SIAM J. Comput.
Vol. 28, No. 4, pp. 1298–1309

GUILLOTINE SUBDIVISIONS APPROXIMATE POLYGONAL SUBDIVISIONS: A SIMPLE POLYNOMIAL-TIME APPROXIMATION SCHEME FOR GEOMETRIC TSP, k-MST, AND RELATED PROBLEMS [*]

JOSEPH S. B. MITCHELL[†]

Abstract. We show that any polygonal subdivision in the plane can be converted into an "m-guillotine" subdivision whose length is at most $(1+\frac{c}{m})$ times that of the original subdivision, for a small constant c. "m-Guillotine" subdivisions have a simple recursive structure that allows one to search for the shortest of such subdivisions in polynomial time, using dynamic programming. In particular, a consequence of our main theorem is a simple polynomial-time approximation scheme for geometric instances of several network optimization problems, including the Steiner minimum spanning tree, the traveling salesperson problem (TSP), and the k-MST problem.

Key words. approximation algorithms, polynomial-time approximation scheme, traveling salesperson problem, k-MST, Steiner minimal trees, guillotine subdivisions, computational geometry

AMS subject classifications. 68Q25, 68R10, 68U05

PII. S0097539796309764

1. Introduction.

We obtain a simple polynomial-time approximation scheme for geometric instances of some network optimization problems, including the Steiner minimum spanning tree, the traveling salesperson problem (TSP), and the k-MST problem.

The method is based on the concept of an "m-guillotine subdivision," a simple extension of the recent approximation method of Mitchell [9], which considered the case $m = 1$. Roughly speaking, an "m-guillotine subdivision" is a polygonal subdivision with the property that there exists a line ("cut"), whose intersection with the subdivision edges consists of a small number ($O(m)$) of connected components, and the subdivisions on either side of the line are also m-guillotine. The upper bound on the number of connected components allows one to apply dynamic programming to optimize over m-guillotine subdivisions, as there is a succinct specification of how subproblems interact across a cut.

Key to our method is a theorem showing that any polygonal subdivision can be converted into an m-guillotine subdivision by adding a set of edges whose total length is small: at most $\frac{c}{m}$ times that of the original subdivision (where $c = 1, \sqrt{2}$, depending on the metric). Then, using dynamic programming to optimize over an appropriate class of m-guillotine subdivisions, we obtain, for any fixed m, $(1+\frac{c}{m})$-approximation algorithms that run in polynomial-time ($n^{O(m)}$) for various network optimization problems.

Related work. Over the last few decades, there has been a wealth of research on the problems studied here, both in the graph versions of the problems and in the geometric versions. Almost any standard textbook on algorithms and networks discusses them; e.g., see [4, 6, 12]. For a survey of work on the TSP, refer to the

[*] Received by the editors September 25, 1996; accepted for publication (in revised form) July 18, 1997; published electronically March 22, 1999. This research was supported in part by Hughes Research Laboratories and NSF grants CCR-9204585 and CCR-9504192.

http://www.siam.org/journals/sicomp/28-4/30976.html

[†] Department of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY 11794-3600 (jsbm@ams.sunysb.edu).

book [7] edited by Lawler et al. For a survey on approximation algorithms, refer to the recent book [5] edited by Hochbaum.

All of the geometric optimization problems considered here are known to be NP-hard. Polynomial-time approximation algorithms were known, allowing one to get within a constant factor of optimal. However, it has been open as to whether or not one can, in polynomial time, achieve an approximation factor of $(1+\epsilon)$ for any fixed $\epsilon > 0$; i.e., no polynomial-time approximation scheme (PTAS) was known. In particular, no factor better than the Christofides bound of 1.5 was known for the Euclidean TSP.

In this paper, we point out how a minor modification to a previous result [9, 11] (see also [3]) leads to a PTAS for various geometric optimization problems, including the TSP, Steiner tree, and k-MST.

In an exciting recent development, Arora [1] announced that he had found a PTAS for the Euclidean TSP, as well as the other problems considered in this paper, thereby achieving essentially the same results that we report here, using decomposition schemes that are somewhat similar to our own. Arora's remarkable results predate this paper by several weeks. Arora has generalized his method also to higher dimensions, obtaining a running time of $n^{O(\log^{d-2} n)/\epsilon^{d-1}}$ in d dimensions. For the two-dimensional TSP, Arora estimates that his analysis yields a time bound of roughly $n^{100/\epsilon}$; he adds that a more careful analysis should yield roughly $n^{30/\epsilon}$. In this paper, we give analysis giving an explicit exponent: time $O(n^{20m+5})$ to get within factor $(1+\frac{2\sqrt{2}}{m})$ of optimal in the Euclidean planar TSP; in terms of ϵ, the time bound is $O(n^{\frac{40\sqrt{2}}{\epsilon}+5})$.

2. m-Guillotine subdivisions.

We follow most of the notation of [9], only somewhat generalized. We consider a polygonal subdivision ("planar straight-line graph") S that has n edges (and hence $O(n)$ vertices and facets). Let E denote the union of the edge segments of S, and let V denote the vertices of S. We can assume (without loss of generality) that S is restricted to the unit square, B (i.e., $E \subset \text{int}(B)$). Then, each facet (2-face) is a bounded polygon, possibly with holes. The *length* of S is the sum of the lengths of the edges of S. If all edges E are horizontal or vertical, then we say that S is *rectilinear*.

