数理解析研究所講究録
第 853 巻 1993 年 55-65

Special cases of the Euclidean Traveling Salesman Problem

Vladimir G. Deineko
Department of Applied Mathematics, Dnepropetrovsk State University, Av. Gagarin 72, 320625 Dnepropetrovsk, Ukraine

René van Dal*
Department of Econometrics, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands

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

Abstract

We present a survey of results on solving special cases of the Euclidean Traveling Salesman Problem. In particular, a sketch is given of an $\mathcal{O}(mn)$ time and $\mathcal{O}(n)$ space algorithm for solving the special case of the n-city 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.

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

*Current address: Railway Technical Research Institute, 2-8-38 Hikari-Cho, Kokubunji-Shi, Tokyo 185, Japan. Electronic mail: `van_dal@ipc02.rtri.or.jp`. Financial support by the Japan International Science & Technology Exchange Center (JISTEC) is gratefully acknowledged.

Introduction[1]

The n-city Euclidean Traveling Salesman Problem (TSP) 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 [8] proved the Euclidean TSP to be $\mathcal{NP}$-hard. Therefore, it is interesting to investigate whether special cases of this problem are solvable in polynomial time. The main reference on special cases of the (general) TSP is the excellent survey by Gilmore, Lawler and Shmoys [7] (at this moment, Burkard, Deineko, Van Dal and Van der Veen [2] are working on a survey of recent results). However, Gilmore, Lawler and Shmoys do not consider special cases of the Euclidean TSP.

Therefore, first a survey of results on special cases of the Euclidean Traveling Salesman Problem will be presented. Thereafter, we give a sketch of 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 Euclidean TSP is a generalization of several special cases of the Euclidean TSP that will be considered in the next section.

[1]This paper is based on the paper 'The convex-hull-and-line Traveling Salesman Problem: A solvable case' which has been submitted to *Computational Geometry: Theory and Applications*, and which in turn is an updated version of Chapter 4 of the Ph.D. thesis of R. van Dal, 'Special Cases of the Traveling Salesman Problem', published by Wolters-Noordhoff bv, Groningen, The Netherlands.

1 Survey of results on the Euclidean TSP

A well-known result with respect to the Euclidean TSP, presumed to be first mentioned explicitly by Flood [6], 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 [9].

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. On the contrary, if all n cities do lie on one line and are labeled according to their order on the line, then a tour is optimal if and only if it is pyramidal, where a tour on n cities is called pyramidal if it is of the form $(1, i_1, \ldots, i_r, n, j_1, \ldots, j_{n-r-2})$ with $i_1 < i_2 < \cdots < i_r$ and $j_1 > j_2 > \cdots > j_{n-r-2}$. Several authors have formulated conditions on the distance matrix under which an optimal tour is at least as long as a shortest pyramidal tour, see e.g. Demidenko [4],[5], Van der Veen [11], and Gilmore, Lawler and Shmoys [7]. Moreover, a shortest pyramidal tour can be found in polynomial time, despite the fact that their number is exponential. 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 [3] 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 [10] 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 to be considered in the next section is another extension of the 3-line TSP. It is easy to see that the class of so-called 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. Finally, another interesting special case of the Euclidean TSP, already mentioned by Cutler [3], is the case where all cities lie on two perpendicular lines. This problem is still open.

2 The convex-hull-and-line TSP

In this section a sketch of an $\mathcal{O}(mn)$ time algorithm is given 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 (see Figure 1). This special case of the Euclidean TSP will be called the convex-hull-and-line TSP.

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

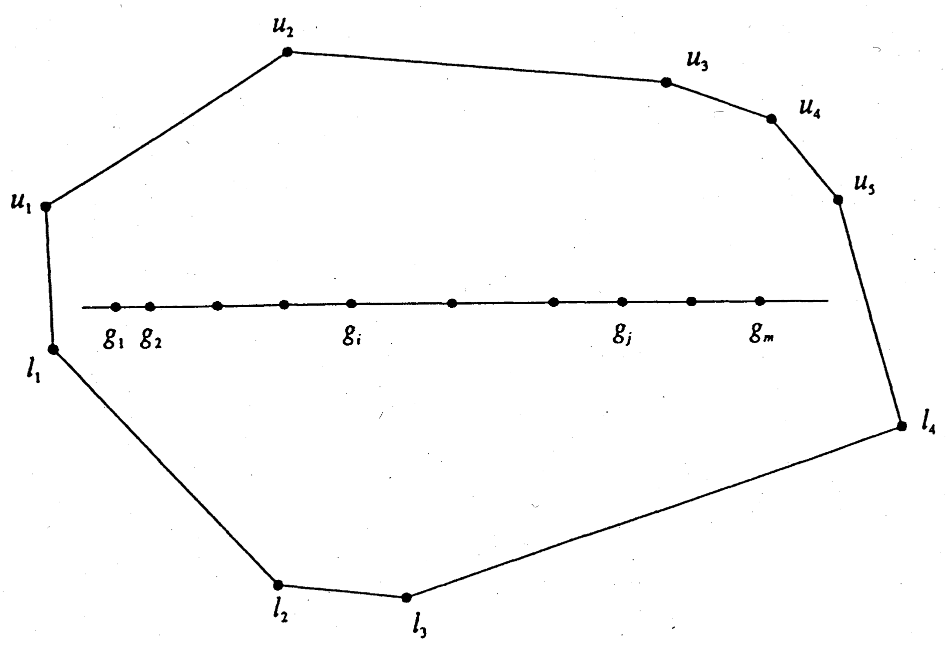

Figure 1: An instance of the convex-hull-and-line TSP.

