Hamiltonian Triangulations and Circumscribing Polygons of Disjoint Line Segments *

Andranik Mirzaian

Department of Computer Science
York University
North York, Ontario
Canada M3J 1P3

ABSTRACT

Let $\Sigma = \{\ S_1\ ,\ \cdots\ ,\ S_n\ \}$ be a finite set of disjoint line segments in the plane. We conjecture that its visibility graph, *Vis*(Σ), is hamiltonian. In fact, we make the stronger conjecture that *Vis*(Σ) has a hamiltonian cycle whose embedded version is a simple polygon (i.e., its boundary edges are non-crossing visibility segments). We call such a simple polygon a *spanning polygon* of Σ. Existence of a spanning polygon of Σ is equivalent to the existence of a hamiltonian triangulation of Σ. A spanning polygon P is said to be a *circumscribing polygon* of Σ, if it has the additional property that no segment in Σ lies in the exterior of P. We prove circumscribing polygons exist for the special case when Σ is *extremally situated*, i.e., when each segment S_i touches the convex hull boundary of Σ. Furthermore, for this special case we give an algorithm that constructs a circumscribing polygon in $O(n \log n)$ time and this is optimal.

1. Introduction

Throughout this paper the domain of discussion is with respect to the Euclidean plane. It is well known that any finite set of points admits a simple polygon with the given points as its vertices

Gra72 We may call such a polygon a *spanning polygon* of the point set. What is a

---

Mir90b

suitable generalization of this fact from points to (disjoint) line segments? Unless otherwise stated, throughout this paper let $\Sigma = \{ S_1 , \cdots , S_n \}$ be a set of n pairwise disjoint line segments. Call each endpoint of a segment $S_i \in \Sigma$ a vertex of S_i, and also a vertex of Σ. The visibility graph, $Vis(\Sigma)$, of Σ (see e.g.,

Ede87 is the (embedded) graph whose vertices are the set of vertices of Σ, and whose edges are those line segments between pairs of vertices that do not cross any segment in Σ. By definition, the segments in Σ are considered to be edges of the visibility graph. In what follows, we will be considering the existence of various simple polygons that are hamiltonian cycles of $Vis(\Sigma)$.

    Rappaport

Rap89 defined a *simple circuit* of Σ to be a simple polygon Q whose vertices are the vertices of Σ, and every segment in Σ is an edge of Q. He showed that not every such Σ has a simple circuit, and to decide whether it does is NP-complete. (Rappaport actually proved this NP-completeness result assuming some segments in Σ may have common vertices and left as an open problem the complexity of the case where segments in Σ are pairwise disjoint.) The set Σ is said to be *extremally situated* if each segment in Σ has at least one of its endpoints on the boundary of the convex hull of Σ. Rappaport et al.

RIT90 showed that even an extremally situated Σ may or may not admit a simple circuit. Furthermore, for this special case they gave an $O(n \log n)$ time algorithm that constructs a simple circuit, if one exists.

Definition 1. We define a *spanning polygon* of Σ to be a simple polygon P such that vertices of P are exactly the vertices of Σ, and no edge of P crosses any segment in Σ. In other words, each segment in Σ is either an edge, an internal diagonal, or an external diagonal of P.

Definition 2. We define a *circumscribing polygon* of Σ to be a spanning polygon P of Σ with the added property that no segment in Σ lies in the exterior of P. In other words, each segment in Σ is either an edge or an internal diagonal of P. See Figure 1.

[Figure 1 about here.]

Definition 3. A *hamiltonian triangulation* of Σ is a triangulation of Σ such that, when viewed as a graph, it contains a hamiltonian cycle. (Recall that a triangulation of Σ is a subdivision of the convex hull of Σ whose edge-set contains the segments in Σ, plus a maximal set of additional noncrossing visibility segments. See, e.g.,

Ede87

We can now mention the following simple facts.

