The Parameterized Complexity of Local Search for TSP, More Refined

Jiong Guo[1⋆], Sepp Hartung[2], Rolf Niedermeier[2], and Ondřej Suchý[1⋆]

[1] Universität des Saarlandes, Saarbrücken, Germany
{jguo, suchy}@mmci.uni-saarland.de
[2] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Berlin, Germany
{sepp.hartung, rolf.niedermeier}@tu-berlin.de

Abstract. We extend previous work on the parameterized complexity of local search for the Travelling Salesperson Problem (TSP). So far, its parameterized complexity has been investigated with respect to the distance measures (which define the local search area) "Edge Exchange" and "Max-Shift". We perform studies with respect to the distance measures "Swap" and "m-Swap", "Reversal" and "m-Reversal", and "Edit", achieving both fixed-parameter tractability and W[1]-hardness results. Moreover, we provide non-existence results for polynomial-size problem kernels and we show that some in general W[1]-hard problems turn fixed-parameter tractable when restricted to planar graphs.

1 Introduction

The Travelling Salesperson Problem (TSP) is probably the most studied combinatorial optimization problem. Almost all algorithm design techniques have been applied to it or were even specifically developed for it [12]. Many heuristic algorithms for TSP follow the paradigm of *local search*: Incrementally try to improve a solution by searching within its local neighborhood defined by a *distance measure*. Perhaps the most prominent and best examined distance measure for TSP is the *k-Edge Exchange* neighborhood (also called *k-Opt neighborhood* in some literature), where one is allowed to exchange at most k edges of the Hamiltonian cycle forming the tour. Implementations of this strategy for $k = 2, 3$ belong to the best performing heuristic algorithms for real-world instances [13]. However, for larger k, for which one would expect a strong increase of quality, the running time becomes infeasible since until now no algorithm is known which significantly beats the trivial $O(n^k)$ running time needed for a brute-force exploration of the local distance-k neighborhood. In an important step forward, considering the problem within the framework of *parameterized complexity* [7, 17], Marx [15] has shown that, by proving W[1]-hardness, there is no hope for an algorithm running in $f(k) \cdot n^c$ time for any function f (solely depending on k) and any constant c.

⋆ Supported by the DFG Cluster of Excellence on Multimodal Computing and Interaction (MMCI) and DFG project DARE (GU 1023/1-2).

	m-Swap	Swap	Edit	m-Reversal	Reversal	Edge
general graphs	FPT (Thm. 2)	W[1]-h (Thm. 1)	W[1]-h (Thm. 1)	FPT (Thm. 3)	W[1]-h (Thm. 1)	W[1]-h [15]
planar graphs	*FPT*	FPT (Thm. 5)	FPT (Thm. 5)	*FPT*	?	?

Table 1. Overview of our results using k as the parameter and assuming m to be a constant. The results written in italics are a direct consequence of a more general result.

Note that such an algorithm is desirable since the degree of the polynomial in the input size does not depend on the parameter k. Moreover, assuming that the ETH (*exponential time hypothesis*) [5, 10] does not fail, Marx has shown that there is no algorithm running in $O(n^{o(\sqrt[3]{k})})$ time.

In this work, besides the k-Edge Exchange neighborhood (briefly, *Edge* distance measure), we consider various other distance measures such as the Reversal distance (which is also widely studied in bioinformatics in the context of genome rearrangements [4]), the Swap distance where one is allowed to exchange two vertices, and the Edit distance where one can move a vertex to an arbitrary new position. For λ being any of these distance measures, we study the following problem.

LocalTSP(λ)
Input: An undirected graph $G = (V, E)$ with vertices labeled $v_1, \ldots, v_n$ such that the identical permutation (id) $v_1, v_2, \ldots, v_n, v_1$ is a Hamiltonian cycle in G, an edge weight function $\omega : E \to \mathbb{R}_0^+$, and a positive integer k.
Question: Is there a permutation π with $\lambda(\pi, \mathrm{id}) \leq k$ that yields a Hamiltonian cycle with $\omega(\pi) < \omega(\mathrm{id})$, where $\omega(\pi) = \sum_{i=1}^{n-1} \omega(\{v_{\pi(i)}, v_{\pi(i+1)}\}) + \omega(\{v_{\pi(n)}, v_{\pi(1)}\})$?

