Symposium on Theoretical Aspects of Computer Science 2010 (Nancy, France), pp. 239-250
www.stacs-conf.org

THE TRAVELING SALESMAN PROBLEM UNDER SQUARED EUCLIDEAN DISTANCES

MARK DE BERG[1] AND FRED VAN NIJNATTEN[1] AND RENÉ SITTERS[2] AND GERHARD J. WOEGINGER[1] AND ALEXANDER WOLFF[3]

[1] Department of Mathematics and Computer Science, TU Eindhoven, the Netherlands.
E-mail address: mdberg@win.tue.nl
E-mail address: f.s.b.v.nijnatten@tue.nl
E-mail address: gwoegi@win.tue.nl

[2] Faculty of Economics and Business Administration, VU Amsterdam, the Netherlands.
E-mail address: rsitters@feweb.vu.nl

[3] Lehrstuhl für Informatik I, Universität Würzburg, Germany.
URL: http://www1.informatik.uni-wuerzburg.de/en/staff/wolff_alexander

Abstract. Let P be a set of points in $\mathbb{R}^d$, and let $\alpha \geqslant 1$ be a real number. We define the distance between two points $p, q \in P$ as $|pq|^\alpha$, where $|pq|$ denotes the standard Euclidean distance between p and q. We denote the traveling salesman problem under this distance function by $\textsc{Tsp}(d,\alpha)$. We design a 5-approximation algorithm for $\textsc{Tsp}(2,2)$ and generalize this result to obtain an approximation factor of $3^{\alpha-1} + \sqrt{6}^{\,\alpha}/3$ for $d = 2$ and all $\alpha \geqslant 2$.

We also study the variant Rev-Tsp of the problem where the traveling salesman is allowed to revisit points. We present a polynomial-time approximation scheme for Rev-Tsp$(2,\alpha)$ with $\alpha \geqslant 2$, and we show that Rev-Tsp(d,α) is apx-hard if $d \geqslant 3$ and $\alpha > 1$. The apx-hardness proof carries over to $\textsc{Tsp}(d,\alpha)$ for the same parameter ranges.

1. Introduction

Motivated by a power-assignment problem in wireless networks (see below for a short discussion of this application) Funke et al. [12] studied the following special case $\textsc{Tsp}(d,\alpha)$ of the Traveling Salesman Problem (Tsp) which is specified by an integer $d \geqslant 2$ and a real number $\alpha > 0$. The cities are n points in d-dimensional space $\mathbb{R}^d$, and the distance between two points p and q is $|pq|^\alpha$, where $|pq|$ denotes the standard Euclidean distance between p and q.

- The objective in problem $\textsc{Tsp}(d,\alpha)$ is to find a shortest tour (under distances $|\cdot|^\alpha$) that visits every city *exactly* once.

1998 ACM Subject Classification: I.1.2 Algorithms, F.2.2 Nonnumerical Algorithms and Problems.
Key words and phrases: Geometric traveling salesman problem, power-assignment in wireless networks, distance-power gradient, NP-hard, APX-hard.

27th Symposium on Theoretical Aspects of Computer Science, Nancy, 2010
Editors: Jean-Yves Marion, Thomas Schwentick
Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany
Digital Object Identifier: 10.4230/LIPIcs.STACS.2010.2458

- In the closely related problem Rev-TSP(d, α), the objective is to find a shortest tour that visits every city *at least* once; thus the salesman is allowed to revisit cities.

Note that TSP$(2, 1)$ is the classical two-dimensional Euclidean TSP and that TSP(d, ∞) is the so-called *bottleneck* TSP in $\mathbb{R}^d$, where the goal is to find a tour whose longest edge has minimum length. We are, however, mainly interested in the case where α is some small constant, and we will not touch the case $\alpha = \infty$.

Similarities and differences to the classical Euclidean TSP. The classical Euclidean TSP is NP-hard even in two dimensions, but it is relatively easy to approximate. In particular, it admits a polynomial-time approximation scheme: Given a parameter $\varepsilon > 0$ and a set of n points in d-dimensional Euclidean space, one can find in $2^{(d/\varepsilon)^{O(d)}} + (d/\varepsilon)^{O(d)} n \log n$ time a tour whose length is at most $1 + \varepsilon$ times the optimal length [23].

A crucial property of the Euclidean TSP is that the underlying Euclidean distances satisfy the triangle inequality. The triangle inequality implies that no reasonable salesman would ever revisit the same city: Instead of returning to a city, it is always cheaper to skip the city and to travel directly to the successor city. All positive approximation results for the Euclidean TSP rely heavily on the triangle inequality. In strong contrast to this, for exponents $\alpha > 1$ the distance function $|\cdot|^\alpha$ does not satisfy the triangle inequality. Thus the combinatorial structure of the problem changes significantly—for example, revisits may suddenly become helpful—and the existing approximation algorithms for Euclidean TSP cannot be applied.

Another nice property of the classical Euclidean problem TSP$(2, 1)$ is that, sloppily speaking, instances with many cities have long optimal tours. Consider for instance a set P of n points in the unit square. Then there exists a tour whose Euclidean length is bounded by $O(\sqrt{n})$ [15]. This bound is essentially tight since there are point sets for which *every* tour has Euclidean length $\Omega(\sqrt{n})$. Interestingly, these results do not carry over to TSP$(2, 2)$ with *squared* Euclidean distances. Problem #124 in the book by Bollobás [8] shows that there always exists a tour for P such that the sum of the squared Euclidean distances is bounded by 4, and that this bound of 4 is best possible. Since, as a rule of thumb, large objective values are easier to approximate than small objective values, this already indicates a substantial difference in the approximability behaviors of TSP(2,1) and TSP(2,2).

