Technische Universität Graz Institut für Mathematik
Christian-Doppler-Laboratorium „Diskrete Optimierung“

The Convex-Hull-and-Line Traveling Salesman Problem: A Solvable Case

Vladimir G. Deineko
René van Dal
Günter Rote

Report 233 September 1992 CDLDO-4

Technische Universität Graz, Steyrergasse 30, A-8010 Graz, Austria

Version 1
This paper appeared in *Information Processing Letters* **51** (1994), 141–148.
Title page processed by TeX on January 31, 1995

The Convex-Hull-and-Line Traveling Salesman Problem: A Solvable Case

Vladimir G. Deineko

Department of Applied Mathematics, Dnepropetrovsk University, Av. Gagarin 72, 320625 Dnepropetrovsk, Ukraine

René van Dal

Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Universitätsrechenzentrum, Universität Heidelberg, Im Neuenheimer Feld 293, D-69120 Heidelberg, Germany
Electronic mail: `Rene.van_Dal@IWR.Uni-Heidelberg.De`

Günter Rote

Institut für Mathematik, Technische Universität Graz, Steyrergasse 30, A-8010 Graz, Austria
Electronic mail: `rote@ftug.dnet.tu-graz.ac.at`

September 1992

Abstract

We solve the special case of the Euclidean Traveling Salesman Problem where $n-m$ cities lie on the boundary of the convex hull of all n cities, and the other m cities lie on a line segment inside this convex hull by an algorithm which needs $\mathcal{O}(mn)$ time and $\mathcal{O}(n)$ space.

Keywords: Euclidean Traveling Salesman Problem, shortest path, well-solvable case, polynomial time algorithm.

Introduction[1]

The n-city Euclidean Traveling Salesman Problem is the TSP where each city i is represented as a point $p_i = (x_i, y_i)$, $x_i, y_i \in \mathbb{R}$, in the plane and the distance $c(p_i, p_j)$ between any pair of cities i and j is computed according to the Euclidean metric, $i, j = 1, \ldots, n$. Papadimitriou [1977] proved the Euclidean TSP to be $\mathcal{NP}$-hard. We give an $\mathcal{O}(mn)$ time and $\mathcal{O}(n)$ space algorithm for solving the special case of the n-city Euclidean TSP where $n - m$ cities lie on the boundary of the convex hull of the n cities, and the other m cities lie on a line segment inside this convex hull. This special case of the n-city Euclidean TSP will be called the *convex-hull-and-line TSP*.

A well-known result with respect to the Euclidean TSP, presumed to be first mentioned explicitly by Flood [1956], states that 'in the euclidean plane the minimal (or optimal) tour does not intersect itself'. An *intersection* of a tour τ is defined as a common point $v \notin \{p_1, \ldots, p_n\}$ that is shared by two (or more) edges of τ, or a common point $w \in \{p_1, \ldots, p_n\}$ that is shared by three (or more) edges of τ. A proof of Flood's result was given by Quintas and Supnick [1965].

An important consequence of this is the following. Assuming that not all cities lie on one line, an optimal tour has the property that the cities on the boundary of the convex hull of the cities are visited in their cyclic order. Note that the case where all cities lie on two parallel lines corresponds to the case where all cities lie on the boundary of their convex hull.

Cutler [1980] has given an $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space dynamic programming algorithm for solving the so-called 3-line TSP, i. e., the Euclidean TSP where all points lie on three distinct parallel lines in the plane. Rote [1992] extended the results of Cutler by considering the N-line TSP, i. e., the Euclidean TSP where all points lie on N parallel lines in the plane, with N a small integer. He gave a dynamic programming algorithm which is polynomial for a fixed number of lines. Moreover, conditions are given such that the algorithm can also be applied in the case that all points lie on 'almost parallel' lines. Real-world problems that can be formulated as an N-line TSP arise in the manufacturing of printed circuit boards and related devices. However, because the running time of the algorithm is rather high (the exponent of the polynomial time bound is the number of lines), the algorithm seems to be of theoretical interest only.

