The traveling salesman problem for lines and rays in the plane

Adrian Dumitrescu

Abstract

In the Euclidean TSP with neighborhoods (TSPN), we are given a collection of $n$ regions (neighborhoods) and we seek a shortest tour that visits each region. In the path variant, we seek a shortest path that visits each region. In this note we present several linear-time approximation algorithms with improved ratios for these problems for two cases of neighborhoods that are (infinite) lines, and respectively, (half-infinite) rays.

Keywords: Traveling salesman problem with neighborhoods, lines, rays, approximation algorithm.

1 Introduction

In the Euclidean traveling salesman problem (TSP), given a set of points in the plane, one seeks a shortest tour that visits each point. If now each point is replaced by a (possibly disconnected) region, one obtains the so-called TSP with neighborhoods (TSPN), first studied by Arkin and Hassin [1]. A tour for a set of neighborhoods is also referred to as a TSP tour. A path for a set of neighborhoods is also referred to as a TSP path.

For the case of neighborhoods that are (infinite) straight lines, an optimal tour can be computed in $O(n^5)$ time [2, 8, 9] (see also [5]), and a $\sqrt{2}$-approximation can be computed in $O(n)$ time [5]. For the case of neighborhoods that are (half-infinite) rays, no polynomial time algorithm is known for computing an optimal tour, and a $\sqrt{2}$-approximation can be computed in $O(n)$ time [5]. In this paper we present several linear-time approximation algorithms with improved ratios for these problems. The obvious motivation is to provide faster and conceptually simpler algorithmic solutions. As mentioned above, while for the case of rays no polynomial time algorithm is known, for the case of lines, the only known algorithms reduce the problem of computing an optimal tour of the lines to that of computing an optimal watchman tour in a simple polygon for which the known algorithms are quite involved and rather slow for large $n$ [2, 8, 9].

In this paper, we present four improved linear-time approximation algorithms for TSP, for two cases of neighborhoods, that are straight lines, and respectively, straight rays in the plane. Our algorithms are all based on solving low-dimensional linear programs. Our results are summarized in Table 1.

Theorem 1 Given a set of $n$ lines in the plane: (i) A TSP tour that is at most $1.28$ times longer than the optimal can be computed in $O(n)$ time. (ii) A TSP path that is at most $1.61$ times longer than the optimal can be computed in $O(n)$ time.

For lines, the previous best approximations obtained in linear time were $\sqrt{2} \approx 1.41$ and $2\sqrt{2} \approx 2.82$, respectively [5].

Theorem 2 Given a set of $n$ rays in the plane: (i) A TSP tour that is at most $1.28$ times longer than the optimal can be computed in $O(n)$ time. (ii) A TSP path that is at most $2.55$ times longer than the optimal can be computed in $O(n)$ time.

For rays, the previous best approximation for tours was $\sqrt{2} \approx 1.41$ [5] (obtained also in linear time, however this was the only approximation known), while for paths there was no approximation known.

Preliminaries. For a polygon $P$, let $\text{per}(P)$ denote its perimeter. For a rectangle $Q$, let $\text{long}(Q)$ denote the length of a longest side of $Q$. For a ray $\rho$, let $\ell(\rho)$ denote its supporting line. Let $\mathcal{L}$ be a given set of $n$ lines, and let $T^*(\mathcal{L})$ be an optimal tour (circuit) of the lines in $\mathcal{L}$. Let $\mathcal{R}$ be a given set of $n$ rays, and let $T^*(\mathcal{R})$ be an optimal tour (circuit) of the rays in $\mathcal{R}$.

Following the terminology from [3, 7], a polygon is an intersecting polygon of a set of regions in the plane if every region in the set intersects the interior or the boundary of the polygon. The problem of computing a minimum-perimeter intersecting polygon (MPIP) for the case when the regions are line segments was first considered by Rappaport [7] in 1995. As of now, MPIP (for line segments) is not known to be polynomial, nor it is known to be NP-hard.

Since both lines and rays are infinite (i.e., unbounded regions) finding optimal tours $T^*(\mathcal{L})$ and $T^*(\mathcal{R})$ are equivalent to finding minimum-perimeter intersecting polygons (MPIPs) for $\mathcal{L}$ and $\mathcal{R}$ respectively. We can assume without loss of generality that not all lines in $\mathcal{L}$ are concurrent at a common point (this can be easily checked in linear time), thus $\text{per}(T^*(\mathcal{L})) > 0$. The
same assumption can be made for the rays in \( \mathcal{R} \), thus \( \text{per}(T^*(\mathcal{R})) > 0 \).

The following two facts are easy to prove; see also [3, 4, 7].

