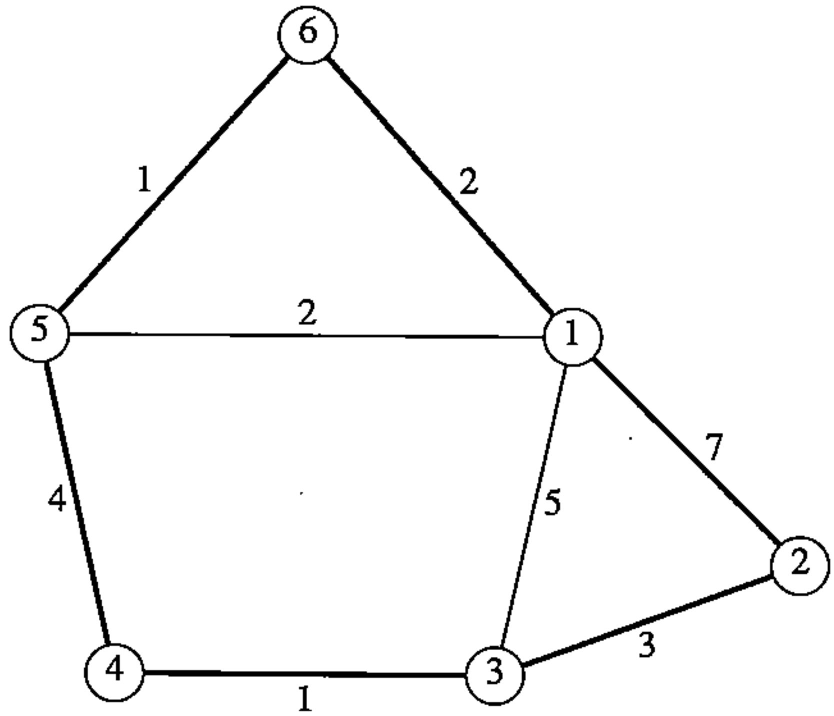

Figure 1a. A clockwise naming of the nodes
of a weighted outerplanar graph.

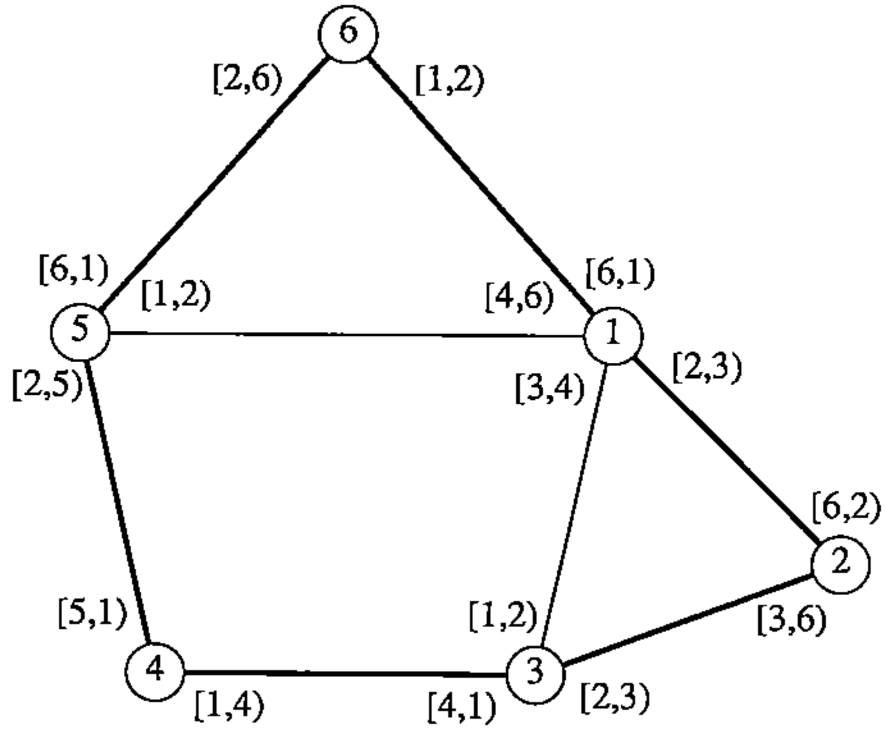

Figure 1b. A labeling of the edges of the outerplanar graph of Figure 1a with intervals encoding shortest paths.

