On Polynomial Kernels for Traveling Salesperson Problem and Its Generalizations

Václav Blažej
Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Czech Republic

Pratibha Choudhary
Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Czech Republic

Dušan Knop
Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Czech Republic

Šimon Schierreich
Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Czech Republic

Ondřej Suchý
Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Czech Republic

Tomáš Valla
Department of Theoretical Computer Science, Faculty of Information Technology,
Czech Technical University in Prague, Czech Republic

Abstract

For many problems, the important instances from practice possess certain structure that one should reflect in the design of specific algorithms. As data reduction is an important and inextricable part of today's computation, we employ one of the most successful models of such precomputation – the kernelization. Within this framework, we focus on Traveling Salesperson Problem (TSP) and some of its generalizations.

We provide a kernel for TSP with size polynomial in either the feedback edge set number or the size of a modulator to constant-sized components. For its generalizations, we also consider other structural parameters such as the vertex cover number and the size of a modulator to constant-sized paths. We complement our results from the negative side by showing that the existence of a polynomial-sized kernel with respect to the fractioning number, the combined parameter maximum degree and treewidth, and, in the case of Subset TSP, modulator to disjoint cycles (i.e., the treewidth two graphs) is unlikely.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Fixed parameter tractability

Keywords and phrases Traveling Salesperson, Subset TSP, Waypoint Routing, Kernelization

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.22

Related Version *Full Version*: `https://arxiv.org/abs/2207.01109`

Funding The authors acknowledge the support of the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics". This work was additionally supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS20/208/OHK3/3T/18.

© Václav Blažej, Pratibha Choudhary, Dušan Knop, Šimon Schierreich, Ondřej Suchý, and Tomáš Valla;
licensed under Creative Commons License CC-BY 4.0

30th Annual European Symposium on Algorithms (ESA 2022).
Editors: Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman; Article No. 22; pp. 22:1–22:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The Traveling Salesperson Problem is among the most popular and most intensively studied combinatorial optimization problems in graph theory[1]. From the practical point of view, the problem was already studied in the 1950s; see, e.g., [15]. We define the problem formally as follows.

Traveling Salesperson Problem (TSP)

Input: An undirected graph $G = (V, E)$, edge weights $\omega \colon E \to \mathbb{N}$, and a budget $\mathscr{b} \in \mathbb{N}$.
Question: Is there a closed walk in G of total weight at most $\mathscr{b}$ that traverses each vertex of G at least once?

TSP is known to be NP-hard [24]. It is worth mentioning that the input to TSP is usually a complete graph and it is required that each vertex is visited exactly once (single-visit version). In this paper, we aim to study the impact of the structure of the input graph on the complexity of finding a solution to TSP. Towards this, we employ parameterized analysis [16] and therefore we also consider the decision version of TSP. Many variants of the problem have been studied in detail [29]. Our formulation is also known as Graphical TSP [14].

In this paper, we further study two natural generalizations of TSP– Subset TSP and Waypoint Routing Problem. In Subset TSP (subTSP) the objective is to find a closed walk required to traverse only a given subset W of vertices (referred to as *waypoints*) instead of the whole vertex set as is in TSP. An instance of TSP can be interpreted as an instance of subTSP by letting $W = V(G)$. In Waypoint Routing Problem (WRP) the edges of the input graph have capacities (given by a function $\kappa \colon E(G) \to \mathbb{N}$) and the objective is to find a closed walk traversing a given subset of vertices respecting the edge capacities, i.e., there is an upper limit on the number of times each edge can be traversed in a solution. As we show later, an instance of subTSP can be interpreted as an instance of WRP by letting $\kappa(e) = 2$ for every $e \in E(G)$. Note that both subTSP and WRP are NP-complete.

Related Work. TSP (and its variants) were extensively studied from the viewpoint of approximation algorithms. The single-visit version of TSP in general cannot be approximated, unless P = NP [51]. For the metric single-visit version of the problem Christofides [13] provided a $\frac{3}{2}$-approximation, while it is known that unless P = NP, there is no $\frac{117}{116}$-approximation algorithm [12]. This is so far the best known approximation algorithm for the general case, despite a considerable effort, e.g., [54, 26, 10, 23, 8, 30, 37].

The problem remains APX-hard even in the case of weights 1 and 2, however, a $\frac{7}{6}$-approximation algorithm is known for this special case [50]. A PTAS is known for the special case of Euclidean [4, 47] and planar [27, 3, 38] TSP. Also, the case of graph metrics received significant attention. Gharan et al. [25] found a $\frac{3}{2} - \varepsilon_0$ approximation for this variant. Mömke and Svensson [48] then obtained a combinatorial algorithm for graphic TSP with an approximation factor of 1.461. This was later improved by Mucha [49] to $\frac{13}{9}$ and then by Sebö and Vygen [53] to 1.4. See, e.g., the monograph of Applegate et al. [2] for further information.

A popular practical approach to the single-visit version of TSP is to gradually improve some given solution using local improvements – the so-called q-OPT moves (see, e.g., [35, 36]). Marx [45] proved that TSP is W[1]-hard w.r.t. q. Later, Bonnet et al. [7] studied the q-OPT

[1] TSP Homepage: `http://www.math.uwaterloo.ca/tsp/index.html`.

technique on bounded degree graphs. They also investigated the complexity in the case where q is a fixed constant. The first significant theoretical improvement in the general case was by de Berg et al. [6] (announced at ICALP '16) who gave an $\mathcal{O}(n^{\lfloor 2q/3 \rfloor +1})$ time dynamic programming based algorithm. Furthermore, they showed that improving upon the $\mathcal{O}(n^3)$ time for the case of 3-OPT would yield a breakthrough result by a connection with the All Pairs Shortest Paths (APSP) problem (see, e.g., [11]). Lancia and Dalpasso [40] observed that the DP-based approach is indeed slow in practice and implemented a practical fast algorithm tailored for 4-OPT. Later, Cygan et al. [17] improved the result of de Berg et al. [6] by presenting an $\mathcal{O}(n^{(1/4+\varepsilon_q)q})$ time algorithm. Moreover, they showed a connection of 4-OPT to APSP. Guo et al. [28] considered other suitable measures (parameters) for local search (e.g., r-swap or r-edit) and provided a W[1]-hardness result in general graphs (for both parameters). Furthermore, they gave an FPT algorithm for planar graphs and showed that the existence of polynomial kernels is unlikely even in this case.

Subset TSP can be solved in time $2^\ell \cdot n^{\mathcal{O}(1)}$ by first computing the distance between every pair of waypoints, and then solving the resulting ℓ-waypoint instance using the standard Bellman-Held-Karp dynamic programming algorithm [5, 31]. Klein and Marx [39] showed that if G is an undirected planar graph with polynomially bounded edge weights, then the problem can be solved significantly faster, in $2^{\mathcal{O}(\sqrt{\ell}\log \ell)} \cdot n^{\mathcal{O}(1)}$ time. It is also known that Subset TSP can be solved in $2^{\mathcal{O}(\sqrt{\ell}\log \ell)} \cdot n^{\mathcal{O}(1)}$ on edge-weighted directed planar graphs [46].

