Searching the k-change neighborhood for TSP is W[1]-hard

Dániel Marx[*]

Abstract

We show that searching the k-change neighborhood is W[1]-hard for metric TSP, which means that finding the best tour in the k-change neighborhood essentially requires complete search (modulo some complexity-theoretic assumptions).

Keywords: Traveling Salesperson Problem, W[1]-hardness, parameterized complexity, local search

1 Introduction

The Traveling Salesperson Problem (TSP) is one of the most studied optimization problems. Given the NP-hardness of this practically important problem, many heuristic approaches were proposed in the literature (c.f., [8, 9]). Many of these algorithms are based on local search, which means that the algorithm produces better and better solutions by iteratively improving the current best solution. The improvement is local: only those tours are considered that are "close" to the current tour, that is, those tours that can be reached by applying certain operations on the current tour. The effectiveness of this search procedure will largely depend on the set of allowed operations that is used to find the improved tour. Presumably, having a larger set of operations increases our chances of finding a better tour, but the time required to search the local neighborhood increases.

Perhaps the most studied local search heuristic is the *k-change neighborhood* rule, where those tours are considered that can be reached from the current tour by replacing at most k arcs. Computational experiments are mostly done for $k = 2, 3$, or 4. For larger k, however, the method becomes impractical, as there are $n^{O(k)}$ possible tours that have to be considered for improvement. A brute force search of the k-change neighborhood is not feasible for large k; therefore, unless we have a more efficient algorithm for finding the best tour in the local neighborhood, local search based on the k-change neighborhood cannot

[*]Institut für Informatik, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany. dmarx@informatik.hu-berlin.de

be implemented efficiently. On the other hand, an algorithm with running time of the form $O(2^k \cdot n)$ would make it possible to search the k-change neighborhood for, say, $k = 10$.

The hardness of searching the k-change neighborhood can be studied very naturally in the framework of parameterized complexity, as suggested in [3]. In classical complexity theory, the hardness of a problem is usually analyzed as a function of the input size. Parameterized complexity studies the hardness of a problem in finer detail: certain parameters of the problem instances are defined, and the running time is expressed not only as a function of the input size, but also as a function of these parameters. For example, in classical complexity, the problems Maximum Clique and Minimum Vertex Cover have the same complexity: both are NP-hard. In the decision version of these problems, each instance has a very natural parameter: the size k of the clique/vertex cover that we are looking for. Both problems can be solved in $n^{O(k)}$ time by complete enumeration. This means that the problems are polynomial-time solvable for every fixed value of k, but the algorithms become practically useless for large values of n, even if k is as small as 10. The problem is that k appears in the exponent of n, which means that the polynomial degree increases as k increases. However, Minimum Vertex Cover has better algorithms, for example it can be solved in $O(2^k \cdot n^2)$ time. It follows that Minimum Vertex Cover has a quadratic algorithm for every fixed value of k. The central issue of parameterized complexity is the question whether it is possible to find algorithms where the exponent of the input size does not increase as parameter k increases. We say that a parameterized problem is *fixed-parameter tractable* if it can be solved in time $f(k) \cdot n^c$, where c is a constant and f is a (possibly exponential) function depending only on k. This means that the exponential increase of the running time can be restricted to the parameter k. It turns out that several NP-hard problems are fixed-parameter tractable, for example Minimum Vertex Cover, Longest Path, and Disjoint Triangles. Analogously to NP-completeness in classical complexity, the theory of W[1]-hardness can be used to show that a problem is unlikely to be fixed-parameter tractable, i.e., for every algorithm the parameter has to appear in the exponent of n.

Returning to the TSP problem, what we would like to have is an algorithm that, given a tour T, finds the best tour in the k-neighborhood of T. If k is part of the input, then this problem is clearly NP-hard: if $k = n$, then the problem is equivalent to finding the best possible tour. On the other hand, the problem is polynomial-time solvable for every fixed value of k (in $n^{O(k)}$ time by complete search). In this paper we investigate whether the brute force search can be improved to an algorithm where the degree of n does not depend on k, i.e., whether the problem is fixed-parameter tractable. The main result of the paper is that finding the best tour in the k-change neighborhood is W[1]-hard, which implies that the problem is not fixed-parameter tractable, unless W[1] = FPT. This answers an open question of Fellows [3]. The hardness result holds even if the distance matrix is symmetric and satisfies the triangle inequality. However, our result does not say anything about the important special case when the cities are points in the plane, and the distance is Euclidean distance. It remains an interesting open question whether the hardness result can be extended to this case as well.

