

Parameterized Complexity of Connected Even/Odd Subgraph Problems*

Fedor V. Fomin¹ and Petr A. Golovach²

1 Department of Informatics, University of Bergen
PB 7803, 5020 Bergen, Norway
fedor.fomin@ii.uib.no

2 School of Engineering and Computing Sciences, Durham University
Science Laboratories, South Road, Durham DH1 3LE, UK
petr.golovach@durham.ac.uk

Abstract

Cai and Yang initiated the systematic parameterized complexity study of the following set of problems around Eulerian graphs. For a given graph G and integer k , the task is to decide if G contains a (connected) subgraph with k vertices (edges) with all vertices of even (odd) degrees. They succeed to establish the parameterized complexity of all cases except two, when we ask about

- a connected k -edge subgraph with all vertices of odd degrees, the problem known as k -EDGE CONNECTED ODD SUBGRAPH; and
- a connected k -vertex induced subgraph with all vertices of even degrees, the problem known as k -VERTEX EULERIAN SUBGRAPH.

We resolve both open problems and thus complete the characterization of even/odd subgraph problems from parameterized complexity perspective. We show that k -EDGE CONNECTED ODD SUBGRAPH is FPT and that k -VERTEX EULERIAN SUBGRAPH is $W[1]$ -hard.

Our FPT algorithm is based on a novel combinatorial result on the treewidth of minimal connected odd graphs with even amount of edges.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases Parameterized complexity, Euler graph, even graph, odd graph, treewidth

Digital Object Identifier 10.4230/LIPIcs.STACS.2012.432

1 Introduction

An *even* graph (respectively, *odd* graph) is a graph where each vertex has an even (odd) degree. Recall that an Eulerian graph is a connected even graph. Let Π be one of the following four graph classes: Eulerian graphs, even graphs, odd graphs, and connected odd graphs. In [4], Cai and Yang initiated the study of parameterized complexity of subgraph problems motivated by Eulerian graphs. For each Π , they defined the following parameterized subgraph and induced subgraph problems:

* This work is supported by EPSRC (EP/G043434/1), Royal Society (JP100692), and the European Research Council (ERC) via grant Rigorous Theory of Preprocessing, reference 267959.

k -EDGE Π SUBGRAPH (resp. k -VERTEX Π SUBGRAPH)	
<i>Instance:</i>	A graph G and non-negative integer k .
<i>Parameter:</i>	k .
<i>Question:</i>	Does G contain a subgraph with k edges from Π (resp. an induced subgraph on k vertices from Π)?

Cai and Yang established the parameterized complexity of all variants of the problem except k -EDGE CONNECTED ODD SUBGRAPH and k -VERTEX EULERIAN SUBGRAPH, see Table 1. It was conjectured that k -EDGE CONNECTED ODD SUBGRAPH is FPT and k -VERTEX EULERIAN SUBGRAPH is W[1]-hard. We resolve these open problems and confirm both conjectures.

	EULERIAN	EVEN	ODD	CONNECTED ODD
k -EDGE	FPT [4]	FPT [4]	FPT [4]	FPT Thm. 3
k -VERTEX	W[1]-hard Thm. 4	FPT [4]	FPT [4]	FPT [4]

■ **Table 1** Parameterized complexity of k -EDGE Π SUBGRAPH and k -VERTEX Π SUBGRAPH.

The remaining part of the paper is organized as follows. In Section 2, we provide definitions and give preliminary results. In Section 3, we show that k -EDGE CONNECTED ODD SUBGRAPH is FPT. Our algorithmic result is based on an upper bound for the treewidth of a minimal connected odd graphs with an even number of edges. We show that the treewidth of such graphs is always at most 3. The proof of this combinatorial result, which we find interesting in its own, is non-trivial and is given in Section 4. The bound on the treewidth is tight—complete graph on four vertices K_4 is a minimal connected odd graph with an even number of edges and its treewidth is 3. In Section 5, we prove that k -VERTEX EULERIAN SUBGRAPH is W[1]-hard and observe that the problem remains W[1]-hard if we ask about (not necessary induced) Eulerian subgraph on k vertices. We conclude the paper in Section 6 with some open problems.

