A Class of Weighted TSPs with Applications

David Kempe      Mark Klein
University of Southern California

August 3, 2018

Abstract

Motivated among others by applications to the prevention of poaching or burglaries, we define a class of weighted Traveling Salesman Problems on metric spaces. The goal is to output an infinite (though typically periodic) tour that visits the \( n \) points repeatedly, such that no point goes unvisited for “too long.” We consider two objective functions for each point \( x \). The maximum objective is simply the maximum duration of any absence from \( x \), while the quadratic objective is the normalized sum of squares of absence lengths from \( x \). The overall minimization objective is the weighted maximum of the individual points’ objectives. When a point has weight \( w_x \), the absences under an optimal tour should be roughly a \( 1/w_x \) fraction of the absences from points of weight 1. Thus, the objective naturally encourages visiting high-weight points more frequently, and at roughly evenly spaced intervals.

We give a polynomial-time combinatorial algorithm whose output is simultaneously an \( O(\log n) \) approximation under both objectives. We prove that up to constant factors, approximation guarantees for the quadratic objective directly imply the same guarantees for a natural security patrol game defined in recent work.

1 Introduction

In the classic metric Traveling Salesman Problem (TSP), we are given \( n \) points in a metric space, and the goal is to compute a tour that visits each point (at least) once, returns to the starting point, and (approximately) minimizes the total distance traveled. The name is derived from the story of a traveling salesman who needs to sell his product in each of \( n \) cities, and wants to return home as quickly as possible.

In reality, the need for the salesman’s product or service will not typically arise once at time 0 and be forever met with the salesman’s visit. Instead, demand for the product will arise over time in the population, and the salesman will need to return to the same cities repeatedly to serve the demand that has accrued since his most recent visit. Naturally, demand in larger cities accrues more quickly, so larger cities should be visited more frequently. Unserved demand leads to disutility, and the salesman’s objective function in choosing an (infinite) tour is to minimize the overall disutility.

This view motivates the following class of weighted Traveling Salesman Problems (defined more
formally in Section 2). Each of the $n$ points in the metric space has a weight $w_x \geq 0$. For each point $x$, a (infinite) visit schedule induces a distribution of times (total distance traveled) between consecutive visits to $x$ (which we call absence lengths). For the two versions of the problem we study, the cost $c_x$ of a point $x$ is either (1) the maximum absence length from $x$ (we call this version the weighted Max-TSP), or (2) the expected absence length at a time $t$ that is chosen uniformly randomly from the schedule\footnote{This interpretation has meaning only when the schedule is periodic. We define a more general notion in Section 2.} which equals the normalized sum of squares of absence lengths (we call this version the weighted Quadratic TSP). The goal is then to find an infinite schedule minimizing $\max_x w_x c_x$. The objective function encourages the salesman to visit high-weight points more frequently; and not only more frequently, but “roughly evenly spaced.”

In terms of the traveling salesman, this class of problems is motivated as follows. After the salesman has committed to his tour, a city $x$ is chosen adversarially, and develops a need for the salesman’s product/service. The cost incurred is the duration until this need is met, times the population of the city incurring the need. If the time is also chosen adversarially, we exactly obtain the weighted Max-TSP objective; if the time is random, then the expected cost is (half) the weighted Quadratic TSP objective.

An alternative objective — discussed briefly in Sections 1.3 and 5 — is to minimize the sum $\sum_x w_x c_x$ instead of the maximum. These objective functions are motivated when the city is not chosen adversarially, but rather, a random person from the population develops a need for the product.

1.1 A Second Application: Security Games

A second important motivation arises from a class of security games, especially in the prevention of poaching, illegal harvesting, or burglary.

In general, in security games [25, 4, 27], a defender needs to use limited resources to protect a set of $n$ targets of non-uniform value $w_x$ from attack by a rational attacker. The attacker can observe the (typically mixed) strategy chosen by the defender, but usually not the defender’s coin flips. A wide literature has analyzed different variants of this problem, with different combinatorial constraints on the actions of either or both players; see [25, 27] for a comprehensive survey and a general optimization treatment.

Motivated among others by anti-poaching [13, 14] and anti-burglary efforts, a recent paper [19] proposed a variant in which the defender commits to a distribution over (infinite) tours of the $n$ targets; in [19], the metric is assumed to be uniform, i.e., all targets are at distance 1 from each other. The attacker performs a single attack, choosing a target $x$ to attack, as well as an attack duration $t$. The attacker’s reward if not caught is $w_x \cdot t$, and if caught, is 0. In Section 4 we show a reduction from this problem to the weighted quadratic TSP objective discussed above; the reduction only loses a constant factor in the approximation.

If the attacker can observe the defender’s coin flips, then the defender may as well not randomize. A rational attacker will attack a target $x$ immediately after the defender has left it for the longest possible interval, and stay until right before the defender returns to it. Thus, the attacker’s reward
for the best possible attack is exactly the weighted Max-TSP objective.

1.2 Our Results

Our main result is an efficient (combinatorial) \( O(\log n) \) approximation algorithm for both the weighted Max-TSP and the weighted Quadratic TSP. We state the result informally here; a formal statement and proof are given in Section 3.

**Theorem 1 (Main Theorem (stated informally))** There is a polynomial-time approximation algorithm which returns a solution that is simultaneously an \( O(\log n) \) approximation for both the weighted Max-TSP and weighted Quadratic TSP objectives.

As a corollary, we obtain \( O(\log n) \) approximation algorithms for the security game scheduling problems discussed in Section 1.1.

In our algorithm, we first round weights to powers \( 2^{-i} \). The points of weight \( 2^{-i} \) are partitioned (approximately optimally) into \( 2^{i} \) spanning trees \( T_{i} \), such that no spanning tree is too large. Each spanning tree is shortcut into a tour, and those tours are carefully sequenced, such that tours for points of higher weight are repeated correspondingly more frequently.

In Section 2 we give an approximation-preserving reduction from the standard metric TSP to the weighted Max-TSP, implying APX-hardness of the latter by \[23\]. However, there is an obvious gap between the constant lower bound for approximability and the \( O(\log n) \) upper bound we obtain. The question of whether the approximation guarantee can be improved to a constant is a natural question for future work.

1.3 Related Work

The motivation for our problem is similar to that of the well-known Minimum Latency Problem \[3, 8, 16\] (also called the Traveling Repairman Problem). In the Minimum Latency Problem, the objective is to minimize the sum of first arrival times to each point in a given (finite) metric space; a natural motivation is that a task is to be performed at each point, and the goal is to serve each point as quickly as possible on average. The high-level approach of most approximation algorithms for this problem is to consider trees of exponentially increasing total cost, each spanning (approximately) as many points as possible, then to execute the sequence of Eulerian tours of those trees. The arrival time at a point that first appears in tree \( i \) is dominated by the cost of the \( i \)-th tree, and the fact that the point was in no earlier tree serves as a certificate that not many more points could have been visited faster. A constant-factor approximation algorithm for the minimum-cost tree spanning \( k \) vertices \[15\] therefore yields a constant-factor approximation for the Minimum Latency Problem.