Our results. Table 1 summarizes our results. For $\lambda \in \{\text{Swap, Edit, Reversal, Edge}\}$, we show that the result of Marx [15] for LocalTSP(Edge) can be extended and even strengthened, that is, we show that LocalTSP(λ) is W[1]-hard, implying that it is probably not fixed-parameter tractable for the "locality parameter" k. Furthermore, again assuming that the ETH holds, there cannot be an algorithm with running time $O(n^{o(\sqrt{k})})$. In addition, exploring the limitations of polynomial-time preprocessing, we indicate that, unless NP $\subseteq$ coNP/poly, there is no polynomial-size problem kernel for LocalTSP(λ) for any of the considered distance measures λ.

On the positive side, for the Swap distance we show that, restricting by a parameter m the distance of two vertices that are allowed to swap, makes LocalTSP(m-Swap) fixed-parameter tractable with respect to the combined parameter (k, m). Furthermore, we show that an analogously restricted Reversal distance, called m-Reversal, again leads to fixed-parameter tractability. Continuing to chart the border of tractability, we show that LocalTSP(λ) for $\lambda \in \{\text{Swap, Edit}\}$ is fixed-parameter tractable on planar graphs. Due to space limitations most details are deferred to a full version of the paper.

Related Work. The most important reference point is Marx' study of LocalTSP(Edge) [15] (using different notation). In addition, long before Marx, Balas [1] studied LocalTSP(Max-Shift), where Max-Shift distance k means that in order to obtain an improved Hamiltonian cycle the maximum number of positions that a vertex is allowed to shift is k. Contrasting the parameterized hardness result of Marx [15], Balas showed that LocalTSP(Max-Shift) is fixed-parameter tractable by providing an algorithm running in $O(4^{k-1} \cdot k^{1.5} \cdot |V|)$ time.

2 Basic Notation and Distance Measures

Notation. Let $\mathcal{S}_n$ denote the set of all bijective mappings of the set $\{1, \ldots, n\}$ to itself and let $\mathrm{id} \in \mathcal{S}_n$ be the identity. A Hamiltonian cycle through a graph $G = (V, E)$ with vertices labeled $v_1, v_2, \ldots, v_n$ is expressed by a permutation $\pi \in \mathcal{S}_n$ such that the edge set $E(\pi)$ of π, defined as $E(\pi) = \{\{v_{\pi(i)}, v_{\pi(i+1)}\} \mid 1 \leq i < n\} \cup \{\{v_{\pi(n)}, v_{\pi(1)}\}\}$, is a subset of E. For a weight function $\omega : E \to \mathbb{R}_0^+$ we define the weight of π by $\omega(\pi) = \sum_{e \in E(\pi)} \omega(e)$. The Hamiltonian cycle π is called *improved* compared to id when $\omega(\pi) < \omega(\mathrm{id})$. In this sense, LocalTSP(λ) is the question whether there is an improved Hamiltonian cycle π with $\lambda(\pi, \mathrm{id}) \leq k$.

A parameterized problem is said to be *fixed-parameter tractable* if there is an algorithm that solves every instance (I, k) (where k is the parameter) within $f(k) \cdot |I|^c$ time for a constant c and a function f which solely depends on k [7, 17]. A *kernelization algorithm* computes for a given instance (I, k) in polynomial time a new instance (I', k') (called kernel) such that (I', k') is a yes-instance iff (I, k) is a yes-instance, $k' \leq g(k)$, and $|I'| \leq g(k)$ for a function g which solely depends on k [2, 11]. The function g measures the size of the kernel.

