Canonical Cyclic Orderings for Point Sets in the Plane

Alan Saalfeld
Bureau of the Census*

Abstract

For points in general position in the plane, we describe a family of cyclic orderings which are invariant under isometries. We prove that the family can contain at most 60 orderings. The entire family of orderings can be built in $O(n \log n)$ time in $O(n)$ space, where n is the number of points to be ordered. The method used to generate the cyclic orderings of points works for the vertex set of any free tree embedded in the plane. We apply the method to the Euclidean minimum spanning tree for the points in general position to obtain our family of cyclic orderings.

1 Motivation

The technique we will describe in this paper was inspired by and imitates procedures of systematic sampling from lists [KISH]. Coincidentally, one of the important applications of our ordering procedure is the systematic selection of subsets of points in the plane for sampling purposes. Point sets chosen by our procedure will exhibit excellent spatial representativeness properties; and, morever, our sample subsets will be independent of any coordinate system that may affect and bias other sampling strategies.

2 Algorithm Overview

We will describe and illustrate the algorithm by performing it on a simple example. Suppose we are given n points in the plane in general position, such as shown in figure 1.

*The views expressed herein are the author's and not necessarily the views of the Bureau of the Census.

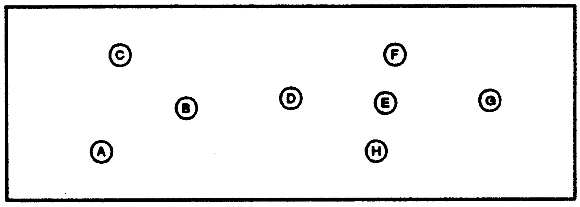

Figure 1: Eight Points in "General Position"

2.1 Build EMST

Build their Euclidean minimum spanning tree, as shown in figure 2, in time $O(n \log n)$, [AHO] and simultaneously sort the edges at each vertex in clockwise order. (General position of the points is used to guarantee uniqueness of the EMST.)

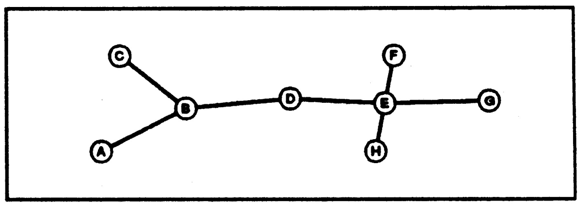

Figure 2: The Euclidean Minimum Spanning Tree for those Eight Points

2.2 Walk Eulerian Tour

Start anywhere on some edge and perform a two-sided Eulerian tour of the tree. A two-sided Eulerian tour walks every edge twice and visits each vertex p deg(p) times, as shown in figure 3. Having the edges sorted in clockwise order permits the Eulerian tour to be made in linear time.[1]

[1]Note: Garey and Johnson [GAREY] and others [PREPARATA], [EDELSBRUNNER] describe an ordering based on a two-sided Eulerian tour which they use to ap-

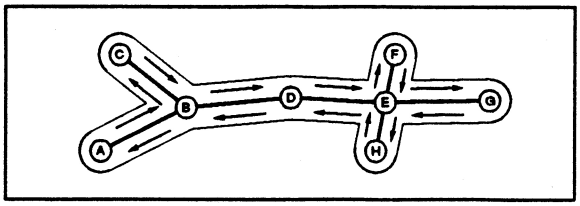

Figure 3: A Two-sided Eulerian Tour of the EMST

2.3 Build Auxiliary Interval

While making the Eulerian tour, build an interval of total length n units as follows: each time vertex p is visited on the tour, attach to the current right-most end point r of the partially built interval $[0, r)$, a half-open interval $[r, r + 1/\deg(p))$ of length $1/\deg(p)$ labeled "p" as shown in figure 4. Since the number of visits any vertex receives during the Eulerian tour is equal to its degree, the total length of subintervals that correspond to any individual vertex p will be $\deg(p) \cdot 1/\deg(p) = 1$ unit; and the total length of all the subintervals will be n, the total number of vertices.

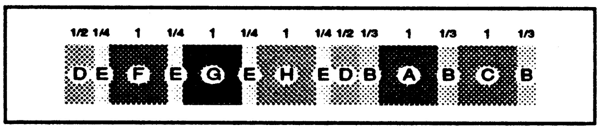

Figure 4: The Successive Weighted Vertex Visits of the Eulerian Tour

Attach the two ends of the composite interval to make it cyclic. Notice that the cyclic interval we have built up to this point is completely independent of our initial starting point and of the position of the point set in space—any translation, rotation, scaling, or other transformation that preserves the EMST will give us the same circular ordering of half-open intervals, as shown in figure 5.

