Geometric Tours to Visit and View Polygons Subject to Time Lower Bounds

Su Jia* Joseph S. B. Mitchell†

October 16, 2019

Abstract

Vehicle routing problems in geometric domains often seek shortest paths and tours that visit a given set of regions. The TSP seeks to visit points, the TSP with neighborhoods (TSPN) seeks to visit regions, and the watchman route problem (WRP) seeks to visit the visibility polygons of all points in a specified domain, so that the domain is seen by the vehicle. We introduce a new feature to these problems: The impact of requiring a lower bound on the amount of time that a region is visited (in the TSPN) or a point is seen (in the WRP) by a robot moving with bounded speed. Without this constraint, optimal solutions have instantaneous visitations; in robotics and sensing applications, this feature makes the solutions unrealistic and potentially ineffective, e.g., for search problems. Our model with time lower bounds addresses this deficiency and requires the introduction of new ideas to address the resulting optimization problems. In the TSPN with time lower bounds, we are given a set $\mathcal{R}$ of N regions in the plane (given as N polygons having, in total, n vertices), and we wish to find a minimum makespan trajectory for a bounded-speed robot that spends at least time b_R in each region $R \in \mathcal{R}$. Our results include: (1) a PTAS for disjoint fat convex regions in the plane; (2) an $O(\log n)$-approximation for fat regions that obey a certain "thickness" property; (3) $O(\log^2 N, \epsilon)$ and $O(\log^2 n, \epsilon)$ dual-approximation algorithms[1] for general (even disconnected) polygonal regions, with and without preemption, respectively, for any $\varepsilon > 0$, which requires the tour to come very close (within distance $\varepsilon \cdot diam(R)$) to each $R \in \mathcal{R}$; and, (4) approximation algorithms for several versions of the watchman route problem with time lower bounds (WRP-TLB) in certain classes of polygons.

*Carnegie Mellon University, sjia1@andrew.cmu.edu

†Stony Brook University, Joseph.Mitchell@stonybrook.edu

[1] An (α, β) dual approximation guarantees a trajectory with makespan at most α times optimal, while meeting the visitation requirements approximately, getting *close* (within distance $\beta \cdot diam(R_i)$ of R_i) to the regions for the required amounts of time.

1 Introduction

A classic problem in computational geometry is the watchman route problem (WRP): Compute a shortest path/tour for a mobile agent (an observer, or *robot*) within a polygonal domain (polygon with holes) P so that every point of P is seen from some point of the path/tour: the robot must move within P to visit the visibility region $Vis(p)$ of every point $p \in P$. A robot that can travel with maximum speed $v_{\max}$ can complete the task of observing every point within P in the minimum possible time by traveling at speed $v_{\max}$ on an optimal watchman path/tour. However, such a trajectory results in many portions of P being seen instantaneously, as the optimal trajectory "reflects" off of "essential cuts". In real search or data collection tasks it does not suffice to see or visit points instantaneously; instead, there is a requirement that the robot, which moves at a bounded speed, must see or visit a point or a region for at least a certain minimum amount of time, with this amount depending, e.g., on the scanning speed of a searcher, or the data collection speed (bandwidth) of a traveling data mule. Thus, we desire trajectories that meet time lower bound constraints.

More formally, let P be the subset of the plane in which the mobile agent (the *robot*) is allowed to travel; we are particularly interested in the cases in which P is a polygonal domain or in which $P = \mathbb{R}^2$. We are given a set of N polygonal regions $\mathcal{R} = \{R_1, ..., R_N\}$. We let n denote the total number of vertices of the input polygons P and $\mathcal{R}$. Each region $R_i \subseteq P$ has a given *time lower bound* (TLB) b_i; let $b_{\max} = \max_i b_i$. We have a mobile robot that can travel at maximum speed $v_{\max} = 1$; the unit speed assumption is without loss of generality, by an appropriate choice of distance units. The TLB constraint is that the robot must spend at least time b_i in region R_i. Refer to Fig. 1. The *TSP with neighborhoods and time lower bounds* (TSPN-TLB) seeks a path/tour trajectory that minimizes the total time (the *makespan*) necessary to complete the task of visiting each region for at least its TLB, while traveling at speed at most $v_{\max} = 1$; i.e., the cost we seek to minimize is the makespan of the robot's motion schedule. If the regions R_i are the visibility polygons of a discrete set T of target points within a polygonal domain P, then we have the *discrete watchman route problem with time lower bounds* (dWRP-TLB). In the case that $\mathcal{R}$ is the set of visibility polygons of *all* points (a continuum) within P, we obtain the *watchman route problem with time lower bounds* (WRP-TLB).

With respect to the TLB constraints, we distinguish between two cases. In the case that we allow *preemption*, we require only that the total time the robot is in region R_i is $\geq b_i$, not that it be contiguous; the time spent in region R_i may consist of many intervals of time, as the robot may leave and re-enter R_i over and over. In the case *without preemption*, we require that the robot visit each R_i for at least one contiguous time interval of length b_i.

Our Contributions

We introduce and study the impact of putting lower bounds on the visit/view times in the TSPN and watchman route problems. Our results include:

Approximations for the TSPN-TLB Since the TSPN-TLB generalizes the NP-hard problems TSP and TSPN, we seek approximation algorithms. We obtain: (i) A PTAS for disjoint convex fat regions; (ii) An $O(d)$-approximation (with preemption) when the regions overlap in depth $\leq d$; (iii) An $O(\log n)$-approximation (with preemption) for "θ-thick" regions of arbitrary sizes in $poly(n^{O(1/\theta^2)})$ time; (iv) A polylog approximation for general regions, with

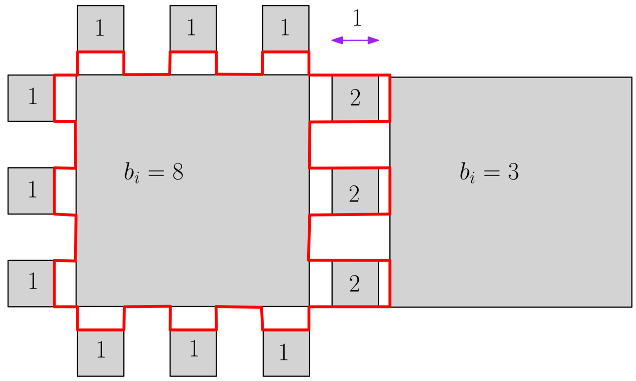

Figure 1. TSPN-TLB example (with preemption): Regions (gray) are labeled with TLB's b_i. An optimal tour (red) has the robot moving at full speed always, using distances traveled to satisfy the TLB's b_i. Note that regions can be revisited. (Gaps between nearby regions are exaggerated for clarity.)

preemption; (v) For any fixed $\epsilon > 0$, and general (even disconnected) regions in the plane, an $O(\log^2 N, \epsilon)$ dual-approximation[2] algorithm, with running time[3] $poly(n, (N \log n)^{O(1/\epsilon)})$ for the non-preemptive version, and an $O(\log^2 n, \epsilon)$ dual-approximation with running time $poly(n^{O(1/\epsilon)})$ for the preemptive version.

To put these results into context, we note that the TSPN that we generalize has best-known approximations that are $O(\log n)$-factor for connected regions in the plane and polylog for disconnected regions; a PTAS is known for "fat", disjoint (or nearly so) regions. Our results for TSPN-TLB come very close to these results for the more special TSPN. The challenges for TSPN-TLB comes from several sources, but most notably arise because we need to assess and take into consideration alternative "longer" routes that spend more time traveling in regions that "need it", in order to satisfy their TLB's. Thus, local optimality of truly optimal tours is much more complex than the simple "reflection principle" for shortest routes visiting regions. Our results build on prior work (of course), significantly extending and devising techniques; we utilize linear programming (for fractional set cover), network optimization (for covering Steiner trees), potential function arguments (previously used for k-MST), recursive optimization (dynamic programming and grid-based methods), and PTAS techniques from geometric optimization.

Approximations for the WRP-TLB and dWRP-TLB: The WRP-TLB, while intimately connected with the TSPN-TLB, presents significant additional challenges. It seems to be a very challenging open problem to solve even the most basic WRP-TLB problem: Find an optimal tour for a simple polygon, with the TLB requirement to see every point for at least time b. We make first progress in this direction by studying some special cases (1.5D terrain, monotone orthogonal polygons, orthogonal polygons), often under the much simpler (though admittedly "unrealistic") vision model of "rectangle vision" (p sees q if their axis-aligned bounding box lies within P). Even these cases capture much of the complexity of WRP-TLB and the tradeoffs between travel distance and view time that we address (e.g., using linear programming in

[2] An (α, β) dual approximation guarantees a trajectory with makespan at most α times optimal, while meeting the visitation requirements approximately, getting *close* (within distance $\beta \cdot diam(R_i)$ of R_i) to the regions for the required amounts of time.

[3] We write $poly(n)$ and $poly(m, n)$ to mean polynomial in n and polynomial in m and n, respectively.

combination with structural results).

Our results include: (i) A general polylog approximation for dWRP-TLB in polygonal domains (under usual visibility), based on our general TSPN-TLB results above; (ii) An $O(1)$-approximation (with preemption) above a 1.5-D terrain; (iii) An exact algorithm for dWRP-TLB in monotone orthogonal polygons (with preemption), under rectangle vision and axis-parallel motion, with targets at a subset of the convex vertices; (iv) An $O(1)$-approximation for WRP-TLB (with or without preemption) for simple orthogonal polygons, under rectangle vision.

Related Work

The *watchman route problem* (WRP) seeks a shortest tour/path for a mobile sensor/robot to be able to see every point of an input region P. If P is a simple polygon, polynomial-time solutions are known [5, 9, 11, 21, 30, 31]; if P is a polygon with holes, the problem is NP-hard and an $O(\log^2 n)$-approximation is known [13, 27]. The WRP has been studied extensively, both in the offline and online setting; see the surveys [24, 29]. In the related *vision points problem* [6], one is given a curve within P and the goal is to place a small number of guards at "vision points" along the curve in order to see all of P. The WRP is a visibility coverage problem closely related to the traveling salesperson problem (TSP) and the *TSP with neighborhoods* (TSPN), in which we desire a shortest path/tour to visit a given set of regions. The TSPN has been studied extensively, and many approximation results are known for geometric instances in the plane [12, 25, 28, 26], in higher dimensions [1, 14], and in doubling metrics [7]; see the survey [29].

The variant of the WRP we study here, in which we lower bound the view time, is related to the problem of using mobile robots with range scanners to build digital indoor models. In the *myopic watchman problem with discrete vision*, we are to compute a min-time path for a robot to scan all of the domain with a limited-range scanner, considering that the robot must travel *and* must pause when taking a detailed image; in [16], this problem is shown to be NP-hard, an $O(1)$-approximation algorithm is given, and a PTAS is given for the special case of orthogonal simple polygons.

The TSPN-TLB is related also to the *Covering Steiner Tree problem* (CST), introduced in [22]. Given an undirected graph $G = (V, E)$ with weighted edges and n vertices, and k subsets of V (called *groups*), each with a *requirement* r_i, one must find a minimum-length tree that has at least r_i vertices from each of the groups. The current best approximation bound, $O(\log n \log N \log k)$, where N is the size of the maximum group, is obtained by Gupta and Srinivasan [19], based on the tree-rounding method for the group Steiner tree problem in [18], and can be derandomized [8]. This problem cannot be approximated within ratio $O(\log^{2-\epsilon} k)$ for any $\epsilon > 0$ (see [20]).

Preliminaries

We seek optimal *trajectories* for a robot that moves at bounded speed (at most $v_{\max} = 1$); thus, our solution is a schedule that stipulates not just the path to execute in the physical space, but the speed profile as well, including places where the robot may stop (pause). We often use γ to denote both the trajectory (path in space-time) and the path (in space), hoping that the context will make it clear when we speak of a trajectory (with speed profile/schedule) versus speak of a geometric path in space. We let OPT denote the total cost (makespan) of an optimal trajectory, γ^*, i.e., the total time needed to execute γ^*.

We let $BB(X)$ denote the *(axis-aligned) bounding box* of a region $X \subseteq \Re^2$, and let $diam(X)$ (resp., $area(P)$) denote the diameter (resp., area) of X. Some of our results exploit the geometric structure of "fatness" (see, e.g., [10]). We say that a region X is *ϕ-fat* (or simply *fat*) if for any disk $\mathcal{D}$, with center within X and with X *not* contained within $\mathcal{D}$ ($X \not\subseteq \mathcal{D}$), the area of $\mathcal{D} \cap X$ is at least $(1/\phi)area(\mathcal{D})$. For convex regions (the case considered in our PTAS result), this notion of fatness is equivalent (up to constant factors of the fatness parameter ϕ) to other simple notions, including bounded aspect ratio (diameter over width), bounded ratio of radius of minimum enclosing disk to radius of largest enclosed disk, etc. We consider ϕ to be a fixed constant throughout.

One approach to meeting time lower bounds is to compute a shortest path/tour that visits the regions $\mathcal{R}$ (i.e., a solution to the TSPN or the WRP), and have the robot stop and pause for time b_i at some point within each region R_i. This *pause point solution* yields a feasible path/tour, but it may have makespan that is arbitrarily large compared to the optimal makespan ($\Omega(n \cdot OPT)$), in general; see Fig. 2 (TSPN-TLB) and Fig. 3 (WRP-TLB). Even for the case that $\mathcal{R}$ is a set of disjoint unit-diameter disks, optimal use of pause points along an optimal TSPN tour is suboptimal by (roughly) a factor 2; refer to Fig. 2(b).

For the TSPN-TLB, we can assume, without loss of generality, that the TLB's b_i are integers, $b_i \in \{1, \ldots, N^2\}$, for purposes of approximation, by the following reasoning. First, by appropriate scaling of the time units, we can assume that $b_{\max} = N^2$. (Length units are scaled so that $v_{max} = 1$.) Second, by adding one pause point per region, and pausing at most 1 time unit at it, a trajectory that satisfies nonintegral TLB's can be made to satisfy the rounded up integral TLB's, with a total increase of makespan of at most N; since $OPT \geq b_{\max} = N^2$, this adjustment causes the makespan to increase by at most OPT/N, not impacting the approximation factor significantly (adding at most $1/N \leq 1$ to the factor).