Permissive algorithms and distance measures. So far, the distance between Hamiltonian cycles was usually measured in terms of *Edge* distance, counting the number of edges used by one cycle but not used by the other. Another measure considered is the *Max-Shift* distance, which equals the maximum shift of the position of a vertex between the two permutations. We consider several further measures based on the following operations on permutations.

Definition 1. For a permutation $1, 2, \ldots, n$, we define the following operations:

$$\begin{aligned} \textit{reversal } \rho(i,j) &\text{ results in } 1, \ldots, i-1, j, j-1, \ldots, i+1, i, j+1, \ldots, n; \\ \textit{swap } \sigma(i,j) &\text{ results in } 1, \ldots, i-1, j, i+1, \ldots, j-1, i, j+1, \ldots, n; \\ \textit{edit } \epsilon(i,j) &\text{ results in } 1, \ldots, i-1, i+1, \ldots, j-1, j, i, j+1, \ldots, n. \end{aligned}$$

We do not consider the elements 1 and n to be anyhow special and, therefore, the operations can also be applied "over them". For a constant m and $0 < j - i \leq m-1$ or $n + j - i \leq m - 1$ we speak about m-swaps and m-reversals.

The distance measures *Swap, m-Swap, Edit, Reversal*, and *m-Reversal* count the minimum number of the appropriate operations to apply to one permutation in order to obtain the other.

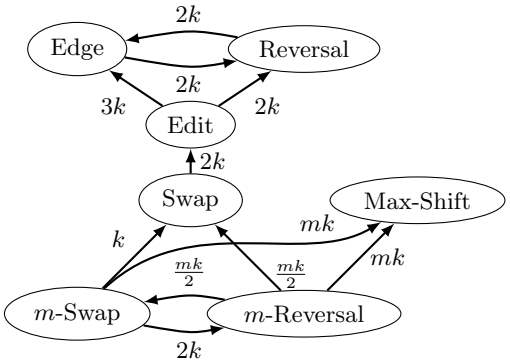

Fig. 1. Hasse diagram of the relations between the distance measures. Let $f : \mathbb{N} \to \mathbb{N}$. An arrow from a distance measure λ to a measure τ labeled "$f(k)$" means that τ is λ-bounded with function $f(k)$.

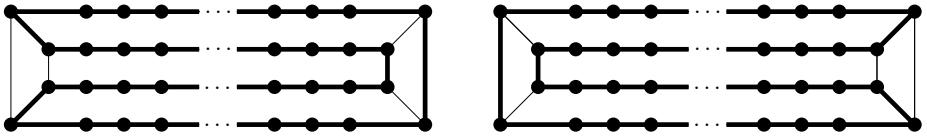

Fig. 2. A planar graph with two different Hamiltonian cycles. The cycles are only four edge modifications and four reversals from each other, while they can be made arbitrary far apart for any other of the measures by extending the horizontal lines. Furthermore, the differences between the cycles are also very far apart considering the distance in the graph and, as the graph has no other Hamiltonian cycles, there is no other solution with changes concentrated in a constant distance to one particular vertex.

Definition 2. A distance measure λ is *bounded* by a distance measure τ (or τ-bounded) if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for any two permutations $\pi, \pi' \in \mathcal{S}_n$ it holds that $\lambda(\pi, \pi') \leq f(\tau(\pi, \pi'))$.

It is easy to see that the relation of boundedness is reflexive and transitive and, therefore, forms a quasi-order on the distance measures. Figure 1 depicts all relations between the measures, omitting relations that can be deduced from the transitivity, in this sense showing a "Hasse diagram" of this quasi-order. The shown relations are easy to check. It is also not hard to come up with examples showing that no further boundedness relations hold between the distance measures. See Figure 2 for an interesting case of two Hamiltonian cycles which are close for Reversal and Edge distances, but far apart for all the other distances considered.

