

Guillotine subdivisions approximate polygonal subdivisions: Part II – A simple polynomial-time approximation scheme for geometric k -MST, TSP, and related problems*

Joseph S. B. Mitchell[†]

April 30, 1996; Last revision: July 24, 1996

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 salesman 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 [6], 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 ($O(n^{O(m)})$), 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 [2, 4, 8]. For a survey of work on the traveling salesman problem, refer to the book [5] edited by Lawler et al. For a survey on approximation algorithms, refer to the recent book [3] 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 whether or not one can, in polynomial time, achieve an approximation factor of $(1 + \epsilon)$, for any fixed $\epsilon > 0$; i.e., no PTAS (Polynomial-Time Approximation Scheme) was known. In particular, no factor better than the Christofides bound of 1.5 was known for the Euclidean TSP.

*Color slides for a talk on this subject are available at <http://ams.sunysb.edu/~jsbm/jsbm.html>

[†]jsbm@ams.sunysb.edu, Dept. of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY 11794-3600. Supported in part by Hughes Research Laboratories and NSF Grant CCR-9504192.

In this note, we point out how a minor modification to a previous result [6, 7] leads to a PTAS for various geometric optimization problems, including the TSP, Steiner tree, and k -MST.

In an exciting recent development, Sanjeev 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, and his discovery was done independently of this work.

2 m -Guillotine Subdivisions

We follow most of the notation of [6], 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 $\ell \cap \text{int}(W) \neq \emptyset$. 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, \dots, p_ξ , 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 m th endpoint, p_m , with the m th-from-the-last endpoints, $p_{\xi - m + 1}$. (See Figure 1.)

A line ℓ is an m -*perfect cut with respect to W* if $\sigma_m(\ell) \subseteq E$ (which implies that $\xi = 2m$ or $\xi = 2m - 1$ (in case $\sigma_m(\ell)$ is a single point)). See Figure 2 for an example.

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 halfplanes 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 4, 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 [6] 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 [6] used $m = 1$; so we copy below the proof from [6], with “ m ” replacing “1”.

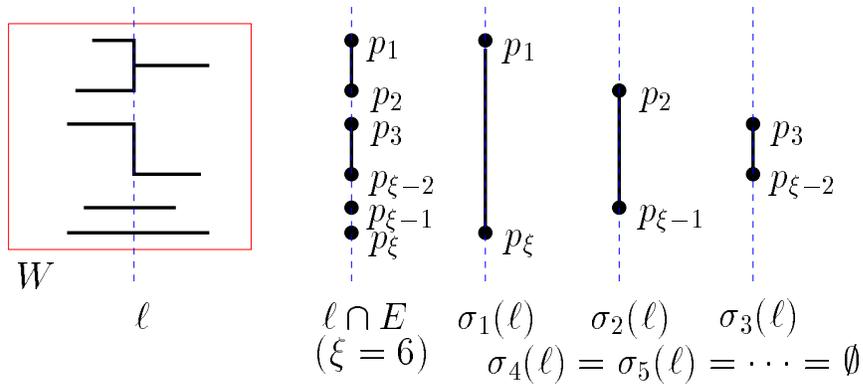


Figure 1: Definition of m -span.

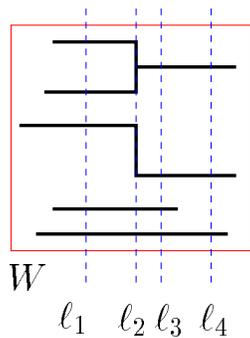


Figure 2: The vertical cuts l_1, l_2, l_3, l_4 are 3-perfect (also m -perfect, for $m \geq 4$). The cut l_4 is also 2-perfect (but not 1-perfect).

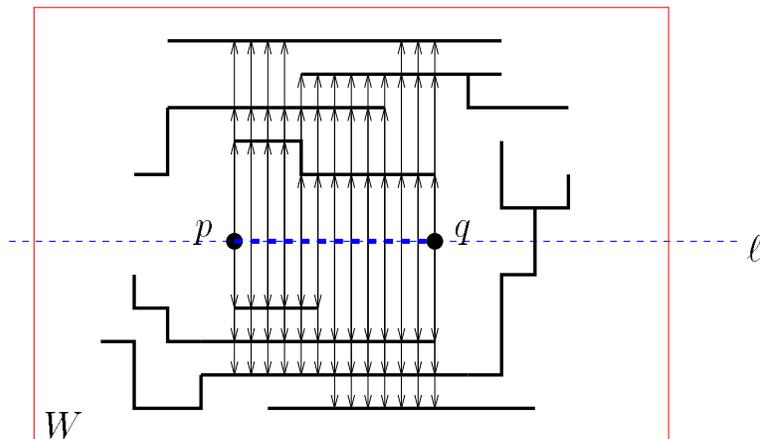


Figure 3: Subsegment $pq \subset l$ is 3-dark; its length is charged to the 3 levels of E that lie above/below.