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 2.1 *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 that lie between g_i and g_j on $\mathcal{G}$ are visited between v and w.*

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 2.2 *An optimal tour can be obtained by splitting the set of points $\mathcal{B}$ 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 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, in almost all cases one way should be discarded because it yields an intersection. Furthermore, inserting a segment between u_1 and l_1 may also result in an intersection in the tour, as the following lemma shows.

Lemma 2.3 *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 $j = m$.*

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 possible splittings of $\{g_1, g_2, \ldots, g_m\}$ can be associated with paths in an acyclic digraph D with vertex set $\{0, 1, \ldots, m\}$ (the additional vertex 0 acts as a source) and arcs (i, j) with costs d_{ij} for all $0 \leq i < j \leq m$, where d_{ij} is the minimum cost of inserting the segment $\{g_{i+1}, g_{i+2}, \ldots, g_j\}$ between two admissible adjacent points in $\mathcal{B}$.

It is easy to see that if we associate with the arc (i, j) the segment $\{g_{i+1}, g_{i+2}, \ldots, g_j\}$, 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 2.1 *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 $\mathcal{B}$ 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.

In the first phase of our algorithm 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.

Let us briefly discuss how to compute the values d_{ij}. Clearly, for a fixed value of j, and for all i, $0 \leq i < j$, 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 $\mathcal{O}(mn)$ time, but we will show that the time complexity can be improved to $\mathcal{O}(n)$ time.

Let $A = (a_{ik})_{0 \leq i < j,\ 1 \leq k < p}$ 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 $j(i_1) \leq j(i_2)$. A is totally monotone if every submatrix of A is monotone. Aggarwal et al. [1] have shown that all row minima can be computed in $\mathcal{O}(j + p)$ time if the matrix A is totally monotone. It is easy to prove that $A = (a_{ik})$ is totally monotone (actually, in our paper we prove that A has an even stronger property).

Finally, we only need to store the values d_{ij} for a fixed j for each of the m iterations in the first phase, and hence in this phase the algorithm needs only $\mathcal{O}(n)$ space. Finally, we state our main theorem.

Table 1: Coordinates of points.

1	(177,177)	11	(1000,32)
2	(355,355)	12	(1000,268)
3	(381,381)	13	(1000,681)
4	(457,457)	14	(1000,822)
5	(632,632)	15	(1000,992)
6	(789,789)	16	(993,993)
7	(164,0)	17	(794,1000)
8	(171,0)	18	(57,1000)
9	(387,0)	19	(0,1000)
10	(409,0)	20	(0,329)

Theorem 2.2 *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.*

Example Let $n = 20$ and the coordinates of the points 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 20 is 0,1,20 and the length of this path is 43 + 801 = 844 (rounded to integers). 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

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

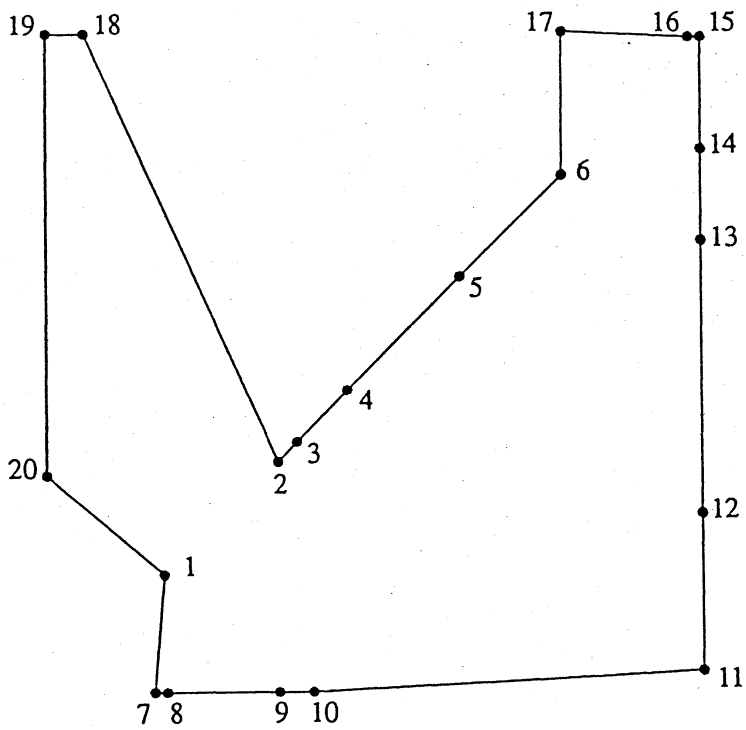

Figure 2: An instance of the convex-hull-and-line TSP and its solution.

and the length of this tour is 4694 (see Figure 2).

References

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

[2] Burkard, R.E., V.G. Deineko, R. van Dal and J.A.A. van der Veen, "Solvable cases of the TSP: a survey of recent results", in preparation.

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

[4] Demidenko, V.M., "A special case of traveling salesman problems (in Russian)", *Izv. Akad. Nauk BSSR Ser. Fiz.-Mat. Nauk* 5 (1976) 28–32.

[5] Demidenko, V.M., "The traveling salesman problem with asymmetric matrices (in Russian)", *Vestsi Akad. Nauk BSSR Ser. Fiz.-Mat. Nauk* 1 (1979) 29–35.

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

[7] Gilmore, P.C., E.L. Lawler and D.B. Shmoys, "Well-solved special cases", in: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds., *The Traveling Salesman Problem*, (John Wiley and Sons, Chichester, 1985) 87–143.

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

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

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

[11] van der Veen, J.A.A., "Solvable cases of the traveling salesman problem with various objective functions", Ph.D. thesis, University of Groningen (Groningen, The Netherlands, 1992).