Marx and Schlotter [16] proposed to distinguish between *strict* and *permissive* local search algorithms. Strict local search algorithms find an improved solution (or prove that it does not exist) within some limited distance from the given solution. Permissive local search algorithms find *any* improved solution (potentially, with

unbounded distance to the given solution), provided that an improved solution exists within the limited distance of the given solution. The motivation for this distinction is that finding an improved solution within a bounded distance of a given solution may be hard even for problems where an optimal solution can easily be found, e. g., Minimum Vertex Cover on bipartite graphs [14]. The following lemma indicates a tight relationship between our notion of bounded distance measures and the existence of permissive FPT-algorithms.

Lemma 1. *If a distance measure λ is τ-bounded, then a (permissive) FPT-algorithm for* LocalTSP(λ) *is a permissive FPT-algorithm for* LocalTSP(τ).

3 General Graphs

We first show that LocalTSP(λ) is W[1]-hard for $\lambda \in \{$Swap, Edit, Reversal, Edge$\}$. To this end, we build on the proof given by Marx [15]. In contrast to Marx, who gave a reduction from the k-Clique problem, we reduce from the k-Multicolored Clique problem. This simplifies the construction and makes it even more powerful. Specifically, we show that, if there exists an improved Hamiltonian cycle, then there is also one that can be obtained by $O(k^2)$ swaps. Then, since the other measures are Swap-bounded and, unlike in Marx' construction, *any* improved Hamiltonian cycle in the constructed graph implies the existence of a k-multicolored clique, the hardness result holds true even for them.

Theorem 1. LocalTSP(λ) *is W[1]-hard with respect to k for $\lambda \in \{$Swap, Edit, Reversal, Edge$\}$.*

It was shown that k-Clique cannot be solved within $O(n^{o(k)})$ time unless the ETH fails [5, 10]. It follows from the reduction given by Fellows et al. [8] that if k-Multicolored Clique would be solvable in $O(n^{o(k)})$ time, then k-Clique would also be solvable in $O(n^{o(k)})$ time. In the parameterized reduction given in the proof of Theorem 1 reducing from k-Multicolored Clique to LocalTSP(λ) for $\lambda \in \{$Swap, Edit, Reversal, Edge$\}$ the parameter for LocalTSP is within the square of the parameter of k-Multicolored Clique. This implies the following corollary. For the case of Edge distance, it improves the lower bound $O(n^{o(\sqrt[3]{k})})$ given by Marx [15, Corollary 3.5].

Corollary 1. *Unless the ETH fails,* LocalTSP(λ) *does not admit an algorithm with running time $O(n^{o(\sqrt{k})})$ for $\lambda \in \{$Swap, Edit, Reversal, Edge$\}$.*

We have shown that on general graphs there is no hope to obtain an FPT-algorithm for LocalTSP(Swap) parameterized by k. However, restricting the distance measure to m-swaps makes the problem fixed-parameter tractable. As the exponent of the polynomial in the running time is also independent of m, we state the results with respect to the combined parameter (k, m). The corresponding algorithm is based on the bounded search tree technique and it is mainly based on the observation that the solution can be assumed to be given by a sequence of

swaps that are somehow related. For a formal description we need the following definition.

Let S be a sequence of swaps. We define an auxiliary *swap graph* G^S as follows. There is a vertex for each swap in the sequence S and two swaps $\sigma(i,j)$ and $\sigma(t,l)$ are adjacent if either t or l is contained in $\{i-1,i,i+1,j-1,j,j+1\}$. Furthermore, if a swap $\sigma(i,j)$ is applied, we call the positions i and j and the vertices at these positions *affected*.

Lemma 2. *If a* LocalTSP(λ) *instance for* $\lambda \in \{m\text{-Swap, Swap}\}$ *admits an improved Hamiltonian cycle, it also admits an improved Hamiltonian cycle which can be obtained by swaps (or* m*-swaps) such that their swap graph is connected.*

Proof. Suppose that we are given a sequence S of swaps whose application to a Hamiltonian cycle $\mathrm{id} \in \mathcal{S}_n$ creates an improved Hamiltonian cycle $\pi \in \mathcal{S}_n$. Furthermore, assume that $C_1, \ldots, C_p$ with $p \geq 2$ are the connected components of the corresponding swap graph G^S. For any of these components C, we denote by $\pi^C \in \mathcal{S}_n$ the permutation that results from applying the swaps in C to id preserving their order relative to S.