Amiri et al. [1] initiated the study of the variant of WRP discussed in this paper. They showed that the problem admits an XP algorithm w.r.t. the treewidth of the input graph. They also gave a randomized algorithm for the problem with running time $2^\ell \cdot n^{\mathcal{O}(1)}$, where ℓ is the number of waypoints; the algorithm can be derandomized if $\ell = \mathcal{O}((\log\log n)^{1/10})$. Schierreich and Suchý [52] improved the former algorithm by giving a deterministic algorithm with running time $2^{\mathcal{O}(\mathrm{tw})} \cdot n$, which is tight under ETH [33]. Note that Hamiltonian Path and therefore TSP is W[1]-hard w.r.t. the cliquewidth of the input graph [21].

It is somewhat surprising that, despite its popularity in theoretical computer science, the TSP problem did not receive much attention from the perspective of data reduction (kernelization). A notable exception is the work of Mannens et al. [43, 44] who studied Many Visits TSP and gave a polynomial kernel when the problem is parameterized by the vertex cover number of the input graph. In this work, we significantly expand the kernelization approach to TSP and its variants.

Structural parameters within the class of bounded treewidth. Even though all problems in FPT admit a kernel, not all of them admit a polynomial-sized kernel. It is not hard to see that TSP is AND-compositional and therefore we can (conditionally) exclude the existence of a polynomial kernel w.r.t. treewidth; more precisely, we show the following.

▶ **Lemma 1** ($\star$). *There is no polynomial kernel for unweighted Traveling Salesperson Problem with respect to either the fractioning number or the combined parameter treewidth and maximum degree, unless the polynomial hierarchy collapses.*

It follows that in order to obtain positive results we should investigate parameters within the class of bounded treewidth. Note that, roughly speaking, fractioning number is a *modulator to parameter-sized components.* A modulator is a set of vertices M such that when we remove M from a graph G the components of $G \setminus M$ fall into a specific (preferably simple) graph class (see Section 2 for formal definitions). Since almost all graph classes are closed under disjoint union, it is also popular to speak of addition of k vertices to a graph class (i.e., for a graph class $\mathcal{G}$ we write $\mathcal{G} + kv$ if we are "adding" k vertices). Thus, it naturally models

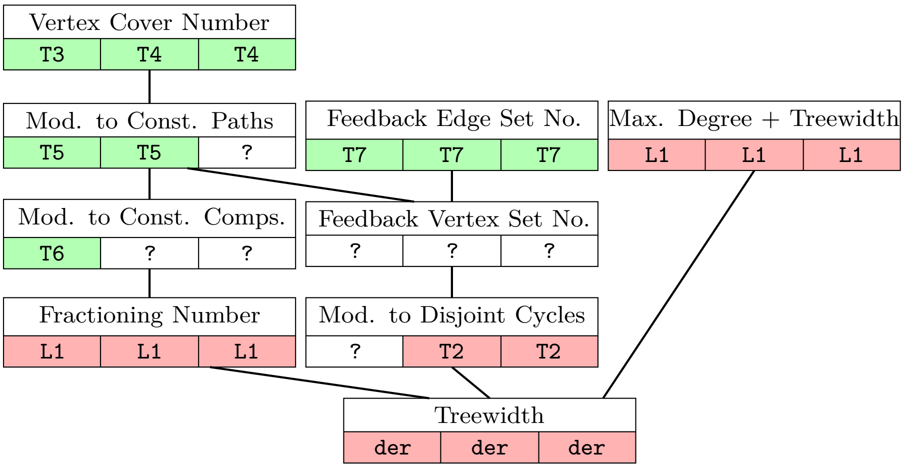

Figure 1 Overview of our results. Each node names a graph parameter and at its bottom it contains boxes representing (from left to right) TSP, SUBTSP, and WRP. If a node A is connected to a lower node B with an edge, then $B(G) = \mathcal{O}(A(G))$ for respective parameters. *Modulator*, *Constant*, and *Components* are shortened. Each box contains either a reference to the theorem which proves existence or non-existence of polynomial kernel (e.g. `T2`), or it contains `?` if the problem is open, or it contains `der` if the result for such setting is derived from another result.

the situation when the input is close to a class of graphs where we can solve the problem efficiently (also called "Distance to $\mathcal{G}$"). Modulator size is a popular structural parameter (see, e.g., [9, 41, 42]). Thus, the above result motivates us to study modulator-based parameters and their effect on kernelization of TSP and its variants. Indeed, many problems admit much more efficient algorithms w.r.t. vertex cover than for treewidth parameterization; see, e.g., [34, 19, 20]. Therefore, this is exactly the point where the positive part of our journey begins.

Before we do so, we discuss one more negative result. For SUBTSP we can, under a standard complexity-theoretic assumption, exclude the existence of a polynomial (Turing) kernel for the parameter size of a modulator to graphs of treewidth two. More specifically, we show the following.

▶ **Theorem 2** ($\star$). *Unweighted* SUBTSP *with respect to the minimum k' such that there is a size k' modulator to cycles of size at most k' and there are at most k' non-terminals is* WK[1]*-hard.*

Therefore, SUBTSP parameterized by the size of a modulator to (disjoint) cycles does not admit a polynomial (Turing) kernel, unless all problems in WK[1] admit a polynomial Turing kernel. Moreover, it does not admit (classical) polynomial kernel, unless the polynomial hierarchy collapses. For more details on WK[1]-completeness, see [32].

Our Contribution. On a positive note, we begin with a polynomial kernel for TSP and WRP w.r.t. vertex cover number to demonstrate the general techniques we employ in this paper. We study the properties of a nice solution to the instance with respect to a particular structural parameter and its interaction with the modulator. We begin the study with TSP as a warm-up. In this case, we show that if there are "many" vertices in $G \setminus M$, where M is the modulator (vertex cover), then all but few of them behave in a "canonical way". Based on the properties and cost of this canonical traversal, we identify which of these vertices can be safely discarded from the instance. In particular, we show the following.

▶ **Theorem 3** (⋆). *Traveling Salesperson Problem admits a kernel with $\mathcal{O}(k^3)$ vertices, $\mathcal{O}(k^4)$ edges, and with a total bit-size $\mathcal{O}(k^{16})$, where k is the vertex cover number of G.*

In the case of subTSP or WRP, we cannot be sure that a solution visits all vertices in the modulator. Quite naturally, this results in the second kind of rule where we see that if there are "many" vertices attached to the modulator in the same way, we can mark some vertices in the modulator as terminals (there is a solution visiting them).

▶ **Theorem 4** (⋆). *Waypoint Routing Problem admits a kernel with $\mathcal{O}(k^8)$ vertices, $\mathcal{O}(k^9)$ edges, and with a total bit-size $\mathcal{O}(k^{36})$, where k is the vertex cover number of G.*

Using similar, yet more involved results, we are able to find a polynomial kernel for subTSP for a modulator to constant-sized paths and for TSP for a modulator to constant-sized components.