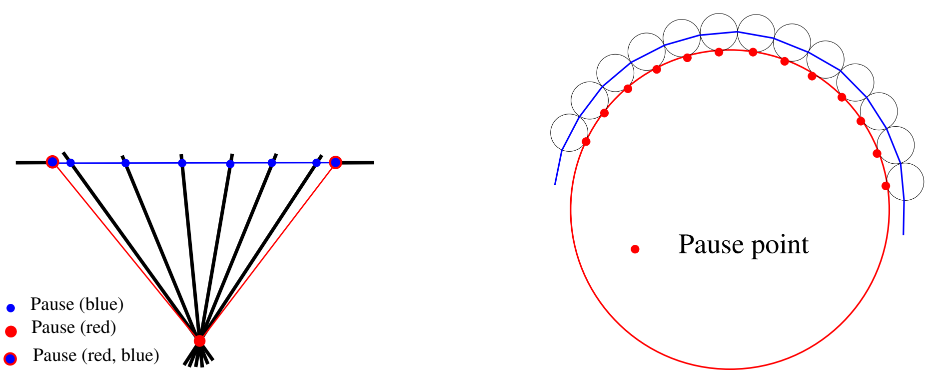

Figure 2. Left (a).: Pause point solution for TSPN-TLB on black regions can cost $\Omega(n \cdot OPT)$: TSPN (blue) requires $\Omega(n)$ pauses, while OPT (red) requires just 3 pauses. **Right (b).**: The pause-point solution (red) to TSPN-TLB is roughly a factor 2 worse than an optimal solution (blue). Here, the time lower bounds $b_i = 2$ are satisfied by slightly lengthening the TSPN (red) tour (which is roughly a circle of radius $r = n/\pi$, perimeter $2n$) to the blue tour (which is roughly a circle of radius $r + 1$, perimeter $2n + 2\pi$). The makespan of the pause point solution is roughly $2n + nb_i = 4n$, while the optimal makespan is about $2n + 2\pi$.

A trajectory γ is called *discrete* if it is specified by a sequence, $(p_1, p_2, \ldots, p_k)$, of *pause points* $p_j \in P$, each with an associated wait time $w(p_j) \geq 0$, and has the following form: The trajectory γ visits the sequence $(p_1, p_2, \ldots, p_k)$ in order, traveling at full speed ($v_{\max} = 1$) from p_j to p_{j+1} along the straight segment $p_j p_{j+1}$, and pausing at each p_j for a time $w(p_j)$. (A pause at p_j of time $w(p_j) = 0$ means that the path simply turns at p_j, while continuing to move at full speed.)

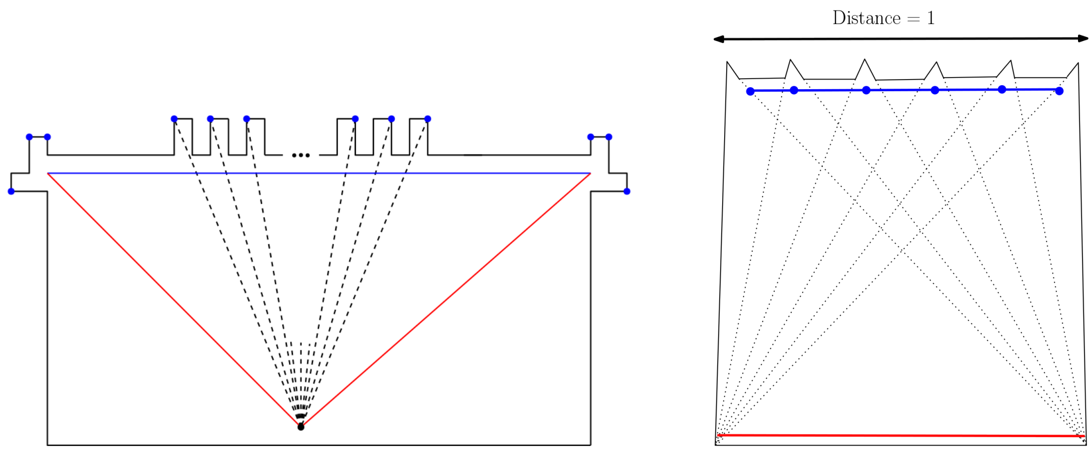

Figure 3. Left (a).: The pause point solution (blue path) for dWRP-TLB, with discrete target points (blue), within a simple orthogonal polygon has cost $\Omega(n \cdot OPT)$, with $\Omega(n)$ pauses, while OPT (red) has one pause. **Right (b).**: The pause point solution (blue) for WRP-TLB in a simple polygon has cost $\Omega(n)$, while OPT (red) has cost $O(1)$.

We say that a discrete trajectory on the sequence $(p_1, \dots, p_k)$ is *pause-point feasible with preemption* for $\mathcal{R}$ if, for each $R_i \in \mathcal{R}$, the sum $\sum_{p_j \in R_i} w(p_j)$ of the waits in R_i is at least b_i. A discrete trajectory is *pause-point feasible without preemption* for $\mathcal{R}$ if, for each $R_i \in \mathcal{R}$, there exists a pause point p_j of the trajectory such that $w(p_j) \geq b_i$.

While we know (Figure 2) that the pause point solution along a TSPN tour can be arbitrarily bad with respect to OPT, the following lemma (whose proof is in the Appendix) shows that there is a discrete trajectory that is pause-point feasible (with preemption) and close to optimal. We let $\mathcal{A}(\mathcal{R})$ denote the *arrangement* of the N polygonal regions $R_i \in \mathcal{R}$.

Lemma 1.1. *[A][4] For the TSPN-TLB, there exists a discrete trajectory that is pause-point feasible with preemption, of makespan at most $2 \cdot OPT$, with at most one pause point of γ per face of $\mathcal{A}(\mathcal{R})$.*

In light of Lemma 1.1, our optimization problem (in the case with preemption), up to a constant factor approximation, can be cast as that of seeking a **weighted** point set (V, w), with a set $V \subset P$ of points and weights $w(v)$ on $v \in V$, in order that the weights satisfy the TLB's (with preemption) and in order to minimize $|TSP(V)| + \sum_{v \in V} w(v)$, i.e. the *connection cost* $|TSP(V)|$ (the length of a TSP path/tour on V), plus the *covering cost* $\sum_{v \in V} w(v)$. The connection cost of the TSP can be approximated, of course, using the minimum spanning tree, $MST(V)$, or other Euclidean TSP approximation.

If only covering cost is important, and we have a set $\{g_1, \dots, g_k\}$ of "guard points" lying within faces of an arrangement of the regions $\mathcal{R}$, with all regions containing at least one guard, then the problem of assigning the minimum total "on" times, t_i, to each g_i so that each region $R_j \in \mathcal{R}$ is seen for at least time b_j (not necessarily contiguously – with preemption) is simply a linear program $(\min_{t_1,\dots,t_k} \sum_i t_i$, subject to $\sum_{j: g_j \in R_i} t_j \geq b_i$, for each i), which is a *fractional set cover* LP. The goal in our problems is to combine considerations of covering costs *and* connection costs.

We often do not distinguish between the problems of seeking a path, a cycle, or a tree with TLB's, since, up to a constant factor, approximation results are the same.

[4]Throughout, we use "[A]" to indicate a result whose proof is deferred to the Appendix.

2 TSPN with Time Lower Bounds

TSPN, as a generalization of TSP, is NP-hard, even in the Euclidean plane; thus, TSPN-TLB is NP-hard as well. In seeking approximation algorithms for TSPN-TLB, a challenge is that travel within a region can be counted towards meeting *two* goals: getting from one place to another, *and* fulfilling the TLB for the region. It may be that a slightly longer route is extremely beneficial in terms of fulfilling the TLB constraints; we saw this to be true for the TSPN-TLB for overlapping regions (Fig. 2), where the choice of route could impact the makespan by a factor $\Omega(n)$, and even for the TSPN-TLB for disjoint unit disks (Fig. 2(b)), where the impact is roughly a factor 2.

For *disjoint* regions $\mathcal{R}$ in the plane, we observe that TSPN-TLB has a straightforward algorithm with approximation factor $(\beta + 1)$, where β is the approximation factor for TSPN: Compute a β-approximate TSPN on $\mathcal{R}$, and then add pauses totaling $\sum_i b_i$. (The factor follows from the facts that $OPT \geq OPT_{TSPN}$ and $OPT \geq \sum_i b_i$ for *disjoint* regions.) Naively, then, one obtains a $(2 + \epsilon)$-approximation for those instances of TSPN on disjoint regions for which a PTAS is known (e.g., for nearly disjoint fat regions [25]). One of our main results is a PTAS for TSPN-TLB for disjoint fat convex regions in the plane.

For the case of *bounded overlap* (no point lies in more than d regions of $\mathcal{R}$), we have two approaches: (1) we give a $(\beta + d)$-approximation algorithm (Theorem 4.7, in the Appendix), based on using a linear programming formulation (for fractional set cover); (2) our PTAS methods generalize to fat convex regions of bounded depth d, with the parameter d going into the exponent of the running time ($n^{O(d)}$).

For (arbitrarily) *overlapping* regions $\mathcal{R}$, the best approximations for TSPN on connected regions in the plane have factor $O(\log n)$ (and polylog factor for disconnected regions). Thus, our goal is to get close to a log-factor for TSPN-TLB. We consider (Section 2.2) the case of regions $\mathcal{R}$ that are "thick" (a concept we introduce, related to fatness), where we obtain an $O(\log n)$-approximation. We then give a polylog approximation for TSPN-TLB on quite general (even disconnected) regions (Section 2.3). In Section 2.4 we improve on the approximation factor for the general case, at the expense of slightly relaxing the TLB constraints, obtaining a *dual-approximation* by means of a potential function approach.

2.1 A PTAS for Disjoint Fat Convex Regions

We now consider the special case of the TSPN-TLB in which the N input regions $\mathcal{R}$ are disjoint, convex and fat, a case for which a PTAS is known for TSPN [25, 28]. As we already mentioned, a $(2 + \epsilon)$-approximation follows from the disjointness of the regions $\mathcal{R}$, together with the PTAS for TSPN on disjoint, fat regions [25, 28]. (See also Theorem 4.7.) In fact, the factor $(2 + \varepsilon)$ is essentially tight for the pause-point strategy, even for a set $\mathcal{R}$ of disjoint unit disks (recall Fig. 2(b)).

To obtain a PTAS we apply dynamic programming to optimize over a certain class of special, recursively structured subproblems. Key to the polynomial running time is that subproblems be "succinct" in their specification. Towards this goal, we introduce a class of solutions based on the "m-guillotine" paradigm, specifically to address the issues imposed by having time lower bounds on the regions R_i. As with the PTAS for TSPN on fat regions [25, 28], we have to deal with regions that cross a cut in the recursive partitioning of rectangular subproblems. For TSPN, this was addressed with the concept of "(m, M)-guillotine subdivisions" (defined below), which added the idea of "bridging" all but $O(\log n)$ of the regions that cross a cut; with only a logarithmic number of unbridged regions straddling the cut, we can afford to specify for each which subproblem, among the two, inherits the responsibility to visit that region.

The added challenge for TSPN-TLB is that an optimal tour may have multiple portions within a region that are there in order to satisfy the TLB's, so it is not enough just to build bridges that *contact* regions that cross a cut; we must explicitly take into account *all* contributions of subpaths in R_i to meeting the TLB of R_i. The key insight is that, exploiting fatness, when a fat convex region R_i is cut and is part of the "M-region span" for which we construct a bridge through regions, we can afford (because of fatness) to add the length of a traversal of one of the two components of the (convex) region's boundary (going either clockwise or counterclockwise around it), charging this additional length off to the length of the bridge (which is charged off to a small fraction, εOPT, of the overall optimal solution makespan). Refer to Fig. 4. (Here we use the fact implied by fatness that the shorter of the two boundary components is within a constant factor (depending on the fatness parameter ϕ) of the length of the intersection of the bridge segment with the region.) Further, the $O(M) = O(\log N)$ unbridged regions R_i that are cut must have additional information associated with each of them encoded in the subproblems to account for how the TLB for R_i is partitioned (approximately) across the two subproblems on either side of the cut.

We review some terminology about the guillotine method. We will speak of an axis-aligned rectangle window W and a set of line segments, E. The intersection, $\ell \cap (E \cap int(W))$, of an axis-parallel line (a cut) ℓ with $E \cap int(W)$ (the restriction of E to the interior of window W) consists of a (possibly empty) set of subsegments (possibly singleton points) of ℓ. Let ξ be the number of endpoints of such subsegments 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, $p_{\xi-m+1}$. Line ℓ is an *m-good cut with respect to W and E* if $\sigma_m(\ell) \subseteq E$. (In particular, if $\xi \leq 2(m-1)$, then ℓ is trivially an m-good cut.) The intersection of the line segment $ab = \ell \cap W$ with the disjoint convex polygonal input regions $\mathcal{R}$ restricted to W consists of a (possibly empty) set of subsegments. Let ξ denote the number of regions $\mathcal{R}$ that segment ab crosses. We define the *M-region-span*, $\Sigma_M(\ell)$, of ℓ analogously to the m-span: If $\xi < 2M-1$, then $\Sigma_M(\ell)$ is defined to be empty; otherwise, if $\xi \geq 2M-1$, then $\Sigma_M(\ell)$ is the line segment $a_M b_M$, along ℓ, with a_M defined to be the Mth entry point where segment ab enters a (crossed) bounding box, when going from a towards b along ab, and b_M defined similarly to be the Mth entry point where segment ab enters a (crossed) bounding box, when going from b towards a along ab. Line ℓ is an *M-good cut with respect to W, E, and $\mathcal{R}$* if $\Sigma_M(\ell) \subseteq E$. We now say that E satisfies the *(m, M)-guillotine property with respect to window W and regions $\mathcal{R}$* if either (1) no edge of E lies (completely) interior to W; or (2) there exists a cut ℓ, that is m-good with respect to W and E and M-good with respect to W, E, and $\mathcal{R}$, that splits W into W_1 and W_2, and, recursively, E satisfies the (m, M)-guillotine property with respect to both W_1 and W_2, and regions $\mathcal{R}$.

The first step in obtaining a PTAS is proving some structural results about optimal solutions and approximately optimal solutions (those with makespan at most $(1+\varepsilon)OPT$). See Section 4.2 in the appendix for details. In particular, we show that it suffices to restrict attention to computing polygonal solutions restricted to lie on a polynomial-size grid $\mathcal{G}$, with edges linking grid points, all travel happening at full speed, and all pausing done at grid points. To do so requires computing upper and lower bound estimates on OPT that are within a *poly*(N) factor of each other.