The special case of the Euclidean TSP considered here is another extension of the 3-line TSP. It is easy to see that the class of convex-hull-and-line TSPs contains the 3-line TSP as a special case. Furthermore, we obtain an improvement in both running time and space requirement. In the final section we discuss the extension to other metrics than the Euclidean metric.

[1] The revision of this paper started during the visit of R. van Dal to the Dnepropetrovsk University in July–August 1991. Financial support by the Netherlands Organization for Scientific Research (NWO) and by the Japan International Science & Technology Exchange Center (JISTEC) is gratefully acknowledged. This paper is part of the Ph. D. thesis of R. van Dal, 'Special Cases of the Traveling Salesman Problem', Wolters-Noordhoff bv, Groningen, The Netherlands.

1 Properties of optimal solutions

In this section an $\mathcal{O}(mn)$ time algorithm is given for the special case of the n-city Euclidean TSP where $n-m$ cities lie on the boundary of the convex hull of the n cities, and the other m cities lie on a line segment inside this convex hull (see Figure 1).

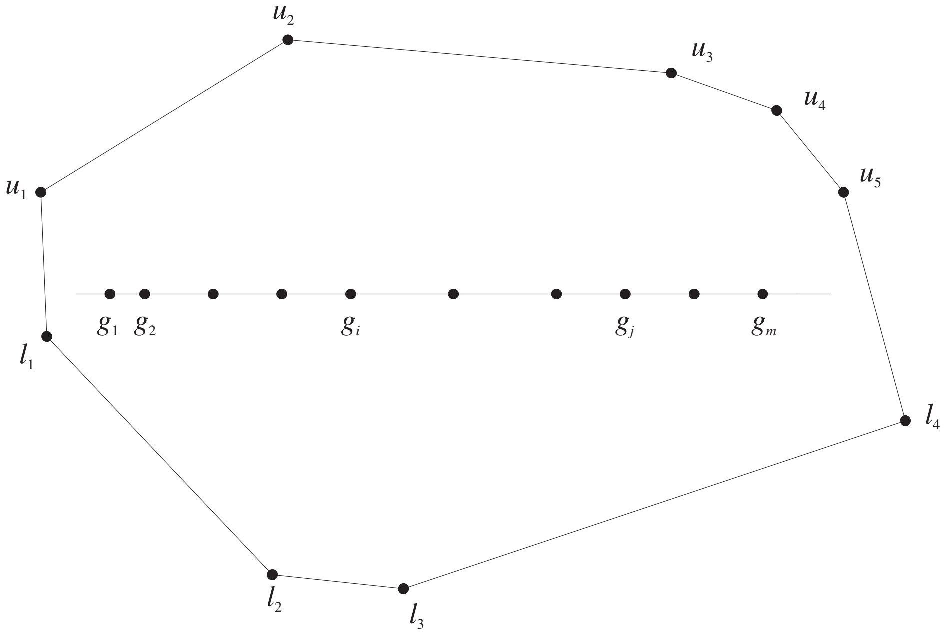

Figure 1: An instance of the convex-hull-and-line TSP.

The points on the line segment inside the convex hull will be labeled consecutively $g_1, g_2, \dots, g_m$. We assume $m \geq 1$. The set of points $\{g_1, g_2, \dots, g_m\}$ will be denoted by $\mathcal{G}$. We will also speak of the line through these points as the line $\mathcal{G}$. For $m = 1$ we can take any line through g_1 as the line $\mathcal{G}$. The points that lie on the boundary of the convex hull of the cities and above or on the line $\mathcal{G}$ will be labeled consecutively $u_1, u_2, \dots, u_p$. The points that lie on the boundary of the convex hull and below the line $\mathcal{G}$ will be labeled consecutively $l_1, l_2, \dots, l_q$. The set of points $\{u_1, \dots, u_p, l_1, \dots, l_q\}$ will be denoted by $\mathcal{B}$.