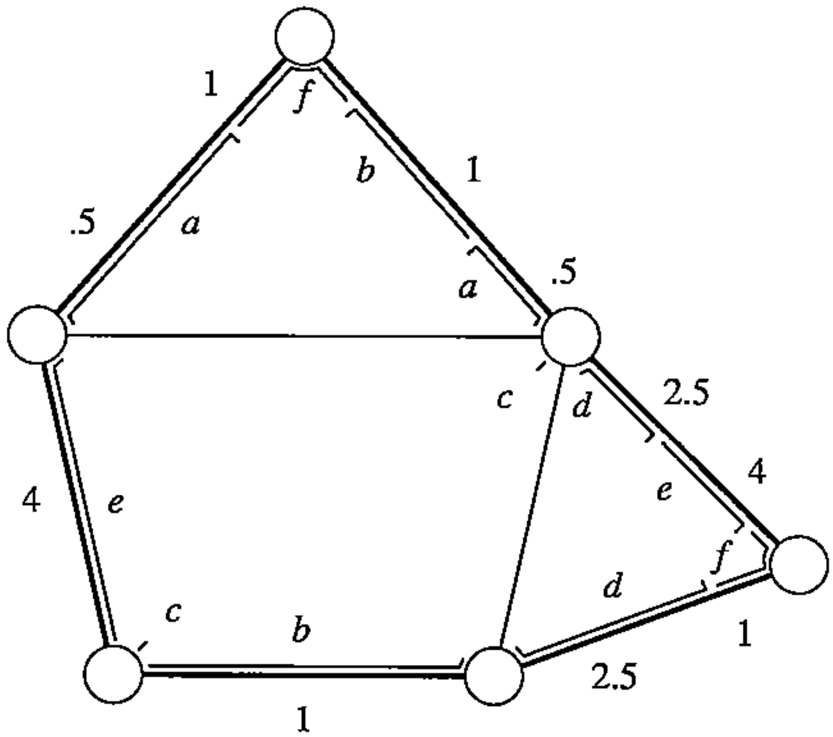

Figure 2. Matching edge sequences generated
by the edge labeling algorithm.

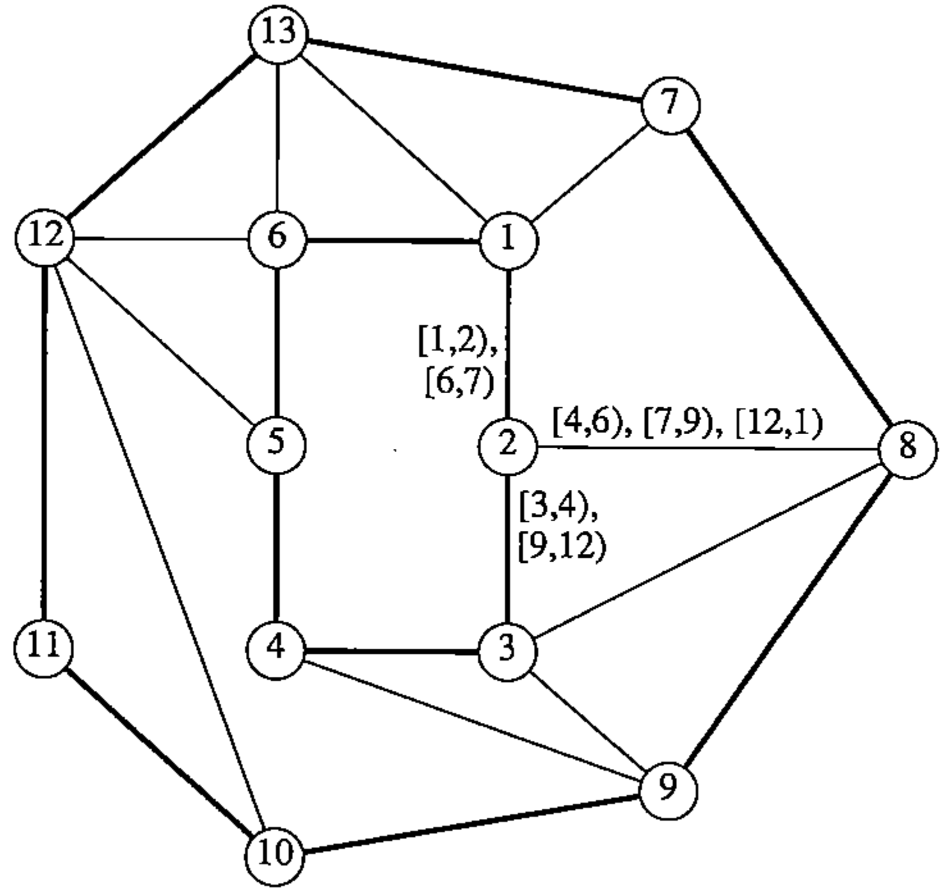

Figure 3. A graph with a face covering of two faces
(whose boundaries are shown in bold),
with the labels assigned to edge ends at node 2.