2 Parameterized complexity

We follow [5] for the standard definitions of parameterized complexity. Let Σ be a finite alphabet. A decision problem is represented by a set $Q \subseteq \Sigma^*$ of strings over Σ. A *parameterization* of a problem is a polynomial-time computable function $\kappa : \Sigma^* \to \mathbb{N}$. A *parameterized decision problem* is a pair (Q, κ), where $Q \subseteq \Sigma^*$ is an arbitrary decision problem and κ is a parameterization. Intuitively, we can imagine a parameterized problem as a decision problem where each input instance $x \in \Sigma^*$ has a positive integer $\kappa(x)$ associated with it. A parameterized problem (Q, κ) is *fixed-parameter tractable (FPT)* if there is an algorithm that decides whether $x \in Q$ in time $f(\kappa(x)) \cdot |x|^c$ for some constant c and computable function f. An algorithm with such running time is called an *fpt-time algorithm* or simply *fpt-algorithm.*

Many NP-hard problems were investigated in the parameterized complexity literature, with the goal of identifying fixed-parameter tractable problems. There is a powerful toolbox of techniques for designing fpt-algorithms: kernelization, bounded search trees, color coding, well-quasi ordering—just to name some of the more important ones. On the other hand, certain problems resisted every attempt at obtaining fpt-algorithms. Analogously to NP-completeness in classical complexity, the theory of W[1]-hardness can be used to give strong evidence that certain problems are unlikely to be fixed-parameter tractable. We omit the somewhat technical definition of the complexity class W[1], see [2, 5] for details. Here it will be sufficient to know that there are several problems, including Maximum Clique, that were proved to be W[1]-hard. Furthermore, we also expect that there is no $n^{o(k)}$ (or even $f(k) \cdot n^{o(k)}$) algorithm for Maximum Clique: recently it was shown that if there exists an $f(k) \cdot n^{o(k)}$ algorithm for n-vertex Maximum Clique, then n-variable 3-Sat can be solved in time $2^{o(n)}$ (see [1] and [4]).

To prove that a parameterized problem (Q', κ') is W[1]-hard, we have to present a parameterized reduction from a known W[1]-hard problem (Q, κ) to (Q', κ'). A *parameterized reduction* from problem (Q, κ) to problem (Q', κ') is a function that transforms a problem instance x of Q into a problem instance x' of Q' in such a way that

- $x' \in Q'$ if and only if $x \in Q$,
- $\kappa'(x)$ can be bounded by a function of $\kappa(x)$, and
- the transformation can be computed in time $f(\kappa(k)) \cdot |x|^c$ for some constant c and function $f(k)$.

It is easy to see that if there is a parameterized reduction from (Q, κ) to (Q', κ'), and (Q', κ') is fixed-parameter tractable, then it follows that (Q, κ) is fixed-parameter tractable as well. The most important difference between parameterized reductions and classical polynomial-time many-to-one reductions is the second requirement: in most NP-completeness proofs the new parameter is not a function of the old parameter. Therefore, finding parameterized reductions is usually more difficult, and the constructions have somewhat different flavor than classical reductions.

3 Hardness result

The input of the Traveling Salesperson Problem (TSP) consists of a set V of cities and a distance matrix. The distance matrix contains a positive integer $d(u,v)$ for each ordered pair (u,v) of cities. For convenience, we allow $d(u,v)=\infty$. The task is to find a tour of minimal length that visits all the cities. That is, we have to find an ordering $v_1, \ldots, v_n$ of the cities such that the total length

$$\sum_{i=1}^{n-1} d(v_i, v_{i+1}) + d(v_n, v_1)$$

is minimal (where $n = |V|$). A tour can be also considered as a set of n ordered pairs of cities; this interpretation will be more convenient for our purposes. If X and Y are two tours on the same set of cities, then the distance of X and Y is $|X \setminus Y| = |Y \setminus X|$. We study the parameterized complexity of the following problem:

k-CHANGE TSP
Input: A set V of cities, a distance matrix, a tour C, and an integer k.
Parameter: k
Question: Is there a tour C' that has length strictly less than C, and is at distance at most k from C?