We will need one elementary lemma:

Lemma 1. *Let p, q, r, and s be four distinct points which form the vertices of a convex quadrilateral, in the given order. We allow that three points are collinear but not that all four points are collinear. Then we have*

$$c(p, q) + c(r, s) < c(p, r) + c(q, s).$$

In other words, the sum of the lengths of the diagonals is greater than the sum of the lengths of two opposite sides.

Proof. Let v be the intersection point of the diagonals. We get

$$c(p, r) + c(q, s) = c(p, v) + c(v, r) + c(q, v) + c(v, s) > c(p, q) + c(r, s),$$

using the triangle inequality for the triangles pvq and rvs. Since at least one of these triangles is nondegenerate we obtain a strict inequality. ∎

In fact, it follows from this lemma that an optimal tour does not intersect itself.

As already stated, in an optimal tour the cities in $\mathcal{B}$ have to be visited in their cyclic order, otherwise there is an intersection. Therefore, for each city $g_i \in \mathcal{G}$, it remains to determine between which two adjacent cities in $\mathcal{B}$ it is visited. The following lemmas give a necessary condition for an optimal tour of the convex-hull-and-line TSP.

Lemma 2. *Let $g_i, g_j \in \mathcal{G}$ and let v and w be two adjacent cities in $\mathcal{B}$. If in an optimal tour τ both g_i and g_j are visited between v and w, then all cities on $\mathcal{G}$ that lie between g_i and g_j are visited between v and w.*

Proof. If there is a city between g_i and g_j that is not visited between v and w, then τ contains an intersection. Therefore, τ is not an optimal tour. ∎

As a consequence of this lemma and the fact that the cities in $\mathcal{B}$ are visited in their cyclic order we obtain the following lemma.

Lemma 3. *An optimal tour can be obtained by splitting the set of points $\mathcal{G}$ into $k+1$ segments*

$$\{g_1, g_2, \ldots, g_{i_1}\}, \{g_{i_1+1}, \ldots, g_{i_2}\}, \ldots, \{g_{i_k+1}, \ldots, g_m\},$$

for $0 \leq k < m$, $0 = i_0 < i_1 < i_2 < \cdots < i_k < m$, and inserting each segment between two adjacent points in $\mathcal{B}$. ∎

The algorithm to be described will first determine for each possible segment $\{g_i, g_{i+1}, \ldots, g_{j-1}, g_j\}$, $1 \leq i < j \leq m$, the cheapest possible way to insert it between two adjacent cities in $\mathcal{B}$, and then it will determine the best way to split $\{g_1, g_2, \ldots, g_m\}$ into segments.

In principle, the insertion of a segment $\{g_i, g_{i+1}, \ldots, g_{j-1}, g_j\}$ between two adjacent points v and w in $\mathcal{B}$ can be done in two ways. However, when v and w are on the same side of $\mathcal{G}$ we can discard one possibility since either the path $v, g_i, g_{i+1}, \ldots, g_{j-1}, g_j, w$ or the path $v, g_j, g_{j-1}, \ldots, g_{i+1}, g_i, w$ intersects itself. In these cases we say that a segment is inserted in the correct orientation when the insertion does not result in such an intersection. Note that neither the path $u_1, g_i, g_{i+1}, \ldots, g_{j-1}, g_j, l_1$ nor the path $u_1, g_j, g_{j-1}, \ldots, g_{i+1}, g_i, l_1$ intersects itself. However, inserting a segment between u_1 and l_1 may also result in an intersection in the tour, as the following lemma shows.