A closed, axis-aligned rectangle W is a *window* if $W \subseteq B$. In the following definitions, we fix attention on a given window, W.

A line ℓ is a *cut* for E (with respect to W) if ℓ is horizontal or vertical and $\ell \cap \text{int}(W) \neq \emptyset$. The intersection, $\ell \cap (E \cap \text{int}(W))$, of a cut ℓ with $E \cap \text{int}(W)$ (the restriction of E to the window W) consists of a discrete (possibly empty) set of subsegments of ℓ. (Some of these "segments" may be single points, where ℓ crosses an edge.) The endpoints of these subsegments are called the *endpoints along* ℓ (with respect to W). (The two points where ℓ crosses the boundary of W are not considered to be endpoints along ℓ.) Let ξ be the number of endpoints along ℓ, and let the points be denoted by $p_1, \ldots, p_\xi$, in order along ℓ.

For a positive integer m, we define the *m-span*, $\sigma_m(\ell)$, of ℓ (with respect to W) as follows. If $\xi \leq 2(m-1)$, then $\sigma_m(\ell) = \emptyset$; otherwise, $\sigma_m(\ell)$ is defined to be the (possibly zero-length) line segment, $p_m p_{\xi-m+1}$, joining the mth endpoint, p_m, with the mth-from-the-last endpoints, $p_{\xi-m+1}$. (See Figure 2.1.)

A (horizontal or vertical) cut ℓ is an *m-perfect cut with respect to* W if $\sigma_m(\ell) \subseteq E$. In particular, if $\xi \leq 2(m-1)$, then ℓ is trivially an m-perfect cut (since $\sigma_m(\ell) = \emptyset$). Similarly, if $\xi = 2m-1$, then ℓ is m-perfect (since $\sigma_m(\ell)$ is a single point). Otherwise, if ℓ is m-perfect, and $\xi \geq 2m$, then $\xi = 2m$. See Figure 2.2 for an example.

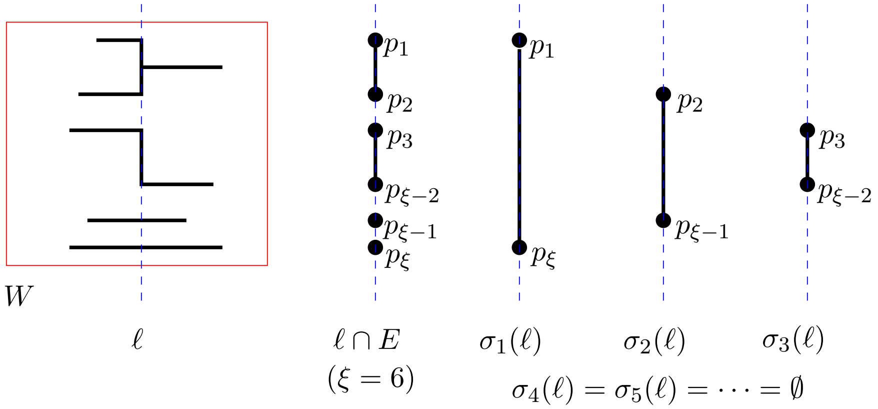

FIG. 2.1. *Definition of m-span.*

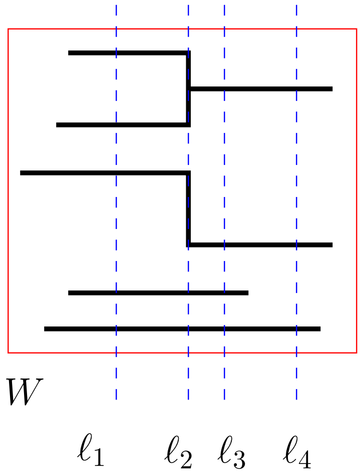

FIG. 2.2. *The vertical cuts $\ell_1, \ell_2, \ell_3, \ell_4$ are 3-perfect (also m-perfect, for $m \geq 4$). The cut ℓ_4 is also 2-perfect (but not 1-perfect).*

Finally, we say that S is an *m-guillotine subdivision with respect to window W* if either (1) $E \cap \text{int}(W) = \emptyset$; or (2) there exists an m-perfect cut, ℓ, with respect to W, such that S is m-guillotine with respect to windows $W \cap H^+$ and $W \cap H^-$, where H^+, H^- are the closed half-planes induced by ℓ. We say that S is an *m-guillotine subdivision* if S is m-guillotine with respect to the unit square, B. A 1-guillotine subdivision is illustrated in Figure 2.3, where "cut" is used to indicate where a 1-perfect cut can be made.

3. The main theorem.

For rectilinear subdivisions, the proof of our main theorem directly follows that of [9] with only a very minor change: we use the concept of "m-darkness," in which we require m walls to block light from the boundary. The proof in [9] used $m = 1$; so we copy below the proof from [9] with "m" replacing "1".