In the *symmetric* version of the problem it is assumed that $d(x,y)=d(y,x)$ for every $x,y \in V$. In the *asymmetrical* version of the problem we do not make this assumption. In the *metric* version of the problem, the distance matrix satisfies the triangle inequality, i.e., $d(x,y) \le d(x,z) + d(z,y)$ for any three cities x, y, z.

It will be more convenient to consider the problem in terms of graphs, especially if the distance matrix contains many infinite values. An instance of the symmetric TSP problem can be transformed into a graph where the cities are the vertices, and if $d(x,y)$ is finite, then there is an edge of weight $d(x,y)$ between x and y. In the asymmetrical case the graph is directed and there is a directed edge from x to y with weight $d(x,y)$ if $d(x,y)$ is finite. It is clear that the finite-weight tours correspond to the (directed) Hamiltonian cycles in the graph.

We show that k-CHANGE TSP is W[1]-hard (Theorem 3.2). The W[1]-hardness proof is given for the asymmetrical version of the problem. However, the asymmetrical problem can be reduced to the symmetric case, thus W[1]-hardness follows for the symmetric problem as well (Lemma 3.1). The reduction is the same as the standard reduction from the directed Hamiltonian cycle problem to the undirected version (c.f., [6])

Lemma 3.1. *Given a directed graph D, one can construct in polynomial time an undirected graph G such that there is a one-to-one correspondence between the Hamiltonian cycles of the two graphs. Moreover, this correspondence preserves the weight of the cycles, and the distance between the cycles.*

Proof. Each vertex v in D is be replaced by three vertices v^{in}, v^*, v^{out}. There is a zero-weight edge between v^{in} and v^*, and between v^{out} and v^*. Furthermore, if there is a directed edge from a to b in D, then G has an edge between a^{out} and b^{in} with the same weight. Now assume that there is a directed Hamiltonian cycle v_1, v_2, $\dots$, v_n, v_1 in D. The corresponding cycle in G is v_1^{out}, v_2^{in}, v_2^*, v_2^{out}, v_3^{in}, $\dots$, v_n^{out}, v_1^{in}, v_1^*, v_1^{out}; clearly it has the same weight. On the other hand, assume that there is a Hamiltonian cycle C in G. For every v, the two neighbors of v^* in C have to be v^{in} and v^{out}. Now a Hamiltonian cycle C' of D can be defined as follows: if the edge between a^{out} and b^{in} is used in C, then C' uses the directed edge from a to b. It can be easily verified that this results in a Hamiltonian cycle with the same weight. Moreover, this correspondence preserves the distance between the cycles, since every tour in G contains the edges $v^{\text{in}}v^*$ and v^*v^{out}, and there is a one-to-one correspondence between the edges $\overrightarrow{ab}$ and $a^{\text{out}}b^{\text{in}}$. □

Theorem 3.2. *Given a directed weighted graph D with a Hamiltonian cycle C, it is* W[1]*-hard to decide whether there is a Hamilton cycle C' such that the weight of C' is strictly less than the weight of C and the distance between C and C' is at most k.*

Proof. The proof is by reduction from Maximum Clique. Given a graph G where we have to find a clique of size t, we construct an equivalent instance of k-change TSP on a directed graph D.

The switch gadget. The constructed graph is built from several copies of the *switch gadget* shown in Figure 1. The gadget is connected to the rest of the graph at the vertices α, β, γ, δ: there are edges entering α and γ; there are edges leaving β and δ. It is easy to see that if a switch gadget is part of a larger graph, and there is Hamiltonian cycle in the larger graph, then this cycle traverses the gadget in one of the three ways presented in Figure 1b–d. Either the cycle enters at α and leaves at β (b); or it enters at γ and leaves at δ (c); or it visits the gadget twice: once entering at α and leaving at β, and once entering at γ and leaving at δ (d). In the first two cases, only one edge of weight 1 is used, while in the third cases both of these edges are in the cycle. The parameters of the reduction will be set in such a way that if a Hamilton cycle C' uses two edges with weight 1 from a gadget, then the total weight of the cycle will be too large. Thus only cases (b) and (c) have to be considered for the Hamiltonian cycle that we are looking for. Therefore, the gadget effectively acts as a switch: either it is used as a $\alpha \to \beta$ path or as a $\gamma \to \delta$ path. In the first case, we say that the cycle uses the *upper path* of the gadget, in the second case we say that the cycle uses the *lower path*.