One ingredient for our approach is that we may assume that all vertices outside the modulator are terminals, and therefore every solution visits them. It is not clear how to use our approach in the presence of capacities already in the case of modulator to paths, since the previous observation does not hold already in this simple case. Indeed, we show this by contracting some of the edges, which in the presence of capacities might have a side effect on the set of possible traversals of the component. The other and more important ingredient is the so-called blending of solutions (see Lemma 30).

▶ **Theorem 5** (⋆). *Let r be a fixed constant. Subset TSP admits a kernel of size $k^{\mathcal{O}(r)}$, where k is the size of a modulator to paths of size at most r.*

When we allow general constant-sized components outside the modulator, we have not yet succeeded in dealing with the highly connected non-terminals in the modulator. At this point, it is not clear whether many components with the same combination of canonical transversal and attachment to the modulator set ensure that we can safely mark a vertex in the modulator as a terminal.

▶ **Theorem 6** (⋆). *Let r be a fixed constant. Traveling Salesperson Problem admits a kernel of size $k^{\mathcal{O}(r)}$, where k is the size of a modulator to components of size at most r.*

We conclude our algorithmic study by providing a rather simple polynomial kernel for WRP parametrized by the feedback edge set number. On the one hand, this result is an application of rather straightforward local reduction rules. On the other hand, these rules are heavily based on the use of edge capacities. It should be pointed out that we get a polynomial kernel for TSP as it has a polynomial compression to WRP, however, it can be shown that "local reduction rules" do not exist in the case of TSP.

▶ **Theorem 7** (⋆). *Waypoint Routing Problem admits a kernel with $\mathcal{O}(k)$ vertices and edges and bit size $\mathcal{O}(k^4)$, where k is the feedback edge set number of G.*

For an overview of our results, please, refer to Figure 1.

Organization of the paper. We summarize the notation and technical results we rely on in Section 2. Section 3 contains a few useful technical lemmas and simple reduction rules. In Section 4, we begin with the core concepts applied in the case of TSP and vertex cover number. The similar approach is then applied in Section 5 to more general types of modulators; this is the most technical part of this manuscript. Finally, in Section 6 we conclude the results and discuss future research directions.
Statements where proofs or details are omitted due to space constraints are marked with ⋆. All missing proofs and details are available in the full version of the paper.

2 Preliminaries

We follow the basic notation of graph theory by Diestel [18]. In parameterized complexity theory, we follow the monograph of Cygan et al. [16].

A *walk* in a graph G is a non-empty alternating sequence of vertices and edges $S = v_1, e_1, \ldots, e_{\ell-1}, v_\ell$ such that $v_i \in V(G)$, $e_i \in E(G)$, and $e_i = \{v_i, v_{i+1}\}$, $\forall i \in \{1, \ldots, \ell-1\}$. It is *closed* if $v_\ell = v_1$. The weight of walk S is $\omega(S) = \sum_{i=1}^{\ell-1} \omega(e_i)$. If all vertices in a walk are distinct, it is called a *path*.

A *solution* to our problems is a closed walk S visiting every vertex in W of total weight at most $\mathscr{b}$ (traversing each edge e at most $\kappa(e)$ times). The least cost such walk is called *an optimal solution*. Given such a walk, we can construct the *corresponding multigraph G_S* which is a multigraph with vertex set being the set of vertices visited by S and each edge occuring as many times as traversed by S. We naturally extend ω to this multigraph, which yeilds $\omega(G_S) = \sum_{e \in E(G_S)} \omega(e) = \omega(S)$. The degree of a vertex in a multigraph is the number of the edges incident with it. Conversely, if a multigraph G_S is Eulerian (connected with all degrees even), then it admits a walk visiting every vertex of the graph and traversing each edge exactly as many times as occuring in G_S.

Structural Graph Parameters. Let $G = (V, E)$ be a graph. A set of edges $F \subseteq E$ is a *feedback edge set* of the graph G if $G \setminus F$ is an acyclic graph. *Feedback edge set number* $\mathrm{fes}(G)$ of a graph G is the size of a smallest feedback edge set F of G. A set of vertices $C \subseteq V$ is called *vertex cover* of the graph G if it holds that $\forall e \in E$ we have $e \cap C \neq \emptyset$. The *vertex cover number* $\mathrm{vc}(G)$ of a graph G is the least size of a vertex cover of G.

▶ **Definition 8.** *Let $\mathcal{G}$ be a graph family, let G be a graph, and $M \subseteq V(G)$. We say that M is a* modulator *of the graph G to the class $\mathcal{G}$ if each connected component of $G \setminus M$ is in $\mathcal{G}$. The* distance of G to $\mathcal{G}$, *denoted* $\mathrm{modul}(G, \mathcal{G})$, *is the minimum size of a modulator of G to $\mathcal{G}$.*

Let r be a fixed constant. Let $\mathcal{G}_{\mathrm{rc}}$ be the class of graphs with every connected component having at most r vertices. The *distance of G to r-components* is $\mathrm{modul}(G, \mathcal{G}_{\mathrm{rc}})$. Similarly, if $\mathcal{G}_{\mathrm{rp}}$ is a class of graphs where every connected component is a path with at most r vertices, then the *distance of G to r-paths* is $\mathrm{modul}(G, \mathcal{G}_{\mathrm{rp}})$.

A set of vertices $C_r \subseteq V$ is called *r-fractioning set* of the graph G if $|C_r| \leq r$ and every connected component of $G \setminus C_r$ has at most r vertices. The *fractioning number* $\mathrm{fn}(G)$ is a minimum $r \in \mathbb{N}$ such that there is an r-fractioning set C_r in graph G.

3 The Toolbox

In this section, we give formal proofs to a few technical statements we use throughout the rest of the paper. This yields a useful set of assumptions that allow us to present less technical proofs in the subsequent sections. We begin with a technical lemma we were not able to find in literature.

▶ **Lemma 9** ($\star$). *Let G be a graph with more than $2|V(G)| - 2$ edges. Then there is a cycle C in G such that the graph $G' = G \setminus E(C)$ has the same connected components as G.*

When proving the safeness of our reduction rules, it is easier to work with a solution that does not use many edges and traverses each edge at most twice. We show that we can always assume to work with such a solution.

▶ **Definition 10** (Nice Solution). *Let $(G, W, \omega, \kappa, \mathscr{b})$ be an instance of TSP, Subset TSP, or WRP. We call a solution* nice *if it uses every edge at most twice and contains at most $2|V|$ edges (edge traversals) in total.*

Indeed, we may always assume that we work with a nice solution.

▶ **Lemma 11** ($\star$). *Let $(G, W, \omega, \kappa, \mathscr{b})$ be an instance of TSP, Subset TSP, or WRP. There is a nice optimal solution.*

We continue with a remark about our instances. It is important to note that item (b) is applicable in general, however, one should be careful when doing so, since it increases the weights in the instance.

▶ Remark 12 ($\star$). Let $(G, W, \omega, \kappa, \mathscr{b})$ be an instance of TSP, Subset TSP, or WRP. We assume that a) G is a connected graph and b) $\omega(e) > 0$ for all $e \in E(G)$.

We apply the following lemma to reduce the weights of the kernelized instances. It follows in a rather straightforward way from results of Frank and Tardos [22].