Lemma 4. *For any optimal tour, the segment $\{g_i, g_{i+1}, \ldots, g_{j-1}, g_j\}$ cannot be inserted between u_1 and l_1 unless $i = 1$. Similarly, the segment $\{g_i, g_{i+1}, \ldots, g_{j-1}, g_j\}$ cannot be inserted between u_p and l_q unless $j = m$.*

Proof. If $i > 1$, then inserting the segment $\{g_i, g_{i+1}, \ldots, g_{j-1}, g_j\}$ between u_1 and l_1 would separate point g_1 from the points right of the chain $[u_1, g_i, g_j, l_1]$ or $[u_1, g_j, g_i, l_1]$, thereby creating an intersection. Similarly, if $j < m$ then the segment cannot be inserted between u_p and l_q without introducing an intersection. ■

Remark. The above lemma corresponds to Cutler's Triangle Theorem with respect to the 3-line TSP; see Cutler [1980]. The Triangle Theorem states that at least one of the edges $\{u_1, l_1\}$, $\{u_1, g_1\}$ and $\{l_1, g_1\}$ is an edge of an optimal tour. Similarly, at least one of the edges $\{u_p, l_q\}$, $\{u_p, g_m\}$ and $\{l_q, g_m\}$ is an edge of an optimal tour.

Therefore, the cases mentioned in the above lemma have to be excluded. Two adjacent points v and w in $\mathcal{B}$ will be called *admissible* for a segment $\{g_i, g_{i+1}, \ldots, g_j\}$, $1 \leq i < j \leq m$, if

- v and w lie strictly on the same side of the line $\mathcal{G}$, or if
- $\{v, w\} = \{u_p, l_q\}$ and $j = m$, or if
- $\{v, w\} = \{u_1, l_1\}$ and $i = 1$.

The cost of inserting the segment $\{g_{i+1}, g_{i+2}, \ldots, g_j\}$ between v and w will be denoted by

$$\begin{aligned} e_{ij}^{v,w} &= \min\{c(v, g_{i+1}) + c(g_{i+1}, g_j) + c(g_j, w) - c(v, w), \\ &\qquad c(v, g_j) + c(g_j, g_{i+1}) + c(g_{i+1}, w) - c(v, w)\}. \end{aligned} \tag{1}$$

The cost of the best possible insertion of the segment $\{g_{i+1}, g_{i+2}, \ldots, g_j\}$ is

$$d_{ij} = \min\{e_{ij}^{v,w} \mid \{v, w\} \text{ is admissible for } \{g_{i+1}, g_{i+2}, \ldots, g_j\}\}. \tag{2}$$

The possible splittings of $\{g_1, g_2, \ldots, g_m\}$ can be associated with paths in an acyclic digraph D with vertex set $\{0, 1, \ldots, m\}$ and arcs (i, j) with costs d_{ij} for all $0 \leq i < j \leq m$.

It is easy to see that if we associate with the arc (i, j) the segment $\{g_{i+1}, g_{i+2}, \ldots, g_{j-1}, g_j\}$, then there is a one-to-one correspondence between the splittings of $\{g_1, g_2, \ldots, g_m\}$ into segments and the paths in D from 0 to m. For example, the splitting $\{\{g_1, g_2, g_3\}, \{g_4, g_5\}, \{g_6\}, \{g_7, g_8, g_9, g_{10}\}\}$ corresponds to the path $0, 3, 5, 6, 10$. Moreover, the length of a path $0, i_1, i_2, \ldots, i_k, m$ in D represents the minimal total costs of inserting the corresponding segments $\{g_1, g_2, \ldots, g_{i_1}\}, \{g_{i_1+1}, \ldots, g_{i_2}\}, \ldots, \{g_{i_k+1}, \ldots, g_m\}$. Evidently, a shortest path from 0 to m in D determines an optimal tour for the convex-hull-and-line TSP, as the following theorem shows.