The fact that each point only needs to be visited once sets the Minimum Latency Problem apart from our problem. Once a point has been visited, it can be ignored, whereas in our problem, a high-weight point must be visited repeatedly. Following the “exponentially increasing trees” approach outlined above, the final tour (of largest weight) visits all nodes, and will therefore result in a long
absence from each point. In our weighted TSP, if one or a few points have high weight, this can cause the objective function to be a factor of $\Omega(n)$ from optimal (e.g., in a star in which the center node has weight $n$). As a result, it appears unlikely that algorithms based on this approach can yield good approximation guarantees for our problem. Indeed, our algorithm in Section 3 is based instead on a decomposition into multiple spanning trees, each of which is sufficiently small and gives rise to a tour; these tours are then carefully sequenced to ensure that high-weight points are visited regularly enough.

As further evidence of the difference between the problems, notice that in the Minimum Latency Problem, a point $x$ of weight $w_x$ can simply be replaced by $w_x$ points of weight 1; i.e., the unweighted problem is fully general. By contrast, for the Weighted Max-TSP objective, the objective becomes exactly the standard TSP objective when all weights are 1, and for the Weighted Quadratic TSP with uniform weights, the optimum TSP solution is within a constant factor of optimal (see Section 2).

Closely related to the Minimum Latency Problem are the broad classes of problems in vehicle routing and maintenance/machine scheduling. In vehicle routing, one or more vehicles must visit locations to serve demands. Different variants have different constraints on the number or initial locations of vehicles, time windows during which the cities must be visited, etc. The objectives are typically to minimize the total length/distance of a tour, or maximize the demand served within a given amount of time [22]. Constant-factor approximations are known for many variants of the problem, but as with the Minimum Latency Problem, the techniques do not appear to carry over to our problem, because each city must only be visited once.

One recent approach that has improved approximation guarantees for (multi-vehicle) Minimum Latency problems is the use of time-indexed configuration LPs [10, 24]. Here, for each time $t$ and path $P$ (starting from an appropriate origin node), a variable $z_{t,P}$ captures whether $P$ is the path taken up to time $t$ (possibly with another index for the vehicle). In the fractional relaxation, one obtains a “distribution” of arrival times at nodes $v$; it can be shown that sufficiently strong versions of these configuration LPs (which capture joint paths for all vehicles) have small integrality gaps. The analysis exploits more sophisticated variants of the arguments outlined above for trees of expanding total cost.

While configuration LPs are an approach that explicitly models time in the routing problem, it is not clear that it can be extended to our weighted TSPs. The configuration LPs do not (and do not need to) enforce any consistency between paths up to time $t$ and paths up to time $t + 1$. It is not at all clear how one would express the time between consecutive visits to a point in terms of the LP variables of configuration LPs, though it is an intriguing possibility that another type of configuration LP could yield improved approximation guarantees.

The variant of vehicle routing most similar to our problem is the vehicle routing problem with time windows, or TSP with time windows (TSPTW). Here, each city has a single time window, and in order to serve its demand, the city must be visited during this window. The objective is usually phrased as maximizing the number/weight of cities whose time window is met by the vehicle(s). The approximability of this problem has been well studied for different metric spaces. For general metrics, [9] showed that TSPTW does not admit a constant-factor approximation unless $P = NP$. The best known approximation for general metric spaces is an $O(\log^2(n))$ approximation by [5].
They first give an $O(\log(n))$ approximation for the deadline TSP problem, and then shows that any $\alpha$-approximation for deadline TSP leads to an $\alpha^2$ approximation for TSPTW. In addition, [5] give a bicriteria approximation scheme that allows for a parameter $\epsilon$ indicating by what factor deadlines may be exceeded.

The bicriteria approximation scheme of [5] could possibly be leveraged to address our weighted TSPs. After guessing the objective value, one obtains absence times for each point, and can — for each point — partition the timeline into intervals of approximately that length, each interval corresponding to a new “city” in the TSPTW problem. It might be possible to search for a suitable $\epsilon$ and use the algorithm from [5] as a certifier of feasibility or infeasibility of an objective value. However, it is not clear whether or how the approximation guarantees from [5] will translate into approximation guarantees for our weighted TSPs, especially because our problem effectively requires an infinite number of time windows and approximation guarantees of [5] are given in terms of the number of time windows (since there is one time window for each city).

An objective function more closely related to ours is minimized in machine repair scheduling problems [6, 1]. Here, each day, a limited number of machines can be repaired, and the goal is to choose a schedule that minimizes the expected value of a randomly selected machine’s weight, scaled linearly by the number of days since it was last repaired. This of course necessitates that each machine be visited repeatedly. However, any machine can be repaired in any time step, which implicitly corresponds to the uniform metric also studied in [19]. For the uniform metric, constant approximation can be easily obtained, but it does not appear that these can be extended easily to general metrics.

Much recent work has focused on improving the approximation guarantee for the metric TSP below the factor $\frac{3}{2}$ which one obtains from Christofides’s Algorithm. This approach is orthogonal to ours, as we currently do not even know of a constant-factor approximation for our class of problems. The techniques used in this line of work are typically based on rounding the Held-Karp relaxation. Given that our problems are much more sensitive to the specific sequencing of the visits, it appears unlikely that any modification of Held-Karp (or other standard LP rounding approaches) would be useful.

As we mentioned above, our work is directly motivated by recent work [19] on security games for poaching or burglary prevention. [19] studied only the uniform metric, and gave optimal algorithms for a natural class of objectives. Because we show that the objective function of [19] is within a constant factor of the Weighted Quadratic TSP objective, our result gives an approximation guarantee for the objective of [19] for arbitrary metrics, as shown in Section 4. A similar motivation of preventing wildlife poaching underlies recent work [18] on recharging bandits. Among others, [18] define a maximization version in which the reward at each point is an increasing concave function of the time since the last visit, and the goal is to find a tour (approximately) maximizing the total reward collected. As with [19], [18] only considers the case of uniform distances; obtaining provable approximation guarantees for non-uniform distances in the maximization version is an interesting direction for future work.

One special class of security games most closely related to our objective is that of adversarial patrolling games [7, 26]. Similar to our work, the goal there is to define a tour such that an attacker who gets to observe the tour will cause as little damage as possible. The exact objective
functions differ from ours, and the literature focuses on heuristics or exponential-time algorithms rather than polynomial-time algorithms with provable approximation guarantees.

2 Definitions and Preliminary Results

The metric space consists of a set $\mathcal{X}$ of $n \geq 3$ points $x \in \mathcal{X}$ with positive distances $d(x, y) > 0$. For the purpose of this paper, a tour is a mapping $\ell : \mathbb{N} \rightarrow \mathcal{X}$. The interpretation is that the salesman visits the points in the order $\ell(1), \ell(2), \ldots$. Tours will typically visit points infinitely often, and in principle may skip some points completely or for long periods of time; however, such tours are far from optimal.

The total distance or time (we use the terms interchangeably) traveled by the salesman under $\ell$ from step $t$ to $t'$ is $d_\ell(t, t') := \sum_{\tau=t}^{t'-1} d(\ell(\tau), \ell(\tau+1))$. In order to evaluate the quality of a tour, we are interested in the maximum absence from any point, and in the normalized sum of squares of absences. Fix a point $x$, and let $\ell_x^{-1}(k)$ be the time step in which the $k$th visit to point $x$ occurs under $\ell$. Precisely, $\ell(\ell_x^{-1}(k)) = x$, and there are exactly $k$ times $t \leq \ell_x^{-1}(k)$ with $\ell(t) = x$. If a schedule visits point $x$ only $k$ times, then $\ell_x^{-1}(k') = \infty$ for $k' > k$. For notational convenience, we define $\ell_x^{-1}(0) = 1$. We then define the following two cost functions:

$$c_\ell^{(\infty)}(\ell) = \max_{k=0,1,2,...} d_\ell(\ell_x^{-1}(k), \ell_x^{-1}(k+1)), \quad (1)$$

$$c_\ell^{(2)}(\ell) = \limsup_{k \to \infty} \sum_{k'=0}^{k} \left( \frac{d_\ell(\ell_x^{-1}(k'), \ell_x^{-1}(k'+1))}{d_\ell(1, \ell_x^{-1}(k+1))} \right)^2. \quad (2)$$

Remark 2 A few remarks are in order about these definitions:

1. The first objective — which we call the weighted Max-TSP objective — simply measures the longest absence from $x$, which will be infinite if $x$ is only visited finitely many times.