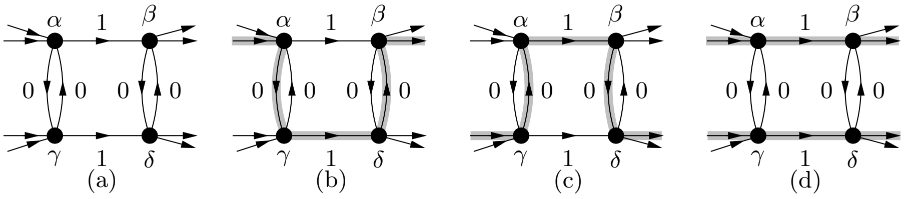

Figure 1: The switch gadget.

Vertex and edge segments. Let n be number of vertices in G, and let m be the number of edges. We construct a directed graph D that consists of $(2n+m)t(t-1)$ copies of the switch gadget and some additional vertices. We have the following switch gadgets:

- *vertex gadget* $V^1_{i,(j_1,j_2)}$ for each $1 \le i \le n$ and $1 \le j_1, j_2 \le t$ with $j_1 \ne j_2$,
- *vertex gadget* $V^2_{i,(j_1,j_2)}$ for each $1 \le i \le n$ and $1 \le j_1, j_2 \le t$ with $j_1 \ne j_2$, and
- *edge gadget* $S_{i,(j_1,j_2)}$ for each $1 \le i \le m$ and $1 \le j_1, j_2 \le t$ with $j_1 \ne j_2$.

We form *segments* from one or more gadgets. The *vertex segment* $V_{i,j}$ ($1 \le i \le n$, $1 \le j \le t$) consists of the *entrance vertex* $a_{i,j}$, the *exit vertex* $b_{i,j}$, the $(t-1)$ gadgets $V^1_{i,(j,j')}$ (for $j' \ne j$), and the $(t-1)$ gadgets $V^2_{i,(j',j)}$ (for $j' \ne j$) (see Figure 2). To simplify the notation, let $W_1, \dots, W_{2(t-1)}$ be an arbitrary ordering of these gadgets. For every $1 \le \ell < 2(t-1)$, there is a zero-weight edge from vertex β of W_ℓ to vertex α of $W_{\ell+1}$. Furthermore, there is a zero-weight edge from $a_{i,j}$ to vertex α of gadget W_1, and there is a zero-weight edge from vertex β of gadget $W_{2(t-1)}$ to $b_{i,j}$. Finally, there is a *bypass edge* with zero weight from $a_{i,j}$ to $b_{i,j}$.

The *edge segment* $E_{i,(j_1,j_2)}$ ($1 \le i \le m$, $1 \le j_1, j_2 \le t$ with $j_1 \ne j_2$) contains two vertices $z_{i,(j_1,j_2)}$ (the *entrance*), $q_{i,(j_1,j_2)}$ (the *exit*), and the gadget $S_{i,(j_1,j_2)}$

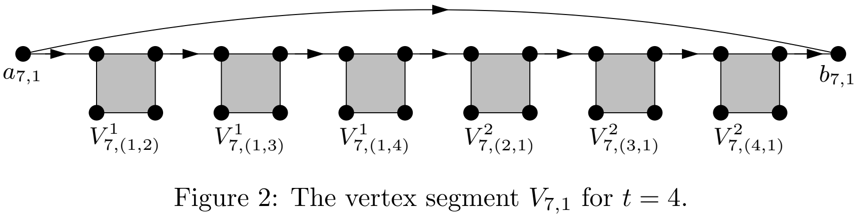

Figure 2: The vertex segment $V_{7,1}$ for $t = 4$.

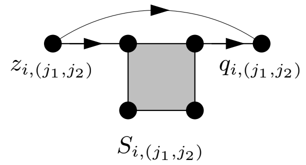

Figure 3: The edge segment $E_{i,(j_1,j_2)}$.

(see Figure 3). There is a zero-weight edge from $z_{i,(j_1,j_2)}$ to vertex α of $S_{i,(j_1,j_2)}$, and a zero-weight edge from vertex β of $S_{i,(j_1,j_2)}$ to $q_{i,(j_1,j_2)}$. Moreover, there is a *bypass edge* with zero weight from $z_{i,(j_1,j_2)}$ to $q_{i,(j_1,j_2)}$.