We shall show that the sets $E(\pi^{C_1}) \triangle E(\mathrm{id}), \ldots, E(\pi^{C_p}) \triangle E(\mathrm{id})$ form a partition of the set $E(\pi) \triangle E(\mathrm{id})$ ($\triangle$ denotes the symmetric difference). Having proved this, the rest of the argumentation is as follows. Since $\omega(\pi) < \omega(\mathrm{id})$ or equivalently $\omega(E(\pi) \setminus E(\mathrm{id})) < \omega(E(\mathrm{id}) \setminus E(\pi))$, it follows that there is at least one component C of G^S with $\omega(E(\pi^C) \setminus E(\mathrm{id})) < \omega(E(\mathrm{id}) \setminus E(\pi^C))$. This implies that $\omega(\pi^C) < \omega(\mathrm{id})$ and thus applying only swaps contained in C also results in an improved Hamiltonian cycle π^C.

It remains to prove that $E(\pi^{C_1}) \triangle E(\mathrm{id}), \ldots, E(\pi^{C_p}) \triangle E(\mathrm{id})$ is a partition of $E(\pi) \triangle E(\mathrm{id})$. First, for all $1 \leq i < j \leq p$ by definition of the swap graph, it follows that the positions, and thus also the vertices, affected by C_i are disjoint from the positions and vertices that are affected by C_j. Formally, $E(\pi^{C_i}) \triangle E(\mathrm{id}) \cap E(\pi^{C_j}) \triangle E(\mathrm{id}) = \emptyset$. For any component C, we next argue that $E(\pi^C) \triangle E(\mathrm{id}) \subseteq E(\pi) \triangle E(\mathrm{id})$. Clearly, for an edge $e = \{i,j\} \in E(\pi^C) \triangle E(\mathrm{id})$, either vertex i or j has to be affected by at least one swap in C. Then, no swap in $S \setminus C$ also affects either i or j, because such a swap would be adjacent to at least one swap in C. Hence, $e \in E(\pi) \triangle E(\mathrm{id})$. Finally, consider an edge $e = \{i,j\} \in E(\pi) \triangle E(\mathrm{id})$. By the same argument as above, all swaps that affect either the vertex i or j belong to the same component of G^S. Thus, since either vertex i or j is affected by a swap, it follows that there is a component C of G^S such that $e \in E(\pi^C) \triangle E(\mathrm{id})$. □

Theorem 2. LocalTSP*(*m*-Swap) is fixed-parameter tractable with respect to the combined parameter* (k,m)*. It is solvable in* $O(m^{2k}(m-1)^{2k} \cdot 4^k \cdot (k^2+n) \cdot n)$ *time.*

Proof. Let (G, ω, k) be an instance of LocalTSP(m-Swap). Furthermore, let S be a sequence of at most k m-swaps such that applying S to id results in an improved Hamiltonian cycle π. By Lemma 2 we can assume that G^S is connected. The algorithm consists of two parts. First, the algorithm guesses the positions of all swaps in S and, secondly, it finds the correct order of them.