2. The second objective — which we call the weighted Quadratic TSP objective — intuitively captures the expected length of an absence interval, if intervals are selected with probability proportional to their length. If $\ell$ is periodic, this is exactly the definition.

3. The normalization in the definition of $c_\ell^{(2)}(\ell)$ is necessary to ensure that the cost is invariant under concatenating a periodic schedule. In other words, we want to ensure that the cost is the same for a finite schedule repeated infinitely, and the schedule repeated twice repeated infinitely.

4. In the definition of $c_\ell^{(\infty)}(\ell)$, unless $c_\ell^{(\infty)}(\ell) = \infty$, the maximum is actually attained. The reason is the following. The length of any subtour is the sum of a multiset of pairwise distances. Because the distances are lower-bounded by the shortest distance, there are only finitely many different lengths of tours below any given bound $B$. Hence, if the supremum is $B < \infty$, it can only occur as a maximum.
Each point \( x \) has a weight \( w_x > 0 \). Without loss of generality (by rescaling), we assume that the largest weight is 1. Using these weights, the overall cost function is the weighted maximum of the individual points’ cost functions. Formally, it is defined as follows (for \( p \in \{2, \infty\} \)):

\[
C^{(p)}(\ell) = \max_x w_x c^{(p)}_x(\ell).
\]

### 2.1 Preliminary Results

We begin with an easy approximation-preserving reduction from the metric TSP to the weighted Max-TSP. This result establishes APX-hardness using the APX-hardness of the metric TSP [23], and the bound will also be important for our analysis in Section 3.

**Lemma 3** Consider an instance in which all weights are the same: \( w_x = 1 \) for all \( x \). Let \( \text{OPT}_{\text{TSP}} \) be the cost of the optimum TSP tour, and \( C^* \) the cost of the optimum tour with respect to the \( C^{(\infty)}(\ell) \) objective. Then, \( \text{OPT}_{\text{TSP}} = C^* \).

**Proof.** First, repeating the optimum TSP tour gives a tour in which each absence from each point \( x \) (and in particular the maximum absence) is exactly \( \text{OPT}_{\text{TSP}} \). Therefore, \( C^* \leq \text{OPT}_{\text{TSP}} \).

For the converse inequality, let \( \ell^* \) be the optimum tour. Let \( x \) be a point maximizing the length of any absence between consecutive visits to \( x \). Let \( t, t' \) be time steps of consecutive visits to \( x \) under \( \ell^* \) achieving this maximum, and let \( D = d_{\ell^*}(t, t') \) be the total time between these consecutive visits. Between \( t \) and \( t' \), all points must be visited at least once. If not, then a point \( y \) not visited between these steps would have a longer absence interval than \( x \), namely, at least from step \( t - 1 \) to \( t' + 1 \). Because each point is visited at least once, the subtour of \( \ell^* \) from step \( t \) until \( t' \) is a candidate TSP tour, showing that \( \text{OPT}_{\text{TSP}} \leq C^* \).

**Lemma 4** Under the weighted Quadratic TSP objective, for every tour \( \ell \), there is a periodic tour \( \ell' \) such that \( C^{(2)}(\ell') \leq 3C^{(2)}(\ell) \).

**Proof.** The claim is trivial if \( C^{(2)}(\ell) = \infty \), so we focus on the case \( C^{(2)}(\ell) < \infty \). Let \( k_0 \) be large enough such that for all \( k \geq k_0 \) and all points \( x \), we have

\[
\sum_{k'=0}^{k}(d_\ell(\ell^{-1}_x(k'), \ell^{-1}_x(k' + 1)))^2 \leq \frac{3/2}{d_\ell(1, \ell^{-1}_x(k + 1))}.
\]

Let \( t_0 = \max_x \ell^{-1}_x(k_0 + 1) \), and let \( t_1 \) be large enough such that for all \( x \) and all \( k \) with \( \ell^{-1}_x(k) \geq t_1 \), we have that \( d_\ell(1, \ell^{-1}_x(k + 1)) \leq \sqrt{3/2} \cdot d_\ell(1, \ell^{-1}_x(k)) \). Such a \( t_1 \) exists: otherwise, there would be arbitrarily large \( k \) such that \( d_\ell(\ell^{-1}_x(k), \ell^{-1}_x(k + 1)) > (\sqrt{3/2} - 1) \cdot d_\ell(1, \ell^{-1}_x(k)) \); because this means that a constant fraction of arbitrarily long subtour lengths is spent on the last absence from \( x \), the quadratic objective for \( x \) would diverge to \( \infty \).

Let \( \hat{\ell} = \max(t_0, t_1) \), and let \( \ell' \) be the periodic schedule repeating \( \ell(1), \ell(2), \ldots, \ell(\hat{\ell}) \).

Fix a point \( x \), and let \( k_x \geq k_0 \) be the number of absence intervals from \( x \) in \( \ell \) that are contained in \( \ell' \). For \( k = 1, \ldots, k_x - 1 \), the absence intervals have lengths \( d_\ell(\ell^{-1}_x(k), \ell^{-1}_x(k + 1)) \). The final absence
interval, in which the tour wraps around, has length at most \( d_\ell(\ell_x^{-1}(k_x), \ell_x^{-1}(k_x+1)) + d_\ell(1, \ell_x^{-1}(1)) \).

In the quadratic objective, we are interested in the square of this length, which is at most
\[
(d_\ell(\ell_x^{-1}(k_x), \ell_x^{-1}(k_x+1)) + d_\ell(1, \ell_x^{-1}(1)))^2 \leq 2d_\ell^2(\ell_x^{-1}(k_x), \ell_x^{-1}(k_x+1)) + 2d_\ell^2(1, \ell_x^{-1}(1)).
\]

Thus, the objective function value for \( x \) under \( \ell' \) is at most
\[
\frac{2d_\ell^2(1, \ell_x^{-1}(1)) + 2d_\ell^2(\ell_x^{-1}(k_x), \ell_x^{-1}(k_x+1)) + \sum_{k=1}^{k_x} d_\ell^2(\ell_x^{-1}(k), \ell_x^{-1}(k+1))}{d_\ell(1, \ell_x^{-1}(k_x))} \\
\leq \frac{2 \sum_{k=0}^{k_x} d_\ell^2(\ell_x^{-1}(k), \ell_x^{-1}(k+1))}{\sqrt{2/3} \cdot d_\ell(1, \ell_x^{-1}(k_x+1))} \\
\leq 3c^{(2)}_x(\ell).
\]

Since this holds for all points \( x \), the proof is complete.

The following lemma compares the optimum TSP cost to the (unweighted) Quadratic TSP objective. It is central to the proof of our main theorem.