▶ **Lemma 13** ($\star$). *There is a polynomial time algorithm, which, given an instance $(G, W, \omega, \kappa, \mathscr{b})$ of TSP, subTSP, or WRP with at most d edges, produces ω' and $\mathscr{b}'$ such that the instances $(G, W, \omega, \kappa, \mathscr{b})$ and $(G, W, \omega', \kappa, \mathscr{b}')$ are equivalent and $(G, W, \omega', \kappa, \mathscr{b}')$ is of total bit-size $\mathcal{O}(d^4)$.*

Finally, we present two simple reduction rules; we always assume that Reduction Rule 1 is not applicable.

▶ **Reduction Rule 1** ($\star$). *Let an instance of TSP, subTSP, or WRP be given. If $\mathscr{b} < 0$, then answer* no. *Otherwise, if $|W| \le 1$, then answer* yes.

▶ **Reduction Rule 2** ($\star$). *Let I be an instance of Subset TSP and $v \notin W$. For each pair of vertices $u, w \in N(v)$ we introduce a new edge $\{u, w\}$ into the graph with $\omega(\{u, w\}) = \omega(\{u, v\}) + \omega(\{v, w\})$ (if this creates parallel edges, then we only keep the one with the lower weight). Finally, we remove v together with all its incident edges.*

While Reduction Rule 2 is easy to apply, its application may destroy the structure of the input graph. Therefore, we only apply it to some specific vertices as explained in the subsequent sections.

4 Polynomial Kernel with Respect to Vertex Cover Number for TSP

In this (warm-up) section, we argue that TSP admits a polynomial kernel with respect to the vertex cover number. That is, we are going to present the most simple use-case of our reduction rules and therefore we can focus on the introduction of the core concept – the natural behavior. We begin with the definition of a (natural) behavior, which is a formal description of how a vertex "can behave" in a solution. Let M be a vertex cover of G of size $k = \mathrm{vc}(G)$ and let $R = V \setminus M$.

▶ **Definition 14.** *For a vertex $r \in R$* a behavior of r *is a multiset $F \subseteq \{\{r, m\} \mid \{r, m\} \in E, m \in M\}$ containing exactly two edges (edge occurrences). We let $B(r)$ be the set of all behaviors of r. We naturally extend the weight function such that for a behavior $F \in B(r)$ we set its weight to $\omega(F) = \sum_{e \in F} \omega(e)$. For a vertex $r \in R$ its* natural behavior *$b^{\mathrm{nat}}(r)$ is a fixed minimizer of $\min\{\omega(F) \mid F \in B(r)\}$ that takes two copies of a minimum weight edge incident with r.*

The following lemma shows that in an optimal solution, most of the vertices of R actually use some behavior.

▶ **Lemma 15** ($\star$). *Let S be an optimal solution. Then the number of vertices $r \in R$ that are traversed more than once by S is at most k.*

Now, that we know what a behavior is, we can observe that each vertex can have one of two roles in the sought solution – they are either "just attached" using the natural behavior or they provide some connectivity between (two) vertices in the vertex cover. The role of a vertex is formalized as follows.

▶ **Definition 16.** *Let $r \in R$ and $F \in B(r)$ be a behavior of r. We say that F* touches *a vertex m of M if m has nonzero degree in $(\{r\} \cup M, F)$. We call the set* $\operatorname{imp}(F)$ *of touched vertices* the impact *of behavior F.*

Let $\mathcal{I}_r = \{\operatorname{imp}(F) \mid F \in B(r)\}$ be the set of all possible impacts of behaviors of r. Let $\mathcal{I} = \bigcup_{r \in R} \mathcal{I}_r$.

Note that we have $1 \le |\operatorname{imp}(F)| \le 2$ for each behavior F. Therefore, we get $|\mathcal{I}| \le \binom{k}{2} + k \le k^2$. Note also that for each $r \in R$ we have $|\operatorname{imp}(b^{\text{nat}}(r))| = 1$.

Now, we show that, in large instances, most of the vertices of R fall back to their natural behavior in an optimal solution. Towards this, we take a solution that differs from this in the fewest possible vertices $r \in R$. Then, we observe that if r is not in the natural behavior, then it is attached to (at least) two vertices in M. If there are many such "extra" edges in a solution, we can apply Lemma 9 – this would yield a contradiction. Thus, we get the following.

▶ **Lemma 17** ($\star$). *There is an optimal solution S such that for all but at most $3k$ vertices $r \in R$ the solution contains exactly the edges of $b^{\text{nat}}(r)$ among the edges incident with r.*

Now, we know that there are only a few vertices that behave unnaturally in an optimal solution, we would like to keep these in the kernel. We arrive at the question of which vertices to keep. To resolve this question, we observe that if a vertex deviates from its natural behavior, we (possibly) have to pay for this some extra price (which comes in the exchange of the provided connectivity). Thus, we would like to keep sufficiently many vertices for which this deviation is cheap. To this end, we first formalize the price.

▶ **Definition 18.** *For $r \in R$ and an impact $I \in \mathcal{I}$ let* the price $P(r, I)$ of change *from $b^{\text{nat}}(r)$ to I at r be $\omega(F_I) - \omega(b^{\text{nat}}(r))$ if there is a behavior $F_I \in B(r)$ with* $\operatorname{imp}(F_I) = I$ *and we let it be ∞ otherwise.*

Note that if there is a behavior $F_I \in B(r)$ with $\operatorname{imp}(F_I) = I$, then it is unique.

Now, based on this, we can provide the following reduction rule we employ.

▶ **Reduction Rule 3.** *For each $I \in \mathcal{I}$ if there are at most $3k$ vertices $r \in R$ with finite $P(r, I)$, then mark all of them. Otherwise mark $3k$ vertices $r \in R$ with the least $P(r, I)$.*

For each unmarked $r \in R$, remove r and decrease $\mathscr{b}$ by $\omega(b^{\text{nat}}(r))$.

Safeness. Let $(G, \omega, \mathscr{b})$ be the original instance and $(\widehat{G}, \widehat{\omega}, \widehat{\mathscr{b}})$ be the new instance resulting from the application of the rule. Note that $\widehat{\omega}$ is just the restriction of ω to $\widehat{G}$ which is a subgraph of G. Let R^- be the set of vertices of R removed by the rule. Note that $\widehat{\mathscr{b}} = \mathscr{b} - \sum_{r \in R^-} \omega(b^{\text{nat}}(r))$. We first show that if the new instance is a *yes*-instance, then so is the original one.

Let $\widehat{S}$ be a solution walk in the new instance and let $\widehat{G}_S$ be the corresponding multigraph formed by the edges of $\widehat{S}$. Note that the total weight of $\widehat{G}_S$ is at most $\widehat{b}$. We construct a multigraph G_S by adding to $\widehat{G}_S$ for each $r \in R^-$ the vertex r together with the edge set $b^{\mathrm{nat}}(r)$. Since $\widehat{G}_S$ is connected and each $b^{\mathrm{nat}}(r)$ is incident with a vertex of $M \subseteq V(\widehat{G}_S)$, G_S is also connected. As each addition increases the degree of the involved vertex by exactly 2 and the degree of each vertex is even in $\widehat{G}_S$, it follows that the degree of each vertex is even in G_S. Hence G_S is Eulerian and contains all vertices of G. Since the weight of the added edges is exactly $\sum_{r \in \mathcal{R}^-} \omega(b^{\mathrm{nat}}(r))$ it follows that (G, ω, b) is a *yes*-instance.