Theorem 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 .*

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 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. Thus, we assume that no m -perfect cut exists with respect to a given window, W .

We say that a point p on a cut ℓ is *m -dark with respect to ℓ and W* if, along $\ell^- \cap \text{int}(W)$, there are at least m endpoints (strictly) on each side of p , where ℓ^- 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. Then, if all points on subsegment pq of ℓ are m -dark, 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 we illustrate how a 3-dark subsegment, pq , has its length charged off to the 3 levels of E that are above/below it.

We call a cut ℓ *favorable* if the m -dark portion of ℓ is at least as long as the m -span, $\sigma_m(\ell)$. The lemma below shows that a favorable cut always exists (even one in the special discrete set). 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 $1/m$, 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 halfplanes 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 spans for all m -perfect cuts is at most L , we are done. \square

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 [6], but we repeat it here for completeness:

¹We 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 m th 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 ℓ^\perp .

Lemma 1 *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 . Then, $\int_0^1 f(x)dx$ is simply the area, A_x , of the (x -monotone) region R_x of points of B 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. \square

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.

In earlier drafts of this manuscript, our approach was 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 follows².

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 line segments of E_W that intersect it. 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, 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

²I thank G. Rote for encouraging me to write the current version, avoiding discretizations, yielding a hopefully cleaner presentation.

closed halfplanes 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 2 *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 1): We use a recursive construction, together with our charging scheme, and Lemma 1 applied 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; 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. \square

4 Some Applications

We now discuss how our main theorem can be applied, leading to polynomial-time approximation schemes for some optimization problems on a set P of n sites in the plane — Steiner tree (and Steiner k -MST), TSP, and k -MST.

4.1 Steiner Tree

The dynamic programming algorithm of [6], 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 [6] (which did 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 [6]). 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 $O(k^2 n^{10m+5})$ (or $O(n^{10m+5})$, in case $k = n$).

For the Euclidean metric, essentially the same algorithms work, but we first add a set of potential (approximate) Steiner points for P (e.g., we can use a regular grid of points, with spacing $\frac{\text{diam}(P)}{nM}$, for some $M \geq m$).

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

4.2 TSP

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

Corollary 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; earlier drafts gave details for the L_1 metric, which is very 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 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 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 hits a point (in P); the result is a *pinned doubled bridge* that is “pinned” in the sense that it lies on a vertical/horizontal line through some point of 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 $(1 + \frac{2\sqrt{2}}{m})L^*$.

(b). Dynamic Programming Algorithm.

Input to Subproblem: (see example in Figure 5)

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 (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 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 Euler cycle on the edges of this subdivision is a tour of P , whose length is at most $(1 + \frac{2\sqrt{2}}{m})L^*$. \square

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 assures 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 Euler 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

It is likely that there are many other applications of our main theorem. We do not yet know a characterization of the general class of problems for which it leads to a PTAS.

It may be interesting to try to tighten our analysis and to try to bring down some of the constants in the polynomial time bounds, through more clever use of dynamic programming.

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.