2 Definitions and Preliminary Results

Graphs. We consider finite undirected graphs without loops or multiple edges. The vertex set of a graph G is denoted by $V(G)$ and its edge set by $E(G)$. A set $S \subseteq V(G)$ of pairwise adjacent vertices is called a *clique*. For a vertex v , we denote by $N_G(v)$ its (*open*) *neighborhood*, that is, the set of vertices which are adjacent to v . Distance between two vertices $u, v \in V(G)$ (i.e., the length of the shortest (u, v) -path in the graph) is denoted by $\text{dist}_G(u, v)$. For a vertex v and a positive integer k , $N_G^{(k)}[v] = \{u \in V(G) \mid \text{dist}_G(u, v) \leq k\}$. The *degree* of a vertex v is denoted by $d_G(v)$, and $\Delta(G)$ is the maximum degree of G . For a set of vertices $S \subseteq V(G)$, $G[S]$ denotes the subgraph of G induced by S , and by $G - S$ we denote the graph obtained from G by the removal of all the vertices of S , i.e. the subgraph of G induced by $V(G) \setminus S$.

Parameterized Complexity. Parameterized complexity is a two dimensional framework for studying the computational complexity of a problem. One dimension is the input size n and another one is a parameter k . It is said that a problem is *fixed parameter tractable* (or FPT), if it can be solved in time $f(k) \cdot n^{O(1)}$ for some function f . One of basic assumptions of the Parameterized Complexity theory is the conjecture that the complexity class $W[1] \neq \text{FPT}$,

and it is unlikely that a W[1]-hard problem could be solved in FPT-time. We refer to the books of Downey and Fellows [6], Flum and Grohe [7], and Niedermeier [8] for detailed introductions to parameterized complexity.

Treewidth. A *tree decomposition* of a graph G is a pair (X, T) where T is a tree and $X = \{X_i \mid i \in V(T)\}$ is a collection of subsets (called *bags*) of $V(G)$ such that:

1. $\bigcup_{i \in V(T)} X_i = V(G)$,
2. for each edge $\{x, y\} \in E(G)$, $x, y \in X_i$ for some $i \in V(T)$, and
3. for each $x \in V(G)$ the set $\{i \mid x \in X_i\}$ induces a connected subtree of T .

The *width* of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is $\max_{i \in V(T)} \{|X_i| - 1\}$. The *treewidth* of a graph G (denoted as $\text{tw}(G)$) is the minimum width over all tree decompositions of G .

Minimal odd graphs with even number of edges. We say that a graph G is *odd* if all vertices of G are of odd degree. Let r be a vertex of G . We assume that G is *rooted* in r . Let G be a connected odd graph with an even number of edges. We say that G is a *minimal* if G has no proper connected odd subgraphs with an even number of edges containing r .

The importance of minimal odd subgraphs with even numbers of edges is crucial for our algorithm because of the following combinatorial result.

► **Theorem 1.** *Let G be a minimal connected odd graph with an even number of edges with a root r . Then $\text{tw}(G) \leq 3$.*

For non-rooted graphs, we also have the following corollary.

► **Corollary 2.** *For any minimal connected odd graph G with an even number of edges, $\text{tw}(G) \leq 3$.*

Let us remark that the bound in Theorem 1 is tight—complete graph K_4 with a root vertex r is a minimal odd graph with even number of edges and of treewidth 3. The proof of Theorem 1 is given in Section 4. This proof is non-trivial and technical, and we find the combinatorial result of Theorem 1 to be interesting in its own. From algorithmic perspective, Theorem 1 is a cornerstone of our algorithm; combined with color coding technique of Alon, Yuster and Zwick in [1] it implies that k -EDGE CONNECTED ODD SUBGRAPH is FPT. We give this algorithm in the next section.

3 Algorithm for k -Edge Connected Odd Subgraph

To give an algorithm for k -EDGE CONNECTED ODD SUBGRAPH, in addition to Theorem 1, we also need the following result of Alon, Yuster and Zwick from [1] obtained by a powerful color-coding technique.