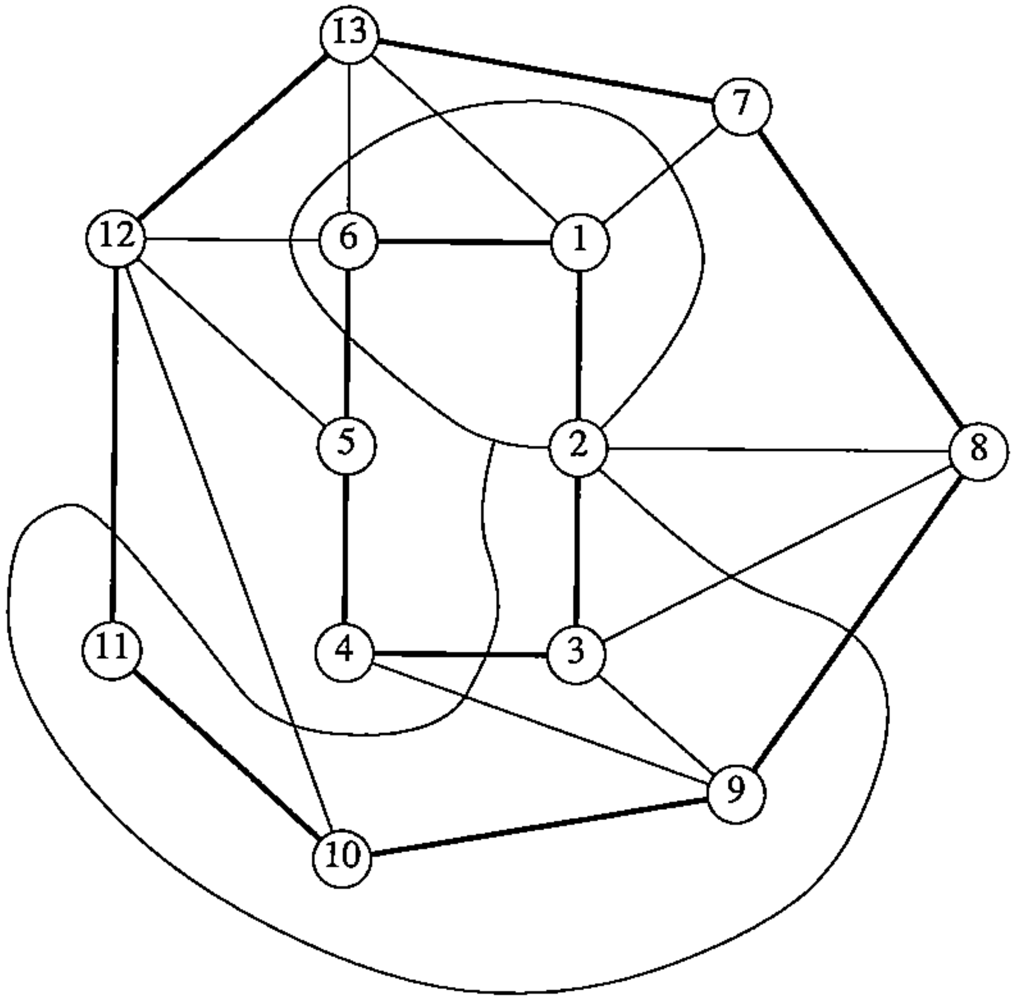

Figure 4a. Collection of simple curves
which separate the sets of vertices
claimed by vertex 2 in Figure 3.