Lemma 5 Consider an instance in which all weights are the same: \( w_x = 1 \) for all \( x \). Let \( \text{OPT}_{\text{TSP}} \) be the cost of the optimum TSP tour, and \( C^* \) the cost of the optimum tour with respect to the \( C^{(2)}(\ell) \) objective. Then, \( \text{OPT}_{\text{TSP}} \leq 480 \cdot C^* \).

Proof. By Lemma 4 we can focus, at a loss of a factor of 3, on periodic tours. Consider a periodic tour \( \ell \) of \( \hat{\ell} \) steps, with \( \ell(\hat{t}+1) = \ell(1) \). Let \( q = \text{OPT}_{\text{TSP}}/16 \). We consider two cases, based on whether \( \ell \) contains many long edges or not. Let \( T \) be the set of all “long” time steps \( t \), i.e., time steps \( t \) with \( d(\ell(t), \ell(t+1)) \geq q \).

- In the first case, we assume that \( \sum_{t \in T} d(\ell(t), \ell(t+1)) \geq \frac{1}{10^7} \cdot d_\ell(1, \hat{t}+1) \). Consider an arbitrary point \( x \). Because the tour is absent from \( x \) for at least each time interval corresponding to a step \( t \in T \), the numerator of the quadratic TSP objective for \( x \) is at least

\[
\sum_{t \in T} d^2(\ell(t), \ell(t+1)) \geq |T| \cdot \left( \frac{1}{|T|} \cdot \sum_{t \in T} d(\ell(t), \ell(t+1)) \right)^2 = \frac{(\sum_{t \in T} d(\ell(t), \ell(t+1)))^2}{|T|},
\]

where the inequality was using convexity. The denominator of the quadratic objective is \( d_\ell(1, \hat{t}+1) \leq 10 \sum_{t \in T} d(\ell(t), \ell(t+1)) \), so the quadratic TSP objective for \( x \) is at least

\[
\frac{\sum_{t \in T} d(\ell(t), \ell(t+1))}{10|T|} \geq \frac{q}{10} = \frac{1}{160} \cdot \text{OPT}_{\text{TSP}}.
\]

- Otherwise, we will focus exclusively on time steps \( t \) for which \( d(\ell(t), \ell(t+1)) < q \). Produce sub-tours (paths) of length between \( q \) (inclusive) and \( 2q \) (exclusive) as follows:

1. Remove from \( \ell \) all edges of length strictly more than \( q \). This produces maximal subpaths with the property that each such subpath is immediately preceded and followed by an edge of length strictly greater than \( q \) in \( \ell \). (Except in the case that there were no such long edges; this case is trivial.)
2. Discard all subpaths whose total length is strictly less than $q$.

3. For each remaining subpath, of length at least $q$, partition it greedily into segments of length between $q$ (inclusive) and $2q$ (exclusive). If as a result of this greedy partition, there is a (final) segment of length strictly less than $q$, discard it. Notice that such a partitioning is always possible because each remaining edge has length at most $q$.

Let $K$ be the number of such subtours that are produced, and $E_1, E_2, \ldots, E_K$ the corresponding edge sets; each $E_i$ defines a path (though not necessarily a simple path). We first show that the subtours together comprise at least a $4/5$ fraction of the total length of $\ell$. The reason is the following: whenever a subtour is discarded, its length was strictly less than $q$, and it must have been immediately followed by an edge of length greater than $q$. For each long edge, at most one short subtour is discarded due to this edge, so the total length of discarded edges (long edges and short tours) is at most twice the total length of long edges, which is at most $1/10$ the total length of all edges (because of the current case).

We will prove below that at least one point $x$ must be absent from at least $1/4$ of these subtours $E_i$. Assume for contradiction that every point is in more than $3/4$ of the subtours $E_i$. Each $E_i$ induces a single connected component on the nodes it includes, and has total cost at most $2qK$, and the total length of discarded edges is at most another $qK/2$. Thus, the denominator is less than $5qK/2$, and the quadratic TSP objective is at least $q/10 = \text{OPT}_{\text{TSP}}/160$.

It remains to prove that some point $x$ is absent from at least $1/4$ of the subtours. Assume for contradiction that every point is in more than $3/4$ of the subtours $E_i$. Each $E_i$ induces a single connected component on the nodes it includes, and has total cost at most $2q$. By Lemma 6 below (with $\gamma = \frac{1}{4}$ and $C = 2q$), this implies the existence of a tree spanning all points of cost strictly less than $8q$, and thus a TSP tour of total cost strictly less than $16q$, by shortcutting the Eulerian tour of the spanning tree. This is a contradiction to the definition that $q = \text{OPT}_{\text{TSP}}/16$.

Taking the maximum of both cases, and accounting for the factor of 3 we lost by focusing on a periodic schedule, gives us the claimed bound.

Lemma 6 Let $G$ be an undirected graph with non-negative edge weights. Let $E_1, E_2, \ldots, E_K \subseteq E$ be connected edge sets of cost $c(E_i) < C$, i.e., each $E_i$ induces a single connected component (with the remaining vertices being isolated). Assume that each vertex is part of at least one edge in at least a $1 - \gamma$ fraction of the edge sets $E_i$. Then, $G$ has a spanning tree of total cost strictly less than $\frac{2}{1-2\gamma} \cdot C$.

Proof. Let $v_e^i = 1$ if the edge set $E_i$ contains the edge $e$, and $v_e^i = 0$ otherwise. Let $v_e = \frac{1}{(1-2\gamma)K} \cdot \sum_{i=1}^{K} v_e^i$. We claim that the vector $v$ satisfies the fractional Steiner Tree LP

\[
\sum_{e \in (S, \overline{S})} v_e \geq 1 \quad \text{for all cuts } (S, \overline{S}),
\]

\[
v_e \geq 0 \quad \text{for all edges } e.
\]

Consider any cut $(S, \overline{S})$, and let $x \in S, y \notin S$ be arbitrary. Because at most a $\gamma$ fraction of the edge sets $E_i$ exclude each of $x$ and $y$, at least a $1 - 2\gamma$ fraction of the $E_i$ include both $x$ and $y$. 
By the connectivity assumption for the $E_i$, each such $E_i$ must contain a path from $x$ to $y$, and in particular an edge crossing $(S, \overline{S})$. Thus, we get that

$$\sum_{e \in (S, \overline{S})} v_e = \frac{1}{(1 - 2\gamma) \cdot K} \cdot \sum_{i=1}^{K} \sum_{e \in (S, \overline{S})} v_e^i \geq \frac{1}{(1 - 2\gamma) \cdot K} \cdot (1 - 2\gamma) \cdot K = 1.$$ 

The cost of this fractional solution is at most

$$\sum_{e=(x,y)} d(x, y) \cdot v_e = \sum_{e=(x,y)} d(x, y) \cdot \frac{1}{(1 - 2\gamma) \cdot K} \cdot \sum_{i=1}^{K} v_e^i$$

$$= \frac{1}{(1 - 2\gamma) \cdot K} \cdot \sum_{i=1}^{K} \sum_{e=(x,y)} d(x, y) v_e^i \leq \frac{1}{(1 - 2\gamma) \cdot K} \cdot K \cdot C = C \cdot \frac{1}{1 - 2\gamma}.$$ 

Goemans and Williamson [17] show that the integrality gap of the Steiner Forest LP is at most 2; in particular, this means that there is a spanning tree of all the nodes of total cost strictly less than $\frac{2}{1 - 2\gamma} \cdot C$.