Theorem 1. *Let σ be the initial subtour for a convex-hull-and-line TSP that visits only the cities on the boundary of the convex hull in their cyclic order. Then a tour τ is optimal if and only if it can be obtained by inserting the points in $\mathcal{G}$ into σ in such a way that the corresponding path in the digraph D has shortest length. As a consequence, the length of an optimal tour is the length of the initial subtour σ plus the length of a shortest path in D.*

Proof. First, we will show that the length of an optimal tour is at least the length of σ plus the length of a shortest path in D. Indeed, by Lemma 3 we only have to consider tours τ that can be obtained by inserting a number of segments into σ. The length of τ is equal to the length of σ plus the length of the corresponding path in D, and we are done.

Next, we will show that this lower bound is actually attained. To that purpose, we construct for any shortest path $0{=}i_0, i_1, \ldots, i_k, i_{k+1}{=}m$ in D a tour τ with corresponding length. For $j = 1, \ldots, k+1$, let the minimal cost of insertion of the segment $\{g_{i_{j-1}+1}, \ldots, g_{i_j}\}$ be between v_j and w_j. A tour of length equal to the length of σ plus the length of the shortest path in D is obtained if we insert each segment $\{g_{i_{j-1}+1}, \ldots, g_{i_j}\}$ between v_j and w_j, $j = 1, \ldots, k+1$, in the correct orientation. Unless two segments have to be inserted between the same pair of points we obtain a tour with corresponding length. Therefore, we have to show that such a conflict cannot arise.

If v_j and w_j lie strictly on different sides of the line $\mathcal{G}$, the only arcs in D whose corresponding segments might be inserted between v_j and w_j are the arcs starting in 0 (or the arcs ending in m), and a shortest path contains only one such arc.

Now consider two segments $\{g_{i_1+1}, g_{i_1+2}, \ldots, g_{j_1}\}$ and $\{g_{i_2+1}, g_{i_2+2}, \ldots, g_{j_2}\}$ with $i_1 + 1 \leq j_1 < i_2 + 1 \leq j_2$, that are to be inserted between two adjacent points in $\mathcal{B}$ which both lie on the same side of the line $\mathcal{G}$, say, between u_k and u_{k+1}. (The proof of the case l_k and l_{k+1} is similar.) We will prove that the arcs (i_1, j_1) and (i_2, j_2) cannot both be arcs of a shortest path from 0 to m in D because the cost of the arc (i_1, j_2) is less than the total cost of the arcs (i_1, j_1) and (i_2, j_2):

$$\begin{aligned}
& c(u_k, g_{i_1+1}) + c(g_{i_1+1}, g_{j_1}) + c(g_{j_1}, u_{k+1}) - c(u_k, u_{k+1}) + \\
+ \quad & c(u_k, g_{i_2+1}) + c(g_{i_2+1}, g_{j_2}) + c(g_{j_2}, u_{k+1}) - c(u_k, u_{k+1}) \\
> \quad & c(u_k, g_{i_1+1}) + c(g_{i_1+1}, g_{j_2}) + c(g_{j_2}, u_{k+1}) - c(u_k, u_{k+1}).
\end{aligned}$$

If we cancel the common terms $c(u_k, g_{i_1+1})$, $c(g_{j_2}, u_{k+1})$ and $c(u_k, u_{k+1})$ and subtract the equation

$$c(g_{i_1+1}, g_{j_1}) + c(g_{j_1}, g_{i_2+1}) + c(g_{i_2+1}, g_{j_2}) = c(g_{i_1+1}, g_{j_2}),$$

we get the equivalent inequality

$$c(g_{j_1}, u_{k+1}) + c(u_k, g_{i_2+1}) > c(g_{j_1}, g_{i_2+1}) + c(u_k, u_{k+1}),$$

which follows from Lemma 1 applied to the quadrangle $g_{j_1} g_{i_2+1} u_{k+1} u_k$. ■

2 The algorithm

Now we are ready to present our algorithm. In the first phase we compute the cost of a shortest path of the acyclic digraph by a dynamic programming recursion, and in the second phase we use this path to construct an optimal tour.