To describe the first part, for convenience, we assume for all swaps $\sigma(i,j)$ that $j \in \{i+1, i+2, \ldots, i+m-1\}$. Furthermore, we define an ordering relation $\leq$ on swaps with $\sigma(i,j) \leq \sigma(t,p)$ iff $i < t$ or $i = t \wedge j \leq p$. Let $\sigma_1, \sigma_2, \ldots, \sigma_s$ with $s \leq k$ be the swaps of S sorted with respect to $\leq$ in ascending order. In the first part of the algorithm, by branching into all possibilities for the positions of the swaps, the algorithm guesses all swaps in the order given above. At the beginning, the algorithm branches into all possibilities to find the position i_1 for $\sigma_1(i_1, j_1)$ and then into the $m-1$ possibilities to find the position j_1. Now, suppose we have already found the swap $\sigma_t(i_t, j_t)$, we next describe how to find the swap $\sigma_{t+1}(i_{t+1}, j_{t+1})$. By the ordering we know that $i_1 \leq \ldots \leq i_t \leq i_{t+1}$ and, since all swaps are m-swaps, for all $1 \leq p \leq t$ with $j_p > i_t$ it holds that $j_p - i_t \leq m-1$. From this and since G^S is connected (Lemma 2), it follows that $i_t - i_{t+1} \leq m$. Thus, we can guess the position of i_{t+1} by branching into $m+1$ possibilities. Afterwards, by branching into $m-1$ possibilities we find the position j_{t+1}. Overall, we can guess the positions of σ_{t+1} by branching into at most m^2 possibilities and thus the positions of all swaps can be guessed in $O(m^{2k-1} \cdot n)$ time.

In the second part, the algorithm guesses the order of the m-swaps. Clearly, the trivial way to do that is by trying all permutations of the swaps, resulting in a total running time of $O(m^{2k-1}k! \cdot n)$. This already shows that the problem is fixed-parameter tractable for (k, m). We next describe how this can be accelerated in case that $4m^2 < k$. To this end, let $\sigma^{(1)}, \sigma^{(2)}, \ldots, \sigma^{(s)}$ be all swaps in S in the order of their application resulting in π. Clearly, if there are two subsequent swaps $\sigma^{(t)}(i,j)$ and $\sigma^{(t+1)}(i',j')$ such that $\{i,j\} \cap \{i',j'\} = \emptyset$, then reversing their order in the application of the swaps also results in π. More generally, instead of finding a total order of the swaps, it is sufficient to find a partial order of the swaps that defines the order for any pair of swaps $\sigma(i,j)$ and $\sigma(t,p)$ where $|\{i,j\} \cap \{t,p\}| = 1$. Clearly, we do not have to define the order of two swaps which are of the same *type*, that is, where $\{i,j\} = \{t,p\}$. Thus, for a position i, consider all swaps which affect position i. Since all these swaps are m-swaps, there can be at most $2m-2$ different types that affect position i. Hence, if there are k_i swaps that affect position i, then there are at most $(2m-2)^{k_i}$ different permutations of these swaps. Combining the number of possibilities of all affected positions, since each swap affects exactly two positions, it follows that there are at most $(2m-2)^{2k}$ permutations of all swaps yielding different Hamiltonian cycles. Once the partial orders at all relevant positions are determined, we check whether this can be obtained by some total order of the swaps, and find this order in $O(k^2)$ time, by representing the partial orders by some arcs in a directed graph on the set of swaps and finding a topological order for that graph. Then we apply the swaps in this order in $O(k)$ time and check whether we obtain an improved Hamiltonian cycle in linear time. Together with the first part, the whole algorithm runs in $O(m^{2k}(m-1)^{2k} \cdot 4^k \cdot (k^2+n) \cdot n)$ time. □

Since the m-Swap distance is bounded by the m-Reversal distance, the above theorem implies also the existence of an $O(m^{mk}(m-1)^{mk} \cdot 2^{mk} \cdot ((mk)^2+n) \cdot n)$-time permissive algorithm for LocalTSP(m-Reversal), i.e., an algorithm that returns an improved Hamiltonian cycle whenever there is an improved Hamiltonian cycle in m-Reversal distance at most k from the given cycle. By modifying the

algorithm from Theorem 2, we can obtain a strict local search algorithm for LocalTSP(m-Reversal) with a better running time.

Theorem 3. LocalTSP*(m-Reversal) is fixed-parameter tractable with respect to the combined parameter (k, m). It is solvable in $O(2^{mk} \cdot m^{2k-1} \cdot (m-1)^k \cdot (k^2+n) \cdot n)$ time.*

We remark that, for LocalTSP(λ) with $\lambda \in \{m$-Swap, m-Reversal$\}$, by applying a standard dynamic programming approach, the algorithms given in the proofs of Theorems 2 and 3 can be extended such that not only *any* improved Hamiltonian cycle is found but also the *best* improved Hamiltonian cycle within the local neighborhood.