► **Proposition 1** ([1]). *Let H be a graph on k vertices with treewidth t . Let G be a n -vertex graph. A subgraph of G isomorphic to H , if one exists, can be found in $O(2^{O(k)} \cdot n^{t+1})$ expected time and in $O(2^{O(k)} \cdot n^{t+1} \cdot \log n)$ worst-case time.*

We are ready to prove the main algorithmic result of this paper.

► **Theorem 3.** *k -EDGE CONNECTED ODD SUBGRAPH can be solved in time $O(2^{O(k \log k)} \cdot n^4 \cdot \log n)$ for n -vertex graphs.*

Proof. Let (G, k) be an instance of the problem. We apply the following algorithm.

Step 1. If k is odd and $\Delta(G) \geq k$, then return YES. Else if k is odd but $\Delta(G) < k$, then go to Step 3.

Step 2. If k is even and $\Delta(G) \geq k$, then we enumerate all odd connected graphs H with k edges of treewidth at most 3. For each odd graph H of treewidth at most 3 and with k edges, we use Proposition 1 to check whether G has a subgraph isomorphic to H . The algorithm returns YES if such a graph H exists. Otherwise, we construct a new graph G by removing from the old graph G all vertices of degree at least k .

Step 3. For each vertex v , check whether there is a connected odd subgraph H with k edges that contains v . To do it, we enumerate all connected subgraphs with $p = 0, \dots, k$ edges that include v using the following observation. For every connected subgraph H of G with $p \geq 1$ edges such that $v \in V(H)$, there is a connected subgraph H' with $p - 1$ edges such that $v \in V(H')$ and H' is a subgraph of H . Hence, given all connected subgraphs with $p - 1$ edges, we can enumerate all subgraphs with p edges by a brute-force algorithm. The algorithm returns YES if a connected odd subgraph H with k edges exists for some vertex v , and it returns NO otherwise.¹

In what follows we discuss the correctness of the algorithm and evaluate its running time.

If k is odd and $\Delta(G) \geq k$, then the star $K_{1,k}$ is a subgraph of G . Hence, G has a connected odd subgraph with k edges.

Let k be even and let $r \in V(G)$ be a vertex with $d_G(r) \geq k$. If G has a connected odd subgraph with k edges containing r , then G has a minimal connected odd subgraph H with even number of edges rooted in r . Let $\ell = |E(H)|$. Graph H contains at most ℓ vertices in $N_G(r)$. It follows that there are $k - \ell$ vertices $v_1, \dots, v_{k-\ell} \in N_G(r) \setminus V(H)$. Denote by H' the subgraph of G with the vertex set $V(H) \cup \{v_1, \dots, v_{k-\ell}\}$ and the edge set $E(H) \cup \{rv_1, \dots, rv_{k-\ell}\}$. Since k and ℓ are even, we have that H' is an odd graph. By Theorem 1, $\text{tw}(H) \leq 3$. Graph H' is obtained from H by adding some vertices of degree 1, and, therefore, $\text{tw}(H') \leq 3$. This means that when G has a connected odd subgraph H with k edges containing r , then there is a connected odd subgraph H' with k edges containing r and of treewidth at most three. But then in Step 2, we find such a graph H' with k edges.

If no connected odd subgraph with k edges was found in Step 2, then if such a graph exist, it contains no vertex of degree (in G) at least k . Therefore all such vertices can be removed from G without changing the solution. Finally, in Step 3, trying all possible connected subgraphs with k edges in the obtained graph of maximum degree at most $k - 1$, we can deduce if G contains an odd subgraph with k edges.

Concerning the running time of the algorithm. There are at most $\binom{k(k-1)/2}{k}$ non-isomorphic graphs with k edges, and we can find all connected odd graphs with k edges in time $2^{O(k \log k)}$ and to check in time $O(k)$ if the treewidth of each of the graphs is at most three by making use of Bodlaender's algorithm [3]. The running time of this part can be reduced to $2^{O(k)}$, see e.g. [2]. Then for each graph H of this type, to check whether H is a subgraph of G , takes time $O(2^{O(k)} \cdot n^4 \cdot \log n)$ by Proposition 1.