Consider an arbitrary ordering of the $nt + mt(t-1)$ segments defined above. Add a zero-weight edge from the exit of each segment to the entrance of the next segment. Moreover, add an edge of weight 1 that goes from the exit of the last segment (denote it by v_{last}) to the entrance of the first segment (denote it by v_{first}). There will be some more edges in the graph D, but before completing the description of D, we first define the Hamiltonian cycle C. The cycle starts at v_{first}, goes through the upper path of the gadget(s) in the first segment, leaves the segment at the exit, enters the second segment at its entrance, etc. The cycle does not use the bypass edges, thus it visits every vertex of every gadget. Finally, when C reaches the exit of the last segment (v_{last}), it goes back to the entrance of the first segment (v_{first}) using the edge with weight 1. The cycle traverses one edge of weight 1 in each gadget, hence the total weight of C is $(2n + m)t(t-1) + 1$.

Encoding the graph. For convenience, we identify the vertices with the integers $\{1, \dots, n\}$ and the edges with the integers $\{1, \dots, m\}$. Consider the ordered pairs (j_1, j_2) ($1 \leq j_1, j_2 \leq t$, $j_1 \neq j_2$), and let P_1, P_2, $\dots$, $P_{t(t-1)}$ be an ordering of these pairs such that the second element of a pair is the same as the first element of the next pair, that is, $P_\ell = (p_\ell, p_{\ell+1})$ for every $1 \leq \ell < t(t-1)$. It is clear that such an ordering exists.

As discussed above, we will ensure that the cycle C' traverses every gadget as either (b) or (c) in Figure 1. In the latter case, we say that the gadget is *active.* We will show that the active gadgets describe a t-clique of graph G. If gadget $S_{i,(j_1,j_2)}$ in edge segment $E_{i,(j_1,j_2)}$ is active, then this means that the edge i is the edge econnecting the j_1-th and j_2-th vertex of the clique. If gadget $V^1_{i,(j_1,j_2)}$ (resp., $V^2_{i,(j_1,j_2)}$) is active, then this means that vertex i is the j_1-th (resp., j_2-th) vertex of the clique. We connect the γ and δ vertices of gadgets in a way that enforces that the active gadgets describe a clique.

For every $1 \leq \ell < t(t-1)$, we add edges as follows. There is a zero-weight edge from vertex δ of gadget $V^2_{i,(p_\ell,p_{\ell+1})}$ to vertex γ of gadget $V^1_{i,(p_{\ell+1},p_{\ell+2})}$. If vertex i is an endpoint of edge r, then there is a zero-weight edge from vertex δ of $V^1_{i,(p_\ell,p_{\ell+1})}$ to vertex γ of $E_{r,(p_\ell,p_{\ell+1})}$, and there is a zero-weight edge from vertex δ of $E_{r,(p_\ell,p_{\ell+1})}$ to vertex γ of $V^2_{i,(p_\ell,p_{\ell+1})}$. Note that only one edge enters

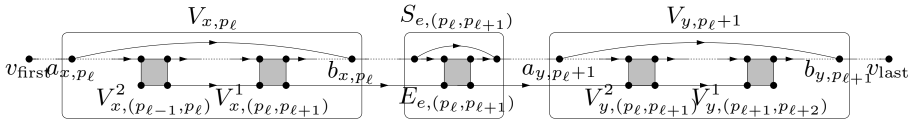
V_{x,p_ℓ}
$S_{e,(p_\ell,p_{\ell+1})}$
$V_{y,p_\ell+1}$
v_{first}
a_{x,p_ℓ}
b_{x,p_ℓ}
$a_{y,p_\ell+1}$
$b_{y,p_{\ell+1}}$
v_{last}
$V^2_{x,(p_{\ell-1},p_\ell)}$
$V^1_{x,(p_\ell,p_{\ell+1})}$
$E_{e,(p_\ell,p_{\ell+1})}$
$V^2_{y,(p_\ell,p_{\ell+1})}$
$V^1_{y,(p_{\ell+1},p_{\ell+2})}$

Figure 4: If x and y are the two end points of edge e, then segments V_{x,p_ℓ}, V_{y,p_ℓ}, and $S_{e,(p_\ell,p_\ell+1)}$ are connected as shown above. The dotted lines represent missing sequences of gadgets and segments.