The overall algorithm is a dynamic program to optimize subproblems that are specified by: (1) a rectangle in the grid; (2) $O(1/\varepsilon)$ edges, with endpoints on the grid, that cross the boundary of the subproblem – these are stipulated edges of the subproblem solution and include possible pause points at the endpoints of the edges, with pause times specified; (3) at most one m-span bridge segment on each of the four sides of the rectangle; (4) $O(\log N)$ regions that cross the boundary of the subproblem, each with additional information specified: the approximate TLB constraint (at the resolution of εb_i) specifying how much time must be spent in the region using edges and pause points within the subproblem; (5) at most one M-region span bridge segment on each of the four sides of the rectangle; (6) an interconnection pattern that must be satisfied within the subproblem, among the $O(1)$ stipulated edges and the m-span bridges.

The number of subproblems is polynomial; in particular, the extra information specified to partition the TLB constraint across subproblems is specified with $(1/\varepsilon)^{O(\log N)}$ information for a subproblem, due to the discretization of the TLB at precision ε. At each step of the dynamic program, we optimize the choice of the cut line ℓ (on the grid), together with all data needed to specify the partition across a cut (the m-span, M-region span, partitions of TLB's, stipulated edges, compatible interconnection patterns).

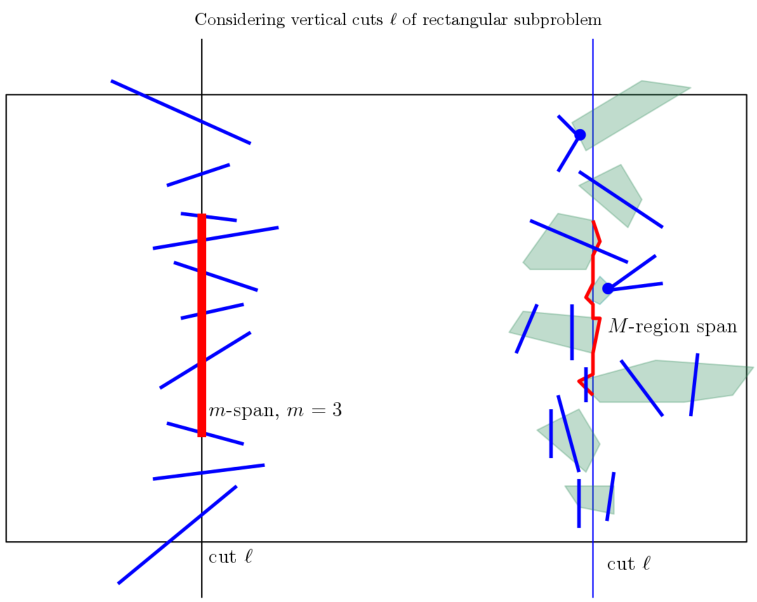

Figure 4. Illustration of a subproblem and the roles of m-spans and M-region spans to summarize information across a cut. For clarity, two vertical cut lines ℓ are shown, each illustrating one type of bridging segment: On left, an m-span, $m = 3$, on edges (only partially drawn), and on right, an M-region span, $M = 3$, on regions (with portions of the tour edges (solid blue) shown). A cut line can have both types of bridging segments, for m-span and for M-region span, depending how many edges/regions are crossed by the defining line ℓ.

Theorem 2.1. *[A] For a set $\mathcal{R}$ of disjoint, fat, convex regions in the plane, the TSPN-TLB with preemption has a PTAS.*

We remark that the PTAS extends immediately to the more general case in which the regions R_i are not necessarily fat, connected, or convex, but there exists a set of disjoint, fat, convex sets Q_i, with $R_i \subseteq Q_i$.

The PTAS for TSPN on fat, disjoint regions generalizes to the case of bounded overlap and weakly disjoint fat regions, at the expense that the depth parameter d goes into the exponent of the running time [7, 25, 28]. We expect that this generalization also applies to TSPN-TLB, giving a PTAS for any fixed depth d of fat, weakly disjoint (even nonconvex) regions, but the details would be very formidable; we leave this to future work.

2.2 Thick Regions

We say that a region R is *θ-thick* $(0 < \theta < 1)$ if for any axis-parallel line ℓ, and for any (maximally) connected component ξ of $R \cap \ell$, we have $\frac{|\xi|}{diam(R)} \geq \theta$, where $diam(R)$ denotes the diameter of region R (considered to be a closed set). The concept of thickness is closely related to the concept of fatness: thickness implies fatness, but the converse is not true.

In this section we consider the case in which the N regions $\mathcal{R}$ are orthogonal, θ-thick simple polygons, and we seek a trajectory path, linking start/end points that are not specified (i.e., they are up to us to compute), of minimum makespan. We obtain an $O(\log n)$-approximation for TSPN-TLB in this case, matching the approximation factor known for TSPN on general connected regions in the plane:

Theorem 2.2. *For N θ-thick orthogonal polygons (possibly overlapping) with n vertices in total, TSPN-TLB has an $O(\log n)$-approximation that runs in $poly(n, (\log N)^{O(\frac{1}{\theta^2})})$ time.*

Consider a regular square grid in the plane, with squares ("pixels") of side length s. An orthogonal polygon that is the (connected, via shared pixel edges) union of k pixels is said to be a *k-polyomino* of *resolution s*.

A point set X is *well clustered* if it can be partitioned into disjoint clusters $C_1, C_2, \ldots, C_k$ such that (a) $diam(C_i) \leq 1$ for each i, and (b) $d(C_i, C_j) \geq 2$ for each $i \neq j$, where $d(A, B) = \min_{a \in A, b \in B} d(a, b)$ is the distance between set A and set B.

Lemma 2.3. *[A] For a well clustered set $X = C_1 \cup \cdots \cup C_k$ of points, there exists a Hamilton path γ visiting the points X such that γ visits the points of each cluster C_i in a contiguous subpath, and the length of γ is at most $O(1)$ times the length of a shortest TSP path on X.*

Lemma 2.4. *[A] For a set $\mathcal{R}$ of k-polyominoes, the TSPN-TLB has an $O(1)$-approximation that runs in $poly(n, (\log N)^{O(k^2)})$ time.*

We now generalize to sets of polyominoes of many different resolutions (pixel sizes) s. We will apply our algorithm to each of the $O(n^3)$ choices of the bounding box (square), W, of the optimal solution, and then select the best choice of W. Let δ be the side length of W (so that $diam(W) = \delta\sqrt{2}$); note that $OPT \geq \delta$. Let $\mathcal{R}_{small}$ be the subset of regions (called *small* regions) with diameter $\leq \frac{\delta}{n^2}$. Let OPT_{small} be the optimal cost (makespan) for TSPN-TLB on the set $\mathcal{R}_{small}$.

Lemma 2.5. *[A] For the set $\mathcal{R}_{small}$ of small k-polyominoes, the TSPN-TLB has a polynomial-time algorithm yielding a solution with cost (makespan) $O(OPT_{small} + \delta)$.*

Lemma 2.6. *[A] Let $\mathcal{R}_W$ be the subset of k-polyominoes in $\mathcal{R}$ that are fully contained in W. Then we can find a solution of cost $O(\log n(OPT + \delta))$ satisfying the TLB of $\mathcal{R}_W$.*

Lemma 2.7. *[A] Let $\mathcal{R}'$ be the subset of k-polyominoes of $\mathcal{R}$ that intersects ∂W. Then we can compute, in polytime, a solution of cost $O(\log n \cdot OPT)$ satisfying all TLBs for $R \in \mathcal{R}'$.*

The following theorem is a consequence of the above lemmas.

Theorem 2.8. *For a set $\mathcal{R}$ of k-polyominoes of various resolutions, for both preemptive and non-preemptive version, the TSPN-TLB has an $O(\log n)$-approximation that runs in $poly(n, (\log N)^{O(k^2)})$ time.*

We are now ready to prove Theorem 2.2.

Proof: (of Theorem 2.2) By Lemma 2.5, we can assume all regions have diameter $\Omega(\delta/n^2)$. Approximate each $R \in \mathcal{R}$ using a polyomino.

Let $\mathcal{R}_i$ be the polygons whose diameter is in $(\delta/2^i, \delta/2^{i-1}]$. We decompose W into a grid of size $\theta/2$. Then the set of cells (pixels) that a region R intersects is a k-polyomino, $\bar{R}$, with $k = O(\frac{1}{\theta^2})$.

Let γ be an $O(1)$-approximation for TSPN-TLB among $\bar{R}$'s. We modify γ into a feasible TSPN-TLB tour for $\mathcal{R}_i$ by inflating the cost by at most a constant factor: create 4 copies of γ, and shift each of them by a distance $\theta\delta/2^i$ in the direction up, down, right, left, respectively, and connect these 4 copies with the original one. By thickness, this new tour will visit all regions in the class $\mathcal{R}_i$. □

2.3 General Regions

We obtain an approximation result for very general regions based on mapping the TSPN-TLB to a related problem, the Covering Steiner Tree (CST) problem: Given a set of N point sets ("groups") in a metric space with n vertices, each group g with a *requirement* k_g, we want to find a tree spanning at least k_g vertices in each group. Konjevod et al. [22] presented an $O(\log n \log(NK_{\max}))$-approximation for CST on tree-metrics, where $K_{\max} = \max_g k_g$. Then, a tree embedding ([3, 15]), with approximation factor $O(\log n \log\log n)$, yields for CST in an arbitrary metric space an overall approximation factor of $O(\log^2 n \log(NK_{\max}) \log\log n)$, in polytime. We use this result to show the following:

Theorem 2.9. *[A] For a set $\mathcal{R}$ of N arbitrary (possibly disconnected) polygonal regions within a (connected) polygonal domain P in the plane, the TSPN-TLB with preemption has a polytime $O(\log^2(nN) \log N \log\log n)$-approximation, where n is the total number of vertices describing $\mathcal{R}$ and P. In particular, this approximation factor carries over to dWRP-TLB with N discrete targets within a (general, nonconvex) polygonal domain of complexity n.*

2.4 Dual-Approximation for General Regions

We now describe a method to improve on the approximation factors for the general case (Theorem 2.9) at the expense of slightly relaxing the TLB constraints. For a region R, let $\bar{R}^\varepsilon$ be the set of points within distance $\varepsilon \cdot diam(R)$ of R. A solution is called an (α, β) *dual-approximation*, if its cost is at most $\alpha \cdot OPT$, and it visits each $\bar{R}_i^\beta$ for at least time b_i.

Define quad-tree $\mathcal{G} = \{G_\ell\}_{\ell=0}^K$ on a side-D square S_0 as follows. Let $G_0 = S_0$, for each $\ell = 1, ..., K-1$ where $K = 2\log_c n$ for some constant c, split each square in G_ℓ into c by c uniform sub-squares (or *children*) to obtain $G_{\ell+1}$. We say a square s is *dyadic* if it is a square in some G_ℓ. The size, $size(s)$, of a dyadic square at level $\ell(s)$ is $D/c^{\ell(s)}$.

Given a point set V, let $z_\ell(V;s)$ be the number of offspring squares of s in G_ℓ that contain at least one point in V. Given a square $s \in G_\ell$, define the *potential of V in s* to be

$$\mathcal{E}(V;s) := \sum_{s' \preceq s} z_\ell(V;s') size(s')$$

where $s' \preceq s$ means that square s' is an offspring (including itself) of s. We call $\mathcal{E}(V) = \mathcal{E}(V;S_0)$, the potential of V in S_0, simply the *potential of V*.

Given integer $\omega \geq 0$, a region R is said to be *ω-well-behaved* if there exists $\ell \leq K - \omega$ and a square $s \in G_\ell$ such that $R \subseteq s$, and R is the union of squares in a subset of squares from the sets $G_{\ell+0}, ..., G_{\ell+\omega}$. In particular, when $\omega = 0$, we simply say that R is *well-behaved*. We say $\mathcal{R}$ is ω-well-behaved if all regions in it are ω-well-behaved.

Lemma 2.10. *[From [17]; A] For any point set V with n points and $diam(V) \geq D/2$, we have (i) $\mathcal{E}(V) = O(\log_c n) \cdot MST(V)$, (ii) $MST(V) = O(c \cdot \mathcal{E}(V))$, for any constant $c \geq 2$.*

For a weighted point set V with weight function w, let $\Phi(V,w) = |MST(V)| + \sum_{v \in V} w(v)$ and $\Lambda(V,w) = \mathcal{E}(V) + \sum_{v \in V} w(v)$. Then, by Lemma 2.10, $\Lambda(V,w) \leq O(\log n) \cdot \Phi(V,w)$, and $\Phi(V,w) \leq O(1) \cdot \Lambda(V,w)$. Define the *profit* $\Pi(\Sigma)$ of a partial[5] solution schedule Σ to be the number of regions satisfied by Σ in the non-preemptive version, and as $\sum_{j=1}^{N} \min\{b_i, t(\Sigma, R_j)\}$ in the preemptive version, where $t(\Sigma, R_j)$ is the amount of time that Σ spends at region R_j. Similarly we can define the profit $\Pi(V,w)$ of a weighted point set.

Lemma 2.11. *[A] Suppose $\mathcal{R}$ is ω-well-behaved. For $L > 0$, let $\Sigma^*(L)$ be a schedule that maximizes profit with cost L and let $\Pi^*(L)$ be its profit.*
(i) For the case without preemption, we can, in poly$(n, (N \log n)^{O(\omega)})$ time and for any $L > 0$, find a partial schedule Σ' with cost $O(c \log_c N \cdot L)$ that earns profit $\Omega(\Pi^*(L))$.
(ii) For the case with preemption, we can, in poly$(n^{O(\omega)})$ time and for any $L > 0$, find a partial schedule Σ' with cost $O(c \log_c n \cdot L)$ that earns profit $\Omega(\Pi^*(L))$.

Lemma 2.12. *[A] If $\mathcal{R}$ is ω-well-behaved, then we can compute*
(i) an $O(\log N)$-approximation without preemption in poly$(n, (N \log_c n)^{c^{2\omega}})$ time, and
(ii) an $O(\log n)$-approximation for TSPN-TLB with preemption in poly$(n^{c^{2\omega}})$ time.

The difficulty of extending this method from ω-well-behaved regions to general regions is that the minimal dyadic square containing a region R can be much larger than $diam(R)$. We overcome this, at a cost of a log-factor in approximation, using a greedy algorithm.

The following lemma draws ideas from the greedy set cover algorithm. A partial schedule with cost L and profit Π is said to have *cost-effectiveness* L/Π. An algorithm is called an *α-greedy-oracle* if it returns a solution with at most α times the optimal cost-effectiveness.

Lemma 2.13. *[A] Let h be an integer constant, and suppose that, for each j, R_j can be written as the union of subregions $R_j^{(i)}$, $i \in \{1, \ldots, h\}$, and let $\mathcal{R}^{(i)} = \{R_j^{(i)}\}$. Suppose we have an α-greedy oracle for each $\mathcal{R}^{(i)}$. Then, we have an $O(\alpha h \log N)$-approximation for TSPN-TLB with or without preemption.*

[5] A solution schedule Σ for TSPN-TLB is said to be *partial* if it does not satisfy all TLB's.