Now suppose that the original instance (G, ω, b) is a *yes*-instance.

▷ Claim 19 (⋆). There is an optimal solution S for (G, ω, b) such that each $r \in R^-$ is incident exactly to the edges of $b^{\mathrm{nat}}(r)$ in S.

Let S be a solution for (G, ω, b) as in the claim and G_S the corresponding multigraph. Let $\widehat{G}_S$ be obtained from G_S by removing the edges of $b^{\mathrm{nat}}(r)$ and vertex r for each $r \in R^-$. This reduces the total weight by exactly $\sum_{r \in R^-} \omega(b^{\mathrm{nat}}(r))$, hence, as G_S is of weight at most b, $\widehat{G}_S$ is of weight at most $\widehat{b}$. Furthermore, as each $r \in R^-$ has only one neighbor in G_S, $\widehat{G}_S$ is connected. Moreover, each removal decreases the degree of exactly one remaining vertex by exactly 2, hence all degrees in $\widehat{G}_S$ are even. Thus $\widehat{G}_S$ is Eulerian, yielding a solution $\widehat{S}$ for $(\widehat{G}, \widehat{w}, \widehat{b})$. This completes the proof. ◀

Since $|\mathcal{I}| \le k^2$, the following observation is immediate.

▶ **Observation 20** (⋆). *After Reduction Rule 3 has been applied, the number of vertices in* R *is bounded by* $3k^3$ *and, hence, the total number of edges is* $\mathcal{O}(k^4)$.

We conclude this section in the following theorem (which follows using Lemma 13).

▶ **Theorem 3** (⋆). *Traveling Salesperson Problem admits a kernel with* $\mathcal{O}(k^3)$ *vertices,* $\mathcal{O}(k^4)$ *edges, and with a total bit-size* $\mathcal{O}(k^{16})$*, where* k *is the vertex cover number of* G.

5 Polynomial Kernels with Respect to Modulator Size

In this section, we move to more general type of modulators. Recall that if M is a vertex cover, the connected components of $G \setminus M$ are single vertices. We are going to relax this a bit – we shall investigate general components of constant size and paths of constant size. For the first, more general one, we obtain a polynomial kernel for TSP. For the second, we obtain a polynomial kernel even for subTSP.

5.1 TSP and the Distance to Constant Size Components

For the rest of the section, we assume that we are given an undirected graph $G = (V, E)$, k is its distance to r-components, M the corresponding modulator, $E_M = \binom{M}{2}$, and $R = G \setminus M$. Let $M = \{m_1, \dots, m_k\}$ be a fixed order of the vertices of the modulator.

We begin with the definition of a behavior.

▶ **Definition 21.** *For a connected component* C *of* R *we consider the graph* $G_C = G[C \cup M] \setminus E_M$. A behavior of C *is a multiset* $F \subseteq E[G_C]$ *of edges of* G_C *(each can be used at most twice) such that in the multigraph* $(C \cup M, F)$*, (i) each vertex* $v \in C$ *has nonzero even degree and (ii) each connected component contains a vertex of* M*, and the set* F *contains at most* $2r$ *edges incident with the vertices of* M*. We let* $B(C)$ *be the set of all behaviors of* C*. For a component* C its natural behavior $b^{\mathrm{nat}}(C)$ *is a fixed minimizer of* $\min\{\omega(F) \mid F \in B(C)\}$.

Now, it is not hard to see, that a solution might "visit" a component of $G \setminus M$ more than once. With this, we arrive at the key notion we introduce in this section – the segments. It is not hard to see that a solution may even revisit some terminals in a component; we shall later prove that this is the case for a few exceptional components only (see Lemma 25).

▶ **Definition 22.** *Let C be a connected component of R and S be a walk starting and ending in $v_0 \in M$ forming a nice optimal solution to the instance. We split S into* segments $S_1, \dots, S_q$ *such that each segment starts and ends in a vertex of M, whereas the inner vertices of each segment are from R. Let $\mathcal{F}(S, C)$ be obtained from the empty set by the following process: For each i, add S_i into $\mathcal{F}(S, C)$ if and only if there is a vertex $c \in C$ visited by S_i and not visited by any of the previous segments. Let $F(S, C)$ be the union of edges of S_i, $S_i \in \mathcal{F}(S, C)$, including multiplicities.*

We observe that the multiset $F(S, C)$ is a behavior of C, i.e, $F(S, C) \in B(C)$.

Again, for a connected component of $G \setminus M$, we want to introduce the notion of touched vertices and the (connectivity) impact of a behavior of the component. For a better understanding of Definition 23, we refer the reader to an example in Figure 2.

▶ **Definition 23.** *Let C be a connected component of R and $F \in B(C)$ be a behavior of C. We say that F* touches *a vertex m of M if m has nonzero degree in $(C \cup M, F)$. Let $T(F)$ be the set of touched vertices.*

Consider the multiset $D(F)$ of edges obtained from the empty set as follows. For each connected component H of $(C \cup M, F)$ containing at least two vertices of M, let m_i be the vertex with the least index in $M \cap V(H)$. For each m_j in $(M \cap V(H)) \setminus \{m_i\}$ add to $D(F)$ a single edge $\{m_i, m_j\}$ if m_j is incident with an odd number of edges of F and a double edge $\{m_i, m_j\}$ if m_j is incident with an even number of edges of F.

We call the pair $\operatorname{imp}(F) = (T(F), D(F))$ the impact *of behavior F.*

Let $\mathcal{I}_C = \{\operatorname{imp}(F) \mid F \in B(C)\}$ be the set of all possible impacts of behaviors of C.

Let $\mathcal{I} = \bigcup_{C \text{ connected component of } R} \mathcal{I}_C$.

From the Handshaking Lemma we get the following observation.

▶ **Observation 24.** *Let m_i be as in Definition 23. Then m_i is incident with an even number of edges of $D(F)$ if and only if it is incident with an even number of edges of F.*

It is not hard to see that $|\mathcal{I}| = k^{\mathcal{O}(r)}$.

Now, we want to give some properties that an optimal solution should have. Namely, we want to see that most of the connected components of $G \setminus M$ fallback to their natural behavior. Towards this, we first prove that there are only a few segments that are not part of a behavior of any component; recall that if the solution "revisited" some terminals in the connected component, this segment might not be part of any behavior.

▶ **Lemma 25** ($\star$). *There is an optimal solution S such that there are at most $2|M|$ segments of S that are not part of any $\mathcal{F}(S, C)$, and, furthermore, for all but at most $2|\mathcal{I}|^2 + 2|M|$ components we have $F(S, C) = b^{\text{nat}}(C)$ and $F(S, C)$ contains all edges of S incident with C.*

Next, we want to introduce the notion of price of a transition between two impacts of a connected component.