gadget $E_{r,(p_\ell,p_{\ell+1})}$ at γ and leaves at δ, thus if this gadget is active, then gadgets $V^1_{i,(p_\ell,p_{\ell+1})}$ and $V^2_{i,(p_\ell,p_{\ell+1})}$ have to be active as well. This is compatible with the intended meaning of the gadgets: if edge r is in the clique, then both endpoints are in the clique. Furthermore, for every $1 \leq i \leq n$, there is a zero-weight edge from v_{last} to vertex γ of gadget $V^1_{i,(p_1,p_2)}$, and there is a zero-weight edge from vertex δ of $V^2_{i,(p_{t(t-1)},p_{t(t-1)+1})}$ to v_{first}. This completes the description of the directed graph D.

k-change TSP⇒Maximum Clique. We claim that if there is a Hamiltonian cycle C' having weight strictly less than $(2n+m)t(t-1)+1$ that is at distance at most $k := 4t^2(t-1)+t+5t(t-1)$ from C, then there is a t-clique in G. As shown in Figure 1, C' has to traverse at least one edge with weight 1 in each gadget, hence the only way the total weight is at most $(2n+m)t(t-1)$ if C' does not use the edge of weight 1 that goes from v_{last} to v_{first}. Furthermore, every gadget has to be traversed either as (b) or (c) of Figure 1, the case of (d) is not allowed.

Let us think about the cycle C' as a path that starts from and returns to v_{first}. Similarly to C, the cycle C' has to go through the segments one by one. It is clear that if C' enters a segment at its entrance, then it has to leave it via its exit. However, inside a segment, C' can do two things: either it goes through the gadget(s) (similarly to C), or it skips the gadget(s) using the bypass edge. In the latter case, we say that the segment is *active*. If vertex segment $V_{i,j}$ is active, then we will take it as an indication that vertex i should be the j-th vertex of the clique. If edge segment $E_{i,(j_1,j_2)}$ is active, then this will mean that the j_1-th and the j_2-th vertices of the clique are connected by edge i. By the time C' reaches v_{last}, every gadget is completely traversed, or not visited at all. The cycle has to return to v_{first} by visiting all the skipped gadgets.

We argue that if the distance between C and C' is at most k, then for every $1 \leq j \leq t$, there is exactly one i such that $V_{i,j}$ is active. This means that there are k selected vertex segments, and these segments can be interpreted as k vertices. We also show that for every pair (j_1, j_2) with $j_1 \neq j_2$, there is exactly one $1 \leq i \leq m$ such that edge segment $E_{i,(j_1,j_2)}$ is active. After v_{last}, cycle C' goes to vertex γ of a gadget $V^1_{i,(p_1,p_2)}$ for some $1 \leq i \leq n$. This implies that the lower path is used for this gadget; therefore, the bypass edge is used in segment V_{i,p_1} for some i, i.e., V_{i,p_i} is active. After leaving gadget $V^1_{i,(p_1,p_2)}$

at vertex δ, cycle C' goes to a gadget $S_{i',(p_1,p_2)}$ for some i', implying that the bypass edge is used in segment $E_{i',(p_1,p_2)}$ for this i'. The next visited gadget is $V^2_{i'',(p_1,p_2)}$ for some i'', and the gadget after that is $V^1_{i'',(p_2,p_3)}$ for the same i''. In a similar fashion, we can show that for every $1 \leq \ell < t(t-1)$ there is an active vertex segment V_{i,p_ℓ} for some i and an active edge segment $E_{i',(p_\ell,p_{\ell+1})}$ for some i'. What remains to be shown is the uniqueness of these i's. To show this, we make use of the fact that the distance of C and C' is at most k. How does the distance between C and C' increase if we make a segment active? We will count $|C \setminus C'|$ to determine the distance. Cycle C uses the upper path in every gadget, thus if C' uses the lower path of a gadget, then we have to delete 3 edges from C. Furthermore, to make a vertex segment active, we have to remove the edges between the gadgets, the edge going from the entrance to the first gadget, and the edge going from the last gadget to the exit. In total, an active vertex segment increases the distance by $4t(t-1)+1$. Similarly, 5 edges have to be removed from C to make an edge segment active. We have seen that there are at least t active vertex segments and at least $t(t-1)$ active edge segments. It follows that the distance can be at most k only if there are exactly t active vertex segments and exactly $t(t-1)$ active edge segments. Now it is clear that for every j (resp., (j_1,j_2)) there is exactly one active $V_{i,j}$ (resp., $E_{i,(j_1,j_2)}$).