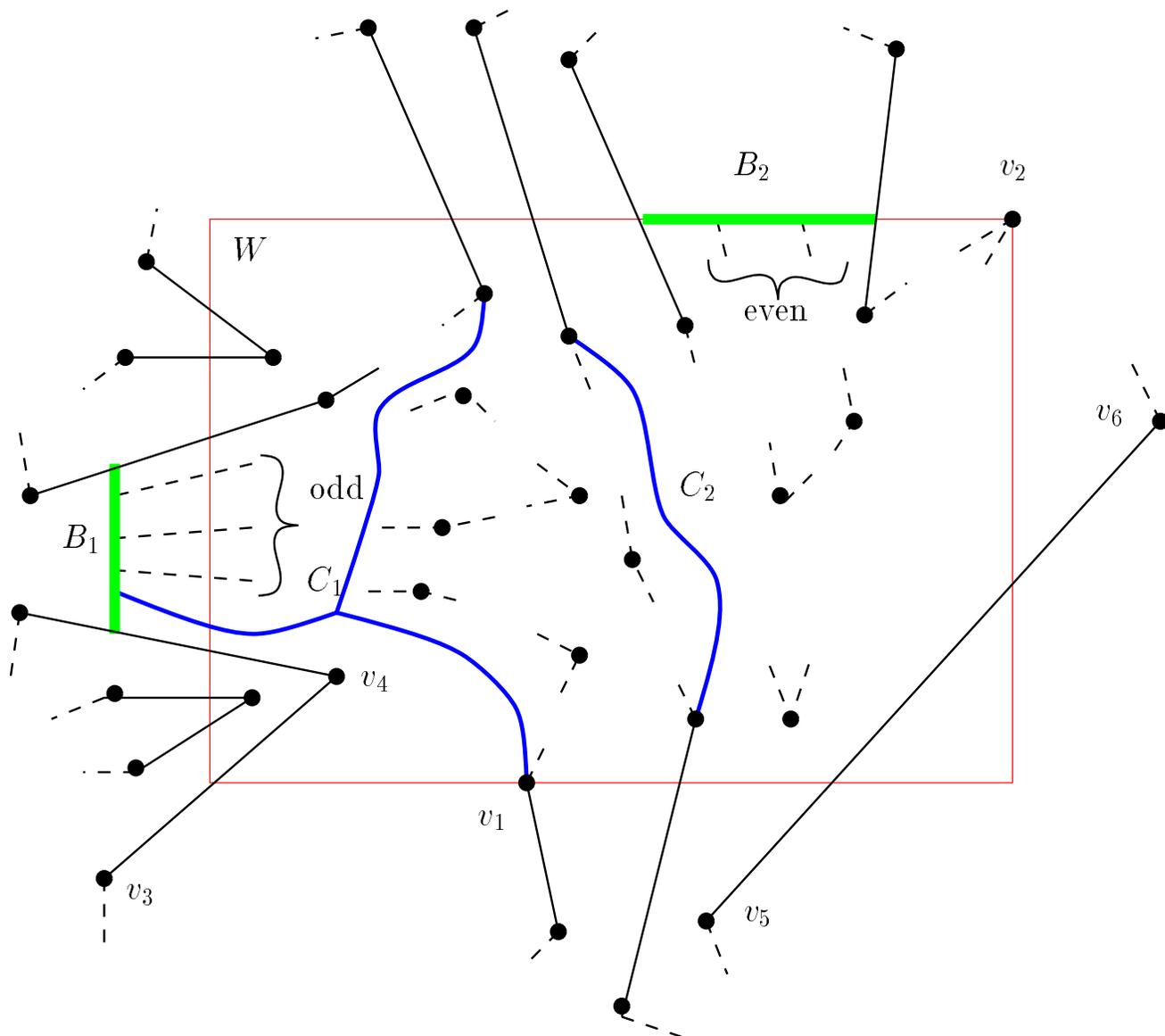


Figure 5: Example subproblem for TSP dynamic program: The window W is determined by the vertices v_1 , v_2 , and the segment v_3v_4 (on which its lower left corner is in contact). There are two bridges, B_1 (which is pinned at some coordinate left of the left boundary of W) and B_2 (pinned at the y -coordinate 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.

Acknowledgements

I thank Rafi Hassin and Günter Rote for 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. Manuscript (to appear, FOCS'96), Princeton University, March 30, 1996.
- [2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Mass., 1990.
- [3] D. Hochbaum, editor. *Approximation Problems for NP-Complete Problems*, PWS Publications, 1996.
- [4] E. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York, 1976.
- [5] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors. *The Traveling Salesman Problem*. Wiley, New York, NY, 1985.
- [6] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple new method for the geometric k -MST problem. In *Proc. Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, Atlanta, GA, January, 1996, pp. 402–408. (See [7] for journal version.)
- [7] J. S. B. Mitchell, A. Blum, P. Chalasani, and S. Vempala. A constant-factor approximation for the geometric k -MST problem in the plane. Manuscript (submitted), University at Stony Brook, 1996. Available at <http://ams.sunysb.edu/~jsbm/jsbm.html>
- [8] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, Englewood Cliffs, NJ, 1982.