Theorem 2.14. *For any fixed $\epsilon > 0$, the TSPN-TLB has an*
(1) $O(\log^2 N, \epsilon)$ dual-approximation with running time poly$(n, (N \log n)^{O(1/\epsilon)})$ for the non-preemptive version, and
(2) $O(\log^2 n, \epsilon)$ dual-approximation with running time poly$(n^{O(1/\epsilon)})$ with preemption.

Proof: Here, we prove the case without preemption; the case with preemption is similar. We assume all regions have diameter $\Omega(D/n^c)$, since the smaller regions can be treated as we did in proving Theorem 2.8. Suppose $diam(R) = D$ and $\tilde{D}$ is the minimum power of 2 above D. Note that R intersects at most 4 dyadic squares $Q_1, ...Q_4$ with side $\tilde{D}$. Denote $R^{(i)} = Q_i \cap R$. Suppose Q_i lies on G_ℓ, then replace it with the union of the squares in $G_{\ell+\omega}$ that intersect $R^{(i)}$. This new region is, by definition of G_ℓ, contained in $\bar{R}^{2^{-\omega}}$. By Lemmas 2.11, 2.13, there is a greedy algorithm that gives the desired factor. □

3 Watchman Route Problem (WRP) with Time Lower Bounds

We consider now the Watchman Route Problem with time lower bounds (WRP-TLB) within a given polyhedral domain P having n vertices. Much of our attention is focused on the discrete version, dWRP-TLB, in which we are given a set $T = \{t_1, \ldots, t_N\}$ of N *target points* $t_i \in P$ that are to be seen (i.e., the robot must visit the visibility region of each t_i), each with a time lower bound b_i.

Since dWRP-TLB generalizes the dWRP (which has $b_i = 0$ for all i), we know that in polygons with holes it is NP-hard, and, in fact, there is no approximation better than a factor $O(\log n)$ unless P=NP [27]. In a simple polygon P, WRP (and dWRP) is solvable in polytime [5, 9, 11, 21, 30, 31]; however, we observe that the WRP-TLB is NP-hard and APX-hard in the version without preemption:

Theorem 3.1. *[A] Without preemption, WRP-TLB and dWRP-TLB are NP-hard (and APX-hard) in a simple polygon P.*

In the version of WRP-TLB *with preemption* in a simple P, hardness is not apparent (and we leave as an open problem), as the corresponding fractional set cover (FSC) problem is a linear program: For each cell σ in the arrangement of the visibility polygons of the targets t_i, let $S(\sigma) = \{t \in T : \sigma \text{ sees } t\}$, and let $\mathcal{S}$ be the collection of sets $S(\sigma)$, for all σ. Then, the FSC problem for the set system $(T, \mathcal{S})$ determines the amounts of time, x_σ to spend in each cell σ in order to satisfy the TLBs using the smallest amount of pausing (and thus forms a lower bound on OPT).

dWRP-TLB in Polygonal Domains. For the NP-hard dWRP-TLB with preemption in a polygonal domain, viewing the problem as an instance of TSPN-TLB on a set $\mathcal{R}$ of visibility regions within P, Theorem 2.9 yields a a general result, a polylog approximation algorithm under ordinary visibility in a polygonal domain of complexity n.

WRP-TLB in Skinny Polygonal Domains. In the Appendix we show a result (Theorem 4.7) about utilizing a β-approximation for TSPN to yield a $(\beta + d)$-approximation for TSPN-TLB on a set of regions having depth (ply) d. A consequence of this (using [27]), is the following: For a skinny polygonal domain P that is the connected union of $O(n)$ line segments of depth $O(1)$, there is an $O(\log^2 n)$-approximation for WRP-TLB.

WRP-TLB Above a 1.5-D Terrain. Assume that P is the region in the plane that lies above an x-monotone polygonal chain (a $1.5D$ terrain). Further, assume we seek a solution to the path version of the WRP-TLB, from a given source point v_s to a given destination v_t.

Theorem 3.2. *[A] For the dWRP-TLB (with preemption) in a region P above a $1.5D$ terrain, there is a polynomial-time exact algorithm for the L_1 metric, and thus a $\sqrt{2}$-approximation algorithm for the Euclidean metric.*

dWRP-TLB in Monotone Orthogonal Polygons. In Section 4.17 of the appendix, we prove, through a series of lemmas:

Theorem 3.3. *[A] For an x-monotone orthogonal polygon P and a set T of targets (with TLB's) at convex vertices of P, an optimal orthogonal (axis-parallel) solution trajectory path to the WRP-TLB can be computed in polynomial time, for both the preemptive and non-preemptive versions. (This implies a $\sqrt{2}$-approximation to the same problem, without the axis-parallel path assumption.)*

3.1 WRP-TLB in a Simple Orthogonal Polygon

We assume now that P is a simple orthogonal polygon (not necessarily monotone). We continue to focus on the case of rectangle vision and we restrict ourselves to orthogonal solution paths/tours (which suffice for approximating (factor $\sqrt{2}$) general polygonal paths/tours).

The *window partition* (WP(s,P)) of an orthogonal polygon P with respect to (source) edge s of P is a partition of P into pieces, using (axis-parallel chords) *windows* according to orthogonal link distance (links must be axis-parallel), with all points in a piece having the same link distance to s. Each piece is a *histogram* with respect to a base segment that is a window of the parent histogram. The *extended window segment* of a window in the WP(s,P) is the supersegment of the window that extends it until it meets the base of the histogram that the window bounds. Let F be the set of extended window segments of WP(s,P). Note that F is connected, and its axis-parallel segments form a tree within P; we call F the *window partition tree*, WPT(s,P), of P with respect to (source) edge s. We define the *truncated window partition tree* (TWPT(s,P)), F', as follows (refer to Fig. 6). Let H_0 be a histogram of the WP(s,P), with base edge e_0, and let $e_1,\ldots,e_t$ be the other window edges of H_0. Further, let H_i be the histogram with base e_i, and let s_i be the point on e_i that is closest to e_0. For each i, let $p_i \in e_i$ be the closest point on e_i to e_0 such that the segment $s_ip_i \subseteq e_i$ can see all of H_i (just as e_i can); we speak of the window e_i being truncated at p_i (to become the segment s_ip_i), and similarly speak of truncating the extended window segments at p_i. Then, the TWPT(s,P), $F' \subseteq F$, is defined to be the union of the truncated extended window segments.

Note that any $g \in P$ can see at most 4 window edges; call them $e_1,\ldots,e_4$. Let $\bar{g}_i$ be the closest point on e_i to g. If $\bar{g}_i$ lies on an edge of TWPT(s,P), we let $g_i = \bar{g}_i$; otherwise, the window e_i was truncated at a point p_i such that $\bar{g}_i$ does not lie on the truncated window, in which case we let $g_i = p_i$, the point of the truncated edge that is closest to g. We refer to the points g_i as *shadow guards* of g in P.

Lemma 3.4. *[A] For any edge s, there is a connected graph H containing TWPT(s,P) whose cost is bounded by a constant times the length of a shortest orthogonal watchman tree for P (under rectangle vision).*

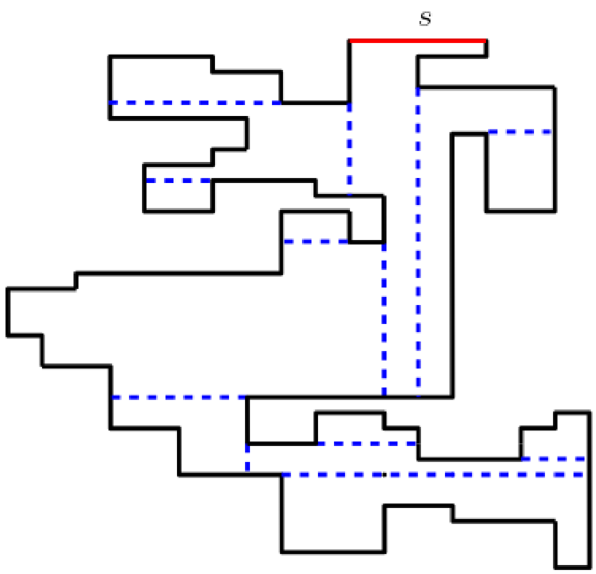

Figure 5. Window partition tree with root edge s.

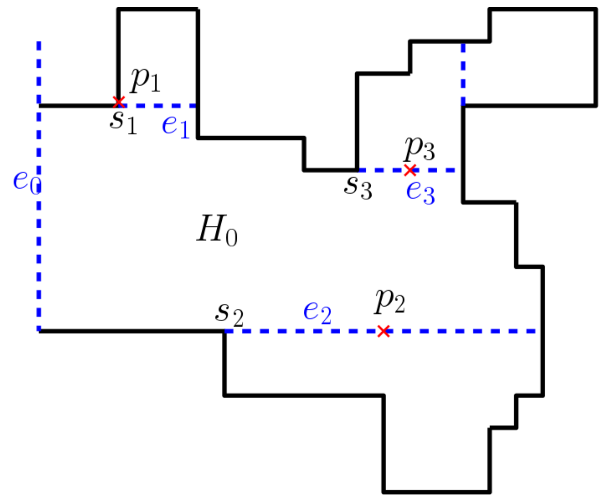

Figure 6. Truncation of windows e_1, e_2, e_3 at points p_1, p_2, p_3.

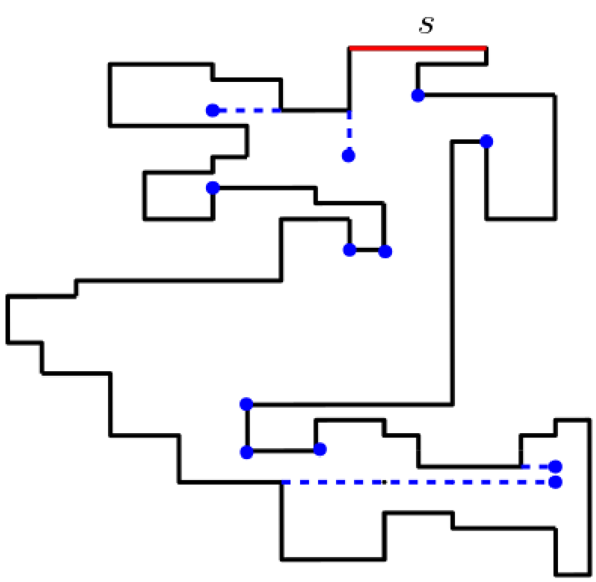

Figure 7. Truncated window partition tree

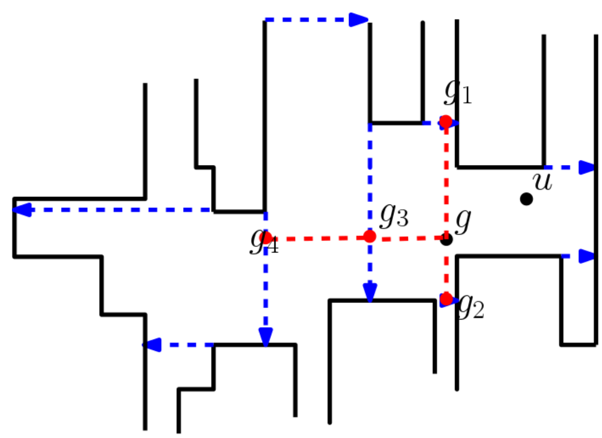

Figure 8. Shadow guards g_1, g_2, g_3, g_4.

Lemma 3.5. *[A] For each $g, u \in P$, if g sees u, then some shadow guard of g also sees u.*

Lemma 3.6. *[A] Let x be any feasible solution to the fractional set cover (FSC) formulation. Then there is another feasible solution $\hat{x}$ s.t. $supp(\hat{x}) \subseteq F'$ and $cost(\hat{x}) \leq O(cost(x))$.*

Combining the above, we obtain:

Theorem 3.7. *There is a polynomial-time $O(1)$-approximation algorithm for the WRP-TLB (with preemption) in simple orthogonal polygons under rectangle vision (L_1 or L_2 metric).*

4 Appendix

4.1 Proof of Lemma 1.1

Proof: Let γ be an arbitrary optimal trajectory; we argue that γ can be transformed into a discrete trajectory, γ', that is pause-point feasible with preemption, at most doubling its makespan.

Consider each polygonal cell σ (possibly with holes) in the overlay arrangement, $\mathcal{A}(\mathcal{R})$, of the regions R_i (each of which is polygonal, possibly with holes). If the intersection of γ with σ is nonempty, it consists of a set of subtrajectories $\gamma_1, \gamma_2, \ldots$. We place a single pause point p_σ on one of these subtrajectories (say, γ_1), and we set the wait, $w(p_\sigma)$, at p_σ to be equal to the total time that γ spends in σ. Then, we define γ' to be the discrete trajectory on the sequence of points p_σ, with the associated waits $w(p_\sigma)$, visiting the points p_σ in the same order that γ does, but now going directly along a straight segment from point to point, at full speed. The makespan of γ' consists of two portions: the waiting cost at the pause points (which totals to exactly the makespan of γ), plus the travel cost, which equals the time spent moving between pause points (which totals at most the makespan of γ, since the trajectory γ' moves at full speed along straight segments linking a finite subset of points along γ). Thus, the makespan of γ' is at most twice the makespan of γ, and γ' is discrete and pause-point feasible with preemption. □

(Remark: The case without preemption can be handled similarly, but the discrete trajectory can be shown (using an interval tree argument) to have makespan at most $O(OPT \log N)$, and this is best possible, as an example with regions that are dyadic intervals shows. Since we do not go into the non-preemptive case further here, we omit details.)

4.2 Structural Properties for TSPN-TLB: Discretization and Localization for PTAS for Disjoint, Convex, Fat Regions

A simple structural result for the PTAS for TSPN among fat regions is that OPT is a simple polygon having at most N vertices (one per region). This already is *not* true for TSPN-TLB, and is related to the source of the challenges in obtaining a PTAS for TSPN-TLB. In particular, while OPT can be assumed to be a simple polygonal trajectory, it may need to have more than N vertices to be even close to optimal: the example in Fig. 1 shows that a tour of N fat convex regions may need to have about $4N$ vertices in order to be very close to optimal. In order to obtain a PTAS, we need to polynomially bound the number of candidate vertices of an optimal route.