Fact 1. *Any simple circuit is a circumscribing polygon, and any circumscribing polygon is a spanning polygon.*

Proof. Obvious. □

Fact 2. Σ *has a hamiltonian triangulation if and only if it admits a spanning polygon.*

Proof. Any hamiltonian cycle in a hamiltonian triangulation of Σ is a spanning polygon of Σ. Conversely, any spanning polygon of Σ can be extended to a hamiltonian triangulation, by adding additional noncrossing visibility segments to it, plus the segments in Σ. □

Note that if some segments in Σ are allowed to have common vertices, then Σ may not have a circumscribing polygon. (See Figure 2.)

[Figure 2 about here.]

We make the following conjectures.

Conjecture 0*: Any finite set of pairwase disjoint line segments has a circumscribing polygon.*

Conjecture 1*: Any finite set of pairwise disjoint line segments has a hamiltonian triangulation.*

Conjecture 2*: The visibility graph of any finite set of pairwise disjoint line segments is hamiltonian.*

Note that by Facts 1 and 2, Conjecture 0 implies Conjecture 1, and Conjecture 1 implies Conjecture 2. In contrast to Conjecture 1, we note that not every triangulation of Σ is hamiltonian. Shamos in his Ph.D. thesis

Sha78 asked whether Delaunay Triangulations are hamiltonian. The negative answer was given in

Kan83 Dil87a Also,

Dil87b raised two related open problems: (a) is it true that "most" Delaunay triangulations are hamiltonian? and (b) how difficult is it to determine whether a given Delaunay triangulation is hamiltonian? (Can it be solved in polynomial time?) A related result is a classical theorem of Whitney

Whi31 which asserts that any maximal 4-connected planar graph is hamiltonian. The proof of this theorem is simplified in

AKS84 and a linear time algorithm is given to find a hamiltonian cycle in such graphs. A

proof of this theorem for a larger class of planar graphs and a linear time algorithm for finding a hamiltonian cycle in such graphs appears in

Dil90 Tutte

Tut56 generalized the result of Whitney by showing that any 4-connected planar graph is hamiltonian. Chiba and Nishizeki

ChN89 gave a linear time algorithm to find a hamiltonian cycle in a 4-connected planar graph.

2. Main Results

The main results of this paper are Theorems 1 and 2 below that show the existence of circumscribing polygons (hence, the existence of spanning polygons and hamiltonian triangulations) for the special case when Σ is extremally situated. The proofs of the theorems appear in subsequent sections. In the rest of the paper assume $\Sigma = \{S_1 , \cdots , S_n\}$ is a set of n extremally situated pairwise disjoint line segments.

Theorem 1. *Any extremally situated Σ admits a circumscribing polygon.*

Theorem 2. *There is an algorithm that constructs a circumscribing polygon of extremally situated Σ in linear space and $O(n \log n)$ time, and this is optimal.*

In a first attempt we may try to construct simple polygons, in $O(n)$ time, that encapsulate the given segments by going around the convex hull in an Euler tour fashion (see Fig. 3), then use a triangulation of these polygons to proceed further

Cha91 The apparent difficulty in this approach is in maintaining convexity as we descend down to subproblems.

[Figure 3 about here.]

Our proofs are based on recursion and employ a novel structure called the *tournament pseudoforest* of Σ. (The term *pseudoforest* has been used in a different context in

GaT88 The convex planar subdivision induced by the tournament pseudoforest of Σ is a linear space data structure and can be constructed in optimal $O(n \log n)$ time using the sweep method. (For an explanation of the sweep method see for example

Ede87 PrS85 Given this data structure, a circumscribing polygon of Σ can be constructed in $O(n)$ additional time.

The organization of the rest of the paper is as follows. Section 3 contains a proof of Theorem 1. Section 4 develops the $O(n \log n)$ algorithm to construct the tournament pseudoforest subdivision. Section 5 gives a proof of Theorem 2 by using the tournament pseudoforest subdivision for a fast implementation of the proof of Theorem 1. Section 6 mentions some applications. Section 7 contains some open problems.

