ARTICLE IN PRESS
YJCSS:3165

Journal of Computer and System Sciences ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Finding even subgraphs even faster ☆

Prachi Goyal [a], Pranabendu Misra [b,1], Fahad Panolan [b,*,1], Geevarghese Philip [d], Saket Saurabh [b,c]

[a] Indian Institute of Science, Bangalore, India
[b] The Institute of Mathematical Sciences, HBNI, Chennai, India
[c] Department of Informatics, University of Bergen, Norway
[d] Chennai Mathematical Institute, India and UMI ReLaX

ARTICLE INFO

Article history:
Received 25 July 2017
Received in revised form 12 December 2017
Accepted 12 March 2018
Available online xxxx

Keywords:
Eulerian edge deletion
FPT algorithm
Representative family
Co-graphic matroid
Dynamic programming

ABSTRACT

In the Undirected Eulerian Edge Deletion problem, we are given an undirected graph and an integer k, and the objective is to delete k edges such that the resultant graph is a connected graph in which all the vertices have even degrees. The corresponding problem in digraphs where the resulting graph should be strongly connected and every vertex should have the same in-degree as its out-degree is called Directed Eulerian Edge Deletion. In this paper, using the technique of computing representative families of co-graphic matroids we design algorithms which solve these problems in time $2^{O(k)}n^{O(1)}$, improving the algorithms by Cygan et al. [Algorithmica, 2014] and affirmatively answer the open problem posed by them. The crucial insight we bring to these problems is to view the solution as an independent set of a co-graphic matroid.

1. Introduction

Many well-studied algorithmic problems on graphs can be phrased in the following way: Let $\mathcal{F}$ be a family of graphs or digraphs. Given as input a graph (digraph) G and a positive integer k, can we delete k vertices (or edges or arcs) from G such that the resulting graph (digraph) belongs to the class $\mathcal{F}$? Recent research in parameterized algorithms has focused on problems of this kind where the class $\mathcal{F}$ consists of all graphs/digraphs whose vertices satisfy certain *parity* constraints [4,10,3,9,5]. In this paper we obtain significantly faster parameterized algorithms for two such problems, improving the previous best bounds due to Cygan et al. [4]. We also settle the parameterized complexity of a third problem, disproving a conjecture of Cai and Yang [3] and solving an open problem posed by Fomin and Golovach [10]. We obtain our results using recently-developed techniques for the efficient computation of representative sets of matroids.

Our problems. An undirected graph G is *even* (respectively, *odd*) if every vertex of G has even (resp. odd) degree. A directed graph D is *balanced* if the in-degree of each vertex of D is equal to its out-degree. An undirected graph is *Eulerian* if it is connected and even; and a directed graph is *Eulerian* if it is strongly connected and balanced. Cai and Yang [3] initiated

☆ The conference version of the paper appeared in the proceedings of FSTTCS 2015.

* Corresponding author.
E-mail address: fahad.panolan@ii.uib.no (F. Panolan).
[1] *Present address*: Department of Informatics, University of Bergen, 5020, Bergen, Norway.

https://doi.org/10.1016/j.jcss.2018.03.001
0022-0000/

Please cite this article in press as: P. Goyal et al., Finding even subgraphs even faster, J. Comput. Syst. Sci. (2018), https://doi.org/10.1016/j.jcss.2018.03.001

the systematic study of parameterized Eulerian subgraph problems. In this work we take up the following edge-deletion problems of this kind:

Undirected Eulerian Edge Deletion **Parameter:** k
Input: A connected undirected graph G and an integer k.
Question: Does there exist a set S of at most k edges in G such that $G \setminus S$ is Eulerian?

Undirected Connected Odd Edge Deletion **Parameter:** k
Input: A connected undirected graph G and an integer k.
Question: Does there exist a set S of at most k edges in G such that $G \setminus S$ is odd and connected?