Define $v_j = i$, if segment $V_{i,j}$ is active, and let $e_{j_1,j_2} = i$ if segment $E_{i,(j_1,j_2)}$ is active. We show that v_1, v_2, $\ldots$, v_t is a clique of size t in G, which proves the first direction of the reduction. Suppose that $x = v_{j_1}$ and $y = v_{j_2}$ are not neighbors. We arrive to a contradiction by showing that edge e_{j_1,j_2} connects these two vertices. By the definition of v_{j_1}, cycle C' uses the lower path in every gadget of vertex segment V_{x,j_1}. In particular, this is true for the gadget $V^1_{x,(j_1,j_2)}$. Thus C' leaves $V^1_{x,(j_1,j_2)}$ at vertex δ, and goes to vertex γ of edge gadget $S_{i,(j_1,j_2)}$ for some i. Therefore, C' uses the lower path in $S_{i,(j_1,j_2)}$, and it follows by the definition of e_{j_1,j_2} that $i = e_{j_1,j_2}$. Now the construction implies that e_{j_1,j_2} is incident to v_{j_1} (otherwise vertex δ of the vertex gadget would not be connected with vertex γ of the edge gadget). After leaving $S_{i,(j_1,j_2)}$ at vertex δ, cycle C' goes to vertex γ of $V^2_{i,(j_1,j_2)}$ for some $1 \leq i \leq n$. This implies that segment V_{i,j_2} is active, hence $i = v_{j_2}$. Therefore, e_{j_1,j_2} is incident to v_{j_2} as well, thus v_{i_1} and v_{i_2} are indeed connected.

Maximum Clique⇒k-change TSP. To prove the other direction, we have to show that if there is a clique K of size t in G, then there is Hamiltonian cycle C' having weight $(2n+m)t(t-1)$ that is at distance at most k from V. Let v_1, v_2, $\ldots$, v_t be the vertices in K, and let e_{j_1,j_2} be the edge connecting v_{j_1} and v_{j_2}. Cycle C' starts from v_{first} and goes through the segments, similarly to C. However, for every $1 \leq j \leq t$, cycle C' traverses vertex segment $V_{v_j,j}$ in a way different from C: after the entrance $a_{v_j,j}$, the cycle goes to exit $b_{v_j,j}$ on the bypass edge, completely avoiding the gadgets in the segment. Similarly, for $j_1 \neq j_2$, in segment $E_{e_{j_1,j_1},(j_1,j_2)}$ the cycle C' uses the bypass edge from the entrance to the exit, and the gadget $S_{e_{j_1,j_2},(j_1,j_2)}$ is avoided. After the cycle reaches v_{last}, it has to return to v_{first} by visiting the skipped gadgets. We do it as follows. First we go from v_{last} to vertex γ of the gadget $V^1_{v_{p_1},(p_1,p_2)}$. Now assume that we are at vertex γ of $V^1_{v_{p_\ell},(p_\ell,p_{\ell+1})}$ for some $1 \le \ell < t(t-1)$. The cycle uses the lower path to visit $V^1_{v_{p_\ell},(p_\ell,p_{\ell+1})}$, leaves the gadget at vertex δ and goes to vertex γ of edge gadget $E_{e_{p_\ell,p_{\ell+1}},(p_\ell,p_{\ell+1})}$. After going through this edge gadget on the lower path, the cycle goes to vertex γ of $V^2_{v_{p_{\ell+1}},(p_\ell,p_{\ell+1})}$ (recall that edge $e_{p_\ell,p_{\ell+1}}$ is incident to both v_{p_ℓ} and $v_{p_{\ell+1}}$), goes through this gadget, and goes to vertex γ of gadget $V^1_{v_{p_{\ell+1}},(p_{\ell+1},p_{\ell+2})}$. We continue this way until vertex δ of gadget $V^2_{p_\ell,(p_\ell,p_{\ell+1})}$ is reached for $\ell = t(t-1)$. At that point the cycle C' is terminated by an edge going to v_{first}. It is clear that every skipped gadget is visited exactly once. By the discussion in the first direction of the proof, the distance of C' from C is exactly k. Moreover, cycle C' uses one edge with weight 1 in each gadget, and it does not use the edge from v_{last} to v_{first}, hence its total weight is $(2n+m)t(t-1)$, which is strictly smaller than the weight of C. □