When we arrive at Step 3, we have that $\Delta(G) \leq k - 1$. We show by induction that for any $p \geq 1$, there are at most $p!k^p$ connected subgraphs with p edges that contain a given vertex v . Clearly, the claim holds for $p = 1$. Let $p > 1$. Any connected subgraph of G with $p - 1$ edges has at most p vertices. Since there are at most pk possibilities to add an edge to this subgraph to obtain a connected subgraph with p edges, the claim follows. Therefore, for each vertex v , we can enumerate all connected subgraphs H with k edges that include v in

¹ The idea of Step 3 is due to anonymous STACS referee. This allows us to improve the running time $O(2^{O(k^2 \log k)} \cdot n^4 \cdot \log n)$ of the algorithm from the original version.

time $O(k!k^k)$. Hence, Step 3 can be done in time $O(2^{O(k \log k)} \cdot n)$. We conclude that the total running time of the algorithm is $O(2^{O(k \log k)} \cdot n^4 \cdot \log n)$. ◀

4 Minimal connected odd graphs with even number of edges

In this section we give a high level description of the proof of Theorem 1, the main combinatorial result of this paper. The proof is inductive, and for the inductive step we identify specific structures in a minimal connected odd graph with an even number of edges.

To proceed with the inductive step, we need a stronger version of Theorem 1. Let G be a graph and let $x \in V(G)$. We say that a graph G' is obtained from G by *splitting x into x_1, x_2* , if G' is constructed as follows: for a partition X_1, X_2 of $N_G(x)$, we replace x by two vertices x_1, x_2 , and join x_1, x_2 with the vertices of X_1, X_2 respectively. The following claim implies Theorem 1.

► **Claim 1.** Let G be a minimal connected odd graph with an even number of edges with a root r . Then $\mathbf{tw}(G) \leq 3$.

Moreover, if $d_G(r) = 1$ and z is the unique neighbor of r , then at least one of the following holds:

- i) there is a tree decomposition (X, T) of G of width at most three such that for any bag $X_i \in X$ with $z \in X_i$, $|X_i| \leq 3$; or
- ii) for any graph G' obtained from $G - r$ by splitting z into z_1, z_2 , $\mathbf{tw}(G') \leq 3$ and there is a tree decomposition (X, T) of G' of width at most three such that there is a bag $X_i \in X$ containing both z_1 and z_2 .

To describe the structures in the graph, we need a notion of a subgraph with terminals. Roughly speaking, a subgraph with terminals is connected to the remaining part of the graph only via terminals. More formally, let H be a subgraph of graph G , and let $s_1, \dots, s_r \in V(H)$. We say that H is a *subgraph of G with terminals s_1, \dots, s_r* if there is a subgraph F of G such that

- $G = F \cup H$;
- $V(F) \cap V(H) = \{s_1, \dots, s_r\}$; and
- $E(F) \cap E(H) = \emptyset$.

Thus every edge of G having at least one endpoint in a non-terminal vertex of H , should be an edge of H . In particular, terminal vertices of H separate non-terminal vertices of H from other vertices of G . We also say that a subgraph H with a given set of terminals is *separating* if the graph obtained from G by the removal of all non-terminal vertices of H and all the edges of H (denoted $G - H$) is not connected.

The specific structures we are looking for in the inductive step are the subgraphs isomorphic to graphs with terminals from the set $\mathcal{H} = \{H_1, H_2, H_3, H_4, H_5, H_6\}$ shown in Fig. 1. We often say that $H_i \in \mathcal{H}$ is contained in graph G (or G has H_i) if G has a subgraph isomorphic to H_i with the terminals shown in Fig. 1. Notice that H_6 is a subgraph of H_4 and H_5 , and we are looking for H_6 only if we cannot find H_4 or H_5 .

The proof of Claim 1 is by induction on the number of edges. The basis case is a graph with 6 edges. Every connected odd graph with an even number of edges has at least 6 edges, and there are only two graphs with 6 edges that have these properties, these graphs are shown in Fig. 2. Trivially, Claim 1 holds for these graphs for any choice of the root. Then we assume that a minimal connected odd graph G with an even number of edges has at least 8 edges.