THEOREM 3.1. *Let S be a rectilinear subdivision, with edge set E, of length L. Then, for any positive integer m, there exists an m-guillotine rectilinear subdivision, S_G, of length at most $(1 + \frac{1}{m})L$ whose edge set, E_G, contains E.*

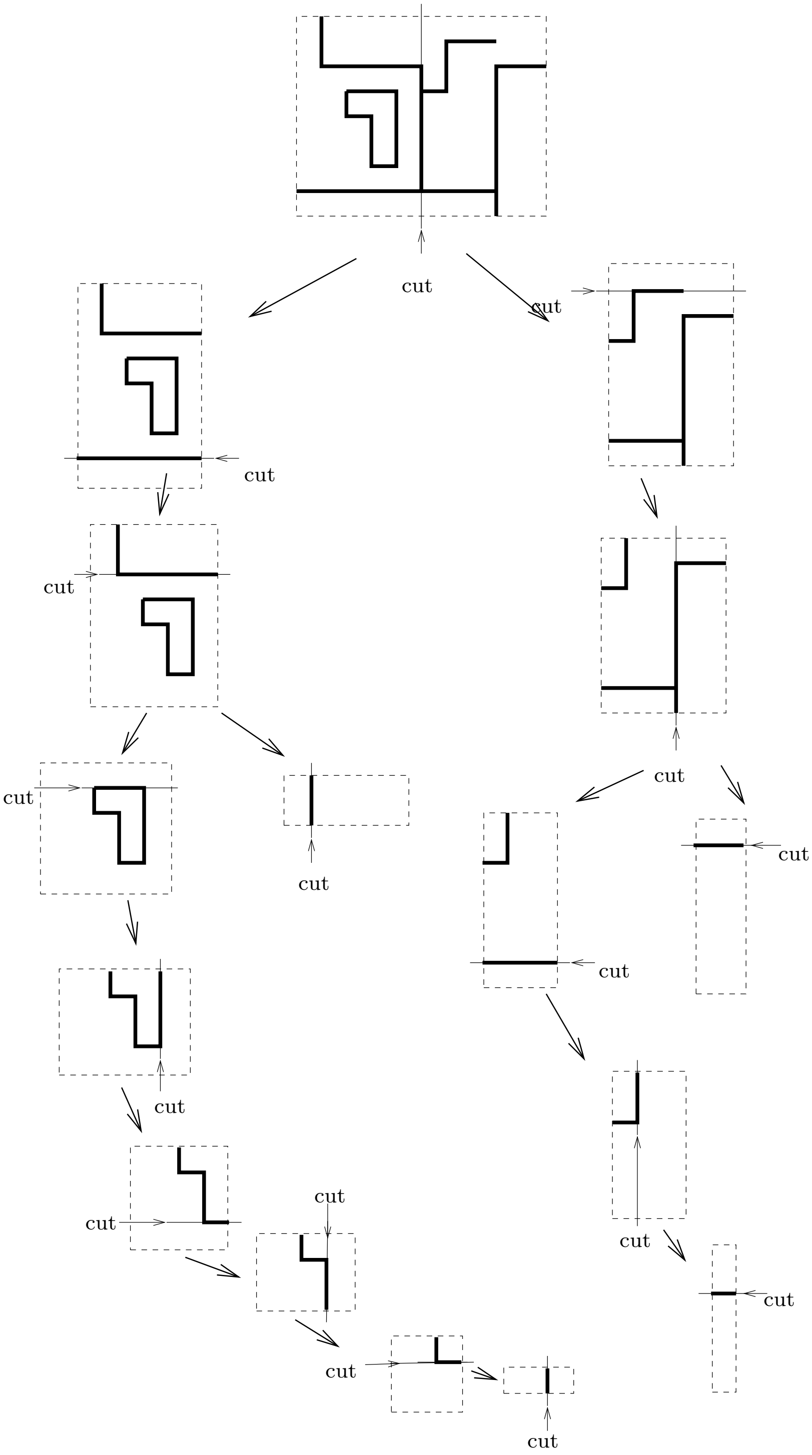

FIG. 2.3. *An example of a* 1*-guillotine rectilinear subdivision.*

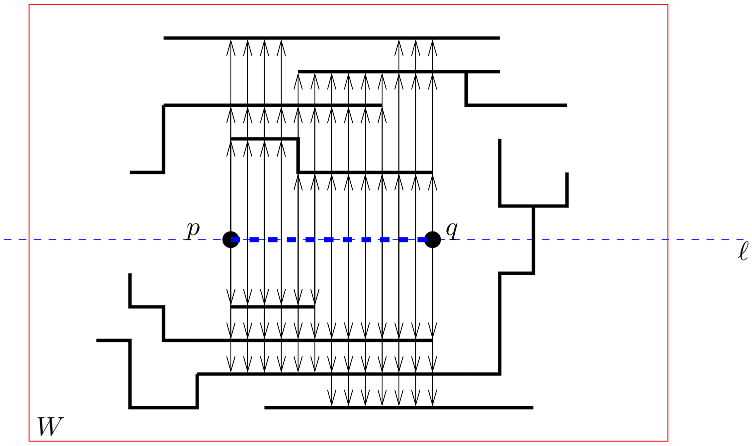

FIG. 3.1. *Subsegment $pq \subset \ell$ is 3-dark; its length is charged to the 3 levels of E that lie above it and the 3 levels of E that lie below it.*

Proof. We will convert S into an m-guillotine subdivision S_G by adding to E a new set of horizontal/vertical edges whose total length is at most $\frac{1}{m}L$. The construction is recursive; at each stage, we show that there exists a cut, ℓ, with respect to the current window W (which initially is the box B), such that we can afford to add the m-span $\sigma_m(\ell)$ to E, while appropriately charging off the length of $\sigma_m(\ell)$. (Once we add $\sigma_m(\ell)$ to E, ℓ becomes an m-perfect cut with respect to W.)