Putting together Theorem 3.2 and Lemma 3.1, it follows:

Theorem 3.3. *k-CHANGE TSP is W[1]-hard, even if the distance matrix is symmetric.* □

The hardness result of Theorem 3.3 applies even if the distance matrix contains only the values 0, 1, and ∞. However, the distance matrix constructed by Theorem 3.2 is not metric. In particular, $d(x,z) + d(z,y)$ can be finite even if $d(x,y) = \infty$. However, this can be easily fixed:

Corollary 3.4. *k-CHANGE TSP is W[1]-hard, even if the distance matrix is metric and symmetric.*

Proof. Let d be a distance matrix on n cities, containing only the values 0, 1, and ∞. Let us define a new distance matrix d':

$$d'(x,y) = \begin{cases} 2n & \text{if } d(x,y) = 0, \\ 2n+1 & \text{if } d(x,y) = 1, \\ 4n & \text{if } d(x,y) = \infty. \end{cases}$$

Clearly, d' satisfies the triangle inequality. Furthermore, if an instance of k-CHANGE TSP is given with distance matrix d and a tour C of finite length, then the answer does not change if we replace d with d'. To see this, observe that new length of any infinite-length tour will be greater than the new weight of any finite-length tour, and the relative ordering of the finite-length tours will remain the same. □

Corollary 3.4 implies that there is no $f(k) \cdot n^{O(1)}$ time algorithm for k-CHANGE TSP (unless W[1] = FPT); k has to appear in the exponent of n. If we make a stronger complexity-theoretic assumption, then we can actually prove a lower bound on the exponent of n:

Corollary 3.5. *There is no $f(k) \cdot n^{o(\sqrt[3]{k})}$ time algorithm for metric k-*CHANGE TSP *with n cities, unless n-variable* 3-SAT *can be solved in* $2^{o(n)}$ *time.*

Proof. The proof in Theorem 3.2 takes an instance (G, t) of MAXIMUM CLIQUE, and turns it into an equivalent instance of k-CHANGE TSP with $k = O(t^3)$. Therefore, an $f(k) \cdot n^{o(\sqrt[3]{k})}$ time algorithm for k-CHANGE TSP would be able to solve MAXIMUM CLIQUE in time $f'(t) \cdot n^{o(t)}$. As shown in [1], this would imply that n-variable 3-SAT can be solved in time $2^{o(n)}$. □

The assumption that n-variable 3-SAT cannot be solved in time $2^{o(n)}$ is also known as the Exponential Time Hypothesis [7], and is equivalent to the parameterized complexity conjecture M[1] $\neq$ FPT (see [4].

Acknowledgments

I'm grateful to Mike Fellows for directing my attention to the problem.

References

[1] J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. In *Proceedings of 19th Annual IEEE Conference on Computational Complexity*, pages 150–160, 2004.

[2] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer-Verlag, New York, 1999.

[3] M. R. Fellows. Parameterized complexity: new developments and research frontiers. In *Aspects of Complexity (Kaikoura, 2000)*, volume 4 of *de Gruyter Ser. Log. Appl.*, pages 51–72. de Gruyter, Berlin, 2001.

[4] J. Flum and M. Grohe. Parameterized complexity and subexponential time. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, (84):71–100, 2004.

[5] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag, Berlin, 2006.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Co., San Francisco, Calif., 1979.

[7] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. System Sci.*, 63(4):512–530, 2001. Special issue on FOCS 98 (Palo Alto, CA).

[8] D. S. Johnson and L. A. McGeoch. The traveling salesman problem: a case study. In *Local search in combinatorial optimization*, Wiley-Intersci. Ser. Discrete Math. Optim., pages 215–310. Wiley, Chichester, 1997.

[9] D. S. Johnson and L. A. McGeoch. Experimental analysis of heuristics for the STSP. In *The traveling salesman problem and its variations*, volume 12 of *Comb. Optim.*, pages 369–443. Kluwer Acad. Publ., Dordrecht, 2002.