▶ **Definition 26.** *For a connected component C of R and an impact $I \in \mathcal{I}$ let $\mathcal{B}(C, I)$ be the set of behaviors $F \in B(C)$ satisfying* $\operatorname{imp}(F) = I$.

For a connected component C of R and a pair $I, I' \in \mathcal{I}$ let the price $P(C, I, I')$ of change *from I to I' at C be* $\min\{\omega(F) \mid F \in \mathcal{B}(C, I')\} - \omega(b^{\text{nat}}(C))$ *if $\mathcal{B}(C, I')$ is non-empty and $I = \operatorname{imp}(b^{\text{nat}}(C))$ and we let it be ∞ if some of the conditions is not met.*

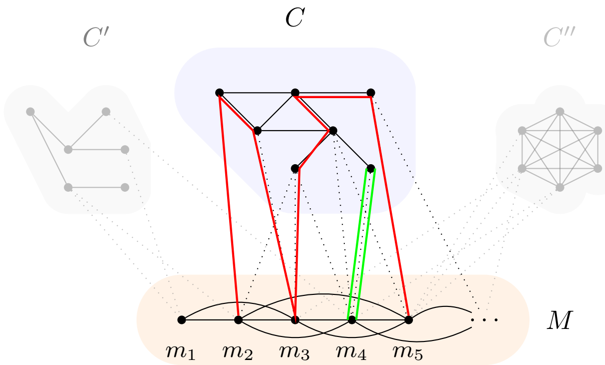

Figure 2 Illustration of Definition 23. A behavior F consists of red and green edges, and vertices $m_2, \dots, m_5$ made the set $T(F)$ of touched vertices. We built the multiset $D(F)$ for behavior F as follows. We start with an empty set. For the red connected component, the vertex $m_i \in M$ with least index is m_2. Thus, for m_3 we add to $D(F)$ double edge $\{m_2, m_3\}$, because the vertex m_3 is incident with even number of edges in behavior F. For the vertex m_5, we add to $D(F)$ the edge $\{m_2, m_5\}$ only once, as the vertex m_5 is incident with odd number of edges in F. For the green connected component, we do not extend $D(F)$ since, in this connected component, there is only one vertex from M.

Now, we are ready to introduce the reduction rule used in this section. To keep a record of the vertices that are marked towards achieving a different goal, we mark them in different colors. *Red* vertices are those that behaves unnaturally in an optimal solution and their deviation from natural behavior is cheap. The task of the vertices marked in *green* is to ensure that at least one vertex with the same impact of natural behavior remains in the instance to provide the connectivity. Finally, there are components marked in *blue*. These are supposed to cover the segments which are not part of any behavior.

▶ **Reduction Rule 4** ($\star$). *For each pair $I, I' \in \mathcal{I}$ if there are at most $2|\mathcal{I}|^2 + 2|M|$ components C with finite $P(C, I, I')$, then mark all of them in red. Otherwise mark $2|\mathcal{I}|^2 + 2|M|$ components C with the least $P(C, I, I')$ in red.*

For each pair of vertices $u, v \in M$, if there is a component C in R such that G_C contains a u-v-path, then mark a component which contains the shortest such path in blue.

For each $I \in \mathcal{I}$, if there are unmarked components C with $\operatorname{imp}(b^{\text{nat}}(C)) = I$, then do the following. If the number of such components is odd, then mark one arbitrary such component in green. If the number of such components is even, then mark two arbitrary such components in green.

For each unmarked component C, remove C from G and reduce $\mathscr{b}$ by $\omega(b^{\text{nat}}(C))$.

To prove Theorem 6 we estimate the number of vertices and edges in the reduced instance; Theorem 6 then follows using Lemma 13.

▶ **Lemma 27** ($\star$). *After Reduction Rule 4 has been applied, the number of components is bounded by $2(|\mathcal{I}|^2 + 2|M|)|\mathcal{I}|^2 + \binom{|M|}{2} + 2|\mathcal{I}| = k^{\mathcal{O}(r)}$. Hence, the number of vertices and edges in the reduced graph G is $k^{\mathcal{O}(r)}$.*

5.2 Subset TSP and the Distance to Constant Size Paths

We start by applying Reduction Rule 2 to all vertices of R. Note that after each application of the reduction rule R remains a disjoint union of paths, because each vertex has degree at most 2 within R. Hence, for the rest of the section we assume that $V(R) \subseteq W$.

We reuse the notions of the (natural) behavior, impact, and price (Definitions 21, 23, and 26) from the previous section.

We define *piece* of $F \in B(C)$ as any connected component of $(C \cup M, F) \setminus M$ to which we add all incident edges in G_S. We define *legs* of a piece as a subset of its edges which are incident with vertices of M. As $G \setminus M$ consists of disjoint union of paths we note that each piece of F consists of a path on C and its legs.

▶ **Lemma 28** ($\star$). *Given a* Subset TSP *instance* $(G, W, \omega, \mathfrak{b})$ *let* S *be a nice optimal solution to the instance. Let* k *be the size of the modulator to disjoint union of paths. There are at most* k *components with pieces of* $F(S, C)$ *with more than* 2 *legs in behaviors* $\{F(S, C) \mid \operatorname{imp}(F(S, C)) = I\}$ *for any fixed* $I \in \mathcal{I}$.

It is the case that every natural behavior has only two-legged pieces.

▶ **Observation 29** ($\star$). *For each* C *component of* R*, each piece of* $b^{\text{nat}}(C)$ *has two legs.*

The most technical part of this section is the following lemma that allows us to, under some conditions, mark a non-terminal in the modulator as a terminal. For this to work, we want to do the following. Take many components which share the impact of the natural behavior which touches the particular non-terminal. Since there are many, many of them also share the impact of the actual behavior and have all pieces 2-legged. For each of them we want to find a behavior that is half-way between the actual and the natural behavior (using the following lemma). This behavior should touch the non-terminal and be of at most the same weight as the actual behavior. Then we find two components for which the half-way behavior has the same impact and change these, so that we obtain a solution that visits the particular non-terminal.

▶ **Lemma 30** (Blending lemma $\star$). *Let* $M' \subseteq M$ *(the set of actually visited vertices) and* C *a component of* R *(therefore a path) such that* $T(b^{\text{nat}}(C)) \nsubseteq M'$. *Let* $v \in T(b^{\text{nat}}(C)) \setminus M'$ *and* $A \in B(C)$ *(the actual behavior) such that* $T(A) \subseteq M'$ *and such that each piece has two legs. Then there is a behavior* $F \in B(C)$ *such that* $v \in T(F)$*,* $T(F) \subseteq T(A) \cup T(b^{\text{nat}}(C))$*, every connected component of* $(C \cup M, F)$ *contains a vertex of* M'*, and* $\omega(F) \leq \omega(A)$.

Now, we are ready to present the reduction rule used in this case. The colors have the same meaning as in Reduction Rule 4. It remains to secure that every vertex in M that is incident with some natural behavior for many vertices in R is in W. Towards this, we prove that if a vertex $m \in M$ is needed for a natural behavior for many vertices in R, it is safe to mark m as a terminal. If this is not the case, we mark the vertices $r \in R$ in *yellow*.