In fact, we will restrict ourselves to a special discrete set, $\mathcal{L}$, of horizontal/vertical cuts, namely those determined by the x- or y-coordinates of original vertices V of the subdivision, or by the midpoints between consecutive x- or y-coordinates of V.

First, note that if an m-perfect cut (with respect to W) exists, then we can simply use it and proceed, recursively, on each side of the cut.

We say that a point p on a cut ℓ is *m-dark with respect to ℓ and W* if, along $\ell^\perp \cap \text{int}(W)$, there are at least m endpoints (strictly) on each side of p, where $\ell^\perp$ is the line through p and perpendicular to ℓ.[1] We say that a subsegment of ℓ is *m-dark* (with respect to W) if all points of the segment are m-dark with respect to ℓ and W.

The important property of m-dark points along ℓ is the following. Assume, without loss of generality, that ℓ is horizontal. The m-dark portion of ℓ is, in general, a union of subsegments of $\ell \cap W$, some endpoints of which may not be themselves m-dark points. (In terms of the *length* (measure) of the m-dark portion, though, the discrete set of endpoints is not relevant.) Let pq be an open subsegment of the m-dark portion of ℓ. Then we can charge the length of pq off to the bottoms of the first m subsegments, $E^+ \subseteq E$, of edges that lie above pq, and the tops of the first m subsegments, $E^- \subseteq E$, of edges that lie below pq (since we know that there are at least m edges "blocking" pq from the top/bottom of W). We charge pq's length half to E^+ (charging each of the m levels of E^+ *from below*, with $\frac{1}{2m}$ units of charge) and half to E^- (charging each of the m levels of E^- *from above*, with $\frac{1}{2m}$ units of charge). In Figure 3.1 we illustrate how a 3-dark subsegment, pq, has its length charged off to the 3 levels of E that are above/below it.

[1] We can think of the edges E as being "walls" that are not very effective at blocking light—light can go through $m-1$ walls but is stopped when it hits the mth wall; then, p on a line ℓ is m-dark if p is not illuminated when light is shone in from the boundary of W, along the direction of $\ell^\perp$.

We call a cut ℓ *favorable* if the total length of the m-dark portion of ℓ is at least as great as the length of the m-span, $\sigma_m(\ell)$. The lemma below shows that a favorable cut always exists (even one in the special discrete set, $\mathcal{L}$). For a favorable cut ℓ, we add its m-span to the edge set (charging off its length, as above), and recurse on each side of the cut, in the two new windows. After a portion of E has been charged on one side, due to a cut ℓ, it will be within m levels of the boundary of the windows on either side of ℓ, and, hence, within m levels of the boundary of any future windows, found deeper in the recursion that contain the portion. Thus, *no portion of E will ever be charged more than once from each side* (top and bottom), so no portion of E will ever pay more than its total length, times $\frac{1}{m}$, in charge ($\frac{1}{2m}$ from each side). Also, the new edges added (the spans $\sigma_m(\ell)$) are never themselves charged, since they lie on window boundaries and cannot therefore serve to make a portion of some future cut m-dark.

(Note too that, in order for a cut ℓ to be favorable but not m-perfect, there must be at least one segment (in fact, at least m segments) of E parallel to ℓ in each of the two open half-planes induced by ℓ; thus, the recursion must terminate in a finite number of steps.)

Since the total length of all m-spans for all favorable cuts is at most $\frac{1}{m}L$, and the total length of all m-spans for all m-perfect cuts is at most L, we are done. □

We now prove our key lemma, which guarantees the existence of a favorable cut. The proof of the lemma uses a particularly simple argument, based on elementary calculus (reversing the order of integration). It appears already in [9], but we repeat it here for completeness.

LEMMA 3.2. *For any subdivision S and any window W, there is a favorable cut.*

Proof. We show that there must be a favorable cut that is either horizontal or vertical.

Let $f(x)$ denote the length of the m-span (with respect to W) of the vertical line through x. (We define $f(x) = 0$, if $x \in [0, 1]$ lies outside of the projection of $W \subseteq B$ onto the x-axis.) Then, $\int_0^1 f(x)dx$ is simply the area, A_x, of the set R_x of points of W that are m-dark with respect to horizontal cuts. Similarly, define $g(y)$ to be the length of the m-span of the horizontal line through y, and let $A_y = \int_0^1 g(y)dy$.

Assume, without loss of generality, that $A_x \geq A_y$. We claim that there exists a horizontal favorable cut; i.e., we claim that there exists a horizontal cut, ℓ, such that the length of its m-dark portion is at least as large as the length of its m-span, $\sigma_m(\ell)$. To see this, note that A_x can be computed by switching the order of integration, "slicing" the region R_x horizontally, rather than vertically; i.e., $A_x = \int_0^1 h(y)dy$, where $h(y)$ is the length of the intersection of R_x with a horizontal line through y (i.e., $h(y)$ is the length of the m-dark portion of the horizontal line through y). Thus, since $A_x \geq A_y$, we get that $\int_0^1 h(y)dy \geq \int_0^1 g(y)dy \geq 0$. Thus, it cannot be that for all values of $y \in [0, 1]$, $h(y) < g(y)$, so there exists a $y = y^*$ for which $h(y^*) \geq g(y^*)$. The horizontal line through this y^* is a cut satisfying the claim of the lemma. (If, instead, we had $A_x \leq A_y$, then we would get a *vertical* cut satisfying the claim.)