3. Proof of Theorem 1

As mentioned before we assume Σ is an extremally situated set of n pairwise disjoint line segments. Let $\partial\Sigma$ denote the boundary of the convex hull of Σ. A segment S_i in Σ is called an *edge segment*, a *diagonal segment*, or an *internal segment*, if it is, respectively, an edge of $\partial\Sigma$, a diagonal of $\partial\Sigma$, or has only one endpoint on $\partial\Sigma$. In the latter case the endpoint of S_i on $\partial\Sigma$ is called its *head* (denoted h_i) and the other endpoint is called its *foot* (denoted f_i). Each endpoint of an edge segment or a diagonal segment is considered as both its head and foot! To simplify the discussion, we assume (a) no segment S_i is horizontal, (b) no three endpoints of segments in Σ are collinear, and (c) no three segments are concurrent if extended.

The proof of Theorem 1 is by induction on the *size* of Σ, which is defined to be the cardinality of Σ plus the number of its internal segments. We need to prove a slightly stronger version in which we allow a pair of edge segments in Σ to have a common head.

In what follows we use the following two observations. (i) Any subset of an extremally situated Σ is also extremally situated. (ii) In any circumscribing polygon P of Σ, any edge segment of Σ must be an edge of P, and any diagonal segment of Σ must be a diagonal of P. The same holds for the inductively defined subproblems.

If all segments of Σ are edge segments, then $\partial\Sigma$ is a circumscribing polygon of Σ. If there is a diagonal segment S_i, then S_i divides the problem into two smaller subproblems one on each of its sides, including S_i. Both subproblems have smaller *size* than the main one, and S_i is an edge segment in both. By induction there is a circumscribing polygon for each of the two subproblems. Paste together these two polygons along S_i to obtain a circumscribing polygon of Σ.

Now suppose Σ has no diagonal segments, but at least one internal segment. Consider a sequence $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_i)$ of internal segments in Σ constructed as follows. The initial segment σ_1 is any internal segment of Σ. Suppose σ_k, for $1 \le k \le j$, have been defined already. Let $\bar{\sigma}_j$ denote the *extension* of σ_j obtained by the following process: extend σ_j along the direction of its supporting line from the side of its foot until it hits either (a) an edge of $\partial\Sigma$ (which may or may not be an edge of Σ), (b) one of $\bar{\sigma}_k$ ($k < j$), or (c) a new segment in $\Sigma - \partial\Sigma$. In case (c) the new segment becomes σ_{j+1}. Also note that in case (b), the extension σ_j stops and does not continue past an extension of a previously considered segment in the sequence. Furthermore, let $t_j = closure(\bar{\sigma}_j - \sigma_j)$. See Figure 4. We construct the sequence σ until $\bar{\sigma}_i$ hits either (i) $\partial\Sigma$, or (ii) some $\bar{\sigma}_k$, $k < i$. In case

[Figure 4 about here.]

(ii), the extended segments $\bar{\sigma}_k$, $\bar{\sigma}_{k+1}$, $\cdots$, $\bar{\sigma}_i$ induce a cycle. With a suitable change of subscript, and without loss of generality, we may assume the subsequence of σ that induces this cycle starts at σ_k with $k = 1$.

Case (i.1). $\bar{\sigma}_i$ hits an edge of $\partial\Sigma - \Sigma$: In this case divide Σ in two parts (two subproblems) one on each side of $\bar{\sigma}_i$ with σ_i belonging to both sides (as in Fig. 5). Note that σ_i was an internal segment, but it becomes an edge segment in each of the two subproblems. So the *size* of each subproblem is strictly smaller than the original problem. Now proceed inductively on each of the two subproblems, then paste together the two resulting polygons along σ_i.

[Figure 5 about here.]