**Observation 1** If \( P_1 \) is an intersecting polygon of \( \mathcal{L} \), and \( P_1 \) is contained in another polygon \( P_2 \), then \( P_2 \) is also an intersecting polygon of \( \mathcal{L} \). The same statement holds for \( \mathcal{R} \).

**Observation 2** \( T^*(\mathcal{L}) \) is a convex polygon with at most \( n \) vertices. Similarly, \( T^*(\mathcal{R}) \) is a convex polygon with at most \( n \) vertices.

A key fact in the analysis of the approximation algorithm is the following lemma. This inequality is implicit in [10]; a more direct proof can be found in [3].

**Lemma 3** [3, 10]. Let \( P \) be a convex polygon. Then the minimum-perimeter rectangle \( Q \) containing \( P \) satisfies \( \text{per}(Q) \leq \frac{1}{2} \text{per}(P) \).

## 2 TSP for lines

In this section we prove Theorem 1.

**TSP tours.** We present a \( \frac{2}{3}(1+\varepsilon) \)-approximation algorithm for computing a minimum-perimeter intersecting polygon of a set \( \mathcal{L} \) of \( n \) lines, running in \( O(n) \) time. If we set \( \varepsilon = 1/200 \), we get the approximation ratio 1.28. For technical reasons (see below) we choose \( \varepsilon \in [1/300, 1/200] \) uniformly at random, and the approximation ratio remains 1.28. The algorithm combines ideas from [3, 4, 5]. As in [3], we first use the fact (guaranteed by Lemma 3) that every convex polygon \( P \) is contained in some rectangle \( Q = Q(P) \) that satisfies \( \text{per}(Q) \leq \frac{1}{2} \text{per}(P) \). In particular, this holds for the optimal tours \( T^*(\mathcal{L}) \) and \( T^*(\mathcal{R}) \). Then we use linear programming to compute a \( (1+\varepsilon) \)-approximation for the minimum-perimeter intersecting rectangle of \( \mathcal{L} \) (as in [3]; see also[5]).

**Algorithm A1.**

Let \( m = \lceil \frac{\pi}{\varepsilon} \rceil \). For each direction \( \alpha_i = i \cdot 2\varepsilon \), \( i = 0, 1, \ldots, m - 1 \), compute a minimum-perimeter intersecting rectangle \( Q_i \) of \( \mathcal{L} \) with orientation \( \alpha_i \). Return the rectangle with the minimum perimeter over all \( m \) directions.

We now show how to compute the rectangle \( Q_i \) by linear programming. By a suitable rotation (by an angle \( \alpha_i \)) of the set \( \mathcal{L} \) of lines in each iteration \( i \geq 1 \), we can assume that the rectangle \( Q_i \) is axis-parallel. This can be obtained in \( O(n) \) time (per iteration). Let \( \{q_1, q_2, q_3, q_4\} \) be the four vertices of \( Q_i \) in counterclockwise order, starting with the lower leftmost corner as in Figure 2. As in [5], let \( \mathcal{L} = \mathcal{L}^+ \cup \mathcal{L}^- \) be the partition of \( \mathcal{L} \) into lines of negative slope and lines of positive slope.

By setting \( \varepsilon \in [1/300, 1/200] \) uniformly at random, in each iteration \( i \), with probability 1 there are no vertical lines in (the rotated set) \( \mathcal{L} \).

Observe (as in [5]), that a line in \( \ell \in \mathcal{L}^+ \) intersects \( Q_i \) if and only if \( q_2 \) and \( q_4 \) are separated by \( \ell \) (points on \( \ell \) belong to both sides of \( \ell \)). Similarly, a line in \( \ell \in \mathcal{L}^- \) intersects \( Q_i \) if and only if \( q_1 \) and \( q_3 \) are separated by \( \ell \). The objective of minimum perimeter is naturally expressed as a linear function. The resulting linear program has \( 4n \) variables \( x_1, x_2, y_1, y_2 \) for the rectangle \( Q_i = [x_1, x_2] \times [y_1, y_2] \), and \( 2n + 2 \) constraints.

\[
\begin{align*}
\text{minimize} & \quad 2(x_2 - x_1) + 2(y_2 - y_1) \\
\text{subject to} & \\
& y_2 \geq ax_1 + b, \quad \ell : y = ax + b \in \mathcal{L}^+ \\
& y_1 \leq ax_2 + b, \quad \ell : y = ax + b \in \mathcal{L}^+ \\
& y_1 \leq ay_1 + b, \quad \ell : y = ax + b \in \mathcal{L}^- \\
& y_2 \geq ay_2 + b, \quad \ell : y = ax + b \in \mathcal{L}^- \\
& x_1 \leq x_2 \\
& y_1 \leq y_2
\end{align*}
\]

Let \( Q^* \) be a minimum-perimeter intersecting rectangle of \( \mathcal{L} \). To account for the error made by discretization, we need the following easy fact; see [3, Lemma 2].