Finally, we note that, in the rectilinear case, f, g, and h are piecewise-constant, with discontinuities corresponding to vertices V of S. Then, we can always select y^* to be at a discontinuity or at a midpoint between two discontinuities. □

General polygonal subdivisions. Consider now a subdivision S whose edges E are *not* rectilinear. A moment's reflection reveals that our charging scheme and the key lemma are quite general and do not really use the rectilinearity of S (or even the straightness of edges in E). In fact, the proof of the main theorem goes through, almost exactly as before, adding "favorable" cuts that are horizontal or vertical, and charging the added length off to the original length of the subdivision. However, we must address the issue of the discretization of cuts (e.g., in order to specify a dynamic program) and, thereby, the termination of the recursion that converts an arbitrary subdivision to an m-guillotine subdivision.

One approach (as in earlier drafts of this paper) is to use discretization of orientations, and/or discretization onto a sufficiently fine grid. Here, we opt instead to modify slightly our previous definition of m-guillotine subdivision as follows.

Assume, without loss of generality, that no two vertices have a common x- or y-coordinate. We let E_W denote the subset of E consisting of the union of segments of E having at least one endpoint inside (or on the boundary of) W. The *combinatorial type* (*with respect to* E) of a window W is an ordered listing, for each of the four sides of W, of the identities of the (closed) line segments of E_W that intersect it (the side of W, each considered to be an open line segment), as well as the identities of the line segments of E_W that intersect each of the four corners of W. We say that W is *minimal* (with respect to E) if there does not exist a $W' \subset W$, strictly contained in W, having the same combinatorial type as W. By standard critical placement arguments,[2] we see that, since it has four degrees of freedom, a minimal window is determined by four *contact pairs*, defined by a vertex $v \in V$ in contact with a side of W or by a corner of W in contact with a segment of E_W. (Thus, there are at most $O(n^4)$ minimal windows.) For any window W containing at least one vertex of E, we let $\overline{W}$ denote a minimal window, contained within W, having the same combinatorial type as W. (At least one such $\overline{W}$ must exist.)

Now, we say that S is an *m-guillotine subdivision with respect to window* W if either (1) $V \cap \text{int}(W) = \emptyset$; or (2) there exists an m-perfect cut, ℓ, with respect to a minimal window, $\overline{W} \subseteq W$, such that S is m-guillotine with respect to windows $W \cap H^+$ and $W \cap H^-$, where H^+, H^- are the closed half-planes induced by ℓ. (Note that, since $\overline{W}$ is minimal, necessarily windows $W \cap H^+$ and $W \cap H^-$ will each have a combinatorial type different from that of W.)

THEOREM 3.3. *Let S be a polygonal subdivision, with edge set E, of length L. Then, for any positive integer m, there exists an m-guillotine polygonal subdivision, S_G, of length at most $(1 + \frac{\sqrt{2}}{m})L$ whose edge set, E_G, contains E.*

Proof. The proof exactly follows that of the rectilinear case (Theorem 3.1): We use a recursive construction, together with our charging scheme, and Lemma 3.2 applied

[2] In particular, an arbitrary window, W, can be made minimal by the following "shrinking" process: Slide the right side of W inward until the first *event*, when (a) the side hits a vertex, or (b) an endpoint of the side strikes a segment of E_W. In case (a), we have "pinned" the right side of W, and we move next to the top side (then the left side, and the bottom side). In case (b), we next move inwards *both* of the two sides incident on the corner (endpoint) that struck a segment, $s \subset E_W$, while keeping that corner in contact with segment s, until another event (of type (a) or (b)) occurs. (Since we know that $s \in E_W$ has one of its endpoints inside W, we know that as we slide the corner along s, the two sides either both move inward or both move outward.) Again, in case of an event of type (a), we are done moving those two sides (they are both pinned), and we proceed to move inwards other sides (processing unpinned sides in the order right, top, left, bottom). In case of an event of type (b), a second corner of the window has struck a segment, $s' \subset E_W$, and we then proceed to slide *three* sides of W inward simultaneously, while keeping both of the pinned corners sliding on their respective segments of E_W. We may end up having all four sides of W moving inwards at once, with three, or possibly even four (in a degenerate case), corners of W restricted to slide on specific segments of E_W. Since each of these segments of E_W must have one endpoint within W, we know that we will eventually have an event of type (a)—a collision with a vertex. When this happens, all four sides of W have been pinned.

to a minimal window $\overline{W} \subseteq W$, to show that we can convert S into an m-guillotine subdivision S_G by adding to E a new set of horizontal/vertical bridge segments whose total length is at most $\frac{\sqrt{2}}{m}L$. (Here, we do not restrict attention to any special subset of horizontal and vertical cuts (such as $\mathcal{L}$); cuts can be classified according to the combinatorial types of the new windows they create.)