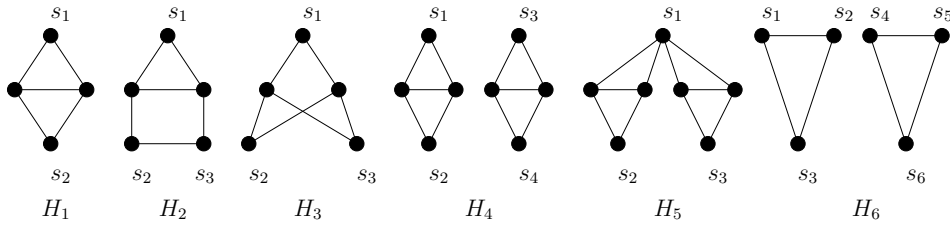


Figure 1 The set \mathcal{H} .

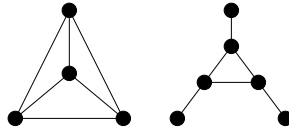


Figure 2 The base of the induction: Minimal graphs with six edges.

If G contains a subgraph R with terminals s_1, s_2 shown in Fig. 3 such that $r \notin V(R) \setminus \{s_1, s_2\}$ and $s_1s_2 \notin E(G)$, then we replace R by edge s_1s_2 . It is possible to show that the resulting graph G' is a minimal connected odd graph with an even number of edges. Since G' has less edges than G , we can use the inductive assumption. Furthermore we assume that G has no R .

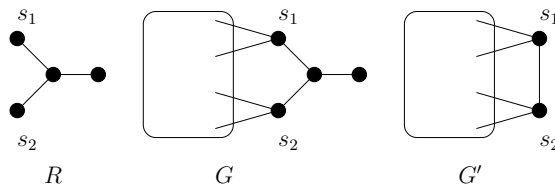


Figure 3 Replacement of R .

Next step is to prove that if G has no subgraph from \mathcal{H} , then G is one of the graphs G_1, G_2, G_3 shown in Fig. 4. For each of these graphs the theorem trivially holds. Actually, we will need a stronger result, saying that if G has no subgraph from H_2, \dots, H_6 and every subgraph of G isomorphic to H_1 is of specific form, namely, this subgraph is not separating and r is not a non-terminal vertex of H_1 , then even in this case, G is one of the graphs G_1, G_2, G_3 shown in Fig. 4. The proof of this claim is not straightforward. With this claim we can proceed further with an assumption that G contains at least one graph from \mathcal{H} .

For the case when r is a non-terminal vertex of a subgraph $H \in \mathcal{H}$, we prove that $H = H_1$. We remove non-terminal vertices of H , identify terminals s_1, s_2 , and add a new root vertex r' adjacent to the vertex obtained from s_1, s_2 . Then we prove that this graph is a minimal connected odd graph with an even number of edges, and then we can apply the induction assumption on this graph, and derive our claim for G . The difficulty here is to ensure that the treewidth of the graph G does not increase when we make the inductive step. This requires the assumptions i) and ii) in Claim 1 on the structure of tree decompositions. From this point, it can be assumed that r is not a non-terminal vertex of a subgraph from \mathcal{H} with the corresponding set of terminals.

All graphs H_2, \dots, H_6 have even number of edges and every terminal vertex of such a graph is of even degree. This means, that G cannot contain a non-separating graph H from $\{H_2, \dots, H_6\}$, because removing edges and non-terminal vertices of H , would result in a

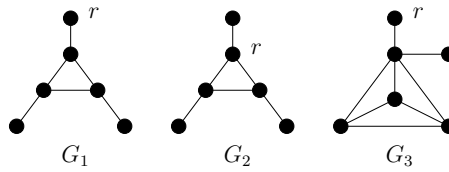


Figure 4 Graphs G_1, G_2, G_3 .

connected odd subgraph of G with even number of edges, which is a contradiction to the minimality of G . Hence, if G contains subgraphs from \mathcal{H} but they are non-separating, G can contain only H_1 . Then as we already have shown, G is one of the graphs G_1, G_2, G_3 shown in Fig. 4.