3 Main Result and Proof

In this section, we prove the main result of our paper.

**Theorem 7** There is a polynomial-time algorithm (Algorithm 1) which outputs a periodic tour $\hat{\ell}$ simultaneously guaranteeing that

$$C^{(2)}(\hat{\ell}) \leq O(\log n) \cdot C^{(2)}(\ell_2^*),$$

$$C^{(\infty)}(\hat{\ell}) \leq O(\log n) \cdot C^{(\infty)}(\ell_\infty^*),$$

where $\ell_2^*$ and $\ell_\infty^*$ are the optimal solutions for the respective objective functions.

In other words, $\hat{\ell}$ is simultaneously an $O(\log n)$ approximation for both of the weighted TSP objectives we consider.

First, by standard arguments, rounding all weights down to the nearest power of 2 and then solving the resulting problem loses at most a factor of 2 in the approximation guarantee. We therefore assume from now on that all weights are powers of 2. For each $i = 0, 1, \dots, M$, let $X_i$ be the set of all points of weight $2^{-i}$, and $n_i = |X_i|$ the number of points of weight $2^{-i}$. Let $M$ be largest such that at least one point has weight $2^{-M}$.

The high-level intuition behind our algorithm is the following: in order to achieve a good objective value, each point should be visited with frequency proportional to its weight. However, those visits should also be “roughly evenly spaced;” many visits in short succession, followed by a long absence,
do not help either of the objectives. To build such a tour systematically, we consider individual
tours for points of the same weight; i.e., tours for points of weight $2^{-i}$, for each $i$. These tours must
then be sequenced carefully, so that tours of high-weight points are more frequent.

More precisely, points of weight $2^{-i}$ are partitioned into $2^i$ sets, with a near-optimal tour through
each set. One phase then consists of following one such tour for each $i$. This means that each tour
for a subset of points of weight $2^{-i}$ is followed roughly once every $2^i$ phases, balancing out the lower
weight. A straightforward implementation of this approach would give an $O(M)$-approximation; a
more careful arrangement of phases and subtours improves this bound to $O(\log n)$.

From the preceding discussion, it is clear that an important part of the algorithm is the ability to
partition all points of a given weight $2^{-i}$ into $2^i$ sets so that the tour lengths through all of these
sets are comparable. Thereto, a core subroutine is an approximation algorithm for the Min-Max
Spanning Tree Problem [11], defined as follows:

**Problem 8 (Min-Max Spanning Tree)** Given an edge-weighted graph $G = (V, E)$ and a pa-
rameter $k$, find $k$ edge sets $E_i \subseteq E$ such that each $E_i$ induces a connected subgraph of $G$, and
each node of $G$ is in at least one of the induced subgraphs. Subject to this constraint, minimize
max $\sum c(E_i)$, where the cost of an edge set is the sum of its edge weights.

While this problem is NP-hard, the main result of [11] provides a polynomial-time constant-factor
approximation algorithm:

**Theorem 9 (Theorem 4 of [11])** There is a polynomial-time 4-approximation algorithm for the
Min-Max Spanning Tree problem. In other words, if $E^*_1, E^*_2, \ldots, E^*_k$ is the optimal solution and
$E_1, E_2, \ldots, E_k$ the solution returned by the algorithm,

$$\max_i c(E_i) \leq 4 \cdot \max_i c(E^*_i).$$

Our algorithm is given formally as Algorithm 1

The tours of high-weight points end up in lists $L_i$ with small values of $i$. They are toured expo-
nentially more frequently than tours in lists $L_i$ with large values of $i$. A good example case to keep
in mind in the following analysis is when there are more than $2^i$ points of weight $2^{-i}$ — then, each
list $L_i$ contains exactly the $2^i$ tours of points of weight $2^{-i}$. The more elaborate construction of
lists $L_i$ is needed to achieve the $O(\log n)$ guarantee. The guarantee on the relationship between $L_i$
and points’ weights is captured by the following lemma:

**Lemma 10** For each $i$, each tour $\ell^i_j$ in the list $L_i$, and any point $x$ occurring in $\ell^i_j$, the weight of
$x$ is at most $w_x \leq 2^{-i}$.

**Proof.** There is at most one tour $\ell_1$ of points of weight 1, at most two tours of points of weight
$\frac{1}{2}$; more generally, there are $\theta_i \leq 2^i$ tours of points of weight $2^{-i}$. Therefore, points of weight $2^{-i}$
must appear in one of the first $2^i - 1$ tours, and therefore be in $L_{i'}$ for $i' \leq i$.

---

3Technically, the result of Even et al. guarantees a $(4 + \epsilon)$-approximation for every $\epsilon$. We ignore the $\epsilon$ term here.
The overall cost of the MST is at least ($kB \leq$ have weight at least 2 $B T$). Because the maximum cost of any tree is always found. When the algorithm terminates successfully, it returns a solution of objective value at most 4 $B$, while guaranteeing that $B \leq \text{OPT}_{\text{Min-Max}}$. The analysis in [11] shows that a successful value of $B$ is always found.

Consider a successful run of Algorithm 2. Let $c$ be the number of connected components $G_i$ after removing the heavy edges. Because the MST of all points is connected, it must contain at least $c - 1$ edges of weight at least $B$. Only the trees $T_0^i$ can have weight less than $2B$, while all other trees $T_j^i$ have weight at least $2B$. By the execution of Kruskal’s Algorithm, each MST$_i$ is a subtree of the overall MST, and because each MST$_i$ is broken into pieces to produce the $T_j^i$, each $T_j^i$ is a subtree of the overall MST as well. Therefore, the cost of the overall MST is at least $(k - c) \cdot 2B + (c - 1)B$. For $c < k$, this quantity is obviously lower-bounded by $kB$. When $c = k$, notice that at least one tree $T_j^i$ must have total edge cost at least $\text{OPT}_{\text{Min-Max}} \geq B$ (since the solution is feasible), so the overall cost of the MST is at least $(k - 1)B + \text{OPT}_{\text{Min-Max}} \geq kB$.

Because the maximum cost of any tree $T_j^i$ is at most $4B$, we obtain the claimed bound.
Then, the preceding argument implies that the total length of an absence from $\hat{x}$ distance between any pair of points from $U$.

Next, we bound $\hat{\ell}$. We bound both terms under the sum separately. First, we upper-bound

$$\max_{x,y \in X} d(x,y).$$

Second, we upper-bound $\ell^*_n$. This implies the first part of the theorem because $C^{(2)}(\hat{\ell}) \leq C^{(\infty)}(\hat{\ell}) \leq O(\log n) \cdot C^{(2)}(\ell^*_n)$, and the second part of the theorem because $C^{(\infty)}(\hat{\ell}) \leq O(\log n) \cdot C^{(2)}(\ell^*_n) \leq O(\log n) \cdot C^{(\infty)}(\ell^*_n)$.

We write $C^* = C^{(2)}(\ell^*_n)$. Consider an arbitrary point $\hat{x}$, of weight $2^{-i}$. We want to upper-bound the absence length under $\hat{\ell}$ between any two consecutive visits to $\hat{x}$. By Lemma 10, $\hat{x}$ must appear in a tour in a list $L_i$ for $i' \leq i$. In Line 14, $\lambda_{i'}$ phases elapse before a return to $\hat{x}$. Each of those phases traverses exactly one tour from each list $L_i$, for $i = 0, \ldots, I$. In addition, it requires moving from each tour to the next.