2.4 Select from Interval

Next choose an arbitrary starting point in this cyclic interval of length n. Then select n points by skipping along the cyclic interval one unit at a time, adding the point p to our ordered list when the skipping point lands in a half-open subinterval corresponding to p, as shown in figure 6.

proximate a Euclidean Travelling Salesman Tour to within a factor of 2, but the tour they describe is not canonical. It depends on the starting point for the Eulerian Tour; and changing the starting point may produce up to $O(n)$ different cyclic orderings.

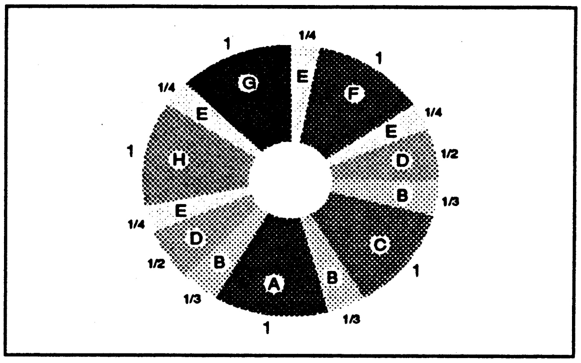

Figure 5: The Successive Weighted Vertex Visits Ordered Cyclically

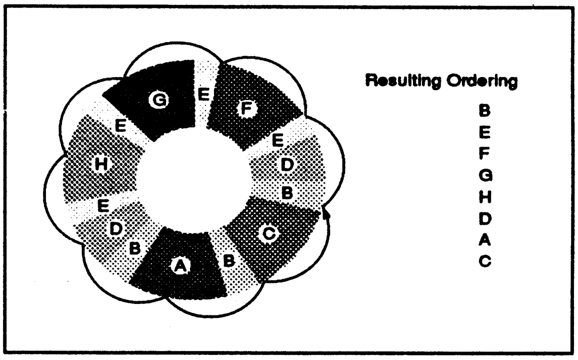

Figure 6: Vertex Ordering Based upon Skipping along Cyclic Interval

The important features of our construction of the cumulative interval are the following:

1. **Every vertex is selected exactly once by this procedure.**
2. **There are at most (up to cyclic permutations) 60 orderings that can arise from this procedure.**

3 Proof Sketches

3.1 Ordering the Vertices

The proof that the selection procedure actually produces an ordering of the vertices follows immediately from the following lemma and its first corollary.

Lemma 3.1 (Integral-Branch-Weights) *The fractional vertex weights accumulated between any two consecutive visits of the Eulerian tour to a multivisited vertex always add up to an integer.*

Proof: The proof of this lemma rests entirely on the observation that between two consecutive visits to any vertex, an entire branch emanating from that vertex is completely consumed by the subwalk of the Eulerian tour (*i.e.* every edge of the branch is traveled twice.) The branch consumed is that branch associated with the edge[2] that was both "exiting" edge for the first visit and then "entering" edge for the second visit of the two consecutive visits in question. Note that the "exiting" edge of the first visit and the "entering" edge of the second visit are always equal on consecutive visits of the Eulerian tour, as shown in figure 7.

In consuming an entire branch, one must visit every vertex in that branch as many times as possible, *i.e.* as many times as the degree of that vertex. Thus each vertex in the branch gets fully counted. In other words, the sum of weights for all the visits for any individual vertex during the walk of the branch is 1. And the sum of weights for *all* the visits of *all* vertices during the walk of the branch is an *integer*, equal to the number of distinct vertices in the branch. □

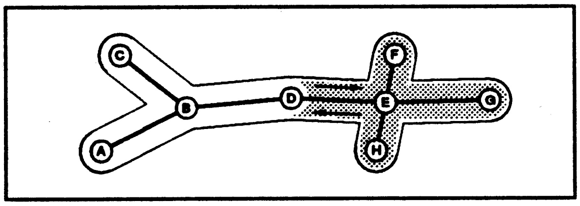

Figure 7: A Consumed Branch Contains All Visits of All Its Vertices

This lemma has two useful corollaries. To prove the first corollary we will want to talk about the fractional part of a number or an interval of numbers. Our meaning is the usual one: the fractional part of 5.35 is 0.35. The fractional part of an interval such as [17.32, 17.84) is just the set of all possible fractional values: [0.32, 0.84).

[2] The branch associated with an edge out of a vertex consists of all of the vertices and edges that can be reached by a *path* from the vertex in question that starts along the edge in question. The branch as we have defined it does not include the starting vertex.