Directed Eulerian Edge Deletion **Parameter:** k
Input: A strongly connected directed graph D and an integer k.
Question: Does there exist a set S of at most k arcs in D such that $D \setminus S$ is Eulerian?

Our algorithms for these problems also find such a set S of edges/arcs when it exists; so we slightly abuse the notation and refer to S as a *solution* to the problem in each case.

Previous work. Cai and Yang [3] listed sixteen odd/even undirected subgraph problems in their pioneering paper, and settled the parameterized complexity of all but four. The first two problems above are among these four; Cai and Yang conjectured that these are both W[1]-hard, and so are unlikely to have fixed-parameter tractable (FPT) algorithms: those with running times of the form $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function f where n is the number of vertices in the input graph. Cygan et al. [4] disproved this conjecture for the first problem: they used a novel and non-trivial application of the color-coding technique to solve both Undirected Eulerian Edge Deletion and Directed Eulerian Edge Deletion in time $2^{\mathcal{O}(k \log k)} n^{\mathcal{O}(1)}$. They also posed as open the question whether there exist $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$-time algorithms for these two problems. It was also posed as an open problem at the School on Parameterized Algorithms and Complexity 2014, Będlewo, Poland [1]. Fomin and Golovach [10] settled the parameterized complexity of the other two problems—not defined here—left open by Cai and Yang, but left the status of Undirected Connected Odd Edge Deletion open.

The related problem of *adding* a small number of edges to obtain an Eulerian graph—called the *Eulerian Extension* problem—can be used to model various scheduling and routing problems such as no-wait flow-shop scheduling and the Rural Postman problem. Different variants of this problem have been shown to be NP-hard [14], and the weighted variant—given a graph G, a weight function $w : V(G) \times V(G) \to \mathbb{N}$, and $w_{max}, k \in \mathbb{N}$, can G be made Eulerian by adding at most k edges of total weight at most w_{max}—has been shown to be FPT in the parameter k [7].

Our results and methods. We devise deterministic algorithms which run in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ for all the three problems defined above. This answers the question of Cygan et al. [4] in the affirmative, solves the problem posed by Fomin and Golovach, and disproves the conjecture of Cai and Yang for Undirected Connected Odd Edge Deletion.

Theorem 1. Undirected Eulerian Edge Deletion, Undirected Connected Odd Edge Deletion, *and* Directed Eulerian Edge Deletion *can all be solved in time* $\mathcal{O}(2^{(2+\omega)k} \cdot n^2 m^3 k^6) + m^{\mathcal{O}(1)}$ *where* $n = |V(G)|$, $m = |E(G)|$ *and* ω *is the exponent of matrix multiplication.*

Our main conceptual contribution is *to view the solution as an independent set of a co-graphic matroid*, which we believe will be useful in other problems where one of the constraints that need to be satisfied is that of connectivity.

We now give a high-level overview of our algorithms. Given a subset of vertices T of a graph G, a T-*join* of G is a set $S \subseteq E(G)$ of edges such that T is exactly the set of odd degree vertices in the subgraph $H = (V(G), S)$. Observe that T-joins exist only for even-sized vertex subsets T. The following problem is long known to be solvable in polynomial time [8].

Min T-Join
Input: An undirected graph G and a set of terminals $T \subseteq V(G)$.
Question: Find a T-join of G of the smallest size.

Consider the two problems we get when we remove the connectivity (resp. strong connectivity) requirement on the graph $G \setminus S$ from Undirected Eulerian Edge Deletion and Directed Eulerian Edge Deletion; we call these problems Undirected Even Edge Deletion and Directed Balanced Edge Deletion, respectively. Cygan et al. show that Undirected Even Edge Deletion can be reduced to Min T-Join, and Directed Balanced Edge Deletion to a minimum cost flow problem with unit costs, both in polynomial time [4]. Thus it is not the local requirement of even degrees which makes these problems hard, but the simultaneous global requirement of (strong) connectivity.