Previous work and our results. Funke et al. [12] note that the distance function $|\cdot|^\alpha$ satisfies the so-called *τ-relaxed triangle inequality* with parameter $\tau = 2^{\alpha-1}$ (see Section 2 for a definition). The classical TSP under the τ-relaxed triangle inequality has been extensively studied [2, 3, 6, 7], and all the corresponding machinery from the literature can be applied directly to TSP(d, α). For instance, Andreae [6] derives a $(\tau^2 + \tau)$-approximation for the classical TSP under the τ-relaxed triangle inequality (Δ_τ-TSP, for short). This result translates into a $(4^{\alpha-1} + 2^{\alpha-1})$-approximation for TSP$(\,\cdot\,, \alpha)$. For $\tau > 3$, it is better to apply Bender and Chekuri's 4τ-approximation [2] for Δ_τ-TSP, which yields a $2^{\alpha+1}$-approximation for TSP$(\,\cdot\,, \alpha)$. Funke et al. derive a $(2 \cdot 3^{\alpha-1})$-approximation algorithm for TSP$(\,\cdot\,, \alpha)$, which for the range $2 < \alpha < \log_{3/2} 3 \approx 2.71$ is better than applying the known results [6, 2]. The best result for $\alpha < 2$ is obtained by Böckenhauer et al. [7] whose Christofides-based $(3\tau^2/2)$-approximation for Δ_τ-TSP yields a $(3 \cdot 2^{2\alpha-3})$-approximation for TSP$(\,\cdot\,, \alpha)$.

We will demonstrate in Section 2 that essentially *every* variant of the original T^3-algorithm by Andreae and Bandelt [3] already gives a $(2 \cdot 3^{\alpha-1})$-approximation for TSP(d, α). The bottom-line of all this, and the actual starting point of our paper, is that the machinery around the τ-relaxed triangle inequality only yields a bound of roughly $2 \cdot 3^{\alpha-1}$. This raises the following questions: How much can geometry help us in getting even better approximation ratios? Can we beat the 6-approximation for Tsp(2, 2) of Funke et al.? We answer these questions affirmatively: We develop a new variant of the T^3-algorithm which we call the *geometric* T^3*-algorithm.* An intricate analysis in Section 3 shows that this yields a 5-approximation for Tsp(2, 2). We then extend our analysis to Tsp$(2, \alpha)$ with $\alpha > 2$, and thus obtain a $(3^{\alpha-1} + \sqrt{6}^{\,\alpha}/3)$-approximation; see Section 4. This new bound is always better than the bound $2 \cdot 3^{\alpha-1}$ of Funke et al. and of our analysis of the T^3-algorithm.

Finally, in Section 5, we turn our attention to the following two questions: (a) How does the approximability of Tsp behave when we make α larger than one? (b) Does allowing revisits change the complexity or the approximability of the problem? As we know, classical Euclidean Tsp (that is, Tsp$(d, 1)$) is NP-hard [19] and has a polynomial-time approximation scheme (PTAS) in any fixed number d of dimensions [4]. On the other hand, Rev-Tsp(d, α) has—to the best of our knowledge—not been studied before. Concerning question (b), complexity behaves as expected: Rev-Tsp(d, α) is NP-hard for any $d \geqslant 2$ and any $\alpha > 0$, and our (straightforward) hardness argument also works for Tsp(d, α). In terms of approximability, we show that whereas the two-dimensional problem Rev-Tsp$(2, \alpha)$ still has a PTAS for all values $\alpha \geqslant 2$, the problem becomes APX-hard for all $\alpha > 1$ in three dimensions. We were surprised that the APX-hardness proof, too, carried over to Tsp$(3, \alpha)$ for all $\alpha > 1$. This inapproximability result stands in strong contrast to the behavior of the classical Euclidean Tsp (the case $\alpha = 1$).

The connection to wireless networks. Consider a wireless network whose nodes are equipped with omni-directional antennas. The nodes are modeled as points in the plane, and every node can communicate with all other nodes that are within its transmission radius. The power (that is, the energy) needed to achieve a transmission radius of r is roughly proportional to r^α for some real parameter α called the *distance-power gradient.* Depending on environmental conditions, α typically is in the range 2 to 6 [13, Chapter 1]. The goal is to assign powers to the nodes such that the resulting network has certain desirable properties, while the overall power consumption is minimized. A widely studied variant has the objective to make the resulting network strongly connected [1, 11, 16]. Other variants (finding broadcast trees; having small hop diameter; etc) have been studied as well. Funke et al. [12] suggest that it is useful to have a tour through the network, which can be used to pass a *virtual token* around. The resulting power-assignment problem is Tsp$(2, \alpha)$.

Another setting related to Tsp$(2, \alpha)$ is the following. Instead of omni-directional antennas, some wireless networks use directional antennas. This achieves the same transmission radius under a smaller energy consumption [17, 22]. To model directional antennas, Caragiannis et al. [9] assume that a node can communicate with other nodes in a circular sector of a given angle (where the sector's radius is still determined by the power of the node's signal). For directional antennas one not only has to assign a power level to each node, but also has to decide on the direction in which each node transmits. If the opening angle tends to zero and the points are in general position, a strongly connected network becomes a tour. Hence, our results on Tsp$(2, \alpha)$ may shed some light on the difficulty of power assignment for directional antennas with small opening angles.

2. Approximating $\textsc{Tsp}(\cdot\,,\alpha)$

In this section we lay the basis for our main contribution, a 5-approximation for $\textsc{Tsp}(2,2)$ in Section 3. We review known algorithms for a related version of $\textsc{Tsp}$, which can be applied to our setting. As it turns out, these algorithms already yield the same worst-case bounds as the algorithm that Funke et al. [12] gave recently.

We recall some definitions. Let S be a set, let $\operatorname{dist}(\cdot,\cdot) : S \times S \to \mathbb{R}_{\geqslant 0}$ be a distance function on S, and let $\tau \geqslant 1$. We say that $\operatorname{dist}(\cdot,\cdot)$ fulfills the *τ-relaxed triangle inequality* if any three elements $p, q, r \in S$ satisfy $\operatorname{dist}(p,r) \leqslant \tau \cdot (\operatorname{dist}(p,q) + \operatorname{dist}(q,r))$. Recall that we denote by Δ_τ-$\textsc{Tsp}$ the $\textsc{Tsp}$ problem on complete graphs whose weight function (when viewed as a distance function on the vertices) fulfills the τ-relaxed triangle inequality. The following lemma, which has been observed by Funke et al. [12], allows us to apply algorithms for Δ_τ-$\textsc{Tsp}$ to our problem. The proof relies on Hölder's inequality.