**Lemma 4** [3]. There exists an \( i \in \{0, 1, \ldots, m - 1\} \) such that \( \text{per}(Q_i) \leq (1 + \varepsilon) \text{per}(Q^*) \).

By Observations 1 and 2, and by Lemmas 3 and 4, the algorithm A1 computes a tour that is at most 1.28 longer than the optimal. The algorithm solves a constant number of 4-dimensional linear programs, each in \( O(n) \) time [6]. The overall time is \( O(n) \).

**TSP paths.** The key to the improvement is offered by the following.

**Observation 3** Let \( Q \) be a rectangle. Then \( Q \) intersects a set of lines \( \mathcal{L} \) if and only if any three sides of \( Q \) intersect \( \mathcal{L} \).

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Tour (old ratio)</th>
<th>Tour (new ratio)</th>
<th>Path (old ratio)</th>
<th>Path (new ratio)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines</td>
<td>( \sqrt{2} = 1.41 \ldots )</td>
<td>1.28</td>
<td>2( \sqrt{2} = 2.82 \ldots )</td>
<td>1.61</td>
</tr>
<tr>
<td>Rays</td>
<td>( \sqrt{2} = 1.41 \ldots )</td>
<td>1.28</td>
<td>-</td>
<td>2.55</td>
</tr>
</tbody>
</table>

Table 1: Old and new approximation ratios. No approximation for paths on rays was reported in [5].
Proof. Fix any three sides of $Q$: $\{s_1, s_2, s_3\}$ (each $s_i$ is a closed segment). Now if $\ell$ is a line intersecting $Q$, then $\ell$ intersects at least two sides of $Q$, hence it intersects at least one element of $\{s_1, s_2, s_3\}$, as required. $\square$

The next lemma gives a quantitative upper bound on the total length of three shorter sides of a rectangle enclosing a curve.

Lemma 5 Any open curve of length $L$ can be included in a rectangle $Q$, so that $\text{per}(Q) - \text{long}(Q) \leq 6(2 - \sqrt{3})L = 1.6076\ldots L$.

Proof. Let $\gamma$ be an open curve of length $|\gamma| = L$, and let $a, b \in \gamma$ be a diameter pair. We can assume w.l.o.g. that $ab$ is a horizontal segment of unit length $|ab| = 1$, where $a = (0, 0)$, $b = (1, 0)$. The two points $a$ and $b$ partition $\gamma$ into three parts, $\gamma_i$, $i = 1, 2, 3$; ($\gamma_1$ or $\gamma_3$ may be empty).

1. $\gamma_1$: from one endpoint of $\gamma$ to $a$.
2. $\gamma_2$: from $a$ to $b$.
3. $\gamma_3$: from $b$ to the other endpoint of $\gamma$.

We show that $\gamma$ can be included in an axis-parallel rectangle $Q$, whose vertical sides are incident to the points $a$ and $b$ respectively, and satisfying the claimed inequality.

Let $l$ and $h$ be the lowest and resp. highest point of $\gamma$. Let $y_1, y_2 \geq 0$ be the $y$-coordinates of $h$ and resp. $l$ in absolute value. Write $y = y_1 + y_2$. Since $\gamma$ has unit diameter, we have $y \leq 1$. Set $Q = [0, 1] \times [y(l), y(h)]$, and observe that $\gamma$ is contained in $Q$. Clearly,

$$\text{per}(Q) - \text{long}(Q) = 1 + 2(y_1 + y_2) = 1 + 2y.$$

We distinguish three cases, depending on which part of $\gamma$ the two extreme points $l$ and $h$ are located. We can assume w.l.o.g. (by symmetry) that $h \in \gamma_1$ or $h \in \gamma_2$. See Figure 1.

Case 1: $h \in \gamma_1$ and $l \in \gamma_3$. By the triangle inequality we have

$$L = |\gamma| = |\gamma_1| + |\gamma_2| + |\gamma_3| \geq 1 + y_1 + y_2 = 1 + y.$$

It follows that

$$\text{per}(Q) - \text{long}(Q) = 1 + 2y \leq 3(1 + y)/2 \leq 3L/2,$$

which is better than required.