ALGORITHM (*convex-hull-and-line TSP*)

PHASE 1:
  $f_0 := 0$; (* f_j is the length of a shortest path from 0 to j. *)
  for $j := 1$ **to** m **do**
    compute d_{ij} for $i = 0, 1, \ldots, j-1$;
    $f_j := \min\{ f_i + d_{ij} \mid i = 0, 1, \ldots, j-1 \}$;
    $p_j := \arg\min f_j$; (* p_j is the predecessor of j on *)
                (* the shortest path from 0 to j. *)
  endfor;
  (* The length of an optimal tour is $f_m + c(\sigma)$. *)

PHASE 2:
  $j := m$;
  repeat
    $i := p_j$;
    Determine $\{v, w\}$ admissible for $\{g_{i+1}, g_{i+2}, \ldots, g_j\}$
      such that $d_{ij} = e_{ij}^{v,w}$;
    Insert the segment $\{g_{i+1}, g_{i+2}, \ldots, g_j\}$ between v and w
      in the correct orientation;
    $j := i$;
  until $j = 0$;

Let us first discuss how to compute the values d_{ij}. The cost of inserting the segment $\{g_{i+1}, g_{i+2}, \ldots, g_j\}$ between u_1 and l_1 (or u_p and l_q) can be computed in constant time, if that insertion is admissible. We will show that, for a fixed value of j, and for all i, $0 \le i < j$, the cost of inserting the segment $\{g_{i+1}, g_{i+2}, \ldots, g_j\}$ between u_k and u_{k+1}, $k = 1, \ldots, p-1$, (between l_k and l_{k+1}, $k = 1, \ldots, q-1$, respectively) can be computed in $\mathcal{O}(n)$ time. This means that in one iteration of the first phase, all the values d_{ij}, for a fixed j, can be computed in $\mathcal{O}(n)$ time, and thus the whole iteration takes $\mathcal{O}(n)$ time. This leads to a time complexity of $\mathcal{O}(mn)$ for phase 1.

Let $A = (a_{ik})_{0 \le i < j,\, 1 \le k < p}$ be the $j \times (p-1)$ matrix with entries

$$a_{ik} = c(u_k, g_{i+1}) + c(g_{i+1}, g_j) + c(g_j, u_{k+1}) - c(u_k, u_{k+1}).$$

Clearly, our problem will be solved if we determine the minimum in each row of the matrix. Let $j(i)$ be the index of the leftmost column containing the minimum value in row i of A. A is called *monotone* if $i_1 < i_2$ implies that $j(i_1) \le j(i_2)$. A is *totally monotone* if every submatrix of A is monotone, or equivalently, if $a_{i_1 k_2} < a_{i_1 k_1}$ implies $a_{i_2 k_2} < a_{i_2 k_1}$ for $i_1 < i_2$ and $k_1 < k_2$. Aggarwal et al. [1987] have shown that all row minima can be computed in $\mathcal{O}(j + p)$ time if the matrix A is totally monotone. To prove that $A = (a_{ik})$ is totally monotone, we show that A has even a stronger property, the so-called *Monge property*:

$$a_{i_1 k_1} + a_{i_2 k_2} \le a_{i_1 k_2} + a_{i_2 k_1}, \quad \text{for } i_1 < i_2 \text{ and } k_1 < k_2.$$

Substituting the definition of a_{ik} and canceling common terms leads to the equivalent inequality

$$c(u_{k_1}, g_{i_1+1}) + c(u_{k_2}, g_{i_2+1}) \le c(u_{k_2}, g_{i_1+1}) + c(u_{k_1}, g_{i_2+1}),$$

which follows from Lemma 1 applied to the quadrangle $u_{k_1} g_{i_1+1} g_{i_2+1} u_{k_2}$, see Figure 2.