Lemma 2.1 ([12]). *Let $\alpha > 0$ be a fixed constant. The distance function $|\cdot|^\alpha : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geqslant 0}, (p,q) \mapsto |pq|^\alpha$ fulfills the τ-relaxed triangle inequality for $\tau = 2^{\alpha-1}$.*

Andreae and Bandelt [3] gave an approximation algorithm for Δ_τ-$\textsc{Tsp}$. Their T^3-algorithm is an adaptation of the well-known double-spanning-tree heuristic for $\textsc{Tsp}$. This heuristic finds a minimum spanning tree (MST) in the given graph G, doubles all edges, finds an Euler tour in the resulting multigraph, and finally constructs a Hamiltonian cycle from the Euler tour by skipping all nodes that have already been visited. The weight of the MST is a lower bound for the length of a $\textsc{Tsp}$-tour since removing any edge from a tour yields a spanning tree whose weight is at least the weight of the MST. Note that this statement holds for arbitrary weight functions. If the triangle inequality holds, the heuristic yields a 2-approximation since then skipping over visited nodes never increases the length of the tour, which initially equals twice the weight of the MST. For the weight function $|\cdot|^\alpha$, however, the heuristic can perform arbitrarily badly—consider a sequence of n equally-spaced points on a line.

The T^3-algorithm of Andreae and Bandelt also creates a Hamiltonian tour by short-cutting the MST, but their algorithm never skips more than two consecutive nodes. It is never necessary to skip more than two consecutive nodes because the cube T^3 of a tree T is always Hamiltonian by a result of Sekanina [24]. Recall that the cube of a graph G contains an edge uv if there is a path from u to v in G that uses at most three edges. The proof of Sekanina is constructive; Andreae and Bandelt use it to construct a tour in MST^3.

The recursive procedure of Sekanina [24] to obtain a Hamiltonian cycle in T^3 intuitively works as illustrated in Fig. 1; for the pseudo-code, see Algorithm 1. The algorithm is applied to a tree T and an edge $e = u_1u_2$ of T. Removing the edge e splits the tree into two components T_1 and T_2. In each component T_i ($i = 1, 2$), the algorithm selects an arbitrary edge $e_i = u_iw_i$ incident to u_i and recursively computes a Hamiltonian cycle of T_i that includes the edge e_i. The algorithm returns a Hamiltonian cycle of T that includes e. The cycle consists of the cycles in T_1 and T_2 without the edges e_1 and e_2, respectively. The two resulting paths are stitched together with the help of e and the new edge w_1w_2.

Note that different choices of the edge e_i in line 5 give rise to different versions of the algorithm. The standard T^3-algorithm takes an arbitrary such edge, while Andreae's refined version [2] makes a specific choice, which gives a better result. (In the next section we will choose e_i based on the local geometry of the MST, which will lead to an improved result for our problem.) Andreae's tour in MST^3 has weight at most $(\tau^2 + \tau)$ times the weight of the MST, which is worst-case optimal [3]. Combining his result with Lemma 2.1 yields that

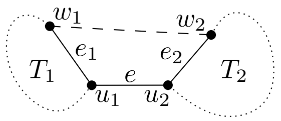


Figure 1: Recursively finding a Hamiltonian cycle in the cube of the tree T.

```
Algorithm 1: CycleInCube($T$, $e = u_1u_2$)
1 for $i \leftarrow 1$ to 2 do
2     $T_i \leftarrow$ component of $T - e$ that contains $u_i$
3     if $|T_i| = 1$ then $P_i \leftarrow \emptyset$; $w_i \leftarrow u_i$
4     else
5         pick an edge $e_i = u_iw_i$ incident to $u_i$ in $T_i$
6         if $|T_i| = 2$ then $\Pi_i \leftarrow e_i$
7         else $\Pi_i \leftarrow$ CycleInCube($T_i, e_i$) $- e_i$
8 return $\Pi_1 + e + \Pi_2 + w_1w_2$
```

the refined T^3-algorithm is a $(4^{\alpha-1} + 2^{\alpha-1})$-approximation for Tsp$(\,\cdot\,, \alpha)$. We now improve on this with the help of a simple argument. We will frequently use the following definition. Let T be a tree and let $v_0, \dots, v_k$ be a simple path in T. Then we call v_0v_k a *k-shortcut* of T. We say that a shortcut *vw uses* an edge e if e lies on the path connecting v and w in T. It is not hard to see that the weight of a k-shortcut can be bounded as follows.

Lemma 2.2. *Let $\alpha \geqslant 1$ and let e be a k-shortcut using edges $e_1, \dots, e_k$. Then $|e|^\alpha \leqslant k^{\alpha-1} \sum_{i=1}^{k} |e_i|^\alpha$.*

Given a tree T, the tour constructed by the T^3-algorithm consists of edges of T and 2- and 3-shortcuts that use edges of T. Note that in this tour each edge of T is used exactly twice. Thus, for $\alpha \geqslant 2$, the original T^3-algorithm does actually better than the bound we obtained above for the refined T^3-algorithm.

Corollary 2.3. *Every version of the* T^3*-algorithm is a* $(2{\cdot}3^{\alpha-1})$*-approximation for* Tsp$(\,\cdot\,, \alpha)$.

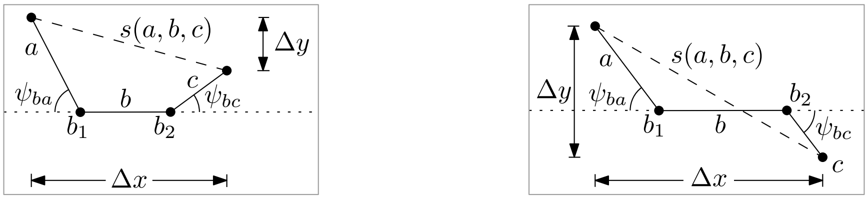
Note that our improved analysis of the T^3-algorithm yields the same result as the algorithm of Funke et al. [12].