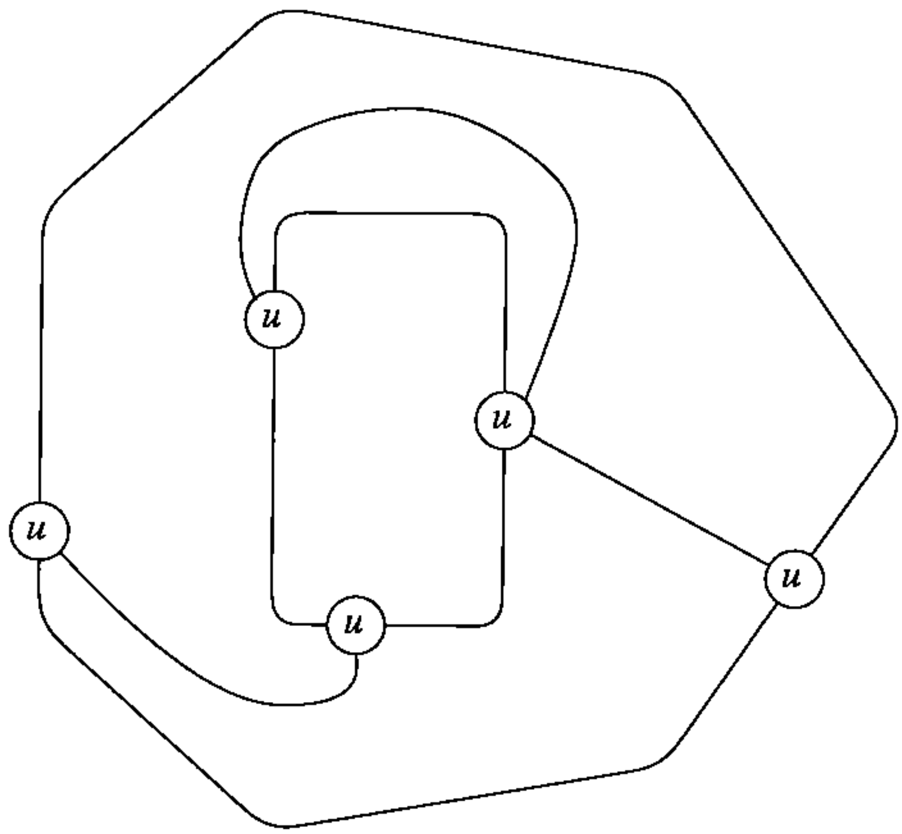

Figure 4b. The mimicking graph for Figure 4a.

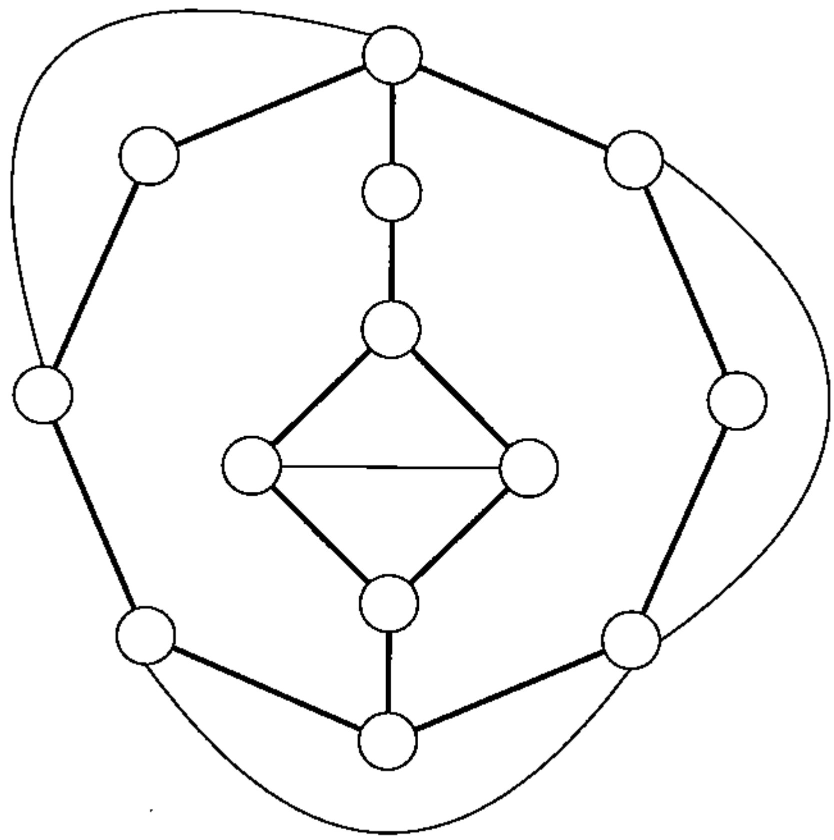

Figure 5a. A graph with a face covering of two faces
(whose boundaries are shown in bold),
which share common vertices.