▶ **Reduction Rule 5** ($\star$). *For each pair* $I, I' \in \mathcal{I}$ *of impacts if there are at most* $2|\mathcal{I}|^2 + 2|M|$ *components* C *with finite* $P(C, I, I')$*, then mark all of them in red. Otherwise mark* $2|\mathcal{I}|^2 + 2|M|$ *components* C *with the least* $P(C, I, I')$ *in red.*

For each pair of vertices $u, v \in M$*, if there is a component* C *in* R *such that* G_C *contains a* u*-*v*-path, then mark the component which contains the shortest such path in blue.*

For each $I \in \mathcal{I}$*, if there are unmarked components* C *with* $\operatorname{imp}(b^{\text{nat}}(C)) = I$*, then let* $I = (T, D)$ *and do the following. If* $T \subseteq W$*, then if the number of such components is odd, then mark one arbitrary such component in green. If the number of such components is even, then mark two arbitrary such components in green.*

If $T \nsubseteq W$ *and there are at most* $\big((r+1)^{4r} \cdot 2^{4r+1} + k\big) \cdot |\mathcal{I}|$ *unmarked components* C *with* $\operatorname{imp}(b^{\text{nat}}(C)) = I$*, then mark them all in yellow.*

If $T \nsubseteq W$ *and there are more than* $\big((r+1)^{4r} \cdot 2^{4r+1} + k\big) \cdot |\mathcal{I}|$ *unmarked components* C *with* $\operatorname{imp}(b^{\text{nat}}(C)) = I$*, then add* T *to* W.

If W was not changed, then for each unmarked component C, remove C from G and reduce $\mathfrak{b}$ by $\omega(b^{\text{nat}}(C))$.

To prove Theorem 5 we estimate the number of vertices and edges in the reduced instance; Theorem 5 then follows using Lemma 13.

▶ **Lemma 31** ($\star$)**.** *After Reduction Rule 5 has been applied, the number of components is bounded by $2(|\mathcal{I}|^2 + 2|M|)|\mathcal{I}|^2 + \binom{|M|}{2} + 2|\mathcal{I}| + \big((r+1)^{4r} \cdot 2^{4r+1} + k\big) \cdot |\mathcal{I}|^2 = k^{\mathcal{O}(r)}$. Hence, the number of vertices and edges in the reduced graph G is $k^{\mathcal{O}(r)}$.*

6 Conclusions

The core focus of this work is kernelization of the Traveling Salesperson Problem. To stimulate further research in this area we would like to promote some follow up research directions. Design of "local" rules might be impossible (as was mentioned in the case of feedback edge set number) and therefore we occasionally have to consider generalizations of this problem that give us more power when designing the reductions. An interesting open problem that remains in this area is whether TSP does admit a polynomial kernel with respect to the feedback vertex number. Towards this we have provided several steps that suggest it should be possible to design a polynomial kernel. It should be noted that in order to do so, one could try to "lift" our arguments to modulator to trees of any size. Note that this is not possible for cycles.

Yet another interesting line of work is the one of q-path vertex cover (q-pvc). This is a generalization of the vertex cover number – a graph G has q-pvc of size k if it has a modulator M (with $|M| = k$) such that in $G \setminus M$ there is no path of length q. It is not hard to see that some of our arguments can be applied for components of $G \setminus M$ that are stars (of arbitrary size). Therefore, it should be possible to give a polynomial kernel for 3-pvc. Thus, the question arises: Is there a polynomial kernel with respect to r-pvc for constant r? In this case we are more skeptical and believe that the correct answer should be negative.

References

1 Saeed Akhoondian Amiri, Klaus-Tycho Foerster, and Stefan Schmid. Walking through waypoints. *Algorithmica*, 82(7):1784–1812, 2020. doi:10.1007/s00453-020-00672-z.

2 David L. Applegate, Robert E. Bixby, Vašek Chvátal, and William J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2011. doi:10.1515/9781400841103.

3 Sanjeev Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science, FOCS '96*, pages 2–11. IEEE Computer Society, 1996. doi:10.1109/SFCS.1996.548458.

4 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, September 1998. doi:10.1145/290179.290180.

5 Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM*, 9(1):61–63, January 1962. doi:10.1145/321105.321111.

6 Mark de Berg, Kevin Buchin, Bart M. P. Jansen, and Gerhard Woeginger. Fine-grained complexity analysis of two classic TSP variants. *ACM Transactions on Algorithms*, 17(1), December 2021. doi:10.1145/3414845.

7 Édouard Bonnet, Yoichi Iwata, Bart M. P. Jansen, and Łukasz Kowalik. Fine-grained complexity of k-OPT in bounded-degree graphs for solving TSP. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *Proceedings of the 27th Annual European Symposium on Algorithms, ESA '19*, volume 144 of *Leibniz International Proceedings in Informatics*, pages 23:1–23:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.23.

8 Sylvia C. Boyd and Robert Carr. Finding low cost TSP and 2-matching solutions using certain half-integer subtour vertices. *Discrete Optimization*, 8(4):525–539, 2011. doi:10.1016/j.disopt.2011.05.002.

9 Leizhen Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127(3):415–429, 2003. doi:10.1016/S0166-218X(02)00242-1.

10 Robert D. Carr, Santosh S. Vempala, and Jacques Mandler. Towards a 4/3 approximation for the asymmetric traveling salesman problem. In David B. Shmoys, editor, *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '00*, pages 116–125. ACM/SIAM, 2000.

11 Timothy M. Chan and R. Ryan Williams. Deterministic APSP, Orthogonal Vectors, and more: Quickly derandomizing Razborov-Smolensky. *ACM Transactions on Algorithms*, 17(1):2:1–2:14, 2021. doi:10.1145/3402926.

12 Miroslav Chlebík and Janka Chlebíková. Approximation hardness of travelling salesman via weighted amplifiers. In Ding-Zhu Du, Zhenhua Duan, and Cong Tian, editors, *Proceedings of the 25th International Conference on Computing and Combinatorics, COCOON '19*, volume 11653 of *Lecture Notes in Computer Science*, pages 115–127, Cham, 2019. Springer. doi:10.1007/978-3-030-26176-4_10.

13 Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon University Pittsburgh, Management Sciences Research Group, 1976.

14 Gérard Cornuéjols, Jean Fonlupt, and Denis Naddef. The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, 33(1):1–27, 1985. doi:10.1007/BF01582008.

15 G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, December 1958. doi:10.1287/opre.6.6.791.

16 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, Cham, 2015. doi:10.1007/978-3-319-21275-3.

17 Marek Cygan, Łukasz Kowalik, and Arkadiusz Socała. Improving TSP tours using dynamic programming over tree decompositions. *ACM Transactions on Algorithms*, 15(4), October 2019. doi:10.1145/3341730.

18 Reinhard Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer, Berlin, Heidelberg, 5th edition, 2017. doi:10.1007/978-3-662-53622-3.