Bender and Chekuri [6] designed a 4τ-approximation for Δ_τ-Tsp using a different lower bound: the optimal Tsp tour is a biconnected subgraph of the original graph. The weight of the optimal Tsp tour is at least that of the minimum-weight biconnected subgraph. The latter is np-hard to compute [10], but can be approximated within a factor of 2 [21]. Moreover, the square of a biconnected subgraph is always Hamiltonian. Thus using only edges of the biconnected subgraph and *two*-shortcuts yields a 4τ-approximation for Δ_τ-Tsp. Combining the result of Bender and Chekuri with Lemma 2.1 immediately yields the following result, which is better than Corollary 2.3 for $\alpha > \log_{3/2} 3 \approx 2.71$.

Corollary 2.4. *The algorithm of Bender and Chekuri is a* $2^{\alpha+1}$*-approximation for* Tsp$(\,\cdot\,, \alpha)$.

3. A 5-Approximation for Tsp(2,2)

In the previous section we have used graph-theoretic arguments to determine the performance of the T^3-algorithm. By Corollary 2.3, the T^3-algorithm yields a 6-approximation for $\alpha = 2$, independently of the dimension of the underlying Euclidean space. We now define what we call the *geometric* T^3-algorithm and show that it yields a 5-approximation for Tsp$(2, 2)$. The geometric T^3-algorithm simply chooses in line 5 of Algorithm 1 the edge e_i that makes the smallest angle with the edge e.



(a) a and c lie on the same side of the line through b

(b) a and c lie on different sides of the line through b

Figure 2: Two cases for computing the length of the 3-shortcut $s(a,b,c)$.

The idea behind taking advantage of geometry is as follows. In Corollary 2.3 we have exploited the fact that each edge is used in two ($\leqslant 3$)-shortcuts. The weight of a 3-shortcut is maximum if the corresponding points lie on a line. For the case of the Euclidean MST it is well-known that edges make an angle of at least $\pi/3$ if they share an endpoint. The same proof also works for the MST w.r.t. $|\cdot|^\alpha$. This guarantees that in line 5 of Algorithm 1, we can pick an edge e_i that makes a relatively small angle with e—if the degree of u_i is larger than 2. Otherwise, it is easy to see that e_i is used by a ($\leqslant 2$)- and a ($\leqslant 3$)-shortcut, which is favorable to being used by two 3-shortcuts, see Lemma 2.2.

Although the intuition behind our geometric T^3-algorithm is clear, its analysis turns out to be non-trivial. We start with the following lemma that can be proved with some elementary trigonometry. Given two line segments s and t incident to the same point, we denote the smaller angle between s and t by $\angle st$ and define $\psi_{st} = \pi - \angle st$.

Lemma 3.1. *Given a tree T, the 3-shortcut $s(a,b,c)$ that uses the edges a, b, c of T in this order has weight*

$$|s(a,b,c)|^2 \quad = \quad |a|^2 + |b|^2 + |c|^2 + 2|a||b|\cos\psi_{ba} + 2|b||c|\cos\psi_{bc} + 2|a||c|\cos(\psi_{ba} + \delta\cdot\psi_{bc}),$$

where $\delta = +1$ if a and c lie on the same side of the line through b, and $\delta = -1$ if a and c lie on opposite sides. Moreover, $|s(a,b,c)|^2 \leqslant 2|a|^2 + |b|^2 + 2|c|^2 + 2|a||b|\cos\psi_{ba} + 2|b||c|\cos\psi_{bc}$.

Lemma 3.1 (illustrated in Fig. 2) expresses the weight of a 3-shortcut in terms of the lengths of the edges and the angles between them. Now we show that if an edge a is used in two 3-shortcuts, two of these angles are related. Note that the T^3-algorithm generates the two 3-shortcuts that use a in two consecutive recursive calls, see Fig. 3. The T^3-algorithm is first applied to edge b and then recursively to edge a. In the recursive call, the shortcut $s(e,a,d)$ is generated where d is an edge incident to both a and b. Then the algorithm returns from the recursion and generates the 3-shortcut $s(a,b,c)$. Thus a is the middle edge in one 3-shortcut and the first or last edge in the other 3-shortcut. We rely on the following.

Lemma 3.2. *If the geometric* T^3*-algorithm generates the two 3-shortcuts $s(a,b,c)$ and $s(e,a,d)$ in two recursive calls and d is incident to both a and b, then $\psi_{ba} \geqslant (\pi - \psi_{ad})/2$.*

Now we are ready to prove the main result of this section.

Theorem 3.3. *The geometric* T^3*-algorithm yields a 5-approximation for* TSP$(2,2)$.

Proof. We express the length of each shortcut s of the T^3-tour in terms of the lengths of the MST edges that s uses. Changing the perspective, for each MST edge a, we use contrib(a) to denote the sum of all terms that contain the factor $|a|$. The edge a is used in at most two shortcuts. Bounding their lengths yields an upper bound on contrib(a). The sum of all contributions relates the length of the T^3-tour to that of the MST (w.r.t. $|\cdot|^\alpha$), which in turn is a lower bound for the length of an optimal TSP tour.

Due to Lemma 2.2, contrib$(a) \leqslant 5|a|^2$ if a is used in a ($\leqslant$2)-shortcut on one side and a ($\leqslant$3)-shortcut on the other side. So we focus on the case that a is used in two 3-shortcuts, see Fig. 3. We rewrite the composite terms in the bound for $s(a,b,c)$ in Lemma 3.1 using Young's inequality with ε, which, given $x, y \in \mathbb{R}$ and $\varepsilon > 0$, states that $xy \leqslant x^2/(2\varepsilon) + y^2\varepsilon/2$.