For a set $\mathcal{R}$ of disjoint convex regions in the plane, we say that a TSPN-TLB tour γ is $\mathcal{R}$-*conforming* if it is a trajectory with the following properties: (a) it follows a polygonal route $\gamma = (v_1, v_2, \ldots, v_k)$; (b) each v_j is on the boundary of one of the regions $R_i \in \mathcal{R}$; (c) it travels at full speed $v_{\max} = 1$ between consecutive vertices; (d) it has pause points among its vertices; and (e) for each i, the sum of the pauses at points (vertices) on the boundary of R_i, plus the (Euclidean) lengths of segments of γ within R_i, is at least the TLB b_i.

Lemma 4.1. *There is an $\mathcal{R}$-conforming optimal solution for TSPN-TLB on a set of N disjoint convex regions $\mathcal{R}$ in the plane.*

Proof: Let γ^* be the route of an optimal TSPN-TLB tour with makespan OPT; the robot may travel at various speeds (at most $v_{\max} = 1$) along γ^*, and may pause at various points. Consider a walk along γ^*, starting at any point $s \in \gamma^*$ that lies inside one of the regions $\mathcal{R}$. During this loop around γ^*, we enter and exit regions $R_i \in \mathcal{R}$. Let $(v_1, v_2, \ldots, v_k)$ be the points where γ^* crosses boundaries of regions $R_i \in \mathcal{R}$. By the triangle inequality, replacing the route γ^* with the polygonal route with vertices $(v_1, \ldots, v_k)$ can only make the route shorter. Consider the trajectory on the route $(v_1, \ldots, v_k)$, traveling at full speed $v_{\max} = 1$ between consecutive vertices, and, for each edge $v_j v_{j+1} \subset R_i$ within a region R_i, pausing at point $v_j \in R_i$ for an amount of time $\tau(v_j, v_{j+1}) - |v_j v_{j+1}|$, where $\tau(v_j, v_{j+1})$ is the length of time the optimal tour spends between its arrival at v_j and its departure from v_{j+1}. (We know that $\tau(v_j, v_{j+1}) - |v_j v_{j+1}| \geq 0$ by the triangle inequality.) Then, this new trajectory, which has the same makespan as the optimal tour, is $\mathcal{R}$-conforming and continues to obey the TLB constraints. □

Further, we bound the complexity of an optimal tour:

Lemma 4.2. *There exists an optimal tour for the TSPN-TLB on a set of N disjoint, fat convex polygonal regions $\mathcal{R}$ in the plane that is a simple polygon (without self-intersections) with $k = O(n^2)$ vertices.*

Proof: Let γ^* be an optimal tour for TSPN-TLB on $\mathcal{R}$. We saw already in the proof of Lemma 4.1 that we can assume that γ^* is polygonal, with vertices on the boundaries of the (fat, convex) polygons $\mathcal{R}$. A simple, standard uncrossing argument (as for Euclidean TSP) shows that γ^* is simple. To bound the complexity of γ^* we argue as follows. Consider region R_i. On its boundary lie one or more vertices of γ^*. If only one vertex, then this vertex is a "reflection point", where γ^* touches R_i (and pauses for time $w(R_i)$). If there are two or more vertices of γ^* on the boundary of R_i, then they must all be endpoints of chords of γ^* crossing R_i, since a reflection point on the boundary of R_i could be shortcut and removed (possibly after reassigning the waiting at the reflection point to another portion of $\gamma^* \cap R_i$), until there is either a single reflection point or a set of chords of γ^* crossing R_i. Now, consider one class of chords – those with endpoints on two particular edges, e and e', of R_i. (Overall, there are only $O(n^2)$ classes of chords for $\mathcal{R}$.) If the number of chords in this class were "huge" (bigger than a constant, depending on the fatness parameter, ϕ, of regions $\mathcal{R}$), then we could improve γ^* as follows. Denote the (noncrossing) chords in this class (with endpoints on e, e' of R_i) by $c_1, \ldots, c_K$. By fatness, we know that either the portion of the boundary of R_i that joins the endpoints of c_1 and includes the endpoints of $c_2, \ldots, c_K$ has length $O(|c_1|)$ or the portion of the boundary of R_i that joins the endpoints of c_K and includes the endpoints of $c_1, \ldots, c_{K-1}$ has length $O(|c_K|)$. Assume, without loss of generality, it is the first case. Then, we can remove the chords $c_2, \ldots, c_K$ and instead include the portion of the boundary joining the endpoints of c_1, and passing through the endpoints of $c_2, \ldots, c_K$, doubling every other portion along the boundary. This can be done so that the degrees of the vertices (endpoints of $c_2, \ldots, c_K$) have even degree (2 or 4). Thus, the resulting modification of γ^* has resulted in a connected Eulerian graph, which results in a tour, and, with shortcutting, an alternative simple tour on the same set of vertices, all but a constant number now being reflection points (which can be removed, as argued above). The modification resulted in shorter overall length and maintained feasibility (the TLB for R_i can be achieved by pausing appropriately at any of the vertices on the boundary of R_i). Thus, overall, there are at most $O(n^2)$ vertices in the final γ^*, after applying this modification to all regions. □

Let S_0 be a minimum-diameter axis-aligned rectangle that intersects or contains all input regions $R_i \in \mathcal{R}$; let $D = diam(S_0)$. Note that S_0 is easily computed in polynomial time by standard critical placement arguments; it is pinned by four contacts between a vertex/edge of the square and a vertex/edge of a region R_i.

Lemma 4.3. *For TSPN-TLB, we have that $OPT \geq b_{\max}$ and $2D \leq OPT \leq c_0 D\sqrt{N} + \sum_i b_i$, for a constant c_0. If the regions $\mathcal{R}$ are disjoint, we have $OPT \geq \sum_i b_i$.*

Proof: The first lower bound is immediate from the TLB of the region having $b_i = b_{\max}$. In the case of disjoint regions, the TLBs collectively imply $OPT \geq \sum_i b_i$. The second lower bound follows from the fact that the shortest tour visiting all four sides of $W_0 = BB(\gamma^*)$ has length at least twice the diameter of W_0 (Fact 1 of [2]); since γ^* visits all four sides of W_0, and S_0 has diameter at most that of W_0, this implies that the length of γ^* is $\geq 2D$, and thus that $OPT \geq 2D$. The upper bound follows from the fact that a feasible solution to the TSPN-TLB is to traverse a TSP tour on N points within a region of diameter D (which, by standard properties of the Euclidean TSP is of length $O(D\sqrt{N})$), with pause points in each region totalling additional time $\sum_i b_i$. □

The above lemma allows us to compute (in polytime) an estimate of OPT within a factor $O(\sqrt{N})$ for the case of disjoint regions, since the upper bound $(c_0 D\sqrt{N} + \sum_i b_i)$ is at most $O(\sqrt{N})$ times the lower bound $(\max\{\sum_i b_i, 2D\})$. For non-disjoint regions, $OPT \geq \max\{2D, b_{\max}\} \geq (2D + b_{\max})/2 \geq (D + b_{\max})/2$ and $OPT \leq c_0 D\sqrt{N} + \sum_i b_i \leq O(\sqrt{N})(D + b_{\max})$, implying a ratio (upper bound divided by lower bound) of at most $O(\sqrt{N})$.

Let $\delta = \varepsilon D/n^2$, and consider a fixed $\varepsilon > 0$. Let $\mathcal{G}$ denote the regular grid (lattice) of points $(i\delta, j\delta)$, for integers i and j. Let Γ_i be the subset of grid points $\mathcal{G}$ at distance at most $\delta/\sqrt{2}$ from region R_i; then, $\Gamma_i \neq \emptyset$ (it may be that $\Gamma_i = \Gamma_j$ for $i \neq j$).

For given regions $\mathcal{R}$ and time lower bounds b_i, we say that a trajectory is *grid-conforming* if it is a trajectory with the following properties: (a) it follows a polygonal route whose vertices are among the grid points $\bigcup_i \Gamma_i$; (b) it travels at full speed $v_{\max} = 1$ between consecutive vertices; (c) it has pause points among its vertices (which are grid points); and (d) for each i, the sum of the pauses at points Γ_i, plus the (Euclidean) lengths of the trajectory edges linking two grid points of Γ_i, is at least the TLB b_i.

Lemma 4.4. *Any $\mathcal{R}$-conforming TSPN-TLB tour $\gamma = (v_1, v_2, \ldots, v_k)$, of makespan τ and $k \leq n^2$ vertices within regions $\mathcal{R}$, can be modified to be a grid-confirming polygonal tour $\gamma_{\mathcal{G}}$, of makespan at most $(1+\varepsilon)\tau$ and having at most k vertices among the grid points $\bigcup_i \Gamma_i$. Similarly, any grid-conforming tour $\gamma_{\mathcal{G}}$, of makespan $\tau_{\mathcal{G}}$ and k vertices among $\bigcup_i \Gamma_i$, for $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_N\}$ and TLBs b_i, can be modified to be a TSPN-TLB $\mathcal{R}$-conforming tour γ, of makespan at most $(1+\varepsilon)\tau_{\mathcal{G}}$ and k vertices, for $\mathcal{R} = \{R_1, R_2, \ldots, R_N\}$.*

Proof: For each of the k vertices of γ we can simply add to γ a detour that goes from the vertex to a grid point and back to the vertex. Since no point of $\bigcup_i R_i$ is further from a grid point than $\delta/\sqrt{2}$, we get that the total detour length is bounded above by $k \cdot 2\delta/\sqrt{2} = \varepsilon D\sqrt{2}(k/n^2) \leq \varepsilon OPT \leq \varepsilon\tau$, utilizing the bound we proved above on k. Thus, shortcutting this new tour to be a polygonal route going directly, at full speed, between consecutive grid points, the increase in the makespan is at most $\varepsilon\tau$. The second claim is proved similarly. □

The next lemma provides a means of "localizing" an optimal solution, so that our search for approximately optimal tours can be restricted to a polynomial-size grid. Let $W_0 = BB(\gamma^*)$ denote the axis-aligned bounding box of γ^*, the polygonal route of an optimal $\mathcal{R}$-conforming tour. We can assume that W_0 contains at least one of the grid points $c_0 \in \bigcup_i \Gamma_i$. (Otherwise, the problem can be solved directly.) Our algorithm enumerates over all choices of c_0.

Lemma 4.5. *There exists an optimal TSPN-TLB tour whose associated route γ^* lies within the ball, $B(c_0, D_0)$, of radius $D_0 = O(D\sqrt{N} + \sum_i b_i)$ centered at c_0, a grid point within $W_0 = BB(\gamma^*)$. Further, there exists a tour $\gamma^*_{\mathcal{G}}$ of the grid sets Γ_i, of makespan at most $(1+\varepsilon)OPT$, that has its vertices at grid points $\mathcal{G}$ that lie within an M-by-M array of grid points centered at c_0, where $M = O(N^{2.5}/\varepsilon)$.*

Proof: Since $W_0 = BB(T^*)$ is a rectangle of diameter at most OPT (in fact, at most $OPT/2$), and c_0 and γ^* both lie within W_0, we know that all of γ^* lies within a ball $B(c_0, D_0)$, of radius $D_0 = O(D\sqrt{N} + \sum_i b_i)$ centered at c_0.

Since grid points of $\mathcal{G}$ are at spacing $\delta = \varepsilon D/N^2$, we see that a grid of size $M = O((\sqrt{N}(D + b_{\max})/\delta) = O(N^{2.5}/\varepsilon)$ suffices, and Lemma 4.4 shows that γ^* can be rounded to $\mathcal{G}$. □

4.3 Proof of Theorem 2.1

Consider an optimal tour, γ^*, of makespan OPT. By our localization and discretization results, we know that there is a grid-rounded tour $\gamma^*_{\mathcal{G}}$ of comparable makespan whose vertices lie on a certain polynomial-size grid. We will consider separately each choice of W_0, the hypothesized bounding box of $\gamma^*_{\mathcal{G}}$.

For a given choice of W_0, we apply the dynamic programming outlined above to compute a minimum-makespan solution that has the properties:

(a) it is (m, M)-guillotine with respect to window W_0 and the "large" regions $\mathcal{R}_{W_0}$ (of diameter greater than $diam(W_0)$, which are not contained within W_0, but intersect its boundary; see [25, 28]), with doubled m-span bridge segments;

(b) it satisfies certain connectivity requirements interconnecting subsets of the specified (unbridged) segments and the m-spans; and,

(c) it visits all of the regions $\mathcal{R}$ while obeying all TLB's.

As shown in [25, 28], the network that is output by the dynamic program can be made to be connected (without increasing its length and the resulting makespan by more than $\varepsilon \cdot OPT$). Further, with the doubling of the bridges and the M-region spans that detour around the fat regions crossing a bridge, it can be made to contain an Eulerian subgraph, which will enable us to extract a feasible tour meeting all TLB's.

The main guillotine structure theorem of [25, 28] shows that there always exists a choice of cut ℓ that allows us to charge off the lengths of m-spans of crossing edges (in fact, the m-spans of the "grid-encasements" of edges) and the M-region spans of crossing regions. The additional structure needed for maintaining TLB's consists of

(i) the (doubled) path along M-region spans, going around each crossed region along the shorter of the two traversals of its boundary; and

(ii) the approximate partitioning of the TLB constraint for each of the $O(M) = O(\log n)$ specified regions (not bridged by the M-region span).

This allows us to convert an optimal solution, γ^*, to be in the class of solutions over which our dynamic programming algorithm optimizes, at a small increase in its makespan. In particular, the length of the path corresponding to a trajectory γ goes up by only a factor $(1+\varepsilon)$ when it is made to be (m, M)-guillotine with respect to the fat regions $\mathcal{R}$, where $m = O(1/\varepsilon)$ and $M = O((\log n)/\varepsilon)$. Further, the total increase in makespan caused by "being sloppy" with the partitioning of the TLB's for specified crossing regions is at most εb_i for region R_i, and $\sum_i \varepsilon b_i \leq \varepsilon \cdot OPT$ overall. Thus, the makespan of the solution produced by the dynamic program is at most $(1 + O(\varepsilon))OPT$. Finally, through the use of bridge doubling, and the doubling of the m-region span paths around the bridged fat regions, the solution produced by the dynamic program is guaranteed to contain an Eulerian subgraph, from which we can extract a closed walk, along with a speed schedule (which is always full speed along edges, and zero speed at pause points), yielding the desired trajectory.

The running time is polynomial because, as discussed above, the number of subproblems is polynomial, as well as the number of choices of cuts and associated information about the specified edges and regions crossing the cut, and the m-span and M-region span along the cut. In particular, the running time of the algorithm is $(1/\varepsilon)^{O(M)} n^{O(1/\varepsilon)}$, which is polynomial in n, for any fixed ε, since $M = \lceil (1/\varepsilon) \log(n/\varepsilon) \rceil$.