Case (i.2). $\bar{\sigma}_i$ hits an edge segment S_j at a point p: Split S_j in two segments S'_j and S''_j at point p. Divide the problem in two, as in Fig. 6. One contains t_i, S'_j, and every other segment of Σ on the same side of $\bar{\sigma}_i$ as S'_j, excluding σ_i. The second contains $\bar{\sigma}_i$, S''_j, and every remaining segment of Σ (and of course excluding σ_i). Note that p is a "new" endpoint in both subproblems, the head of σ_i is not an end point in the first subproblem, and the foot of σ_i is not an endpoint in the second subproblem. The two subproblems are strictly smaller in *size*, since S_j is replaced by another edge segment (namely, S'_j or S''_j), and the internal segment σ_i is replaced by an edge segment (namely, t_i or $\bar{\sigma}_i$). By induction there is a circumscribing polygon for each of the two subproblems. Paste together the two resulting polygons along t_i. Note that segments σ_i and S_j become edges of the resulting polygon and the "extra vertex" p disappears.

[Figure 6 about here.]

Case (ii). We use a similar argument as in case (i.2). See Figure 7.

[Figure 7 about here.]

Suppose the internal segments that induce the cycle, in appropriate order around the cycle, are σ_1, σ_2, $\cdots$, σ_i. (Here the ordering is such that σ_j loses to σ_{j+1}, for $j = 1, 2, \cdots, i$, where index arithmetic is taken modulo i, and this is done in the rest of the proof of the theorem as well.) As suggested in Fig. 7, we will obtain i subproblems. In addition to these subproblems, there is a convex "hole" in the middle which will become part of the eventual circumscribing polygon. The subproblems are similar to the first

subproblem of case (i.2) and are defined as follows. Let S'_{j+1} denote the portion of $\bar{\sigma}_{j+1}$ between the head of σ_{j+1} and the point of intersection between $\bar{\sigma}_j$ and $\bar{\sigma}_{j+1}$. The jth sub-problem consists of segments t_j, S'_{j+1}, and every segment of Σ (other than σ_j and σ_{j+1}) that fall in the convex portion of the convex hull of Σ cut off by $\bar{\sigma}_j$ and S'_{j+1}. Each sub-problem is extremally situated and strictly smaller in *size* than Σ. We proceed inductively for each subproblem, then appropriately paste the resulting polygons of the subproblems and the convex "hole" together as suggested in Fig. 7. $\square$

4. The Tournament Pseudoforest

The idea is motivated from the proof of Theorem 1. Imagine the segments in Σ play a tournament as follows: Extend each segment in Σ along its line of support from the side of its foot until it hits either $\partial\Sigma$, or the extension of another segment. If two segment extensions intersect, they play a match. The extension of the loser ends at the intersection point, while the winner continues to be extended. If the intersection point is in the relative interior of one of the segments, then that segment is declared the winner of that match, otherwise the winner is chosen arbitrarily (so the structure is not unique). If a segment extension intersects $\partial\Sigma$, it loses to $\partial\Sigma$ and its extension terminates at the intersection point. Now let $\bar{\sigma}_i$ denote the extension of σ_i, and $t_i = closure(\bar{\sigma}_i - \sigma_i)$ be called the *trail* of σ_i.

As defined in

GaT88 a *pseudotree* is a tree plus possibly an extra edge, between two of its distinct vertices (that creates a unique simple cycle). A *pseudoforest* is a vertex-disjoint collection of pseudotrees.

The tournament pseudoforest of Σ, denoted $TP(\Sigma)$, is the plane graph that is the union of the segments and their trails. $TP(\Sigma)$ forms a convex partitioning of the convex hull of Σ. We refer to this convex planar subdivision as the tournament pseudoforest sub-division, denoted $TP - subdivision$. (See Figure 8.) The following lemmas and the corollary justify these claims.

[Figure 8 about here.]

Lemma 1. *$TP - subdivision$ is a convex planar subdivision of convex hull of Σ.*

Proof: This follows from the fact that if some face of the partition has a reflex vertex, then both of the two segment extensions that intersect at that vertex are the loser of that match, a contradiction. $\square$

We define a *fan* to be a cyclic sequence $\bar{\sigma}_1, \bar{\sigma}_2, \cdots, \bar{\sigma}_i$ of segment extensions so that σ_j loses to $\bar{\sigma}_{j+1}$, for $j = 1, 2, \cdots, i$, where index arithmetic is done modulo i. This is

related to the notion of convex "hole" mentioned in the proof of Theorem 1.