Let v be the vertex that is incident to edges a and b. If there are multiple 3-shortcuts that use edges that are incident to v then the T^3-algorithm generates these in consecutive recursive calls. We renumber the edges incident to v such that the algorithm is first applied to vv_1, then recursively to vv_2 etc. Then there is some $i \geqslant 1$ such that $b = vv_i$ and $a = vv_{i+1}$ because the algorithm is first applied to b and then recursively to a. We define $\psi_i = \psi_{vv_i, vv_{i+1}} (= \psi_{ba})$. We rewrite the term $2|a||b|\cos\psi_{ba}$ in the bound for $|s(a,b,c)|^2$ in Lemma 3.1 as follows.

$$2|a||b|\cos\psi_{ba} = 2|vv_i||vv_{i+1}|\cos\psi_i \leqslant f(|vv_{i+1}|, |vv_i|, \psi_i), \tag{3.1}$$

where

$$f(|vv_{i+1}|, |vv_i|, \psi_i) = \begin{cases} 0 & \text{if } \psi_i \geqslant \frac{\pi}{2}, \\ |vv_i|^2 + |vv_{i+1}|^2 \cos^2\psi_i & \text{if } \psi_i < \frac{\pi}{2} \text{ and} \\ & \big(i = 1 \text{ or } \big(i > 1 \text{ and } \psi_{i-1} \geqslant \frac{\pi}{2}\big)\big), \\ \big(|vv_i|^2 + |vv_{i+1}|^2\big)\cos\psi_i & \text{if } \psi_i < \frac{\pi}{2} \text{ and } i > 1 \text{ and } \psi_{i-1} < \frac{\pi}{2}. \end{cases}$$

The second case of inequality (3.1) follows from Young's inequality with $\varepsilon = 1/\cos\psi_i$ and the third case from Young's inequality with $\varepsilon = 1$. Replacing $2|b||c|\cos\psi_{bc}$ in the bound for $|s(a,b,c)|^2$ in Lemma 3.1 is analogous. Together, the two replacements yield the bound

$$|s(a,b,c)|^2 \leqslant 2|a|^2 + |b|^2 + 2|c|^2 + f(|a|, |b|, \psi_{ba}) + f(|c|, |b|, \psi_{bc}). \tag{3.2}$$

We use (3.2) to bound the weights of all 3-shortcuts. The weight of the final tour is the sum of the weights of all shortcuts. In this sum we can take the two occurrences of an edge $a = vv_{i+1}$ together and analyze the contribution of a to the tour. Note that the result of (3.2) is still at most $3(|a|^2 + |b|^2 + |c|^2)$. So if an edge a is used in a ($\leqslant$3)-shortcut on one side and a ($\leqslant$ 2)-shortcut on the other side, then we still have that contrib$(a) \leqslant 5|a|^2$. It remains to consider the case that a is used in two 3-shortcuts. Let $s(a,b,c)$ and $s(e,a,d)$ be these 3-shortcuts. The algorithm is first applied to edge b and generates shortcut $s(a,b,c)$, where a is the first or the third edge of the shortcut. Then the algorithm is recursively applied to edge a and generates shortcut $s(e,a,d)$, where a is the middle edge. Fig. 3 shows how the vertices are numbered in this case.

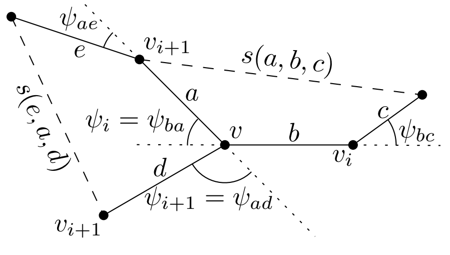


Figure 3: Two 3-shortcuts that use edge a.

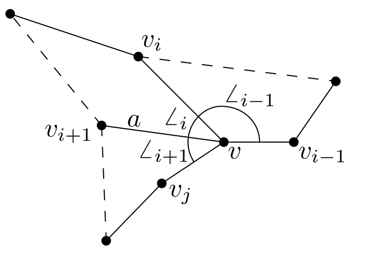


Figure 4: Illustration of case III.

Let σ_a be a function that takes a sum of terms and returns the sum of all terms that contain $|a|$. We derive the following expression for contrib(a).

$$\begin{aligned}\text{contrib}(a) &= \sigma_a(\text{weight}(s(a,b,c))) + \sigma_a(\text{weight}(s(e,a,d)))\\ &\leqslant \sigma_a\left(2|a|^2 + |b|^2 + 2|c|^2 + f(|a|,|b|,\psi_{ba}) + f(|c|,|b|,\psi_{bc})\right)\\ &\quad + \sigma_a\left(2|e|^2 + |a|^2 + 2|d|^2 + f(|e|,|a|,\psi_{ae}) + f(|d|,|a|,\psi_{ad})\right)\\ &\leqslant 4|a|^2 + \sigma_a(f(|vv_{i+1}|,|vv_i|,\psi_i)) + \sigma_a(f(|vv_{i+2}|,|vv_{i+1}|,\psi_{i+1})) \qquad (3.3)\end{aligned}$$

By definition of f we have to consider three cases in (3.3) for contrib(a).

Case I: $\psi_i \geqslant \pi/2$ or $\psi_{i+1} \geqslant \pi/2$.

We assume w.l.o.g. that $\psi_i \geqslant \pi/2$. Then we know that $f(|vv_{i+1}|,|vv_i|,\psi_i)) = 0$ and in the worst case $\sigma_a(f(|vv_{i+2}|,|vv_{i+1}|,\psi_{i+1})) \leqslant |a|^2$. Thus we have that contrib$(a) \leqslant 5|a|^2$.

Case II: $\psi_i < \pi/2$ and $\psi_{i+1} < \pi/2$ and ($i = 1$ or ($i > 1$ and $\psi_{i-1} \geqslant \pi/2$)).

By definition of f we have:

$$\begin{aligned}\sigma_a(f(|vv_{i+1}|,|vv_i|,\psi_i)) &= \sigma_a\left(|vv_i|^2 + |vv_{i+1}|^2\cos^2\psi_i\right) &&= |a|^2\cos^2\psi_i\\ \sigma_a(f(|vv_{i+2}|,|vv_{i+1}|,\psi_{i+1})) &= \sigma_a\left((|vv_{i+1}|^2 + |vv_{i+2}|^2)\cos\psi_{i+1}\right) &&= |a|^2\cos\psi_{i+1}\end{aligned}$$

Lemma 3.2 states that $\psi_i \geqslant (\pi - \psi_{i+1})/2$. We also know that $\psi_i \leqslant \pi$ by definition. Thus we have

$$\text{contrib}(a) \leqslant \left(4 + \cos^2\frac{\pi - \psi_{i+1}}{2} + \cos\psi_{i+1}\right)|a|^2 \;\leqslant\; 5|a|^2.$$

Case III: $\psi_i < \pi/2$ and $\psi_{i+1} < \pi/2$ and $i > 1$ and $\psi_{i-1} < \pi/2$.