Corollary 3.2 *Every vertex gets hit exactly once by skipping one unit at a time through the cyclic n-interval.*

Proof: Consider any vertex v of degree $= k$. Each visit to the vertex will result in an interval of length $1/k$ being added to the cumulative interval. We want to prove that, no matter where we fix a start for our cyclic interval, the fractional parts of the intervals corresponding to v in the total interval of length n have no overlap. From lemma 3.1, it is clear that each successive interval corresponding to v has its fractional part begin where the fractional part of the last interval corrsponding to v left off, since an interval of integer length (*i.e.* having *no* fractional part) corresponding to all of the vertices of the branch consumed, has intervened. In fact, the fractional parts of values assumed in the intervals corresponding to any individual vertex must span all of the values between 0 and 1. Thus any real number r or integral augmentation $r + m$ of r can hit at most one of the k intervals of length $1/k$; and there is exactly one integer m_0 such that $r + m_0$ will hit one of the k intervals. □

The next corollary follows immediately from the proof of the lemma.

Corollary 3.3 *The collection of vertices of* **any branch** *of the EMST always constitute a complete interval (i.e. appear consecutively) for any cyclic ordering produced by our ordering procedure.*

In the above corollary, recall that a *branch* excludes its starting vertex.

3.2 The Orderings are Few

The argument that there are at most sixty such cyclic orders follows from the fact that the degree of the vertices in a Euclidean minimum spanning tree is at most 6 (or 5 if one insists on points being in general position). In either case, the least common multiple of possible vertex degrees is 60; and, therefore, on our cyclic cumulative interval, we can imagine subdividing all of the intervals of length 1/5 or 1/4 or 1/3 or 1/2 or 1 into subintervals of length 1/60 (or of length 1/12 if there are no vertices of degree 5), still labeling the subintervals as before with the appropriate corresponding vertex indentifier (see figure 8). Now it is clear that there can be at most 60 effectively different ways

(12 ways, if no vertices have degree 5) of placing our skip pattern. □

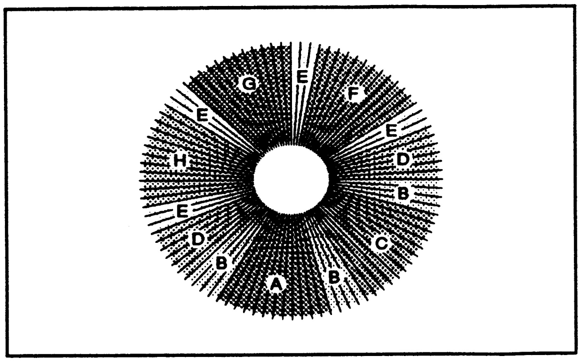

Figure 8: Equal-sized Selection Bands of 1/12

4 Useful Generalizations

4.1 Visits with Unequal Weights

We chose an equal weight of $1/\deg(p)$ for each visit to a vertex so that the total weight of all visits would add up to 1. However, as long as the total weight of all visits adds to 1, the argument given in the proof of Lemma 3.1 still holds. (This observation is due to Walid Aref.) So we have the following stronger theorem:

Theorem 4.1 *While making an Eulerian tour of a tree, build a separate cyclic interval of total length n units by assigning a non-negative weight to each vertex visit in any way so that the total weight for all visits to any individual vertex is one. Then every vertex gets hit exactly once by skipping one unit at a time through the cyclic n-interval.*

4.2 Edge Orderings

Moreover, an identical argument can be made for edges instead of vertices with each edge getting weight exactly 1/2 (since every edge is visited twice in the Eulerian Tour). But giving every edge weight 1/2 amounts to nothing more than skipping every other edge in our selection procedure. So we have the following lemma:

Lemma 4.2 *While making an Eulerian tour of a tree, mark every other edge visited. Then every edge gets marked exactly once.*

If we consider ordering the edges of a tree using this strategy, then consecutive edges in our order are never more than link distance two apart! This strategy may be extended to graphs as follows: Take any connected graph and perform some node-splitting operation to build a tree whose edges correspond to the edges of the original graph. Then the graph edges may be assigned a cyclic order based on selecting alternate hits from an Eulerian tour of the corresponding edges of the derived tree.

Since we can certainly split nodes in $O(n \log n)$ time using sorting and a plane sweep operation, we can accomplish the following ordering for the edges of any connected graph efficiently:

Corollary 4.3 *One may find a cyclic ordering for the edges of any connected graph in $O(n \log n)$ time so that any two edges which are consecutive in the cyclic ordering never have link distance greater than two in the graph.*

5 References

1. Aho, Alfred, John Hopcroft, and Jeffrey Ullman, 1985, **Data Structures and Algorithms**, Addison-Wesley, Reading, MA.

2. Edelsbrunner, Herbert, 1987, **Algorithms in Combinatorial Geometry**, Springer-Verlag, New York.

3. Garey, Michael R., and David S. Johnson, 1979, **Computers and Intractability, A Guide to the Theory of NP-Completeness**, W. H. Freeman, New York.

4. Kish, Leslie, 1965, **Survey Sampling**, John Wiley, New York.

5. Preparata, Franco, and Michael Shamos, 1985, **Computational Geometry, An Introduction**, Springer-Verlag, New York.