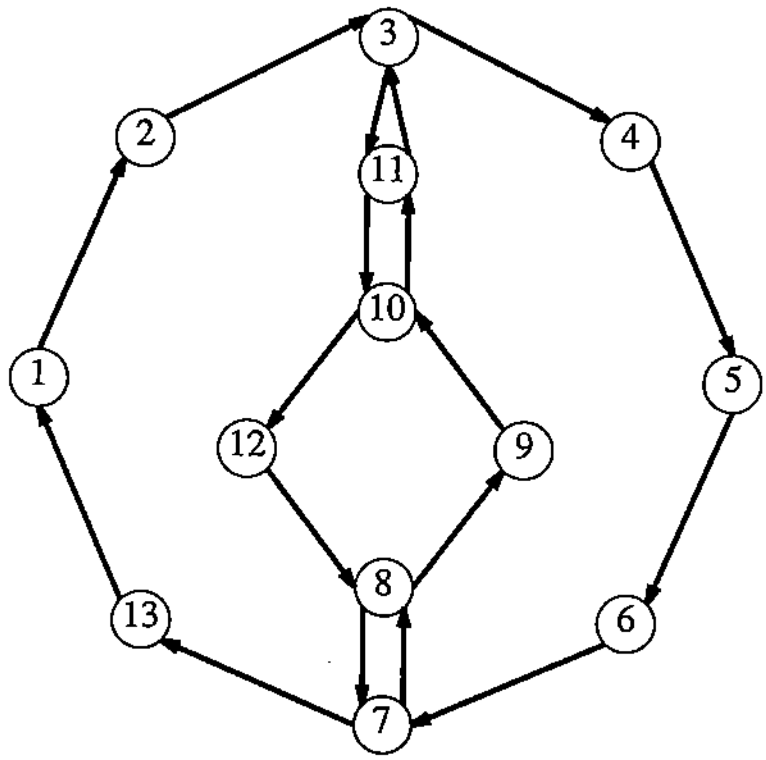

Figure 5b. A restricted walk of an Eulerian multigraph
derived from the face covering of Figure 5a,
with the vertices named accordingly.

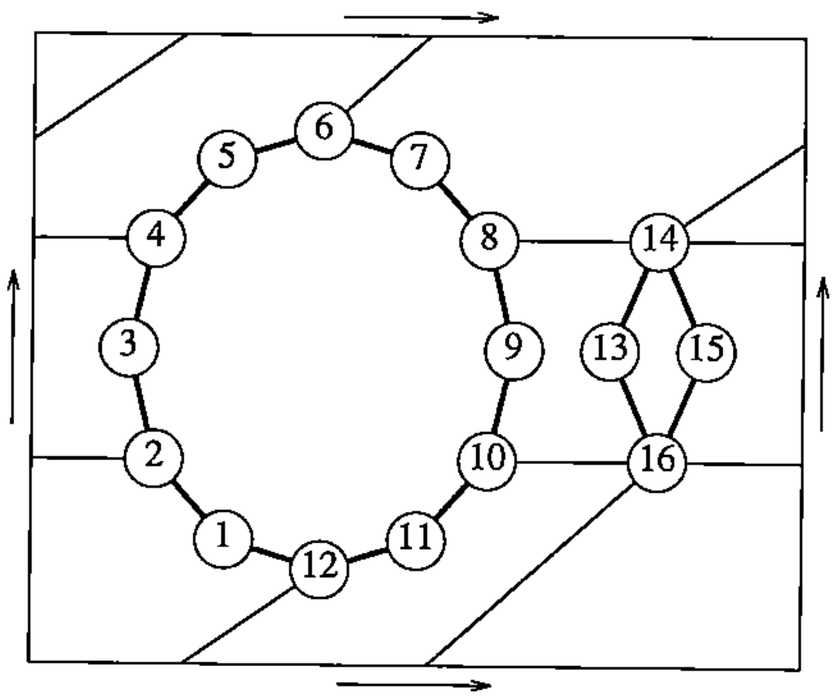

Figure 6. A nonplanar graph embedded on a torus,
with a face covering of two faces
(whose boundaries are shown in bold).