Lemma 2. *A segment extension cannot appear in more than one fan.*

Proof: Suppose $\bar{\sigma}_j$ appears in some fan. Then $\bar{\sigma}_j$ completely determines the segment extensions that form the fan, since each segment extension in the fan (starting with $\bar{\sigma}_j$) loses to a unique segment extension, which is the next one in the sequence. $\square$

Lemma 3. *Let T be a connected component of the tournament pseudoforest. Then T contains one and only one of the following:*

(i) a fan,
(ii) an edge segment of Σ,
(iii) a diagonal segment of Σ, or
(iv) a segment of Σ whose trail hits an edge of $\partial\Sigma - \Sigma$.

Proof: Let C be the set of all the segment extensions that either appear in some fan, or are edge segments, diagonal segments, or extensions of segments in Σ that lose to $\partial\Sigma - \Sigma$. Using Lemma 2, we see that at this point in time each connected component of C is: a fan, an edge segment, a diagonal segment, or extension of a segment that loses to $\partial\Sigma - \Sigma$. Now we add to C, one by one, the remaining segment extensions of Σ and show that the number of connected components of C does not change. While there is a segment extension of Σ not already placed in C, choose one such segment S which loses to a segment already placed in C. Since all the fans are already placed in C, there must be one such segment. Segment S, when placed in C, would belong to the same connected component of C which contains the segment to which S loses, and it will not cause any merge of two connected components. Continue this process until all segment extensions are added to C. When this is done, C is the tournament pseudoforest and each of its connected components contains one and only one of the four kinds (i)–(iv) mentioned in the statement of the Lemma. $\square$

Corollary 1. *$TP(\Sigma)$ is a pseudoforest.*

Before describing the algorithm to construct $TP(\Sigma)$ and the $TP - subdivision$, let us say a few words about the representation of the data structure. We will maintain each connected component of $TP(\Sigma)$ as a rooted pseudotree (with appropriate bidirectional pointers with the $TP - subdivision$), where, by using Lemma 3, the root is chosen to be its unique fan, edge segment, diagonal segment, or its segment which loses to $\partial\Sigma - \Sigma$. Except where the root corresponds to a fan, the rest of the pseudotree is like a standard tree. We refer to this entire data structure collectively as the tournament pseudoforest data structure, denoted $\Psi(\Sigma)$.

In the rest of this section we will describe our algorithm to construct $\Psi(\Sigma)$ in $O(n \log n)$ time using the sweep method. We sweep Σ by a horizontal sweep line along the direction of the y-axis, where the event points of the sweep are the endpoints of the segments of Σ, plus some "tournament intersection points". There are a total of $O(n)$ event points and each contributes $O(\log n)$ to the time complexity.

The algorithm will maintain the portion of the pseudoforest below the sweep line, excluding the extension (below the sweep line) of those segments that are above the sweep line and hence not yet considered. The intersection of the sweep line and the edges of the pseudoforest built so far will be called the *active set*. We use the natural linear ordering of points in the active set, i.e., in increasing order of x-coordinate. The edges of the current pseudoforest that intersect the sweep line (at the points of the active set) are called the *active edges*. We use the same natural ordering on the active edges as their corresponding active points. The *event-queue*, a priority queue, will maintain those endpoints of the segments that are at or above the sweep line, and the intersection points, above the sweep line, of adjacent active edges, in increasing order of their y-coordinates. (Here, two active edges are called adjacent, if they are consecutive in their natural ordering.) These intersection points are potential "tournament points". If the sweep line reaches the bottom end point of a segment which is not on $\partial\Sigma$, its extension will penetrate below the sweep line. To determine the extension end point of this segment below the sweep line in logarithmic time we would need efficient ray shooting to determine the first edge of the current pseudoforest hit by this extension. (To learn about the idea of ray shooting see, for example,

Ede87 To accomplish this, we will extend the natural ordering of the active set as follows. The regions of the current pseudoforest that are just below and intersect the sweep line, form convex *bays*. We will extend the active set of edges, by including in it all boundary edges of all the bays (i.e., their bottom chains). The natural ordering used on this *extended active set* is to order the edges of each bay consecutively counterclockwise (as we go from its left to right intersection points with the sweep line), and order all edges of bay B_1 before all edges of bay B_2, if the left end intersection of B_1 with the sweep line is to the left of that of B_2. (Note that the bottom chain of a bay is not necessarily x-monotone.) We maintain this extended active set, according to its natural linear ordering, in a balanced search tree called the *active-set-tree*. (See Figure 9.)