4.4 Shallow Arrangements of Regions

Consider the case in which the set $\mathcal{R}$ of regions has depth at most d, meaning that no point in P lies in more than d of the regions $R_i \in \mathcal{R}$.

Consider a set system $(U, \mathcal{S})$, having a universe U of elements and a collection $\mathcal{S}$ of subsets of U, and assume that each element u is associated with a number b_u (the *requirement* for u). The *Fractional Set Cover* (FSC) problem is the following linear program (the relaxation of the set cover integer program), which seeks values x_S, associated with each set $S \in \mathcal{S}$, in order to minimize the sum:

$$\begin{aligned} \min \ & \sum_{S \in \mathcal{S}} x_S \\ \text{s.t.} \ & \sum_{S: u \in S} x_S \geq b_u, \quad \forall u \in U, \ (i) \\ & x_S \geq 0, \qquad\quad \forall S \in \mathcal{S}. \ (ii) \end{aligned}$$

For $\mathcal{S}' \subseteq \mathcal{S}$, let $FSC(\mathcal{S}')$ be the problem with the additional constraint that the support of x has to be contained in $\mathcal{S}'$. We let $|FSC|$ and $|FSC(\mathcal{S}')|$ denote the optimal values of the corresponding optimization problems.

Lemma 4.6. *Suppose $(U, \mathcal{S})$ is a set system for which each set $S \in \mathcal{S}$ has cardinality at most d (i.e., $|S| \le d$). Then for any $\mathcal{S}' \subseteq \mathcal{S}$ with $U \subseteq \bigcup_{S \in \mathcal{S}'} S$, it holds that $|FSC(\mathcal{S}')| \le d \cdot |FSC(\mathcal{S})|$.*

Proof: Construct a feasible solution $\hat{x}$ as follows. Initially let $\hat{x}_S = 0$ for all S. Scan elements in U in an arbitrary order, and, for each $u \in U$, pick an arbitrary S containing it, and increase $\hat{x}_S$ by b_u. In the end,

$$\sum_{S \in \mathcal{S}} \hat{x}_S = \sum_{u \in U} b_u.$$

Let x be any feasible solution to $FSC(\mathcal{S})$. Summing over all constraints for $u \in U$, we have

$$\sum_{u \in U} \sum_{S : u \in S} x_S \ge \sum_{u \in U} b_u.$$

Since $|S| \le d$ for all S, each x_S appears $\le d$ times in the left-hand side above, so

$$\sum_{u \in U} \sum_{S : u \in S} x_S \le d \cdot \sum_{S \in \mathcal{S}} x_S.$$

Combining the above, we obtain

$$d \cdot \sum_{S \in \mathcal{S}} x_S \ge \sum_{u \in U} b_u = \sum_{S \in \mathcal{S}} \hat{x}_S.$$

The proof is completed by plugging in the optimal solution, x^*, to $FSC(\mathcal{S})$. □

Theorem 4.7. *Suppose the maximum depth of the regions $\mathcal{R}$ is d (i.e., no point lies in more than d regions) and that the TSPN on $\mathcal{R}$ within P has an β-approximation algorithm. Then there is an $(\beta + d)$-approximation algorithm for the TSPN-TLB problem.*

Proof: Let γ be an β-approximation for TSPN on $\mathcal{R}$. Let $\mathcal{H}$ be the subset of cells of $\mathcal{A}$ that intersect γ, and let $\mathcal{S}' = \{S(\sigma)\}_{\sigma \in \mathcal{H}}$, where $S(\sigma) \subseteq \mathcal{R}$ is the subset of regions containing σ.

Compute (via linear programming) an optimal solution, x^*, for $FSC(\mathcal{S}')$. Then, for each $\sigma \in \mathcal{H}$, add to γ a pause-point at an arbitrary point in $\gamma \cap \sigma$, pausing for time x^*_σ; the resulting trajectory γ now has makespan $|\gamma| + |FSC(\mathcal{S}')|$. Depth at most d implies that $|S(\sigma)| \le d$, for all σ. Then, by Lemma 4.6, we have $|FSC(\mathcal{S}')| \le d \cdot |FSC(\mathcal{S})|$. Using the facts that $|FSC(\mathcal{S})| \le OPT$, $OPT_{TSPN} \le OPT$, and $|\gamma| \le \beta \cdot OPT_{TSPN}$, we get that the makespan of our solution is $|\gamma| + |FSC(\mathcal{S}')| \le \beta \cdot OPT_{TSPN} + d \cdot |FSC(\mathcal{S})| \le (\beta + d) \cdot OPT$. □

4.5 Proof of Lemma 2.3

Proof: Let T be a minimum spanning tree (MST) of the point set X. By the definition of well clustered, we know that any inter-cluster edge is of length greater than any intra-cluster edge; thus, by a standard property of the MST, the restriction of T to any one cluster C_i must be connected (since, e.g., Kruskal's algorithm would consider all intra-cluster edges of C_i before considering any edge linking a point of cluster C_i to a point in another cluster). Since the intra-cluster edges of cluster C_i form a (connected) subtree of T, for each cluster C_i, we can obtain a Hamilton cycle of the points C_i, by the usual tree-doubling (and shortcutting) of the subtree, with the length of the cycle at most twice the weight of the subtree. Additionally, the standard method is known to give a Hamilton *path*, between any two distinct points of C_i, with the length of the path at most twice the weight of the subtree. Now consider the inter-cluster edges of T, and define the *cluster tree*, T_C, whose nodes are clusters C_i and whose edges correspond to the inter-cluster edges of T that interconnect two clusters. By the usual tree-doubling (and shortcutting) of T_C, we obtain a Hamilton cycle on the clusters, with each edge being an inter-cluster edge (either an edge of T or an edge obtained via shortcutting, but still going from an endpoint of an inter-cluster edge of T to another endpoint of an inter-cluster edge, in another cluster) linking some point of one cluster to some point of another cluster. The total weight of the set E_C of edges in this Hamilton cycle is at most twice the sum of the edge lengths of the inter-cluster edges of T. Now, for any cluster C_i for which the two edges of E_C are incident to the same point p_i of C_i, we add to E_C a Hamilton cycle on C_i, and for any cluster for which the two edges of E_C are incident on distinct points, $p_i, p'_i \in C_i$ with $p_i \neq p'_i$, we add to E_C a Hamilton path on C_i from p_i to p'_i. The resulting set of edges span all of X and form a connected set; it becomes a Hamilton cycle on X by shortcutting at each p_i where a Hamilton cycle on C_i was attached. The resulting Hamilton cycle on X has the desired property of visiting all points within a cluster C_i contiguously, while having total length at most twice that of MST(X). □

4.6 Proof of Lemma 2.4

Proof: We compute a minimal square, W, containing all of the k-polyominoes. Assume, without loss of generality, that the k-polyominoes are of resolution $s = 1$ and that W is the square $W = [0, L] \times [0, L]$, where L is a multiple of k. Consider the following two tilings of the plane:

$$\mathcal{T}_1 = \{[(2i-2)k, 2ik] \times [(2j-2)k, 2jk]\}, \qquad \mathcal{T}_2 = [(2i-1)k, (2i+1)k] \times [(2j-1)k, (2j+1)k],$$

for integers i and j. Each tiling consists of $2k$-by-$2k$ squares (called *tiles*); each tiling is a translation (shift) by vector (k, k) of the other tiling. Note that each k-polyomino $R_i \in \mathcal{R}$ is fully contained either in a tile of $\mathcal{T}_1$ or of $\mathcal{T}_2$. We refer to those regions of $\mathcal{R}$ that are fully contained in a tile of $\mathcal{T}_1$ as *red* regions, while the remaining regions (fully contained in a tile of $\mathcal{T}_2$) are *blue*.

Consider the set T_1 of occupied red tiles of $\mathcal{T}_1$ that each contain at least one k-polyomino. Consider one tile of T_1 and its refinement into a subgrid of $2k$-by-$2k$ unit pixels. Each k-polyomino within the tile is a union of k of these $(2k)^2$ pixels. A TSPN-TLB path for the k-polyominoes within this one tile can be $O(1)$-approximated as follows: For each pixel within the tile, decide (approximately) how much pause time (between 0 and $b_{\max} \leq N^2$) to spend in it, and then interconnect the visited pixels (those with pause times > 0) with a minimum spanning tree (MST), computed in $poly(k)$ time per MST. There are $(\log b_{\max})^{O(k^2)} = (\log N)^{O(k^2)}$ choices for these approximate pause times (up to an accuracy of a factor 2), since there are $\log b_{\max} = O(\log N)$ choices for the pause time per pixel. We then optimize over all choices that yield pause times that satisfy the time lower bounds for all k-polyominoes. The result of this optimization within a single tile is a subtour of pause points within (unit) pixels of the tile.

Now, we partition the red tiles of $\mathcal{T}_1$ into 9 classes, according to the values of $i \pmod 3$ and $j \pmod 3$. Note that within one of these classes, the tiles (in the class) along any row or column are spaced with two tiles (not in the class) in between them. Thus, the pause points we computed within the occupied red tiles within one class are well-clustered, so we can apply Lemma 2.3 to yield a tour that connects all of the pause points within the class, while respecting the clustering implied by the tiles. Finally, we interconnect (at cost $O(OPT)$) all of the 9 tours, to yield an overall solution for the red tiling; separately, we do the same for the blue tiling, then merge. □

4.7 Proof of Lemma 2.5

Proof: Partition W into a n^2-by-n^2 grid of squares, each with side length δ/n^2. Let $\mathcal{R}_q$ be the regions in $\mathcal{R}_{small}$ contained in a square q, and let $n_q = |\mathcal{R}_q|$. Since the number of cells in the arrangement of $\mathcal{R}_q$ is $O(n_q^2)$, by formulating the fractional set cover (FSC) problem for $\mathcal{R}_q$ as an LP with n_q constraints and n_q^2 variables, we can solve the FSC exactly in $poly(n_q)$ time. Moreover, since each basic feasible solution has $\leq n_q$ nonzeros, there are $\leq n_q$ pause points we need to connect, so the connection cost within q is $O(\sqrt{n_q} \cdot diam(q)) = O(\sqrt{n_q}\frac{\delta}{n^2})$. Summing over all q's, the total connection cost is $\leq OPT_{small} + \sum_q n_q \frac{\delta}{n^2}$. On the other hand, since the optimum for FSC is a lower bound on OPT, the covering cost is OPT_{small}. Hence the total cost on $\mathcal{R}_{small}$ is $O(OPT_{small} + \delta)$. □

4.8 Proof of Lemma 2.6

Proof: Let $\ell_0 = \lfloor \log \frac{\delta}{n^2} \rfloor$. Let $\mathcal{R}_\ell$ be the k-polyominoes in $\mathcal{R}_W$ of side length 2^ℓ for $\ell = \ell_0, ..., \lceil \log \delta \rceil$. For each ℓ, compute an $O(1)$-approximation using Lemma 2.4. Also find a solution for the negligible regions using Lemma 2.5. The cost for connecting those $O(\log n)$ solutions is $O(\delta \log n)$, so the total cost is $O(\log n(OPT + \delta))$. □

4.9 Proof of Lemma 2.7

Recall that W is the bounding square of OPT, which we "guessed" correctly by enumeration. We classify the regions in $\mathcal{R}$ into the following classes:

Class 1: Those with exactly one polyomino that intersects ∂W. We first claim that the optimal solution for $\mathcal{R}_1$ whose trajectory is ∂W has makespan $O(OPT)$. Observe that since each k-polyomino has only one polyomino intersect ∂W, if there is $p \in W$ is contained in the intersection of a subset of regions $\mathcal{R}'' \subseteq \mathcal{R}_1$, then there is a $p' \in \partial W$ that is also contained in $\cap_{R \in \mathcal{R}''} R$. Therefore, the FSC of $\mathcal{R}_1$ restricted on ∂W has the same optimum as FSC of $\mathcal{R}_1$ in W. Also observe that the optimal solution for $\mathcal{R}_1$ whose trajectory is ∂W has makespan $O(OPT)$, thus the total cost is $O(OPT)$.

Next we show how to find such a solution by solving an LP. Define an equivalence relation "$\sim$": $p \sim q$ if the set of regions in $\mathcal{R}'$ containing p and q are the same. Partition ∂W into cells so that each cell is a maximal connected equivalence class under $\sim$. Let ℓ_σ be the length of cell σ, then the following LP finds the optimal solution w.r.t. $\mathcal{R}_1$ with trajectory ∂W:

$$
\begin{aligned}
& min \sum_\sigma t_\sigma \\
\text{s.t.} \quad & \sum_{\sigma \subset R_i} t_\sigma \geq b_i, \; \forall R_i \in \mathcal{R}_1, \\
& t_\sigma \geq \ell_\sigma, \qquad \forall \sigma, \\
& t_\sigma \geq 0, \qquad \forall S \in \mathcal{S}.
\end{aligned}
$$

Class 2: Those with more than 1 polyomino that intersects ∂W. Then, the enlarged box $[-kL, kL]^2$ will contain such k-polyominoes. Bucket them into $O(\log n)$ classes according to diameters, then apply exactly the same approach as in Lemma 2.4 for each class. The sum of cost of these $\log n$ solutions is clearly $O(\log n \cdot OPT)$.

Once we find an approximate solution for each of the $O(\log n)$ classes, we can paste them by traversing ∂W, which only incurs an extra cost of $O(OPT)$.

4.10 Proof of Theorem 2.9

Proof: We localize and discretize the problem in much the same way we did for the PTAS for disjoint fat convex regions (Section 4.2). Because the polygonal regions R_i are general and overlapping, their arrangement is more complex, but still polynomial in size. We transform an arbitrary tour into its "taut" equivalent of the same homotopy type with respect to the vertices of the arrangement, after which the savings in path length is used at pause points at vertices of the arrangement, in order to restore TLB's after the shortening may have caused violations. With a polynomial number of vertices of an optimal trajectory, and with the upper/lower bounds on OPT shown in Section 4.2, we are able to discretize and localize. This results in a polynomial (in n) size regular square grid within which we search. Let G be this (polynomial-size) grid graph, with each internal grid point connected to its 4 neighbors. Shortest paths in this grid graph approximate (with factor $\sqrt{2}$) Euclidean distances between grid points. Each original region R_i is replaced by a set Γ_i of grid points. If a grid point v lies in multiple sets Γ_i, we replicate it, making a copy for each Γ_i that includes it, and all such copies are embedded as distinct points at the grid point v, with distances of 0 between any two copies, and with grid graph edges linking each copy to neighboring grid graph nodes. (This replication is in order to map to the CST formulation, which assumes the groups of points are disjoint.) In order to model within the CST problem the possibility of pausing at any particular grid node v, for time up to $b_{\max} \leq N^2$, we instantiate $b_{\max} \leq N^2$ dummy nodes associated with each grid node v, attaching each dummy node to v with an edge of length 1, so that the grid graph G has stars of $b_{\max}$ degree-1 dummy nodes attached to each grid node v. (If v is replicated, because it is in multiple (say, m_v) sets Γ_i, we also replicate each dummy node m_v times, and interconnect each such cluster with edges of length 0, while connecting each copy of a dummy node to its respective copy of node v.)