Case 2: $h, l \in \gamma_2$. We can assume w.l.o.g. that $\gamma_2$ reaches point $h$ before reaching point $l$. Let $p \in [a, b]$ be a point where $\gamma_2$ crosses the segment $ab$. Let $x_1 = |ap|/2$, and $x_2 = |pb|/2$, so that $x_1 + x_2 = 1/2$, and $x_1, x_2 \geq 0$. By the reflection principle and the triangle inequality, we have

$$L \geq |\gamma_2| \geq 2\sqrt{x_1^2 + y_1^2} + 2\sqrt{x_2^2 + y_2^2}.$$

Via the Cauchy-Schwarz inequality we derive

$$\sqrt{x_1^2 + y_1^2} + \sqrt{x_2^2 + y_2^2} \geq \sqrt{(x_1 + x_2)^2 + (y_1 + y_2)^2} = \sqrt{1 + y^2},$$

hence

$$L \geq 2\sqrt{1 + y^2} = \sqrt{1 + 4y^2}.$$

It can be checked by squaring and direct expansion that

$$1 + 2y \leq \sqrt{2}\sqrt{1 + 4y^2},$$

hence

$$\text{per}(Q) - \text{long}(Q) = 1 + 2y \leq \sqrt{2}\sqrt{1 + 4y^2} \leq \sqrt{2}L,$$

which is again, better than required.

Case 3: $h \in \gamma_2$ and $l \in \gamma_3$ (another case, when when $h \in \gamma_2$ and $l \in \gamma_3$ is analogous, and leads to the same analysis.) By the reflection principle and the triangle inequality, we have

$$L \geq |\gamma_2| + |\gamma_3| \geq 2\sqrt{1 + y_1^2} + 2\sqrt{1 + y_2^2} = \sqrt{4y_1^2 + 1} + y_2.$$

It remains to show that under the constraints $y_1, y_2 \geq 0$ and $y_1 + y_2 \leq 1$, we have

$$\sqrt{4y_1^2 + 1} + y_2 \leq 6(2 - \sqrt{3}).$$

Let

$$f(y_1, y_2) = \sqrt{4y_1^2 + 1} + y_2.$$

Assume that $y_1$ is fixed, and let $g(y_2) = f(y_1, y_2)$. The derivative $g'(y_2)$ has the same sign as the quadratic function $16y_1^2 - 4y_1 + 3$ which is strictly positive for $y_1 \geq 0$. Therefore $g'(y_2) > 0$ for $y_2 \geq 0$, thus $f(y_1, y_2)$ is an increasing function of $y_2$, and consequently

$$f(y_1, y_2) \leq f(y_1, 1 - y_1) = \frac{3}{\sqrt{4y_1^2 + 1} + 1 - y_1}.$$

Let now $y_1$ be variable, and define

$$k(y_1) = \sqrt{4y_1^2 + 1} + 1 - y_1.$$

The derivative

$$k'(y_1) = -1 + \frac{4y_1}{\sqrt{4y_1^2 + 1}}$$

vanishes at $y_1 = 1/(2\sqrt{3})$, and is negative for $y_1$ smaller than this value, and positive for $y_1$ larger than this value. We conclude that $k(y_1)$ attains its minimum at
this value, and correspondingly \( f(y_1, 1 - y_1) \) attains its maximum at this value:

\[
\hat{f}(y_1, y_2) \leq f(y_1, 1 - y_1) \leq f\left(\frac{1}{2\sqrt{3}}, 1 - \frac{1}{2\sqrt{3}}\right) = 6(2 - \sqrt{3}).
\]

The bottleneck in the analysis is therefore the third case. The proof of Lemma 5 is complete. \( \square \)

To compute a TSP path for a set of \( n \) lines, we use the algorithm \( \text{A2} \) we describe next. This algorithm is similar to algorithm \( \text{A1} \), and computes a rectangle in each direction from a given sequence. The only difference in the linear program is that instead of the angle in each direction from a given sequence, the two coordinates axes, and so the number of directions \( m \), from algorithm \( \text{A1} \), is \( m = \lfloor \frac{n}{2} \rfloor \) in algorithm \( \text{A2} \). Let now \( Q^* \) be an intersecting rectangle of \( L \) with minimum sum of the lengths of three sides. Analogous to Lemma 4 we have

**Lemma 6** There exists an \( i \in \{0, 1, \ldots, m - 1\} \) such that

\[
\text{per}(Q_i) - \text{long}(Q_i) \leq (1 + \varepsilon)(\text{per}(Q^*) - \text{long}(Q^*)).
\]

By Lemma 5 and Lemma 6 the approximation ratio is \( 6(2 - \sqrt{3})(1 + \varepsilon) \), and we set \( \varepsilon = 1/1000 \) (or slightly smaller, as before), to obtain the approximation ratio 1.61. This completes the proof of Theorem 1.