The only difference that we should mention is the origin of the "$\sqrt{2}$" term in the bound. This comes from the fact that each side of an inclined segment of E may be charged *twice*, once vertically and once horizontally. Specifically, the charge assigned to a segment is at most $\frac{1}{m}$ times the sum of the lengths of its x- and y-projections, i.e., at most $\frac{\sqrt{2}}{m}$ times its length. □

4. Some applications. We now discuss how our main theorem can be applied, leading to polynomial-time approximation schemes for some NP-hard optimization problems on a set P of n sites in the plane.

Steiner tree problem: Determine a tree of minimum total length that spans (visits) the set of points P. In this problem, as opposed to the *minimum spanning tree* (MST) problem (solvable exactly in $O(n \log n)$ time for points in the Euclidean plane [13]), the tree is allowed to have vertices ("Steiner points") that are *not* among the points of P.

Steiner k-MST problem: For a given integer k ($k \leq n$), determine a tree, possibly with Steiner points, of minimum total length that spans at least k of the n points in P.

TSP: Determine a *tour* (cycle) of minimum length that visits each point of the set, P, of sites. Because it is of minimum length, a Euclidean TSP tour will necessarily visit each site exactly once, and will be a *simple* polygon (a closed polygonal walk that does not self-intersect). The k-TSP is to determine a minimum-length tour on a subset of at least k of the n points.

k-MST problem: For a given integer k ($k \leq n$), determine a tree, with vertices in the set P, of minimum total length that spans at least k of the n points in P.

4.1. Steiner tree. The dynamic programming algorithm of [9, 11], which computes a 2-approximation to the Steiner k-MST (and hence to Steiner tree, for $k = n$), in the L_1 metric, generalizes immediately to the case of m-guillotine subdivisions. Now, instead of a factor of $1 + \frac{1}{1} = 2$, as in [9] (for the case $m = 1$), we get a factor $1 + \frac{1}{m}$. Exactly the same algorithm works, only now there are up to $2m$ endpoints per side of the rectangular subproblem (instead of 2 per side, as in [9]). Thus, there are $O(k)$ choices of the integer k' that specifies the number of sites that must be visited within a subproblem, $O(n^4)$ choices of rectangle B bounding a subproblem, $O(n^{4 \cdot 2m})$ choices for the endpoints on each side of the rectangle, $O(n^{2m+1})$ choices of cut (and endpoints along the cut), and $O(k)$ choices for how to partition k'. Overall, we get time $O(k^2 n^{10m+5})$. In the case that $k = n$, we do not need to choose the value of k', or how to partition it when we select a cut, since we are forced to visit all sites within a subproblem; thus, the factor of k^2 drops, leaving us with a time bound of $O(n^{10m+5})$.

For the Euclidean metric, essentially the same algorithms work, but we first augment P with a set, G, of candidate (approximate) Steiner points, such that each Steiner point in an optimal tree can be rounded to one of the candidate points, without increasing the size of the tree by much. (In contrast, in the rectilinear case, we can assume that all Steiner points lie (exactly) at vertices of the grid induced by horizontal/vertical lines through points of P; thus, no new candidates need to be added, since these grid points are potential subdivision vertices in the dynamic programming recursion that searches for a shortest possible m-guillotine subdivision.) One choice of set G that works is to let G be a regular (square) grid of $O(n^2M^2)$ points, with spacing $\frac{diam(P)}{nM}$, for some $M \geq m$. Then, each Steiner point in an optimal tree can be rounded to such a grid point by adding a segment of length less than $\frac{diam(P)}{nM}$; this adds an overall length of only $\frac{diam(P)}{M} \leq \frac{diam(P)}{m} \leq (\frac{1}{m})L^*$ to the optimal tree, where L^* is the length of the optimal tree. Our dynamic programming algorithm, then, searches for a minimum-length m-guillotine subdivision that spans all (or at least k) of the points of P, while allowing vertices of the subdivision to be placed at any of the points $P \cup G$.

Corollary 4.1. *Given any fixed positive integer m, and any set of n points in the plane, there is an $n^{O(m)}$-time algorithm to compute a Steiner spanning tree (or Steiner k-MST) whose length is within a factor $(1 + \frac{1}{m})$ of the minimum.*

4.2. TSP. For the TSP, we again apply dynamic programming, since the main theorem gives us an easy way to decompose the problem recursively.

Corollary 4.2. *For any fixed positive integer m, there is an $O(n^{20m+5})$ algorithm to compute an approximate TSP whose Euclidean length is within a factor $(1 + \frac{2\sqrt{2}}{m})$ of optimal. (For the L_1 metric, the factor is $(1 + \frac{2}{m})$, and the time bound $O(n^{10m+5})$.)*

Proof. (We present the details for the Euclidean metric; details for the L_1 metric are similar.) Let T^* be a minimum-length TSP tour for P, and let L^* denote its length.

The structure of the proof is typical of many approximation results: (a) We show that T^* can be transformed into a special type of subdivision of length at most $(1 + \frac{2\sqrt{2}}{m})L^*$; (b) we give a dynamic programming algorithm to compute (exactly) a shortest subdivision, having edge set E^*_G, of this special type; and (c) we show how to obtain a tour of P from the resulting subdivision, with the tour having length at most that of the subdivision.