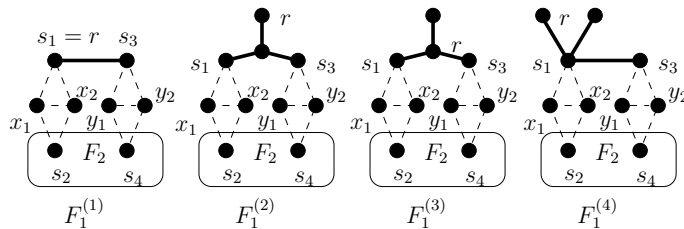


Figure 5 The case $H = H_4$, the trees $F_1^{(1)}, \dots, F_1^{(4)}$ are formed by “bold” edges.

Thus we can assume that G contains a separating subgraph H from \mathcal{H} . Among all such separating subgraphs, we select H such that the number of edges of the component F_1 of the graph $G' = G - H$ containing r is minimum. We prove that G' has exactly two components F_1, F_2 , where F_1 is a tree. We consequently consider the cases $H = H_1, \dots, H_6$ and argue as follows. If $H = H_1$, then $F_1 = K_2$ and we apply induction for F_2 rooted in one of the terminals of H . If $H = H_2$, then we prove that $F_1 = K_2$. If $F_2 = K_2$, then the proof follows directly. Otherwise, we identify terminals s_1, s_3 , and add a new root r' adjacent to the vertex obtained from s_1, s_3 . It is possible to show that the constructed graph is a minimal connected odd graph with an even number of edges, and we can use the induction assumption for this graph. The arguments for the case $H = H_3$ are similar. If $H = H_4$, then we prove that F_1 is one of the trees $F_1^{(1)}, \dots, F_1^{(4)}$ shown in Fig. 5. For F_2 , we prove that $\text{tw}(F_2) \leq 2$, and use this fact to construct a tree decomposition of G of width three. The case $H = H_5$ is similar. Finally, for $H = H_6$, we prove that it can be assumed that $s_1, s_2, s_4, s_5 \in V(F_1)$, $s_3, s_6 \in V(F_2)$, and F_1 is the tree shown in Fig. 6. Then we apply for F_2 the same arguments as in the case $H = H_4$. In each of the cases, we succeed to reduce G to a smaller minimal connected odd graph G' with even number of edges and show that $\text{tw}(G) \leq \text{tw}(G')$, which completes the induction step.

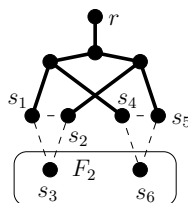


Figure 6 The case $H = H_6$, the tree F_1 is induced by “bold” edges.

5 Complexity of k -Vertex Eulerian Subgraph

In this section we prove that k -VERTEX EULERIAN SUBGRAPH is $W[1]$ -hard.

► **Theorem 4.** *The k -VERTEX EULERIAN SUBGRAPH is $W[1]$ -hard.*

Proof. We reduce from the well-known $W[1]$ -complete k -CLIQUE problem (see e.g. [6]):

k -CLIQUE

Instance: A graph G and non-negative integer k .

Parameter: k .

Question: Does G contain a clique with k vertices?

Notice that the problem remains $W[1]$ -complete when the parameter k is restricted to be odd. It follows immediately from the observation that the existence of a clique with k vertices in a graph G is equivalent to the existence of a clique with $k + 1$ vertices in the graph obtained from G by the addition of a universal vertex adjacent to all the vertices of G . From now it is assumed that $k > 1$ is an odd integer.

Let G be a graph. We construct the graph G' by subdividing edges of G by k^2 vertices, i.e. each edge xy is replaced by an (x, y) -path of length $k^2 + 1$. We say that $u \in V(G')$ is a *branch vertex* if $u \in V(G)$, and u is a *subdivision vertex* otherwise. We also say that u is a *subdivision vertex for an edge $xy \in E(G)$* if u is a subdivision vertex of the path obtained from xy . We claim that G has a clique of size k if and only if G' has an induced Eulerian subgraph on $k' = \frac{1}{2}(k-1)k^3 + k$ vertices.