It can be shown that this leads to a contradiction, see Fig. 4 (on page 245).

In cases I and II, the contribution of any edge $|a|$ to the tour is at most $5|a|^2$. The theorem follows by summing up the contributions of all edges. ■

When using the MST as a lower bound in the analysis, there is not much room for improvement. There are instances of Tsp(2,2) where the T^3-algorithm yields a tour whose weight is $4\frac{4}{11}$ times that of the MST; see also [18, Theorem 4.19].

4. Approximating Tsp$(2,\alpha)$ with $\alpha \geqslant 2$

In this section we generalize the main result of the previous section to $\alpha \geqslant 2$. Our new bound is always better than the bound $2 \cdot 3^{\alpha-1}$ of Funke et al. [12], see also Corollary 2.3. For $\alpha < 3.41$ our bound is better than the bound $2^{\alpha+1}$ that follows from the algorithm of Bender and Chekuri [6], see Corollary 2.4.

Theorem 4.1. *The geometric T^3-algorithm yields a $(3^{\alpha-1} + \sqrt{6}^{\alpha}/3)$-approximation for* Tsp$(2,\alpha)$ *if $\alpha \geqslant 2$.*

Proof. If an edge a is used in a ($\leqslant$2)-shortcut on one side and a ($\leqslant$3)-shortcut on the other side then the total contribution of a to the tour is at most $(2^{\alpha-1} + 3^{\alpha-1})|a|^\alpha$ by Lemma 2.2. So we will focus our analysis again on the case that a is used in two 3-shortcuts. For $\alpha = 2$ we can express the weight of a 3-shortcut by Lemma 3.1 and rewrite the composite terms as in inequality (3.1). For $\alpha > 2$ we apply Hölder's inequality.

$$\begin{aligned} |s(a,b,c)|^\alpha &= \left(|s(a,b,c)|^2\right)^{\alpha/2} \\ &\leqslant \left(2|a|^2 + |b|^2 + 2|c|^2 + f(|a|,|b|,\psi_{ba}) + f(|c|,|b|,\psi_{bc})\right)^{\alpha/2} \\ &= \left(\beta_a|a|^2 + \beta_b|b|^2 + \beta_c|c|^2\right)^{\alpha/2} \qquad (4.1) \\ &\leqslant 3^{\alpha/2-1}\left(\beta_a^{\alpha/2}|a|^\alpha + \beta_b^{\alpha/2}|b|^\alpha + \beta_c^{\alpha/2}|c|^\alpha\right) \qquad (4.2) \end{aligned}$$

We introduced the constants of type β to shorten the expression. Note that the last inequality holds only if $\alpha > 2$.

In order to bound the contribution of an edge a that is used in two 3-shortcuts we follow the proof of Theorem 3.3. Since the assumptions of case III in that proof led to a contradiction, it suffices to consider cases I and II.

Case I: $\psi_i \geqslant \pi/2$ or $\psi_{i+1} \geqslant \pi/2$.

$$\begin{aligned} \operatorname{contrib}(a) &\leqslant 3^{\alpha/2-1}\left((2+\cos\psi_i)^{\alpha/2} + (2+\cos\psi_{i+1})^{\alpha/2}\right)|a|^\alpha \\ &\leqslant 3^{\alpha/2-1}\left(2^{\alpha/2} + 3^{\alpha/2}\right)|a|^\alpha \;=\; \left(3^{\alpha-1} + \sqrt{6}^{\,\alpha}/3\right)|a|^\alpha \end{aligned}$$

Case II: $\psi_i < \pi/2$ and $\psi_{i+1} < \pi/2$ and ($i = 1$ or ($i > 1$ and $\psi_{i-1} \geqslant \pi/2$)).

$$\operatorname{contrib}(a) \leqslant 3^{\alpha/2-1}\underbrace{\left((2+\cos\psi_{i+1})^{\alpha/2} + \left(2+\sin^2\psi_{i+1}/2\right)^{\alpha/2}\right)}_{g_\alpha(\psi_{i+1})}|a|^\alpha$$

Now we use the fact that the function $h : [0, 2\pi] \to \mathbb{R}, x \mapsto (2+\cos x)^k + (2+\sin^2 x/2)^k$ attains its maximum value at $x = 0$. Thus g_α also attains its maximum in the range $[0, \pi/2)$ in $x = 0$. This yields

$$\operatorname{contrib}(a) \;\leqslant\; 3^{\alpha/2-1}\cdot g_\alpha(0)\cdot|a|^\alpha \;\leqslant\; \left(3^{\alpha-1} + \sqrt{6}^{\,\alpha}/3\right)|a|^\alpha.$$

In both cases we showed that $\operatorname{contrib}(a) \leqslant (3^{\alpha-1} + \sqrt{6}^{\,\alpha}/3)|a|^\alpha$. The theorem follows for $\alpha > 2$ by summing up the contributions of all edges. The case $\alpha = 2$ corresponds to Theorem 3.3. ∎

5. The Approximability of Tsp and Rev-Tsp

In this section we discuss complexity and approximability of Tsp and its variant Rev-Tsp, where the salesman is allowed to revisit the cities. Recall that for any fixed dimension $d \geqslant 2$, Tsp$(d, 1)$ is NP-hard [19] and admits a PTAS [4].

Theorem 5.1. *Tsp(d,α) and Rev-Tsp(d,α) are NP-hard for any $d \geqslant 2$ and $\alpha > 0$.*

Proof. Itai et al. [14] showed that, given n points in the unit grid, it is NP-hard to decide whether there is a Tsp tour of Euclidean length n. Thus for both of our problems it is NP-hard to distinguish between OPT $= n$ and OPT $\geqslant n - 1 + \sqrt{2}^{\,\alpha}$. ∎

Theorem 5.2. *T*SP(d,α) *and Rev-*TSP(d,α) *are* APX*-hard for any* $d \geqslant 3$ *and any* $\alpha > 1$.