(a) *Transforming T^*.* We know that T^* is a simple polygon with vertices P. Following the proof of Theorem 3.1, we add segments (m-spans of favorable cuts), called "bridges," to the subdivision T^*, in order to make it m-guillotine; *however*, when we add an m-span segment (corresponding to cut ℓ), we *double* it, creating a second copy that lies on top of the first copy. (The reason for the doubling will be made apparent below, in step (c).) Furthermore, we "slide" the doubled bridge, parallel to itself and in the direction that decreases (or does not increase) the length of the bridge, until it lies on a vertical/horizontal line through some point of P. The result is a *pinned double bridge*, "pinned" by this line through a point in P. We let S denote the resulting subdivision, and let E denote its edge set (including the doubled segments). The length of E is at most $(1+\frac{2\sqrt{2}}{m})L^*$, since, by Theorem 3.1, the bridges that we add have total length, prior to doubling, of at most $\frac{\sqrt{2}}{m}L^*$.

(b) *Dynamic programming algorithm.*

Input to Subproblem (see example in Figure 4.1):

1. A rectangle R, corresponding to a minimal window, determined by (up to) four points of P, together with some subset of the $O(m)$ edges defining the boundary information (below).
2. Boundary information specifying $\leq 2m$ crossing segments (each determined by a pair of points in P) that cross the boundary of R, and at most one

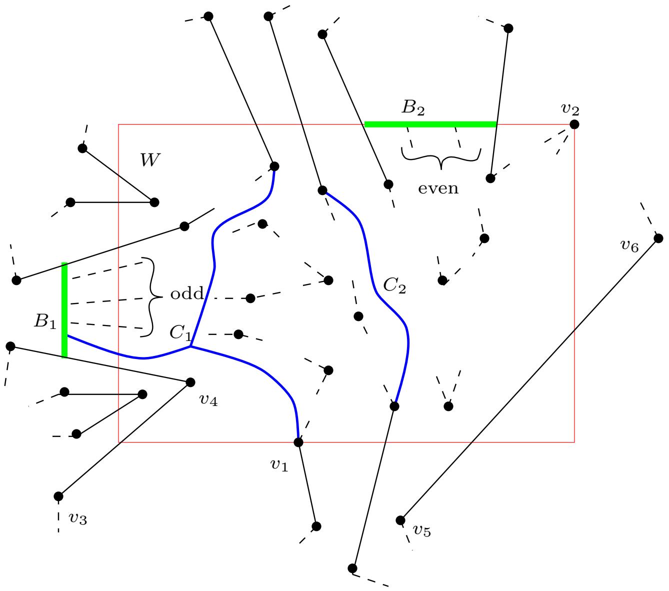

FIG. 4.1. *Example subproblem for TSP dynamic program: The window W is determined by the vertices v_1, v_2, and the segment v_3v_4 (on which its lower left corner is in contact). There are two bridges, B_1 (which is pinned at some coordinate left of the left boundary of W) and B_2 (pinned at the y-coordinate of v_2). Bridge B_1 is required to have an odd number of segments incident on it, while bridge B_2 is required to have an even number. The partition $\mathcal{P}$ specifies the interconnections indicated by trees C_1 and C_2. For vertex v_2, both incident edges are required to be within W; for vertex v_1, only one incident edge must be within W. All points within W, not yet incident on some segment crossing the boundary, are required to have degree two. Short dashed segments indicate a feasible possible set of interconnections.*

(pinned) bridge, per side of R, together with the parity (odd vs. even) of the number of edges of $E_G^* \cap R$ that must be incident on it.

3. Connectivity constraints, given in the form of a partition, $\mathcal{P}$, of the set of crossing segments and bridges, indicating which subsets are required to be connected within subproblem R.

Objective:

Compute a minimum-length m-guillotine subdivision, with edge set E_G^*, such that (1) the subdivision uses only segments joining points of P or *doubled* vertical/horizontal bridge segments that are pinned, (2) all boundary information is satisfied, (3) every point of P within R has degree two, and (4) all connectivity constraints within R are satisfied.

Note that there are $O(n^4 \cdot (n^{4m})^4) = O(n^{16m+4})$ possible inputs (subproblems), since there are $O(n^4)$ choices of R, and $O(n^{4m})$ choices of crossing segments on each of the four sides of R. (The number of possible connectivity constraints and boundary information is constant for fixed m.)

The initial call to the recursion will ask for a solution for the case that R is the bounding box of P, with empty boundary information, and connectivity constraints that simply say that all points of P inside R must be connected (with each point having degree 2).

In the base case, if R has no points of P in its interior, then the subproblem is solved by brute force, since it has only constant size (for fixed m). Otherwise, we can solve the subproblem recursively, optimizing over all choices associated with a cut:

1. $O(n)$ choices of a horizontal/vertical cut.
2. $O(n^{4m})$ choices of new boundary information on the cut. In particular, we select $\leq 2m$ segments (each determined by a pair of points) that cross the cut, together with the information specifying a possible bridge segment. We require that boundary information of the new subproblems be consistent with boundary information of the given problem.
3. $O(1)$ (for fixed m) choices of connectivity constraints for the two new subproblems determined by the cut, subject to the requirement that these constraints be consistent with the constraints $\mathcal{P}$.

Since there are $O(n \cdot n^{4m})$ choices to make in partitioning a subproblem and there are overall $O(n^{16m+4})$ subproblems, we obtain an overall time complexity of $O(n^{20m+5})$.