The tours $\ell_j$ are disjoint, and are themselves partitioned disjointly among the lists $L_i$. Let $U_i$ be the set of points $x$ such that a tour in $L_i$ visits $x$. The $U_i$ form a disjoint partition of $X$ (similar, but typically not identical, to the $X_i$). We let $d(U_i, U_j) = \max_{x \in U_i, y \in U_j} d(x,y)$ be the maximum distance between any pair of points from $U_i, U_j$, and $d(L_i) = \max_{\ell_j \in L_i} d(\ell_j)$ the maximum total length of any tour in $L_i$.

Then, the preceding argument implies that the total length of an absence from $\hat{x}$ under $\hat{\ell}$ is upper-bounded by

$$\lambda_{i'0} \sum_{i=0}^{I} \left( d(L_i) + d(U_i, U_{(i+1) \mod I}) \right).$$

We bound both terms under the sum separately. First, we upper-bound $d(U_i, U_{(i+1) \mod I}) \leq \max_{x,y \in X} d(x,y)$; next, we prove that $d(x,y) \leq C^*$. Let $z$ be a point of weight 1. Because the optimum tour must at some point visit $x$, there is an absence interval from $z$ of length at least $2d(x,z)$. Similarly for $y$. Therefore,

$$C^* \geq 1 \cdot \max(2d(x,z), 2d(y,z)) \geq d(x,z) + d(y,z) \geq d(x,y).$$

Next, we bound $\hat{d}(L_i) \leq O(C^*)$. We distinguish two cases. If $\theta_i = n_i$, then each point of weight
2^{i} is in its own spanning tree and tour, so \( d(\ell'_{j}) = 0 \) for all \( j \). In particular \( d(L_{i}) = 0 \), and the claim holds trivially.

Otherwise, \( \theta_{i} = 2^{i} \), and Lemma 11 implies that \( d(T'_{j}) \leq 4 \cdot 2^{-i} \cdot C_{i}^{\text{MST}} \). By the standard shortcutting argument for the Euclidean tour of a tree, \( d(\ell'_{j}) \leq 2d(T'_{j}) \). We now show that \( 2^{-i} \cdot C_{i}^{\text{MST}} \leq O(C^{*}) \). Consider the subtour \( \ell' \) of \( \ell'_{j} \) induced only by points of weight \( 2^{-i} \), skipping all other points. For any point \( x \) of weight \( 2^{-i} \), the objective value is \( 2^{-i} c_{x}^{(2)}(\ell') \leq 2^{-i} c_{x}^{(2)}(\ell'_{2}) \leq C^{*} \). By Lemma 14, \( \max_{x \in X} c_{x}^{(2)}(\ell') \geq \Omega(C_{i}^{\text{MST}}) \), implying the claimed bound.

Substituting all these bounds into (1), we can now upper-bound the length of absence from \( \hat{x} \) by \( \lambda' \cdot (I + 1) \cdot O(C^{*}) \).

Because \( \lambda' \leq 2^{i} \) and the weight of \( \hat{x} \) is \( 2^{-i} \leq 2^{-i'} \), the weighted objective function value for \( \hat{x} \) is at most \( O((I + 1)C^{*}) \). There are \( J \leq n \) tours, so \( I = \lceil \log_{2}(J + 1) \rceil - 1 \leq \log_{2} n \). Finally, we lost only a factor of \( 2 \) due to rounding the weights to powers of \( 2 \). Because \( \hat{x} \) was chosen arbitrarily, we have shown that Algorithm 1 is an \( O(\log n) \) approximation.

Remark 12

- In the proof, we bounded \( I \) by \( \log_{2} n \). The analysis also directly gives a bound of \( O(M) \). When the range of point weights is small (i.e., \( M \) is small), this bound may be better; in other words, the algorithm exploits favorable point weights.

- The constant in big-\( O \) we obtain is large. A more reasonable bound of \( 17 \log_{2} n \) can be obtained for the objective \( C^{(\infty)}(\hat{\ell}) \), by comparing the tour \( \hat{\ell} \) directly to \( \ell' \), instead of \( \ell'_{2} \), and using Lemma 14 instead of Lemma 15. The constant can be further improved to \( 13 \) by using the improved 3-approximation algorithm for Min-Max Spanning Tree due to [20] instead of the 4-approximation of [11].

4 A Reduction from Security Games

We show that the security game mentioned in Section 1.1 can be reduced to the weighted quadratic TSP objective, at a loss of a constant factor in the objective function. In the security game of [19], there are \( n \) targets with weights \( w_{x} \). A defender commits to a distribution over schedules \( \sigma = (\ell, \tau) \), consisting of a tour \( \ell \) and a real-valued time offset \( \tau \). The intuition is that the defender will execute the tour \( \ell \), but delayed by \( \tau \), to create uncertainty. The attacker then chooses one triple \((x, t_{0}, t)\); here, \( x \) is a target, \( t_{0} \) the start time of the attack, and \( t \) its duration. The attack \((x, t_{0}, t)\) is successful if the defender does not visit target \( x \) in the interval \([t_{0}, t_{0} + t]\) under \( \sigma \). The defender’s cost (and the attacker’s utility) is

\[
U(\sigma, (x, t_{0}, t)) = \begin{cases} 
  w_{x} \cdot t & \text{if the attack is successful}, \\
  0 & \text{otherwise}.
\end{cases}
\]

\footnote{In [19], all targets are at unit distance from each other. Our main goal here is to leverage our approximation algorithm for the weighted Quadratic TSP objective to obtain (weaker) guarantees for the version of the security game in which there is an arbitrary metric on the targets.}

\footnote{To be precise, the model of [19] is entirely defined in continuous time, as a mapping \( \sigma : \mathbb{R}_{\geq 0} \rightarrow X \cup \{\perp\} \), with \( \perp \) denoting that the defender is in transit between targets. Such schedules allow staying at points for a positive amount of time, something that we do not allow in this paper.}
The defender chooses a mixed strategy (distribution $\Sigma$) over schedules $\sigma$ so as to minimize the attacker’s expected utility.

Lemma B.3 of [19] shows that without loss of generality, the optimal mixed defender strategy $\Sigma^*$ is shift-invariant in the following sense: for each $x, t$, the distribution of return times to target $x$ starting from time 0 is the same as starting from time $t$.

**Lemma 13 (Lemma B.3 of [19])** There is a shift-invariant optimal mixed defender strategy.

Lemma 13 allows us to assume w.l.o.g. that the attacker chooses $t_0 = 0$ (or any other fixed time), since no time is better for him than any other. This will be useful in the subsequent analysis.

We now lower-bound the objective function in terms of the cost functions $c_x^{(2)}(\ell)$ for targets $x$. The first step is to use Proposition 3.2 of [19], which does not rely on uniform distances between targets.

**Lemma 14 (Proposition 3.2 of [19])** Consider any shift-invariant strategy $\Sigma$ and target $x$, and let $T_x$ be the expected time until the next return of the defender to target $x$ after time 0 (under $\Sigma$). By choosing an attack duration of $t = \frac{T_x}{2}$, the attacker can achieve expected utility at least $w_x \cdot T_x^3$.