Proof. We only discuss the case $d = 3$ and $\alpha = 2$—all other cases can be settled by slightly modified arguments—TSP. and we only consider Rev-TSP; a similar reduction can be used for TSP. We reduce from $\{1,2\}$-TSP, the TSP on the complete graph where the weight of every edge is either 1 or 2; this problem is APX-hard [20]. An instance of $\{1,2\}$-TSP consists of the complete graph $K_n = (V_n, E_n)$ with vertex set $V_n = \{v_1, \dots, v_n\}$, edge set $E_n = \{e_1, \dots, e_m\}$ where $m = n(n-1)/2$, and edge lengths that are specified by a weight function $w : E_n \to \{1,2\}$. Given K_n and w, we construct a corresponding instance $P_{n,w} \subset \mathbb{R}^3$ of Rev-TSP$(3,2)$.

We start our construction by introducing several auxiliary line segments. For each vertex $v_i \in V_n$ we define its *spine* to be the vertical line segment going from point (ni, ni, n) to point (ni, ni, nm). For each edge $e_k = v_i v_j \in E_n$ with $i < j$, we define two corresponding line segments that are parallel to the xy-plane and that are called *bones*. The first bone connects point (ni, ni, nk) on the spine of v_i to the point (nj, ni, nk). The other bone connects point (nj, nj, nk) on the spine of v_j to the point $(nj, ni - \delta_k, nk)$, where $\delta_k = 1$ if $w(e_k) = 1$ and $\delta_k = \sqrt{2}$ if $w(e_k) = 2$. Note that these two bones do not quite touch; they are separated by a gap of length δ_k.

In order to get the instance $P_{n,w}$ of Rev-TSP$(3,2)$, we subdivide every single (spine or bone) line segment introduced above by a dense, evenly distributed set of points—we call these points *cities* from now on—so that every unit-length piece receives n^5 cities. The distance between adjacent cities is $1/n^5$, and so the cost for going from one city to an adjacent city is $1/n^{10}$. All these cities together form instance $P_{n,w}$, and this completes our construction. Since we have introduced line segments with a total length of at most $n \cdot n(m-1) + m \cdot 2n(n-1) < 2n^4$, the overall number of cities is at most $2n^9$.

For $1 \leqslant i \leqslant n$ we call the cities on the spine of v_i and on all bones incident to this spine the *city cluster* of v_i. Traversing all cities within such a city cluster is very cheap; even if we visit every city twice, this costs at most $2 \cdot 2n^9/n^{10} = 4/n$ for all cities in all city clusters together. In a traveling salesman tour, the only expensive steps occur when the salesman jumps from one city cluster to another city cluster. By the above definition of δ_k, when jumping from bone to bone across the gap corresponding to edge e_k the incurred cost is exactly $w(e_k)$. Note that jumping from city cluster to city cluster in any other way would be much more expensive and would thus not reduce the total cost of the tour.

Finally, let us show that our reduction is approximation preserving. Fix an ε with $0 < \varepsilon < 1$. Consider an instance K_n and w of $\{1,2\}$-TSP, and assume without loss of generality that $n > 4/\varepsilon$. Consider an optimal tour π_0 for this instance. If π_0 uses $\ell \geqslant 0$ edges of length 2 and $n - \ell$ edges of length 1, then it has cost $n + \ell$. Given a PTAS for Rev-TSP, we show how to compute in polynomial time a tour of cost at most $(1+\varepsilon)(n+\ell)$ for K_n and w.

First note that the tour π_0 can be transformed into a tour π_1 through $P_{n,w}$ that makes ℓ jumps of cost 2 and $n - \ell$ jumps of cost 1. That tour π_1 costs at most $n + \ell + 4/n$. Using our hypothetical PTAS for Rev-TSP, we can compute for any $\varepsilon' > 0$ in polynomial time a tour π_2 through $P_{n,w}$ of cost at most $(1+\varepsilon')c_{\text{opt}}$, where c_{opt} is the cost of an optimal Rev-TSP tour. The existence of π_1 yields $c_{\text{opt}} \leqslant n+\ell+4/n$. The tour π_2 can be transformed into a tour π_3 through K_n: Just map the jumps of π_2 to the corresponding edges of K_n. Since this mapping cannot increase the cost, tour π_3 costs at most $(1+\varepsilon')(n+\ell+4/n)$.

Choosing $\varepsilon' = \varepsilon/2$ and using $4/n < \varepsilon < 1$, we can bound the cost of π_3 from above by

$$\left(1+\frac{\varepsilon}{2}\right)(n+\ell)+\left(1+\frac{\varepsilon}{2}\right)\varepsilon \;=\; \left(1+\frac{\varepsilon}{2}\right)(n+\ell)+\frac{\varepsilon}{2}(2+\varepsilon) \;=\; (1+\varepsilon)(n+\ell)$$

as desired. Like π_2, the tour π_3 may visit vertices more than once. This can be fixed by greedily introducing shortcuts. The shortcuts do not increase the cost of the tour since the weight function w (trivially) fulfills the triangle inequality. ∎

Theorem 5.3. *There exists a PTAS for Rev-*TSP$(2,\alpha)$ *for any* $\alpha \geqslant 2$.

Proof. Given a set P of points in the plane, consider the *Gabriel graph* G_P that has a vertex for each point in P. There is an edge between points p and q, if the open disk with diameter pq is empty, in other words, if for all points $r \in P \setminus \{p,q\}$, the angle $\angle prq$ is at most $\pi/2$. The weight of the edge is $|pq|^\alpha$. Note that $|pr|^\alpha + |rq|^\alpha \leqslant |pq|^\alpha$ if $\angle prq$ is at least $\pi/2$. Therefore, there is an optimal TSP tour with revisits through P that only uses the edges of G_P: Indeed, if a tour uses an edge pq for which there is a point r with $\angle prq \geqslant \pi/2$, then replacing pq by pr and rq would shorten the tour. Such a replacement is feasible since revisiting city r is allowed. The Gabriel graph is planar. Hence we end up with an instance of the TSP on weighted planar graphs, for which a PTAS is known [5]. ∎

Recall that a *quasi*-PTAS is an approximation scheme with running time $n^{\mathrm{polylog}\, n}$, where n is the size of the input. The following result follows immediately from the facts that (a) the metric $|\cdot|^\alpha$ has bounded doubling dimension and (b) TSP on metrics of bounded doubling dimension admits a quasi-PTAS [25].