(c) *Obtaining a tour.* Given the solution, S_G^*, to the dynamic program, we claim that there is an Eulerian subgraph of E_G^* (the edge set of S_G^*, *including* doubled bridges) that covers P, so that any Eulerian cycle defines a tour of P, and the length of this tour is at most the length of E_G^*.

Now, by the constraints imposed in the subproblems of the dynamic programming algorithm, all vertices of S_G^* have even degree, *except* possibly those that lie along a bridge segment (that is an m-span of some perfect cut). But the fact that bridge segments have their lengths *doubled* allows us to make all vertices along bridges have even degree as well, by keeping an appropriate subset (selection of subsegments) of the second copy of a bridge: Consider the vertices along a single copy of the bridge, and use subsegments from the second copy of the bridge to link, consecutively, those vertices along the bridge that have odd degree. There will necessarily be an even number of odd-degree vertices along a bridge, since we require that a bridge segment be incident to an even number of other segments.

Thus, by deleting some portions of the second copy of bridge segments, we obtain a subdivision covering P all of whose vertices have even degree. Then, an Eulerian cycle on the edges of this subdivision is a tour of P, whose length is at most $(1 + \frac{2\sqrt{2}}{m})L^*$. □

4.3. *k*-MST. Suppose now that we are given an integer $k \leq n$ and asked to find a minimum spanning tree on some subset of k of the n points P, without using Steiner points. Since we are not allowed to introduce Steiner points, e.g., where segments are incident on bridges, we use the same bridge-doubling trick as we did in writing the dynamic program for the TSP. This ensures that the only odd-degree vertices that we end up with in our optimized subdivision are those at original points of P, thereby allowing us to replace Eulerian paths, linking original sites of P, with shortcut paths that bend only at the sites P. (Naturally, we drop the requirement in the dynamic program that the points in P have degree two, and we add the parameter k to the specification of a subproblem.)

5. Conclusion. We suspect that there are many other potential applications and improvements of the results reported here. We do not yet know a characterization of the general class of problems for which our method leads to a PTAS.

While we have concentrated on L_1 and Euclidean metrics, our results generalize to other metrics on points in the plane. We are currently examining possible extensions to higher dimensions.

Finally, we mention some recent results that have been discovered since the time that this paper was originally submitted. Arora [2] has recently obtained a randomized algorithm with expected running time that is nearly linear in n: $O(n \log^{O(1/\epsilon)} n)$. His new results also allow the d-dimensional problem to be solved in expected time $O(n(\log n)^{(O(\frac{d}{\epsilon}))^{d-1}})$.

Further, in a follow-up to this paper, Mitchell [10] has shown how a variant of m-guillotine subdivisions (termed "grid-rounded m-guillotine subdivisions") yields a (deterministic) PTAS with worst-case time bound $n^{O(1)}$, for any fixed $\epsilon > 0$, for the same set of (two-dimensional) problems we have studied in this paper.

Acknowledgments. I thank Esther Arkin, Rafi Hassin, Samir Khuller, Günter Rote, and the referees for several useful comments and suggestions that improved the presentation of the paper.

REFERENCES

[1] S. Arora, *Polynomial time approximation schemes for Euclidean TSP and other geometric problems*, in Proc. 37th Ann. IEEE Sympos. Found. Comput. Sci., Burlington, VT, IEEE Comput. Soc. Press, Los Alamitos, CA, 1996, pp. 2–12.

[2] S. Arora, *Nearly linear time approximation schemes for Euclidean TSP and other geometric problems*, in Proc. 38th Ann. IEEE Sympos. Found. Comput. Sci., Miami Beach, FL, IEEE Comput. Soc. Press, Los Alamitos, CA, 1997, pp. 554–563.

[3] A. Blum, P. Chalasani, and S. Vempala, *A constant-factor approximation for the k-MST problem in the plane*, in Proc. 27th Ann. ACM Sympos. Theory Comput., Las Vegas, NV, ACM Press, New York, 1995, pp. 294–302.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1990.

[5] D. Hochbaum, editor, *Approximation Problems for NP-Complete Problems*, PWS Publications, Boston, MA, 1997.

[6] E. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart, and Winston, New York, 1976.

[7] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, eds., *The Traveling Salesman Problem*, Wiley, New York, 1985.

[8] C. Mata and J. S. B. Mitchell, *Approximation algorithms for geometric tour and network design problems*, in Proc. 11th Ann. ACM Sympos. Comput. Geom., Vancouver, Canada, ACM Press, New York, 1995, pp. 360–369.

[9] J. S. B. Mitchell, *Guillotine subdivisions approximate polygonal subdivisions: A simple new method for the geometric k-MST problem*, in Proc. 7th ACM-SIAM Sympos. Discrete Algorithms, SIAM, Philadelphia, 1996, pp. 402–408.

[10] J. S. B. Mitchell, *Approximation algorithms for geometric optimization problems*, in Proc. 9th Canadian Conference Comput. Geom., Kingston, Canada, 1997, pp. 229–232.

[11] J. S. B. Mitchell, A. Blum, P. Chalasani, and S. Vempala, *A constant-factor approximation for the geometric k-MST problem in the plane*, SIAM J. Comput., 28 (1999), pp. 771–781.

[12] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, Englewood Cliffs, NJ, 1982.

[13] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.