Our next goal is to lower-bound $T_x$. Suppose that the attacker draws his start time $t_0$ uniformly at random from the interval $[0, \hat{t}]$, for some value of $\hat{t}$. Consider the distribution of the defender’s next return time to $x$ after time $t_0$. By shift-invariance, this distribution is the same for all possible draws $t_0$. We lower-bound the expected return time by pretending that the defender returns to all targets at time $\hat{t}$. Consider a fixed (i.e., non-random) defender schedule $\sigma = (\ell, \tau)$. The expected return time to target $x$ under $\sigma$ for a uniformly random attack start time $t_0$ is exactly half the sum of squares of interval lengths between consecutive visits to $x$ over $[0, \hat{t}]$. Denote this quantity by $s(\sigma, x, \hat{t})$. We can lower-bound $T_x \geq \mathbb{E}_{\sigma \sim (\ell, \tau) \sim \Sigma} \left[ \frac{1}{2} \cdot s(\sigma, x, \hat{t}) \right]$. By taking the limit of $\hat{t} \to \infty$, and noticing that $\lim_{\hat{t} \to \infty} \frac{1}{\hat{t}} \cdot s(\sigma, x, \hat{t}) = c_x^{(2)}(\ell)$, we get that $T_x \geq \mathbb{E}_{\sigma \sim (\ell, \tau) \sim \Sigma} \left[ c_x^{(2)}(\ell) \right]$. Overall, we have shown that the attacker’s expected utility against a mixed defender strategy $\Sigma$ is at least $\Omega \left( \max_x \mathbb{E}_{\sigma \sim (\ell, \tau) \sim \Sigma} \left[ w_x c_x^{(2)}(\ell) \right] \right)$.

Below, we prove Lemma 15 which shows that we can approximate this objective with a single periodic tour and a uniformly random offset.

**Lemma 15** For every mixed defender strategy $\Sigma$, there is a single periodic tour $\ell$ with $C^{(2)}(\ell) \leq O(1) \cdot \max_x \left( \mathbb{E}_{\sigma \sim (\ell, \tau) \sim \Sigma} \left[ w_x c_x^{(2)}(\ell) \right] \right)$.

We can now finish the reduction by upper-bounding the defender’s cost under the periodic tour from Lemma 15.

**Lemma 16** For any periodic tour $\ell$ with uniformly random offset $\tau$, by attacking target $x$, the attacker can obtain expected utility at most $w_x \cdot \frac{1}{2} \cdot c_x^{(2)}(\ell)$.

**Proof.** Because the offset $\tau$ is uniformly random, the strategy is shift-invariant, and the attacker attacks at time 0 without loss of generality. Even if he exactly knew when after time 0 the defender would return next, the attacker could not in expectation attack longer than the expected time until
the defender’s next return. For a periodic schedule \( \ell \), the expected time until the defender’s next return is \( \frac{1}{2} \cdot c^{(2)}_x(\ell) \).

Thus, we have shown that the objective value of the optimum finite periodic tour \( \ell \) is within a constant factor of the optimum randomized defender strategy. Therefore, any approximation guarantee for the \( O^{(2)}(\ell) \) objective for finite periodic tours carries over — at the loss of a constant factor — to the security game objective from [19]. In particular, using the result from Theorem 7, we obtain a polynomial-time \( O(\log n) \) approximation algorithm for finding a defender strategy minimizing a rational attacker’s expected utility.

**Proof of Lemma 15.** By Lemma 14 and the subsequent discussion, the attacker’s utility from attacking \( x \) against a mixed defender strategy \( \Sigma \) is \( \Omega \left( \sum_{\ell \in \mathcal{L}} p_{\ell} \cdot c^{(2)}_x(\ell) \right) \). By Lemma 14 at a cost of a factor 3 in the \( c^{(2)}_x(\ell) \), we can ensure that all tours \( \ell \) in the support of \( \Sigma \) are periodic.

Then, \( \Sigma \) is of the following form: (1) Choose a (finite, periodic) tour \( \ell \) from some distribution. Let \( D \) be the total length of executing the finite tour once. (2) Shift the starting point of the tour to a uniformly random point in the (continuous) interval \( [0, D] \). No matter where the tour is starting, the total time of \( \hat{\ell} \) is

\[
\min \sum_{i=1}^{K} p_i \cdot c^{(2)}_x(\ell_i)
\]

where \( p_i \) are the probabilities of choosing \( \ell_i \). Conditioned on \( \hat{\ell} \), the expected absence length from \( x \) is

\[
\mathbb{E}_{t \sim \Sigma} \left[ c^{(2)}_x(\ell) \right] \geq \sum_{i=1}^{K} p_i c^{(2)}_x(\ell_i).
\]

Now consider \( c^{(2)}_x(\hat{\ell}) \). It is exactly the expected length of the absence interval from \( x \) for a uniformly randomly chosen time \( t \in [0, D] \).

The total time of \( \hat{\ell} \) is

\[
(N_i - 2)D_i c^{(2)}_x(\ell_i) + 2(2D)^2 \leq c^{(2)}_x(\ell_i) + \frac{8D^2}{N_i \cdot D_i} \leq \frac{q c^{(2)}_x(\ell_i)}{p_i} \leq 2c^{(2)}_x(\ell_i).
\]
The probability that the random time $t$ falls into an interval devoted to $\ell_i$ is

$$\frac{N_i D_i}{\sum_j N_j D_j} \leq \frac{2 K p_i \cdot D}{\sum_j K p_j \cdot D} = \frac{2 p_i}{\sum_j p_j} \leq 4 p_i,$$

because the sum of probabilities $p_j$ is at least $\frac{1}{4}$. In summary, the expected absence length from $x$ is at most $\sum_i (4 p_i) \cdot (2 c_x^{(2)}(\ell_i)) \leq 8 E_{(\ell, D)} \left[ c_x^{(2)}(\ell) \right].$

5 Conclusion

Motivated in part by applications in security games such as anti-burglary or anti-poaching patrols, as well as an economic view of the traveling salesman problem, we defined a class of weighted traveling salesman problems in which the goal is to visit the points of a metric space repeatedly, while minimizing the weighted duration of absences from the points. We gave a combinatorial algorithm that simultaneously achieves an $O(\log n)$ approximation guarantee for two natural objective functions.

One can generalize the cost functions for individual points to

$$c_x^{(p)}(\ell) = \lim_{k \to \infty} \frac{\sum_{k'=0}^{k} (d(\ell^{-1}(k'), \ell^{-1}(k' + 1)))^p}{\sum_{k'=0}^{k} (d(\ell^{-1}(k'), \ell^{-1}(k' + 1)))^{p-1}}.$$

Then, the two special cases we analyzed are $p = 2$ and $p \to \infty$. Because for any fixed $x, \ell$, the cost function $c_x^{(p)}(\ell)$ is monotone non-decreasing in $p$, so is $C^{(p)}(\ell)$. We showed in Section 3 that the algorithm’s tour $\hat{\ell}$ satisfies $C^{(\infty)}(\hat{\ell}) \leq O(\log n) \cdot C^{(2)}(\ell^*_2)$. This immediately implies that Algorithm 1 is an $O(\log n)$ approximation for $C^{(p)}(\ell)$ for all $p \geq 2$. However, we are not aware of any natural motivation for studying the objective for $p \in (2, \infty)$. For $p < 2$, the objective function suffers, in special cases, from rewarding repeated visits to the same point (without traveling any distance in between), which is why we did not study it.

The most obvious open question left by our work is whether the approximation guarantee can be improved to a constant. For many special cases, natural modifications of Algorithm 1 do give constant-factor approximations. It seems plausible that a more careful interleaving of subtours could lead to such an improved guarantee.