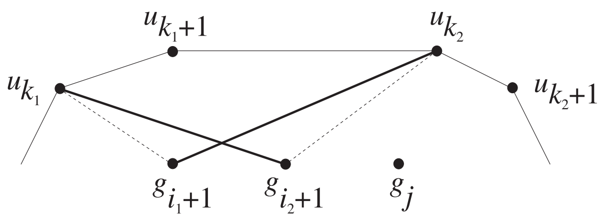

Figure 2: Illustration of the proof.

Finally, we only need to store the values d_{ij} for a fixed j for the current iteration, and hence the first phase needs only $\mathcal{O}(n)$ space.

The search for an admissible pair $\{v, w\}$ in phase 2 can be carried out in $\mathcal{O}(n)$ time, by just using the definition of d_{ij}, equation (2). The loop has to be repeated for each arc of the shortest path, i. e., at most m times, and therefore phase 2 also needs only $\mathcal{O}(mn)$ time and $\mathcal{O}(n)$ space. As a conclusion we have the following theorem.

Theorem 2. *The convex-hull-and-line TSP, i. e., the n-city Euclidean TSP where $n - m$ cities lie on the boundary of the convex hull of the n cities and the other m cities lie on a line segment inside the convex hull, can be solved in $\mathcal{O}(mn)$ time and $\mathcal{O}(n)$ space.* ■

1	$(0.177, 0.177)$	11	$(1.000, 0.032)$
2	$(0.355, 0.355)$	12	$(1.000, 0.268)$
3	$(0.381, 0.381)$	13	$(1.000, 0.681)$
4	$(0.457, 0.457)$	14	$(1.000, 0.822)$
5	$(0.632, 0.632)$	15	$(1.000, 0.992)$
6	$(0.789, 0.789)$	16	$(0.993, 0.993)$
7	$(0.164, 0.000)$	17	$(0.794, 1.000)$
8	$(0.171, 0.000)$	18	$(0.057, 1.000)$
9	$(0.387, 0.000)$	19	$(0.000, 1.000)$
10	$(0.409, 0.000)$	20	$(0.000, 0.329)$

Table 1: Coordinates of points.

Example. Consider the $n = 20$ points whose coordinates are given in Table 1. The $m = 6$ points on the line segment inside the convex hull of the 20 points are

$1, 2, 3, 4, 5$, and 6. The initial subtour is $(7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20)$. A shortest path from 0 to 6 in the network is 0,1,6, and the length of this path is $0.0432 + 0.7983 = 0.8415$. This means that in order to obtain an optimal tour we have to insert point 1 and segment $\{2, 3, 4, 5, 6\}$ into the initial subtour. Point 1 is inserted between 7 and 20 and segment $\{2, 3, 4, 5, 6\}$ is inserted between 17 and 18. So, an optimal tour is

$$(1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 6, 5, 4, 3, 2, 18, 19, 20)$$