We now consider the CST for which the points of the metric space are the $poly(n) \cdot b_{\max}$ grid points of the search grid, together with all associated dummy nodes, and the groups are the individual sets Γ_i (intersected with the search grid rectangle), together with their associated dummy nodes, with each group having its associated requirement given by the TLB b_i. The distances for the metric space are given by shortest path distances in the grid graph, augmented by the "stars" of dummy nodes hanging off of each grid point. Then, the requirement that region R_i be visited for total time (with preemption) at least b_i becomes the requirement in the CST formulation that we construct a covering Steiner tree with requirement to span at least b_i dummy nodes associated with grid points in the group Γ_i. Our formulation then has N groups, $b_{\max} \cdot poly(n)$ vertices, and maximum requirement $b_{\max}$. An f-approximate solution to the CST problem then yields a tree visiting at least b_i dummy nodes associated with grid points Γ_i that represent R_i; traversing this tree yields a trajectory with the required TLBs, and the makespan of the trajectory is within factor $O(f)$ of optimal. □

4.11 Proof of Lemma 2.10

Proof: For simplicity assume the dimension $d = 2$.
(i) We say a dyadic square σ is *nonempty* if $\sigma \cap V \neq \emptyset$. We will show, for any fixed $\ell \in [K]$, it holds $z_\ell(V)D/c^\ell = O(MST(V)$. When $z^\ell(V) \leq 4$, the claim is trivial, so we assume $z^\ell(V) > 4$.

Color the squares in G_ℓ with 4 colors: assign a color to each square so that the cells whose column and row both have the same parity are given the same color. Clearly there exists a color (say, red) with at least $z_l(V)/4$ nonempty cells of that color (red). Since MST(V) spans the vertices in all red squares, and the distance between any two red squares is at least D/c^ℓ,

$$MST(V) = \Omega\big(\frac{D}{c^\ell}(z_\ell(V) - 1)\big),$$

hence $z_\ell(V)D/c^\ell = O\big(MST(V)\big)$.

(ii) We construct a spanning tree in a "bottom-up" (w.r.t. the quadtree) manner.

Start from the bottom level ($\ell = K$). Note that each square in G_K has no children. For each $\sigma \in G_K$, we connect the points in σ by building an MST. This costs us at most $D(m_\sigma - 1)/c^K$, where m_σ is the number of points in σ. Summing over $\sigma \in G_K$, the total connecting cost for G_K is bounded by

$$\frac{D}{c^K} \sum_{\sigma \in G_K} (m_\sigma - 1) \leq n \cdot D/c^K = O(\mathcal{E}(V)).$$

Fix $\ell \leq K - 1$. Suppose for each $\sigma' \in G_{\ell+1}$, the points in σ' are already connected into a tree. Fix a $\sigma \in G_\ell$, let n_σ be its number of nonempty subcells. The points within each of these n_σ subcells are connected into a tree, we connect these n_σ trees by adding $n_\sigma - 1$ edges, each with length D/c^ℓ. Hence, the total cost is at most $(n_\sigma - 1)D/c^\ell$. Summing over $\sigma \in G_\ell$, the total connecting cost in G_ℓ is at most

$$\sum_{\sigma \in G_\ell} n_\sigma \cdot D/c^\ell \leq z_{\ell+1}(V) \cdot D/c^\ell.$$

The proof completes by summing over the connecting costs in G_ℓ for $\ell = 0, ..., K$, which is $O(\mathcal{E}(V) + \sum_{\ell=0}^{K-1} z_{\ell+1}(V) \cdot D/c^\ell) = O\big(c \cdot \mathcal{E}(V)\big)$. □

4.12 Proof of Lemma 2.11

Proof: We prove the non-preemptive version first. For clarity of presentation, we assume $\omega = 0$; there is no essential difference for general ω. Recall that $\Lambda(V, w) = w(V) + \mathcal{E}(V)$. We claim that the problem reduces to finding a weighted point set (V, w) that maximizes profit, subject to a constraint of the form $\Lambda(V, w) \leq L'$, where L' is any given budget.

In fact, let (V^*, w^*) be the weighted point set that realizes schedule $\Sigma^*(L)$. Then by Lemma 2.10,

$$\Lambda(V^*, w^*) = \mathcal{E}(V^*) + w^*(V^*) \leq O(c \log_c N) \cdot MST(V^*) + w^*(V^*) = O(c \log_c N \cdot L).$$

Thus, if (V, w) is the optimal weighted point set with $\Lambda(V, w) \leq O(c \log_c N \cdot L)$, then it has at least the profit of (V^*, w^*), i.e. $\Pi^*(L)$.

On the other hand, we can transform (V, w) into a feasible solution by traversing $MST(V)$, pausing at each $v \in V$ for $w(v)$ time. The cost of this solution is, by Lemma 2.10,

$$O(1) \cdot \Lambda(V, w) = O(c \log_c N \cdot L).$$

Now we present an exact algorithm for finding the optimal weighted point set, with prescribed budget on Λ. Suppose we "guessed" the bounding square of $P^*(L)$ correctly, of diameter D, rescaled to 1 for simplicity. Consider a dyadic square $\sigma \in G_\ell$, let $\Pi(V, w; \sigma)$ be the number of regions *fully* *contained in* σ whose TLB is satisfied by (V, w). Given parameters (numbers) $E, W, M, \nu \in \mathbb{N}$, let $Z(\sigma, W, E, M, \nu)$ be the optimum value of the solution to

$$\begin{aligned} &\max \Pi(V, w; \sigma) \\ &\text{s.t. } \mathcal{E}(V; \sigma) \le E, \\ &\quad \sum_{v \in V} w(v) \le W, \\ &\quad \max_{v \in V} w(v) = M, \\ &\quad Card(V) \le \nu. \end{aligned}$$

Let Q_σ be the regions contained in σ but not in any of $\sigma' \prec \sigma$. Given integers $M_1, ..., M_{c^2}$, let $f(M_1, ..., M_{c^2})$ denote the number of regions in Q_σ we can "cover", by pausing M_i time in the i-th subcell. Formally,

$$f(M_1, ..., M_{c^2}) = |\{R \in Q_\sigma : \max\{M_i : \sigma_i \subset R, i \in [c^2]\} \ge b(R)\}|.$$

Then we obtain the following recursion:

$$Z(\sigma, W, E, M, \nu) = \max\left\{f(M_1, ..., M_{c^2}) + \sum_{j=1}^{c^2} Z(\sigma_j, E_j, \Lambda_j, M_j, \nu_j)\right\},$$

where the maximum is taken over all combinations of integral parameters satisfying

$$\sum_{j=1}^{c^2} W_j \le W,$$

$$E \le \sum_{i \in [c^2]} E_i + |\{i : M_i \ge 1\}| \cdot \frac{D}{2^\ell},$$

$$\max_{i \in [c^2]} M_i = M,$$

$$\text{and} \quad \sum_{j=1}^{c^2} \nu_j \le \nu.$$

By dynamic programming, in $poly(n, (N \log n)^{O(\omega)})$ time, we can find $Z(\sigma_0, W, E, M, \nu)$ for all combinations of (σ, W, E, M, ν). In fact, the integral parameters (W, E, M, ν) range in $[1, Nb_{\max}] \times [1, O(\log_c n)] \times [1, b_{\max}], \times [1, N]$. For the preemptive version, the dynamic programming algorithm is similar, except that the "profit" Π now is defined to be the total amount of "coverage" that V, w contributes, and the parameters (W, E, M, ν) range in $[1, Nb_{\max}] \times [1, O(\log_c n)] \times [1, b_{\max}], \times [1, poly(n)]$, leading to running time $poly(n^{O(\omega)})$.

□

4.13 Proof of Lemma 2.12

Proof: Recall that for a weighted point set, (V, w), we have a "surrogate" $\mathcal{E}(V) + w(V)$ (giving up factor $O(\log n)$ with preemption, or factor $O(\log N)$ without preemption) for the actual cost of the solution obtained by traversing $MST(V)$ and pausing at every $v \in V$ for $w(v)$ time. Therefore, the problem reduces to finding an exact solution for the following problem: find the weighted point set to minimize $\mathcal{E}(V) + w(V)$, subject to satisfying all TLB's for $\mathcal{R}$.

This can be solved by the dynamic programming algorithm described in the proof of Lemma 2.11. Indeed, let σ_0 be a bounding box with side-length $O(diam(OPT))$; then, finding (V, w) with minimal $\mathcal{E}(V) + w(V)$ satisfying all TLB's without preemption is equivalent to finding (V, w) that realizes $\min\{W + E : Z(\sigma_0; W, E, M, \nu) \geq N\}$. (In the case with preemption, this becomes $\min\{W + E : Z(\sigma_0; W, E, M, \nu) \geq \sum_j b_j\}$). □

4.14 Proof of Lemma 2.13

Proof: Here we prove the case without preemption; the proof extends easily to case with preemption. Pick an arbitrary point O as the root. Consider the following greedy algorithm: while there is an "alive" region in $\mathcal{R}$ (i.e., a region whose TLB is not yet satisfied), for each $i \leq h$, invoke the greedy-oracle to compute an approximately most cost-effective schedule. Among these h solutions, choose one with the best cost-effectiveness. Update the set of alive regions, and repeat.

Relabel the regions so that for any $i < j$, R_i is "killed" by greedy no later than R_j. Let k_t be the total number of regions killed at the end of iteration t. At iteration j, since OPT kills $N - k_{j-1}$ regions, there exists an $i \leq h$ such that OPT kills at least $(N - k_{j-1})/h$ regions in $\mathcal{R}^{(i)}$. Run the greedy-oracle on each $\mathcal{R}^{(i)}$; then, we can find a partial schedule with cost-effectiveness no more than $\alpha h OPT/(N - k_{j-1})$. Hence, the total cost of greedy is upper bounded by

$$\sum_{j=1}^{s} (k_j - k_{j-1}) \frac{\alpha h OPT}{N - k_{j-1}} = O(\alpha h \log N \cdot OPT).$$

□

4.15 Proof of Theorem 3.1

Proof: Our reduction is from the Hitting Lines problem: Given a set L of n lines in the plane, and an integer $k \geq 1$, decide if there exist k points ("hitting points"), $p_1, \ldots, p_k$, such that every line of L has at least one of the points on it. The decision problem is NP-complete ([23]) and the optimization problem, to minimize the number of hitting points, is APX-hard ([4]). From L we construct a simple polygon, a "spike box" P, consisting of a square S, large enough to contain all of the crossing point intersections among the lines L, with n very skinny "spikes" (triangular extensions) extending a short distance out from S, along each line $\ell_i \in L$; let t_i denote the vertex at the tip of the ith spike. Refer to Fig. 9. The visibility polygon, $VP(t_i)$, of spike tip t_i is a very skinny triangle of points very close to ℓ_i. With appropriate scaling, the square S can be assumed to have diameter 1 (side length $1/\sqrt{2}$).

Let the TLB for points in P be n^2. We claim that L can be hit with k points if and only if there is a WRP-TLB tour of makespan at most $kn^2 + n$. First, if k hit points suffice, then the following is a valid WRP-TLB of makespan at most $kn^2 + n$: place pause points at each of the k hit points and pause at each for time n^2 (this takes total time kn^2); then, interconnect these $k \leq n$ points

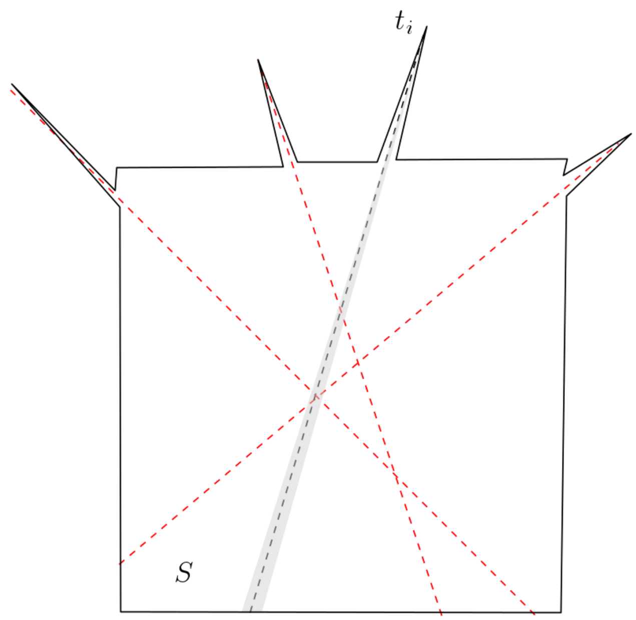

Figure 9. A spike box simple polygon associated with a set L of n lines, the input to HITTING LINES.

within $S \subseteq P$ with a tour of length at most $k \cdot diam(S) \leq k \leq n$. (In fact, a TSP on these k points has length at most $O(\sqrt{k})$.) Conversely, if there is a tour γ with makespan $\tau \leq kn^2 + n$ for the WRP-TLB instance, then the tip t_i of each spike must be seen contiguously (without preemption) for a time of n^2. Consider a traversal, from time 0 to time τ, of the trajectory γ; this traversal involves moving along the corresponding curve, within P, at speed at most 1. (The speed could decrease, even to zero, e.g., at pause points.) Place points $p_1, p_2, \ldots$ at the k locations along γ at times $n^2 - 1, 2(n^2 - 1), 3(n^2 - 1), \ldots, k(n^2 - 1)$ within the makespan time interval $[0, \tau = kn^2 + n]$. Since each spike tip point t_i is seen contiguously for a length of time at least n^2, and the points p_j are spaced in time by $n^2 - 1$, each t_i is seen by at least one of these k points in P. (There is no way for a contiguous interval of time of length n^2 on the time axis, $[0, \tau = kn^2 + n]$, to avoid the set of k points spaced uniformly with separation $n^2 - 1$.) Thus, these k points, slightly perturbed to lie exactly on the respective lines, constitute a hitting set for the n lines L. We conclude that the decision problem is NP-complete; APX-hardness follows similarly, appealing to [4]. □

4.16 Proof of Theorem 3.2

Proof: The visibility polygons of the targets T define an arrangement $\mathcal{A}$ in P. An optimal trajectory will not go above (in y-coordinate) the topmost among the source v_s, the destination v_t, and the vertices of the arrangement $\mathcal{A}$. Thus, we can assume we know $y_{\max}$, the maximum y-coordinate of a point along an optimal trajectory. We then claim that an optimal solution is given by a trajectory γ that goes from v_s upwards to altitude $y_{\max}$, then horizontally at altitude $y_{\max}$ to the point at or above v_t, then downwards to v_t. The agent moves along this path at full speed except to wait at selected pause points along the way. Specifically, for each cell σ of the arrangement through which γ passes, we place one candidate pause point $p_\sigma \in \gamma \cap \sigma$, and we use the FSC linear programming formulation to compute the optimal waiting times, $x_\sigma \geq 0$, to pause at the points p_σ, in order to satisfy the TLBs for all targets. (We are exploiting the special structure of the domain P, that if a point p sees target t_i, then any point directly above p, on the vertical ray upwards, also sees t_i.) $\square$

4.17 Monotone Orthogonal Polygons

We now assume that P is an x-monotone orthogonal polygon and that the target points T are a subset of the convex vertices of P. Here, we restrict our attention to the path version of WRP-TLB, and consider, specifically, orthogonal paths in P; we are to find a minimum-makespan trajectory (starting/ending anywhere in P). Further, we restrict our attention here to the case of *rectangle vision*: under rectangle vision, we say that point $p \in P$ *sees* point $q \in P$ if and only if the axis-aligned rectangle determined by (p, q) lies inside the polygon P (considered to be a closed set).