[Figure 9 about here.]

Now we perform the sweep as follows. Initially the active-set-tree is empty, and the event-queue contains the $2n$ end points of the segments in Σ. As we continue the sweep, suppose p is the next item removed from the event-queue.

If p is a bottom end point of a segment, say S, that is not a vertex of $\partial\Sigma$ (the case that p is a vertex of $\partial\Sigma$ is rather obvious), then we perform a ray shooting on the bays, by consulting the active-set-tree, and split the corresponding bay, say B, into two bays B' and B'' along the extension of S. (Segment S loses to the edge of B hit by the extension of S.) Then we need to update the active-set-tree in the obvious way, and also update the event-queue by removing the "tournament point" corresponding to bay B (if there is any), and inserting the ones for B' and B'' (if there are any). It is clear that it takes $O(\log n)$ time to process p.

If p is the top end of a segment S, which is not a vertex of $\partial\Sigma$, then there is only a status change necessary. We convert from segment S to its extension. Again, this can obviously be done in $O(\log n)$ time.

If p is a "tournament point" and it is the intersection of the "end edges" of a bay, say B_i, then processing of p amounts to "closing up" B_i as follows. Remove all edges of B_i from the active-set-tree; appropriately declare a winner among the two "end edges" of B_i that play the match at p; insert in the event-queue the new tournament point (if any) caused by this winner segment and its new adjacent bay; finally, remove from the event-queue the tournament point between the looser segment of the match at p and its other neighbor (if any). (A similar "closing up" of a bay may occur if p were a vertex of $\partial\Sigma$ and the top end point of a segment.) This step takes $O(\log n + r_i)$ time, where r_i is the number of edges of the bay B_i which is closed up. This is caused by the collective deletion of the edges of B_i from the active-set-tree. This can be done by two split operations followed by a join, assuming the data structure for the active-set-tree is chosen appropriately (e.g., any balanced or self adjusting search tree such as a red-black tree, or a splay tree). Then the nodes of the middle subtree (corresponding to the deactivated bay) that is cut out of the active-set-tree are disposed.

There are a total of $O(n)$ "tournament points" ever processed by the algorithm. Therefore the overall time complexity of the algorithm is $O(n \log n + \sum_i r_i)$, where r_i is the size complexity of bay (or face) B_i. Clearly $\sum_i r_i$ is $O(n)$, since the tournament pseudoforest is a linear size planar subdivision. Therefore, the overall time complexity of the sweep algorithm is $O(n \log n)$. We would need additional $O(n)$ time to traverse the data structure to complete the construction of the pseudoforest and the corresponding subdivision. We conclude the following.

Lemma 4. *The tournament pseudoforest can be constructed in $O(n \log n)$ time.* □

5. Proof of Theorem 2

We use the tournament pseudoforest data structure $\Psi(\Sigma)$ and a *careful* refinement of the proof of Theorem 1. The entire process takes $O(n)$ time, assuming $\Psi(\Sigma)$ is already constructed. We work on the pseudotrees of $TP(\Sigma)$ one by one. We should note that as we proceed to smaller subproblems, we may have edge segments that share common vertices as discussed in the proof of Theorem 1. However, this does not cause significant

difficulty; we appropriately generalize the definition of the tournament pseudoforest and its subdivision to make the necessary accommodation. We omit most details here and sketch only the case corresponding to case (ii) in the proof of Theorem 1 (the most involved case): The root of the pseudotree provides the cyclic sequence of segments (cycle-segments) that form the fan and the convex "hole". Then, for each cycle-segment, we use depth-first-search to go down the pseudotree (akin to the idea of topological ordering of vertices around a plane tree as described in

AGS89 and *clip off* maximally connected trails starting from those that intersect the cycle-segment (but not its trail) or intersect the edge of $\partial\Sigma$ adjacent to the cycle-segment and on the appropriate side. (See Figure 10.) In this way, we cut the problem into *many* subproblems, *all extremally situated*.