Further, analyzing the proofs of Theorems 2 and 3, one can show that if there is an improved Hamiltonian cycle in LocalTSP(m-Swap) or LocalTSP(m-Reversal), then there is also an improved cycle which differs from the given one only on vertices $v_i, v_{i+1}, \ldots, v_{i+mk}$ for some i. Therefore, one can reduce an input instance to polynomially many instances of the same problem, each having its size bounded by a polynomial in k and m. Such a self-reduction is known as polynomial *Turing kernelization*. In contrast to this, in the next section we show that, for any of the distance measures λ considered in this work, LocalTSP(λ) does not admit a polynomial kernel even when restricted to planar graphs.

4 Planar Graphs

In this section we investigate the complexity of LocalTSP on planar graphs. Note that whether LocalTSP(Edge) on planar graphs parameterized by the locality parameter k is fixed-parameter tractable or not is the central open question stated by Marx [15] and it is also mentioned by Fellows et al. [9]. We do not answer this question; however, we show that on planar graphs LocalTSP(λ) for $\lambda \in \{$Swap, Edit$\}$ is fixed-parameter tractable for parameter k. Before that, we show that LocalTSP(λ) on planar graphs does not admit a polynomial kernel for all distance measures λ considered in this work.

Bodlaender et al. [3] have shown that a parameterized problem does not admit a polynomial-size kernel if its unparameterized variant is NP-hard and if it is compositional. A parameterized problem is compositional if there is a polynomial time algorithm that takes as input instances $(I_1, k), \ldots, (I_t, k)$ and computes a new instance (I, k') where k' is upper-bounded by a polynomial in k and (I, k') is a yes-instance iff (I_j, k) is a yes-instance for some $1 \leq j \leq t$.

To show that LocalTSP(m-Swap) on planar graphs has no polynomial kernel, we first consider a more restricted variant, namely LargeLocalTSP(m-Swap), where it is required that the underlying planar graph has more than $2mk$ vertices. We show that LargeLocalTSP(m-Swap) is NP-hard on planar graphs by a many-to-one reduction from Weighted Antimonotone 2-Sat; by exploiting the properties implied by the requirement that there are more than $2mk$ vertices, we then show that it is compositional. Together with the NP-hardness, this implies that LargeLocalTSP(m-Swap) does not admit a polynomial kernel, unless NP $\subseteq$ coNP/poly. Thus, the next theorem follows.

Theorem 4. *Unless NP ⊆ coNP/poly,* LocalTSP*(m-Swap) on planar graphs does not admit a polynomial kernel with respect to the parameter k for any m ≥ 2.*

Figure 1 depicts the relation between the different distance measures. Since there is a directed path from the m-Swap measure to every other measure, the m-Swap measure can be considered as the least powerful measure since all other measures are bounded by it. We thus claim here that by basically the same argumentation as for LocalTSP(m-Swap) one can show that on planar graphs LocalTSP(λ) for $\lambda \in \{m$-Reversal, Swap, Edit, Reversal, Max-Shift, Edge$\}$ does not admit a polynomial kernel with respect to parameter k, unless NP ⊆ coNP/poly.

LocalTSP(Edit) and LocalTSP(Swap) on planar graphs probably do not allow for polynomial kernels; however they admit a permissive FPT-algorithm. In the following we sketch the argumentation for LocalTSP(Swap); the result for the Edit distance can be obtained along the same lines, but it is more technical. The proof relies on the following two lemmas.

Lemma 3. *If a* LocalTSP*(Swap) instance with parameter k admits an improved Hamiltonian cycle, then it also admits an improved Hamiltonian cycle which differs from the given one only within the distance-3k neighborhood around some vertex.*

The following lemma shows that, regardless of the distance measure, on planar graphs it is fixed-parameter tractable to find the best improved Hamiltonian cycle that differs from the given one only within the neighborhood of one specific vertex.