Theorem 5.4. *There exists a quasi-PTAS for Rev-*TSP(d,α) *for any* $\alpha \in (0,1]$ *and* $d \geqslant 1$.

6. Conclusions

In order to construct considerably better approximation algorithms for TSP(d,α), we expect that substantially different methods of analysis have to be found. A result of Van Nijnatten [18, Theorem 4.19] indicates that there is not much room left for improvement as long as we compare to the MST.

The approximability of Rev-TSP$(2,\alpha)$ for $1 < \alpha < 2$ is an interesting open question. We believe that a (quasi)-PTAS may be obtained using the framework of the PTAS for weighted planar graph TSP by Arora et al. [5]. A simple reduction shows that deriving a PTAS for our problem is at least as hard as deriving a PTAS for weighted planar graph TSP. Assume we have a PTAS for Rev-TSP$(2,\alpha)$ for some $\alpha > 1$. Given a weighted planar graph and a planar embedding, we replace each edge by a dense set of points such that traversing a subedge basically costs zero. By making one subedge of each edge e longer, we can make the cost of that subedge (and thus of e) in Rev-TSP proportional to the weight of e. Then, the costs of the optimal solutions of the two problems will be the same up to an arbitrarily small constant factor of $1+\varepsilon$. Such a reduction is polynomially bounded if all weights are polynomially bounded, which can be achieved by a standard rounding scheme.

A PTAS for Rev-TSP$(2,\alpha)$ for any $\alpha > 1$ would be an interesting generalization of the existing PTAS's for weighted planar graphs. Ideally, one would have a PTAS with running time independent of α since it would contain both Euclidean TSP and weighted planar graph TSP as special cases.

References

[1] E. Althaus, G. Călinescu, I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment for symmetric connectivity in static ad hoc wireless networks. *Wireless Networks*, 12(3):287–299, 2006.

[2] T. Andreae. On the traveling salesman problem restricted to inputs satisfying a relaxed triangle inequality. *Networks*, 38(2):59–67, 2001.

[3] T. Andreae and H.-J. Bandelt. Performance guarantees for approximation algorithms depending on parametrized triangle inequalities. *SIAM J. Discrete Math.*, 8(1):1–16, 1995.

[4] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.

[5] S. Arora, M. Grigni, D. Karger, P. Klein, and A. Woloszyn. A polynomial-time approximation scheme for weighted planar graph TSP. In *Proc. 9th Annu. ACM-SIAM Symp. Discr. Algo.*, p. 33–41, 1998.

[6] M. A. Bender and C. Chekuri. Performance guarantees for the TSP with a parameterized triangle inequality. *Inform. Process. Lett.*, 73(1-2):17–21, 2000.

[7] H.-J. Böckenhauer, J. Hromkovič, R. Klasing, S. Seibert, and W. Unger. Towards the notion of stability of approximation for hard optimization tasks and the traveling salesman problem. *Theor. Comput. Sci.*, 285(1):3–24, 2002.

[8] B. Bollobás. *The Art of Mathematics – Coffee Time in Memphis.* Cambridge Univ. Press, 2006.

[9] I. Caragiannis, C. Kaklamanis, E. Kranakis, D. Krizanc, and A. Wiese. Communication in wireless networks with directional antennas. In F. M. auf der Heide and N. Shavit, editors, *Proc. 20th Annu. ACM Symp. Parallel Algorithms Architect.*, p. 344–351, 2008.

[10] K. P. Eswaran and R. E. Tarjan. Augmentation problems. *SIAM J. Comput.*, 5(4):653–665, 1976.

[11] B. Fuchs. On the hardness of range assignment problems. *Networks*, 52(4):183–195, 2008.

[12] S. Funke, S. Laue, R. Naujoks, and Z. Lotker. Power assignment problems in wireless communication: Covering points by disks, reaching few receivers quickly, and energy-efficient travelling salesman tours. In *Proc. 4th Int. IEEE Conf. Distributed Comput. Sensor Systems*, LNCS 5067, p. 282–295, 2008.

[13] L. Godara. *Handbook of Antennas in Wireless Communications.* CRC Press, 2001.

[14] A. Itai, C. Papadimitriou, and J. Szwarcfiter. Hamilton paths in grid graphs. *SIAM J. Comput.*, 4:676–686, 1982.

[15] R. Karp and J. Steele. Probabilistic analysis of heuristics. In E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys, editors, *The Traveling Salesman Problem*, chapter 6, p. 181–205. John Wiley, 1985.

[16] L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoret. Comput. Sci.*, 243(1–2):289–305, 2000.

[17] A. Nasipuri, K. Li, and U. R. Sappidi. Power consumption and throughput in mobile ad hoc networks using directional antennas. In *Proc. 11th IEEE Conf. Comput. Commun. & Networks*, p. 620–626, 2002.

[18] F. van Nijnatten. Range assignment with directional antennas. Master's thesis, TU Eindhoven, 2008. `http://alexandria.tue.nl/extra1/afstversl/wsk-i/nijnatten2008.pdf`.

[19] C. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoret. Comput. Sci.*, 4(3):237–244, 1977.

[20] C. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Math. Oper. Res.*, 18(1):1–11, 1993.

[21] M. Penn and H. Shasha-Krupnik. Improved approximation algorithms for weighted 2- and 3-vertex connectivity augmentation problems,. *J. Algorithms*, 22(1):187–196, 1997.

[22] R. Ramanathan. On the performance of ad hoc networks with beamforming antennas. In *Proc. 2nd ACM Int. Symp. Mobile Ad Hoc Networking & Comput.*, p. 95–105, 2001.

[23] S. B. Rao and W. D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *Proc. 30th Annu. ACM Symp. Theory Comput.*, pages 540–550, 1998.

[24] M. Sekanina. On an ordering of the set of vertices of a connected graph. *Publications of the Faculty of Science, University of Brno*, 412:137–142, 1960.

[25] K. Talwar. Bypassing the embedding: Algorithms for low dimensional metrics. In *Proc. 36th Annu. ACM Symp. Theory Comput.*, pages 281–290, 2004.