Suppose that G has a clique K with k vertices. Let H be the subgraph of G induced by K and the subdivision vertices for all edges xy with $x, y \in K$. It is easy to see that H is a connected Eulerian graph on $k' = \frac{1}{2}(k-1)k^3 + k$ vertices.

Let now H be an induced Eulerian subgraph of G' on $k' = \frac{1}{2}(k-1)k^3 + k$ vertices. Denote by U the set of branch vertices of H , and let $p = |U|$. Let $A = \{xy \in E(G) \mid x, y \in U, \text{ and } H \text{ has a subdivision vertex for } xy\}$ and let $F = (U, A)$. Let also $q = |A|$. Since H is connected, the graph F is connected as well. Observe that if $u \in V(H)$ is a subdivision vertex for an edge $xy \in E(G)$, then all subdivision vertices for xy are vertices of H and $x, y \in V(H)$. It follows that H has $p + q \cdot k^2 = k'$ vertices, and we have $p - k = (\frac{1}{2}(k-1)k - q)k^2$. Since k^2 is a divisor of $p - k$, $p \geq k$. Suppose that $p > k$. Then since k^2 is a divisor of $p - k$, $p \geq k^2 + k$. Any connected graph with p vertices has at least $p - 1$ edges, and it means that $q \geq k^2 + k - 1 > \frac{1}{2}(k-1)k$. We get that $0 < p - k = (\frac{1}{2}(k-1)k - q)k^2 < 0$; a contradiction. We conclude that $p = k$. Then $q = \frac{1}{2}(k-1)k$ and U is a clique with k vertices. ◀

Recall that k -VERTEX EULERIAN SUBGRAPH asks about an *induced* Eulerian subgraph on k vertices. For the graph G' in the proof of Theorem 4, any Eulerian subgraph is induced. It gives us the following corollary.

► **Corollary 5.** *The following problem:*

Instance: A graph G and non-negative integer k .

Parameter: k .

Question: Does G contain an Eulerian subgraph with k vertices?

is $W[1]$ -hard.

6 Conclusion

We proved that k -EDGE CONNECTED ODD SUBGRAPH is FPT and k -VERTEX EULERIAN SUBGRAPH is W[1]-hard. This completes the characterization of even/odd subgraph problems with *exactly* k edges or vertices from parameterized complexity perspective. While it is trivial to decide whether a graph G has a (connected) even or odd subgraph with *at most* k edges or vertices, the question about a subgraph with *at least* k edges or vertices seems to be much more complicated. For AT LEAST k -EDGE ODD SUBGRAPH and AT LEAST k -VERTEX ODD SUBGRAPH, following the lines of the proofs from [4] for k -EDGE ODD SUBGRAPH and k -VERTEX ODD SUBGRAPH, it is possible to show that these problems are in FPT. For other cases, the approaches used in [4] and in our paper, do not seem to work.

Cai and Yang in [4] also considered dual problems where the aim is to find an even or odd subgraph of a graph G with $|V(G)| - k$ vertices or $|E(G)| - k$ edges respectively. Recently, these results were complemented by Cygan et al. [5]. However, the complexity of the dual problem to k -EDGE CONNECTED ODD SUBGRAPH, namely, obtaining connected odd subgraph with $|E(G)| - k$ edges, remains open.

Acknowledgments. The authors are grateful to the anonymous referees for their constructive suggestions and remarks.

References

- 1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.
- 2 Omid Amini, Fedor V. Fomin, and Saket Saurabh. Counting subgraphs via homomorphisms. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP 2009)*, volume 5555 of *Lecture Notes Comp. Sci.*, pages 71–82. Springer, 2009.
- 3 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.
- 4 Leizhen Cai and Boting Yang. Parameterized complexity of even/odd subgraph problems. *J. Discrete Algorithms*, 9(3):231–240, 2011.
- 5 Marek Cygan, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Ildikó Schlotter. Parameterized complexity of eulerian deletion problems. In *Proceedings of the 37th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2011)*, volume 6986 of *Lecture Notes Comp. Sci.*, pages 131–142. Springer, 2011.
- 6 Rodney G. Downey and Michael R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, New York, 1999.
- 7 Jörg Flum and Martin Grohe. *Parameterized complexity theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.
- 8 Rolf Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.