Lemma 4. *For an instance of* LocalTSP *on planar graphs and a vertex v one can find in $O(2^{O(k)} \cdot n + n^3)$ time the best Hamiltonian cycle among those differing from the given one only within distance k from v.*

Theorem 5. *There is a permissive FPT-algorithm for* LocalTSP*(λ) on planar graphs with respect to k for $\lambda \in \{$Swap, Edit$\}$.*

Following the same approach as Fellows et al. [9], Theorem 5 can be easily generalized to any class of graphs with bounded local treewidth. As Lemma 3 does not assume anything about the graph, we only have to modify Lemma 4. The lemma is true in any class of graphs with bounded local treewidth, but the corresponding running time depends on the respective class.

5 Conclusion

We left open the central open problem due to Marx [15] whether LocalTSP(Edge) restricted to planar graphs is fixed-parameter tractable. However, we indicated (see Section 2) that a permissive FPT-algorithm for LocalTSP(Edge) implies a permissive FPT-algorithm for LocalTSP(Reversal) and vice versa. Thus, the question whether the problems are fixed-parameter tractable or not, are equivalent and this might help to shed new light on this question. To this end, it

might be beneficial to explore the connections of LocalTSP(Reversal) to the topic of Sorting by Reversals as studied in bioinformatics [4].

In addition, assuming the Exponential Time Hypothesis [5, 10], we showed that there is no $O(n^{o(\sqrt{k})})$-time algorithm for LocalTSP(λ) for $\lambda \in \{$Swap, Edit, Reversal, Edge$\}$. Is there also a matching upper bound or can the lower bound still be improved?

Finally, our investigations might also be extended by moving from local neighborhoods for TSP to so-called exponential (but structured) neighborhoods as undertaken already in a non-parameterized setting [6].

Bibliography

[1] E. Balas. New classes of efficiently solvable generalized traveling salesman problems. *Ann. Oper. Res.*, 86:529–558, 1999.

[2] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *Proc. 4th IWPEC*, volume 5917 of *LNCS*, pages 17–37. Springer, 2009.

[3] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *J. Comput. System Sci.*, 75(8):423–434, 2009.

[4] A. Caprara. Sorting by reversals is difficult. In *Proc. 1st RECOMB*, pages 75–83. ACM Press, 1997.

[5] J. Chen, B. Chor, M. Fellows, X. Huang, D. W. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inform. and Comput.*, 201(2):216–231, 2005.

[6] V. G. Deineko and G. J. Woeginger. A study of exponential neighborhoods for the travelling salesman problem and for the quadratic assignment problem. *Math. Program., Ser. A*, 87(3):519–542, 2000.

[7] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[8] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009.

[9] M. R. Fellows, F. A. Rosamond, F. V. Fomin, D. Lokshtanov, S. Saurabh, and Y. Villanger. Local search: Is brute-force avoidable? In *Proc. 21th IJCAI*, pages 486–491, 2009.

[10] J. Flum and M. Grohe. Parameterized complexity and subexponential time. *Bulletin of the EATCS*, 84:71–100, 2004.

[11] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

[12] G. Gutin and A. Punnen. *The Traveling Salesman Problem and its Variations*. Combinatorial optimization. 2002.

[13] D. S. Johnson and L. A. McGeoch. Experimental analysis of heuristics for the STSP. *The Traveling Salesman Problem and its Variations*, pages 369–443, 2004.

[14] A. Krokhin and D. Marx. On the hardness of losing weight. *ACM Trans. Algorithms*, 2011. To appear, online available.

[15] D. Marx. Searching the k-change neighborhood for TSP is W[1]-hard. *Oper. Res. Lett.*, 36(1):31–36, 2008.

[16] D. Marx and I. Schlotter. Stable assignment with couples: Parameterized complexity and local search. *Discrete Optim.*, 8(1):25–40, 2011.

[17] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.