and the length of this tour is 4.6772. (see Figure 3). □

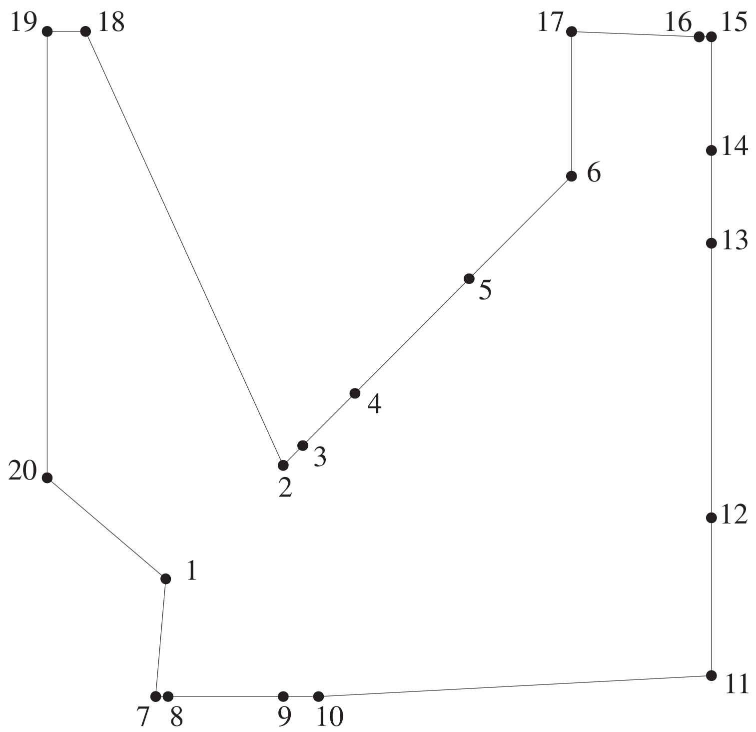

Figure 3: An instance of the convex-hull-and-line TSP and its solution.

3 Other metrics

We have formulated our algorithm and the proof in terms of the Euclidean metric in the plane. Let us discuss how the result can be extended to other metrics. We have used properties of the underlying metric in Lemma 1 and its consequence, the intersection-free property of the optimal solution. Beside the triangle inequality, the proof of Lemma 1 uses the following

Linearity property:
If the point v lies between p and r on the segment pr, then

$$c(p, r) = c(p, v) + c(v, r).$$

This relation is true for all metrics that come from a norm, for example the L_1 (Manhattan) metric and the L_∞ metric.

The linearity property was also used implicitly in assuming that the segment $g_i g_j$ "contains" all cities that lie between g_i and g_j, i. e., $c(g_i, g_j) = c(g_i, g_{i+1}) + c(g_{i+1}, g_{i+2}) + \cdots + c(g_{j-1}, g_j)$. This assumption was for example used in the proof of Lemma 2, in the definition of insertion costs $e_{ij}^{v,w}$ (1) and a_{ik}, and in the proof of Theorem 1.

It is clear that the convex-hull-and-line TSP needs *some* property which links the metric with the geometric structure of the plane, in particular with convexity and with lines. So let us assume that we are given a metric with the linearity property. If equality can hold in the triangle inequality $c(p, v) + c(v, r) \geq c(p, r)$ even for a point v that does not lie on the segment pr, then Lemma 1 still holds, except that the strict inequality cannot be maintained. (This occurs for example in the case of the L_1 and L_∞ metrics.) The other arguments in the paper also go through, but some of the conclusions must be weakened. For example, we cannot claim that *every* optimal solution looks as described in Lemma 2, but only that *there exists* such an optimal solution. Thus, Theorem 1 gives only a sufficient condition for an optimal solution and not a characterization, i. e., "if and only if" has to be replaced by "if". Still, the algorithm is guaranteed to find an optimal solution if we simply compute costs according to the given metric. Let us summarize this:

Theorem 3. *The algorithm solves the convex-hull-and-line TSP in $\mathcal{O}(mn)$ time and $\mathcal{O}(n)$ space for any metric $c(p, q)$ which fulfills the linearity property. In particular, this holds for every metric that is induced by a norm.* ■

References

Aggarwal, A., M. M. Klawe, S. Moran, P. Shor, and R. Wilber (1987), Geometric applications of a matrix-searching algorithm, *Algorithmica* **2**, 195–208.

Cutler, M. (1980), Efficient special case algorithms for the N-line planar traveling salesman problem, *Networks* **10**, 183–195.

Flood, M. M. (1956), The traveling-salesman problem, *Operations Research* **4**, 61–75.

Papadimitriou, C. H. (1977), The Euclidean traveling salesman problem is $\mathcal{NP}$-complete, *Theoretical Computer Science* **4**, 237–244.

Quintas, L. V., and F. Supnick (1965), On some properties of shortest Hamiltonian circuits, *American Mathematical Monthly* **72**, 977–980.

Rote, G. (1992), The N-line traveling salesman problem, *Networks* **22**, 91–108.