**Remark.** The approximation ratio for TSP paths in Theorem 1, 1.61, is determined by the upper bound \( \hat{f}(y_1, y_2) \leq 6(2 - \sqrt{3}) \) in (1). By taking into consideration that \( \gamma \) lies in the intersection of the disks of unit radius centered at \( a \) and \( b \), one can replace the function \( f(y_1, y_2) \) in (2) by the function

\[
\tilde{f}(y_1, y_2) = \frac{1 + 2(y_1 + y_2)}{\sqrt{4y_1^2 + 1 + \sqrt{2 - 2\sqrt{1 - y_2^2}}}}.
\]

We omit the details of this derivation. A numeric calculation indicates that subject to the same constraints on the variables, we have \( \tilde{f}(y_1, y_2) \leq 1.569 \), and this would lead to a slightly better upper bound, 1.57, for the same algorithm. We haven’t found however an analytical proof for the above upper bound on \( \tilde{f}(y_1, y_2) \).

**3 TSP for rays**

As noted in [5]: If the lines are replaced by line segments the problem of finding an optimal tour becomes NP-hard. Should the lines be replaced by rays, we get a variant of the problem that lies somewhere in between the variant for lines and that for line segments, and whose complexity is open.

In this section we prove Theorem 2. The algorithm \( \text{A1} \) from Section 2 can be adapted to compute a \( \frac{1}{3} (1 + \varepsilon) \)-approximate tour for a set \( \mathcal{R} \) of \( n \) rays. As before, assume that in the \( i \)th iteration, the rectangle \( Q_i = \{q_1, q_2, q_3, q_4\} \) is axis-parallel. A ray in \( \mathcal{R} \) is said to belong to the \( i \)th quadrant, \( i = 1, 2, 3, 4 \), if when placed with its apex at the origin, its head belongs to the \( i \)th quadrant. Let \( \mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3 \cup \mathcal{R}_4 \) be the partition of the rays in \( \mathcal{R} \) (after rotation) as dictated by the four quadrants. See Figure 2. Observe that

![Figure 2: The rectangle \( Q_i \), and two rays, one in \( \mathcal{R}_1 \) and one in \( \mathcal{R}_4 \) that intersect it.](image)

- A ray \( \rho \in \mathcal{R}_1 \) intersects \( Q_i \) if and only if \( q_2 \) and \( q_3 \) are separated by \( \ell(\rho) \), and the apex of \( \rho \) is dominated by \( q_3 \).
- A ray \( \rho \in \mathcal{R}_2 \) intersects \( Q_i \) if and only if \( q_1 \) and \( q_3 \)
are separated by \( \ell(\rho) \), and the apex of \( \rho \) lies right and below \( q_4 \).

- A ray \( \rho \in \mathcal{R}_3 \) intersects \( Q_i \) if and only if \( q_2 \) and \( q_4 \) are separated by \( \ell(\rho) \), and the apex of \( \rho \) dominates \( q_1 \).

- A ray \( \rho \in \mathcal{R}_4 \) intersects \( Q_i \) if and only if \( q_1 \) and \( q_3 \) are separated by \( \ell(\rho) \), and the apex of \( \rho \) lies left and above \( q_2 \).

Observe that these intersection conditions can be expressed as linear constraints in the four variables, \( x_1, x_2, y_1, y_2 \). The constraints listed above also correct an error in the old \( \sqrt{2} \)-approximation algorithm from [5], where it was incorrectly demanded that the apexes of the rays must lie in the rectangle \( Q_i \). Indeed, this condition is not necessary, and moreover, may prohibit finding an approximate solution with the claimed guarantee of \( \sqrt{2} \).

The resulting algorithm A3 for computing an approximate tour for \( n \) given rays computes a minimum-perimeter rectangle intersecting all rays over all \( m \) directions. For each of these directions, the algorithm solves a linear program with four variables and \( O(n) \) constraints, as described above. The approximation ratio is \( \frac{4}{\pi}(1+\varepsilon) \), and we set \( \varepsilon = 1/200 \) (or slightly smaller, as before), to obtain the approximation ratio 1.28.

As noted in [5], by walking twice (back and forth) along a path that visits all rays one gets a tour that visits all rays. Obviously, the algorithm A3 finds a (closed) path for \( n \) given rays. The approximation ratio for the path found is not more than twice the ratio achieved for a tour, so in our case \( \frac{2}{\pi}(1+\varepsilon) \), and we set \( \varepsilon = 1/1000 \) (or slightly smaller), to obtain the approximation ratio 2.55. This completes the proof of Theorem 2.

References