Instead of minimizing the maximum of the weighted point objectives, one could study a weighted sum. Such an objective would be motivated if a need for the salesman’s product arises with a uniformly randomly chosen person, and we want to minimize the expected time to serve the demand. First notice that in this version, the weights can be assumed to be uniform, as a point of weight $w$ can be replaced by $w$ points of weight 1, without any change in the objective. The resulting problem becomes more similar to the well-known Traveling Repairman (a.k.a. Minimum Latency) problem \cite{anderson85, christofides76, toth92}. However, even for the (unweighted) sum version of our problem, the standard techniques for the Minimum Latency problem do not appear to carry over, for reasons similar to those discussed in Section 1.3. Algorithm 1 in the case of no weights simply returns a TSP tour of all points. It is not difficult to construct examples where this can be polynomially
far from optimal. Hence, studying the sum (instead of maximum) version of the problem is an interesting direction for future work.

Another natural question, motivated by the security game application, is whether one can also obtain good approximation guarantees for the same objective in the presence of multiple “salesmen;” for example, national parks will typically have multiple anti-poaching patrols simultaneously. The work on Min-Max Tree Covers \[11, 20\] we used as a key tool was motivated by a similar goal of dividing points between multiple TSP tours. See also \[2, 28\], and \[12\] for a discussion of the \(k\)-Traveling Repairman Problem.

We studied two natural cost minimization versions of the weighted TSP. In recent work, motivated by some of the same application domains (such as poaching prevention), Immorlica and Kleinberg \[18\] defined a natural prize-collecting problem with a time component. In their problem, distances between “points” are uniform. Each point \(x\) has an increasing concave reward function \(f_x\). When a point is visited after an absence interval of length \(t\), it yields a reward of \(f_x(t)\). This objective also naturally motivates visiting more valuable points more frequently, and spacing out visits roughly evenly. It is an interesting question whether the techniques in our paper can be adapted to obtain provable approximation guarantees for the prize-collecting version of \[18\] under non-uniform distances. One possible difficulty is that while in our cost-minimization formulation, each point must be visited infinitely often to prevent infinite cost, the prize-collecting version can simply omit some points if they are far away and/or provide little reward.

Acknowledgments

We would like to thank Shaddin Dughmi, Bobby Kleinberg, Kamesh Munagala, Debmalya Panigrahi, David Shmoys, and Omer Tamuz for useful discussions, and anonymous reviewers for useful feedback.

References

[1] Shoshana Anily, Celia A. Glass, and Refael Hassin. The scheduling of maintenance service. *Discrete Applied Mathematics*, pages 27–42, 1997.
[2] Esther M. Arkin, Refael Hassin, and Asaf Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59(1):1–18, 2006.
[3] Sanjeev Arora and George Karakostas. Approximation schemes for minimum latency problems. In *Proc. 31st ACM Symp. on Theory of Computing*, pages 688–693, 1999.
[4] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia. Commitment without regrets: Online learning in Stackelberg security games. In *Proc. 16th ACM Conf. on Economics and Computation*, pages 61–78, 2015.
[5] Nikhil Bansal, Avrim Blum, Shuchi Chawla, and Adam Meyerson. Approximation algorithms for deadline-TSP and vehicle routing with time-windows. In *Proc. 36th ACM Symp. on Theory of Computing*, 2004.
[6] Amotz Bar-Noy, Randeep Bhatia, Joseph (Seffi) Naor, and Baruch Schieber. Minimizing service and operation costs of periodic scheduling. *Mathematics of Operations Research*, 27:518–544, 1998.

[7] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proc. 8th Intl. Conf. on Autonomous Agents and Multiagent Systems*, pages 57–64, 2009.

[8] Avrim Blum, Prasad Chalasani, Don Coppersmith, William R. Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 163–171, 1994.

[9] Hans-Joachim Bockenhauer, Juraj Hromkovic, Joachim Kneis, and Joachim Kupke. The parameterized approximability of TSP with deadlines. *Theory of Computing Systems*, 41:431–444, 2007.

[10] Deeparnab Chakrabarty and Chaitanya Swamy. Facility location with client latencies: LP-based techniques for minimum-latency problems. *Mathematics of Operations Research*, 41(3):865–883, 2016.

[11] Guy Even, Naveen Garg, Jochen Könemann, R. Ravi, and Amitabh Sinha. Min-max tree covers of graphs. *Oper. Res. Lett.*, 32(4):309–315, 2004.

[12] Jittat Fakcharoenphol, Chris Harrelson, and Satish Rao. The $k$-traveling repairman problem. In *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms*, pages 655–664, 2003.

[13] Fei Fang, Thanh H. Nguyen, Rob Pickles, Wai Y. Lam, Gopalasamy R. Clements, Bo An, Amandeep Singh, Brian C. Schwedock, Milind Tambe, and Andrew Lemieux. PAWS — a deployed game-theoretic application to combat poaching. *AI Magazine*, 38(1):23–36, 2017.

[14] Fei Fang, Peter Stone, and Milind Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *Proc. 24th Intl. Joint Conf. on Artificial Intelligence*, pages 2589–2595, 2015.

[15] Naveen Garg. A 3-approximation for the minimum tree spanning $k$ vertices. In *Proc. 37th IEEE Symp. on Foundations of Computer Science*, pages 302–309, 1996.

[16] Michel X. Goemans and Jon M. Kleinberg. An improved approximation ratio for the minimum latency problem. In *Proc. 7th ACM-SIAM Symp. on Discrete Algorithms*, pages 152–158, 1996.

[17] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.

[18] Nicole Immorlica and Robert D. Kleinberg. Recharging bandits. In *Proc. 59th IEEE Symp. on Foundations of Computer Science*, 2018.

[19] David Kempe, Leonard J. Schulman, and Omer Tamuz. Quasi-regular sequences and optimal schedules for security games. In *Proc. 29th ACM-SIAM Symp. on Discrete Algorithms*, 2018.

[20] M. Reza Khani and Mohammad R. Salavatipour. Improved approximation algorithms for min-max tree cover and bounded tree cover problems. *Algorithmica*, 69(2):443–460, 2014.
[21] Eugene L. Lawler, Jan Karel Lenstra, Alexander H. G. Rinnooy Kan, and David B. Shmoys, editors. *The Traveling Salesman Problem: A guided tour of Combinatorial Optimization*. Wiley, 1991.

[22] Jan Karel Lenstra and Alexander H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–227, 1981.

[23] Christos Papadimitriou and Mihalis Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, 1993.

[24] Ian Post and Chaitanya Swamy. Linear programming-based approximation algorithms for multi-vehicle minimum latency problems. In *Proc. 26th ACM-SIAM Symp. on Discrete Algorithms*, pages 512–531, 2015.

[25] Milind Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[26] Yevgeniy Vorobeychik, Bo An, and Milind Tambe. Adversarial patrolling games. In *Proc. 11th Intl. Conf. on Autonomous Agents and Multiagent Systems*, pages 1307–1308, 2012.

[27] Haifeng Xu. The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In *Proc. 17th ACM Conf. on Economics and Computation*, pages 497–514, 2016.

[28] Xiaoming Zheng, Sven Koenig, David Kempe, and Sonal Jain. Multi-robot forest coverage for weighted and unweighted terrain. *IEEE Transactions on Robotics*, 26(6):1018–1031, 2010.