[Figure 10 about here.]

Each step of the clipping process is charged to one of the trails that is clipped off, $O(1)$ time to each. But the number of such trails is in the order of (almost equal to) the number of segments in Σ that were internal but have become edge segments in the smaller subproblems. This is crucial in proving the linear time complexity, given $\Psi(\Sigma)$.

The optimality of the algorithm is implied by an easy linear time reduction from sorting: Suppose we are given n numbers x_1, x_2, $\cdots$, x_n to sort. Let $a = \min_i x_i - 1$ and $b = \max_i x_i + 1$. Let $\Sigma = \{ S_1, S_2, \cdots, S_n \}$ be a set of n vertical line segments, where the x-coordinate of S_i is x_i and the y-coordinates of its two ends are $\pm(x_i - a)(x_i - b)$. (See Figure 11.) The set Σ is clearly extremally situated and $\partial\Sigma$ is its unique circumscribing polygon. Given the order of vertices around $\partial\Sigma$, the sorted ordering of x_i's can be inferred in linear time.

[Figure 11 about here.]

6. Applications

First we mention some applications of circumscribing polygons with respect to Euclidean matching problems. The Euclidean matching problem is: given a set of $2n$ points in the plane, find a perfect matching by choosing n pairwise disjoint line segments, called the matching edges, whose vertices are the given points and whose total length is minimized. Vaidya

Vai89 showed that this problem can be solved in $O(n^{2.5} \log^4 n)$ time. Marcotte and Suri

MaS89 considered a variation of the Euclidean matching problem where the given points are some vertices of a given simple polygon and the matching edges are restricted to be

edges or internal diagonals of the polygon. They showed that this version can be solved in $O(n \log^2 n)$ time.

An open problem in

MaS89 is the following. Let M_o be the weight of a minimum weight Euclidean matching of a set S of $2n$ points; P be a simple polygon that spans the set of points; and M_P be the weight of the minimum weight matching in which all matching edges are constrained to be edges or internal diagonals of the polygon. The question is: what polygon P achieves the minimum ratio M_P/M_o? $M_P/M_o = 1$ is the minimum ratio if and only if a circumscribing polygon P of the segments that form the optimum matching exists.

Circumscribing polygons can also be used in a partial verification algorithm for the Euclidean matching problem: Given a set M of n pairwise disjoint matching edges, we are asked whether M forms a minimum weight matching of the $2n$ points. We first find a circumscribing polygon P of M (assuming this exists and can be found quickly), then apply Marcotte and Suri's $O(n \log^2 n)$ algorithm to find an optimum matching M' in P. M is a minimum weight Euclidean matching of the $2n$ points only if M and M' have the same weight. (Of course the converse does not hold, i.e., if M and M' have the same weight then M is not necessarily optimum.)

Finally, let us mention that this paper introduces the new data structure, namely, the tournament pseudoforest. It would be interesting to explore other applications and extensions of this data structure.

7. Open Problems

Settling Conjectures 0, 1, and 2 are obvious open problems. Another open problem worth mentioning is the following. The $\Omega(n \log n)$ lower bound stated in the proof of Theorem 2 is essentially due to finding the convex hull of Σ. A natural question to ask is: if the convex hull of Σ is given, can we compute the tournament pseudoforest in $o(n \log n)$ time, say, in linear time?

Another open problem akin to Marcotte and Suri's approach is the following. Suppose we are given a simple polygon P with an even number of vertices. Find a minimum weight Euclidean matching between the vertices of P subject to the constraint that no matched segment crosses the boundary of P. Note that P is a spanning polygon of any such matching.

8. Acknowledgment

9. Note Added in Print

We have learned that Urabe and Watanabe have recently found a counter-example to Conjecture 0.