19 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances A. Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Information and Computation*, 209(2):143–153, 2011. doi:10.1016/j.ic.2010.11.026.

20 Jiří Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theoretical Computer Science*, 412(23):2513–2523, 2011. doi:10.1016/j.tcs.2010.10.043.

21 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Intractability of clique-width parameterizations. *SIAM Journal on Computing*, 39(5):1941–1956, 2010. doi:10.1137/080742270.

22 András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987. doi:10.1007/BF02579200.

23 David Gamarnik, Moshe Lewenstein, and Maxim Sviridenko. An improved upper bound for the TSP in cubic 3-edge-connected graphs. *Operations Research Letters*, 33(5):467–474, 2005. doi:10.1016/j.orl.2004.09.005.

24 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

25 Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In Rafail Ostrovsky, editor, *Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science, FOCS '11*, pages 550–559. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.80.

26 Michel X. Goemans. Worst-case comparison of valid inequalities for the TSP. *Mathematical Programming*, 69:335–349, 1995. doi:10.1007/BF01585563.

27 Michelangelo Grigni, Elias Koutsoupias, and Christos H. Papadimitriou. An approximation scheme for planar graph TSP. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science, FOCS '95*, pages 640–645. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492665.

28 Jiong Guo, Sepp Hartung, Rolf Niedermeier, and Ondřej Suchý. The parameterized complexity of local search for TSP, more refined. *Algorithmica*, 67(1):89–110, 2013. doi:10.1007/s00453-012-9685-8.

29 Gregory Gutin and Abraham P. Punnen, editors. *The traveling salesman problem and its variations*. Combinatorial Optimization. Springer, Boston, MA, 2007. doi:10.1007/b101971.

30 Arash Haddadan and Alantha Newman. Towards improving Christofides algorithm for half-integer TSP. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *Proceedings of the 27th Annual European Symposium on Algorithms, ESA '19*, volume 144 of *Leibniz International Proceedings in Informatics*, pages 56:1–56:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.56.

31 Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1:6–25, 1971. doi:10.1007/BF01584070.

32 Danny Hermelin, Stefan Kratsch, Karolina Soltys, Magnus Wahlström, and Xi Wu. A completeness theory for polynomial (Turing) kernelization. *Algorithmica*, 71(3):702–730, 2015. doi:10.1007/s00453-014-9910-8.

33 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, March 2001. doi:10.1006/jcss.2000.1727.

34 Bart M. P. Jansen and Stefan Kratsch. Data reduction for graph coloring problems. *Information and Computation*, 231:70–88, 2013. doi:10.1016/j.ic.2013.08.005.

35 David S. Johnson and Lyle A. McGeoch. Experimental analysis of heuristics for the STSP. In Gregory Gutin and Abraham P. Punnen, editors, *The Traveling Salesman Problem and Its Variations*, pages 369–443. Springer, Boston, MA, 2007. doi:10.1007/0-306-48213-4_9.

36 David S. Johnson and Lyle A. McGeoch. The traveling salesman problem: a case study. In Emile Aarts and Jan Karel Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. Princeton University Press, 2018. doi:10.1515/9780691187563-011.

37 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. An improved approximation algorithm for TSP in the half integral case. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC '20*, pages 28–39. ACM, 2020. doi:10.1145/3357713.3384273.

38 Philip N. Klein. A linear-time approximation scheme for planar weighted TSP. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS '05*, pages 647–657. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.7.

39 Philip N. Klein and Dániel Marx. A subexponential parameterized algorithm for subset TSP on planar graphs. In Chandra Chekuri, editor, *Proceedings of the 2014 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '14*, pages 1812–1830. SIAM, 2014. doi:10.1137/1.9781611973402.131.

40 Giuseppe Lancia and Marcello Dalpasso. Algorithmic strategies for a fast exploration of the TSP 4-opt neighborhood. In Massimo Paolucci, Anna Sciomachen, and Pierpaolo Uberti, editors, *Advances in Optimization and Decision Science for Society, Services and Enterprises*, volume 3 of *AIRO Springer Series*, pages 457–470. Springer, Cham, 2019. doi:10.1007/978-3-030-34960-8_40.

41 Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, Roohani Sharma, and Meirav Zehavi. Brief announcement: Treewidth modulator: Emergency exit for DFVS. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming, ICALP '18*, volume 107 of *Leibniz International Proceedings in Informatics*, pages 110:1–110:4. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.110.

42 Diptapriyo Majumdar, Venkatesh Raman, and Saket Saurabh. Polynomial kernels for vertex cover parameterized by small degree modulators. *Theory of Computing Systems*, 62(8):1910–1951, 2018. doi:10.1007/s00224-018-9858-1.

43 Isja Mannens, Jesper Nederlof, Céline M. F. Swennenhuis, and Krisztina Szilágyi. On the parameterized complexity of the connected flow and many visits TSP problem. In Łukasz Kowalik, Michał Pilipczuk, and Pawel Rzazewski, editors, *Proceedings of the 47th International Workshop on Graph-Theoretic Concepts in Computer Science, WG '21*, volume 12911 of *Lecture Notes in Computer Science*, pages 52–79. Springer, 2021. doi:10.1007/978-3-030-86838-3_5.

44 Isja Mannens, Jesper Nederlof, Céline M. F. Swennenhuis, and Krisztina Szilágyi. On the parameterized complexity of the connected flow and many visits TSP problem. *CoRR*, abs/2106.11689, 2021. arXiv:2106.11689.

45 Dániel Marx. Searching the k-change neighborhood for TSP is W[1]-hard. *Operations Research Letters*, 36(1):31–36, 2008. doi:10.1016/j.orl.2007.02.008.

46 Dániel Marx, Marcin Pilipczuk, and Michał Pilipczuk. On subexponential parameterized algorithms for Steiner tree and directed subset TSP on planar graphs. In Mikkel Thorup, editor, *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science, FOCS '18*, pages 474–484. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00052.

47 Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, March 1999. doi:10.1137/S0097539796309764.

48 Tobias Mömke and Ola Svensson. Approximating graphic TSP by matchings. In Rafail Ostrovsky, editor, *Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science, FOCS '11*, pages 560–569. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.56.

49 Marcin Mucha. 13/9-approximation for graphic TSP. In Christoph Dürr and Thomas Wilke, editors, *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science, STACS '12*, volume 14 of *Leibniz International Proceedings in Informatics*, pages 30–41. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.STACS.2012.30.

50 Christos H. Papadimitriou and Mihalis Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, February 1993. URL: https://www.jstor.org/stable/3690150.

51 Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, July 1976. doi:10.1145/321958.321975.

52 Šimon Schierreich and Ondřej Suchý. Waypoint routing on bounded treewidth graphs. *Information Processing Letters*, 173:106165, 2022. doi:10.1016/j.ipl.2021.106165.

53 András Sebö and Jens Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, 2014. doi:10.1007/s00493-014-2960-3.

54 David B. Shmoys and David P. Williamson. Analyzing the Held-Karp TSP bound: A monotonicity property with application. *Information Processing Letters*, 35(6):281–285, 1990. doi:10.1016/0020-0190(90)90028-V.