Consider the partitioning of P into rectangular cells by the set of edge extensions (horizontal/vertical chords) through all of the edges of P. Let $G = (V, E)$ be the planar graph associated with this arrangement of segments, with vertices V and (axis-parallel) edges E interconnecting them.

Lemma 4.8. *There exists an optimal (orthogonal) trajectory path in G, traveling with full speed $v_{\max} = 1$ on the edges E, pausing only at the vertices V.*

Proof: By local shifting, an optimal (orthogonal) trajectory γ can be moved onto the edges E, segment by segment, without increasing its cost. If trajectory γ travels on $(a, b) \in E$, spending time $|ab| + \Delta t$ on edge (a, b), then we claim that γ can be modified so that it travels at full speed along (a, b) and pauses for time Δt at one of the endpoints, a or b. (This follows from the fact that one of $VP_{cvx}(a)$ and $VP_{cvx}(b)$ contains $VP(u)$ for all points $u \in (a, b)$, where $VP_{cvx}(u)$ denotes the set of convex vertices of P that are seen by point u.) $\square$

Thus, we can assume that we seek a path between two vertical chords, ξ_{left} and ξ_{right}, of P (each being an extension of a vertical edge of P; we can enumerate over all choices of the pair $(\xi_{left}, \xi_{right})$. We truncate P at these chords, and refer, for simplicity, to the remaining x-monotone polygon simply as P.

Let $\partial^+(P)$ and $\partial^-(P)$ be the x-monotone upper and lower boundary chains of P obtained when removing ξ_{left} and ξ_{right} from $\partial(P)$. If there is a horizontal chord joining ξ_{left} and ξ_{right}, separating $\partial^+(P)$ and $\partial^-(P)$, we say that P is *separable*, and we let Strip(P) denote the horizontal strip (rectangle) within P determined by the highest/lowest separating chords.

We can partition P into *separable pieces* by the following natural sweep algorithm. Let the vertical edges of P be denoted $e_1 = \xi_{left}, e_2, \ldots \xi_{right}$, in left to right order. Consider each edge in sequence: if P truncated at e_i is separable while P truncated at e_{i+1} is not separable, then cut P vertically at e_i, and continue in the subpolygon to the right of e_i, cutting P into pieces $P_1, P_2, \ldots$, from left to right. The vertical blue lines in Fig. 10 are the cuts. Note that for any pair of consecutive pieces, the projections of their strips onto the y-axis are disjoint.

A linear programming formulation as a fractional set cover problem, similar to our earlier formulations (see Section 4.4), so not repeated here, yields the following:

Lemma 4.9. *For both the preemptive and non-preemptive versions, one can, for a given path γ, compute in poly(n) time a set of pause points along γ, together with associated wait times, so that the resulting trajectory (along path γ) is optimal for WRP-TLB.*

The following follows directly from the definition of a strip.

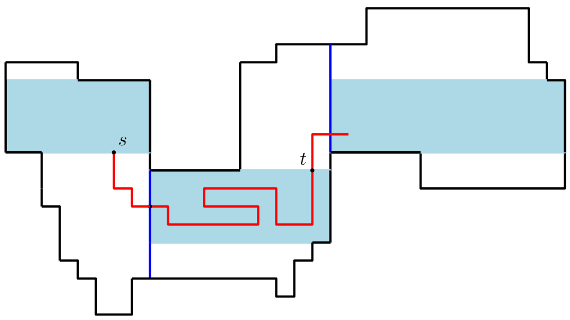

Figure 10. Strips highlighted in blue.

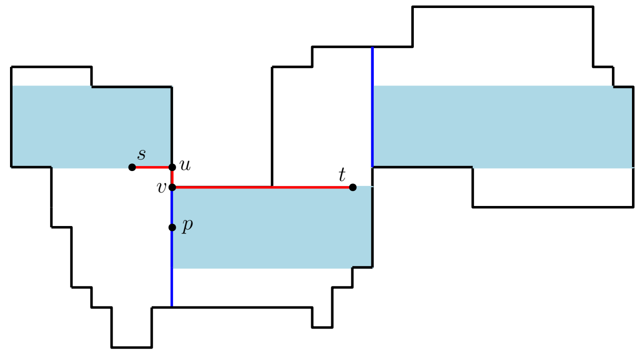

Figure 11. Proof of Lemma 4.11.

Lemma 4.10. *For any i and $p \in Strip(P_i)$, p cannot see any target in P_j for any $j > i$.*

Lemma 4.11. *Consider a minimum-makespan solution whose path, γ, has a minimum number of turns (among optimal solutions). Then, (1) γ enters/leaves $Strip(P_i)$ at most once; (2) the subpath, $\gamma_{s,t}$, of γ from the point s where γ leaves P_{i-1} to the point t where γ leaves P_i is a shortest orthogonal path in P from s to t.*

Proof: Refer to Fig. 10 and Fig. 11. Claim (1) is immediate (by a shortcutting argument). For claim (2), suppose the contrary, and let p be the point where γ enters $Strip(P_i)$, and let uv be the intersection between the shortest path $SP(s,t)$ and the boundary, ∂P, of P. We construct a solution whose trajectory is $SP(s,t)$ and contributes to each target no less than $\gamma_{s,t}$ does. Suppose the time that γ spends on $\gamma_{s,p}, \gamma_{p,t}$ are t_{sp}, t_{pt}, respectively. Then, we travel from s to u with full speed $v_{\max} = 1$ and pause at q for $t_{sp} - d_{L_1}(s,v)$ time, then travel from v to t, with the following pausing rule: for each x-interval $I \subseteq [x(v), x(t)]$ within the interval between the x-coordinate of v and the x-coordinate of v, if γ spends δ time in the slab $I \times \mathcal{R}$, then we spend δ time on I. □

This concludes the proof of Theorem 3.3.

4.18 Proof of Lemma 3.4

Proof: Let T^* be a shortest watchman tree and ℓ^* be its length. Choose s to be any edge shorter than ℓ^*. It is easy to see that the cost for connecting all edges in F' is $O(\ell^*)$, so we only need to show the total length of TWPT(s,P) is $O(\ell^*)$. Consider the following charging scheme: for each infinitesimal horizontal (vertical) segment σ, charge its cost vertically (horizontally) to its projection on T^*. Recall that each point in P sees at most 4 window edges, so every point on T^* is charged at most 4 times. Therefore, the total length of TWPT(s,P) is $O(\ell^*)$. □

4.19 Proof of Lemma 3.5

Proof: Let H be the histogram in WP(s,P) containing g. First, if $u \in H$, then a shadow guard g_i is on the base edge (in Fig. 8 it is g_3), so g_i sees u. Otherwise, $u \in H' \neq H$ for another histogram H' in P. Since g sees u, g is able to see e', the base edge of H', i.e. the rectangle R with diagonal $\overline{gu}$ is contained in P. Note that one of the shadow guards g_i is exactly the intersection point of $\overline{bu}$ and e' (in Fig. 8 it is g_4). Hence, the rectangle with diagonal $\overline{ug_i}$ is contained in P. □

4.20 Proof of Lemma 3.6

Proof: Set $\hat{x} = 0$ initially. For each $g \in P$ with $x_g > 0$, increase $\hat{x}_{g_i}$ for each shadow guard g_i that sees g by x_g respectively. In the end, by Lemma 3.5, for each target v,

$$\sum_{g \in V} \hat{x}_g \geq \sum_{g \sim v, g \in V} x_g \geq b_v,$$

hence $\hat{x}$ is a feasible to FSC. Note that all shadow guards are contained in F', and each $g \in P$ sees at most 4 shadow guards, we have $\sum_{u \in V} \hat{x}_u \leq 4 \sum_{v \in P} x_v$,, completing the proof. □

Acknowledgements. J. Mitchell is partially supported by the National Science Foundation (CCF-1526406), the US-Israel Binational Science Foundation (project 2016116), and DARPA (Lagrange). We thank Estie Arkin, Michael Biro, Chris Bishop, Kan Huang, Christiane Schmidt, Steve Skiena, and other participants of the CG and Algorithms Seminars at Stony Brook for their input and discussions on the problems studied here. We particularly thank the anonymous reviewers of an earlier version of this paper for their insightful comments, questions, and suggestions, which greatly improved the paper.

References

[1] A. Antoniadis, K. Fleszar, R. Hoeksma, and K. Schewior. A PTAS for euclidean TSP with hyperplane neighborhoods. In T. M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1089–1105. SIAM, 2019.

[2] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Appl. Math.*, 55:197–218, 1994.

[3] Y. Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 161–168. ACM, 1998.

[4] B. Brodén, M. Hammar, and B. J. Nilsson. Guarding lines and 2-link polygons is APX-hard. In *Proceedings of the 13th Canadian Conference on Computational Geometry, University of Waterloo, Ontario, Canada, August 13-15, 2001*, pages 45–48, 2001.

[5] S. Carlsson, H. Jonsson, and B. J. Nilsson. Finding the shortest watchman route in a simple polygon. *Discrete & Computational Geometry*, 22(3):377–402, 1999.

[6] S. Carlsson and B. J. Nilsson. Computing vision points in polygons. *Algorithmica*, 24(1):50–75, 1999.

[7] T.-H. H. Chan and S. H.-C. Jiang. Reducing curse of dimensionality: Improved PTAS for TSP (with neighborhoods) in doubling metrics. *ACM Transactions on Algorithms (TALG)*, 14(1):9, 2018.

[8] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: Deterministic approximation algorithms for group Steiner trees and k-median. In *Proc. of the 13th Annual ACM Symposium on Theory of Computing*, pages 114–123. ACM, 1998.

[9] W.-P. Chin and S. Ntafos. Shortest watchman routes in simple polygons. *Discrete & Computational Geometry*, 6(1):9–31, 1991.

[10] M. de Berg, A. F. van der Stappen, J. Vleugels, and M. J. Katz. Realistic input models for geometric algorithms. *Algorithmica*, 34(1):81–97, 2002.

[11] M. Dror, A. Efrat, A. Lubiw, and J. S. B. Mitchell. Touring a sequence of polygons. In *Proc. 35th ACM Symposium on Theory of Computing*, pages 473–482, 2003.

[12] A. Dumitrescu and J. S. B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. *Journal of Algorithms*, 48:135–159, 2003. Special issue devoted to 12th ACM-SIAM Symposium on Discrete Algorithms, Washington, DC, January, 2001.

[13] A. Dumitrescu and C. D. Tóth. Watchman tours for polygons with holes. *Computational Geometry*, 45(7):326–333, 2012.

[14] A. Dumitrescu and C. D. Tóth. The traveling salesman problem for lines, balls, and planes. *ACM Transactions on Algorithms (TALG)*, 12(3):43, 2016.

[15] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.

[16] S. P. Fekete, J. S. B. Mitchell, and C. Schmidt. Minimum covering with travel cost. *Journal of Combinatorial Optimization*, 24(1):32–51, 2012.

[17] N. Garg and D. S. Hochbaum. An $O(\log k)$ approximation algorithm for the k-minimum spanning tree problem in the plane. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 432–438. ACM, 1994.

[18] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000.

[19] A. Gupta and A. Srinivasan. An improved approximation ratio for the covering steiner problem. *Theory of Computing*, 2:53, 2003.

[20] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 585–594. ACM, 2003.

[21] M. Hammar and B. J. Nilsson. Concerning the time bounds of existing shortest watchman route algorithms. In *International Symposium on Fundamentals of Computation Theory*, pages 210–221. Springer, 1997.

[22] G. Konjevod, R. Ravi, and A. Srinivasan. Approximation algorithms for the covering steiner problem. *Random Structures & Algorithms*, 20(3):465–482, 2002.

[23] N. Megiddo and A. Tamir. On the complexity of locating linear facilities in the plane. *Operations Research Letters*, 1(5):194–197, 1982.

[24] J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[25] J. S. B. Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. In *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, pages 11–18, 2007.

[26] J. S. B. Mitchell. A constant-factor approximation algorithm for TSP with pairwise-disjoint connected neighborhoods in the plane. In *Proc. 26th Annual ACM Symposium on Computational Geometry*, pages 183–191, 2010.

[27] J. S. B. Mitchell. Approximating watchman routes. In *Proc. 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 844–855, 2013.

[28] J. S. B. Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. *arXiv preprint arXiv:1703.01646*, 2017.

[29] J. S. B. Mitchell. Shortest paths and networks. In J. E. G. Csaba Tóth, Joseph O'Rourke, editor, *Handbook of Discrete and Computational Geometry (3rd Edition)*, chapter 31, pages 811–848. Chapman & Hall/CRC, Boca Raton, FL, 2017.

[30] X. Tan, T. Hirata, and Y. Inagaki. An incremental algorithm for constructing shortest watchman routes. *Int. J. Comput. Geometry Appl.*, 3(4):351–365, 1993.

[31] X. Tan, T. Hirata, and Y. Inagaki. Corrigendum to "an incremental algorithm for constructing shortest watchman routes". *Int. J. Comput. Geometry Appl.*, 9(3):319–323, 1999.