Efficient Algorithms for Eulerian Extension

Frederic Dorn[1], Hannes Moser[2⋆], Rolf Niedermeier[2], Mathias Weller[2⋆⋆]

[1] Department of Informatics, University of Bergen, Norway.
frederic.dorn@ii.uib.no
[2] Institut für Informatik, Friedrich-Schiller-Universität Jena, D-07743 Jena, Germany. [hannes.moser,rolf.niedermeier,mathias.weller]@uni-jena.de

Abstract. Eulerian extension problems aim at making a given (directed) (multi-)graph Eulerian by adding a minimum-cost set of edges (arcs). These problems have natural applications in scheduling and routing and are closely related to the Chinese Postman and Rural Postman problems. Our main result is to show that the NP-hard Weighted Multigraph Eulerian Extension is fixed-parameter tractable with respect to the number k of extension edges (arcs). For an n-vertex multigraph, the corresponding running time amounts to $O(4^k \cdot n^3)$. This implies a fixed-parameter tractability result for the "equivalent" Rural Postman problem. In addition, we present several polynomial-time algorithms for natural Eulerian extension problems.

1 Introduction

Edge modification problems in graphs have many applications and are well-studied in algorithmic graph theory [4, 14]. The corresponding minimization problems ask to modify as few (potential) edges as possible such that an input graph is transformed into a graph with a desired property. Most studies in this context relate to undirected graphs whereas we are aware of only few studies of "arc modification" problems on directed graphs (digraphs). One example in this direction is given by the NP-hard Transitivity Editing problem, asking to make a digraph transitive by adding and deleting as few arcs as possible [18]. In this work, as part of a larger project on Eulerian graph modification problems, we study the problem of making a (directed) (multi-)graph Eulerian by edge (arc) additions.[1]

A (directed) (multi-)graph is called *Eulerian* if it contains an oriented cycle visiting every edge (arc) exactly once. An *Eulerian extension* is a set of edges (arcs) to add to a (directed) (multi-)graph so that it becomes Eulerian.

Eulerian Extension (EE)
Input: A (directed) graph $G = (V, E)$ and $\omega_{\max} \in \mathbb{N}$.
Question: Is there an Eulerian extension $\mathcal{E}$ for G with $|\mathcal{E}| \leq \omega_{\max}$?

⋆ Supported by the DFG, project AREG (NI 369/9).
⋆⋆ Supported by the DFG, project DARE (NI 369/11).

[1] Here, following previous work, we call this "extension" problem. In the graph modification context, this is also known as "completion" or "addition" problem.

Variants of EE include WEIGHTED EULERIAN EXTENSION (WEE), where an additional weight function $\omega : V \times V \to \mathbb{N}$ is given[2] and the sum of the weights of the arcs in the Eulerian extension we are looking for must not exceed $\omega_{\max}$, and the multigraph variants (where parallel arcs are allowed as input and output) MULTIGRAPH EULERIAN EXTENSION (MEE) and WEIGHTED MULTIGRAPH EULERIAN EXTENSION (WMEE), respectively. This work focuses on the latter problem, which has applications in scheduling [11]. Furthermore, the various applications of RURAL POSTMAN [7] carry over to WMEE since both problems are equivalent.

Related Problems and Previous Work. Lesniak and Oellermann [13] presented an overview of undirected Eulerian graphs. The unweighted and undirected extension problems for graphs and multigraphs were already discussed by Boesch et al. [3], who developed a linear time algorithm for the multigraph case and a matching based algorithm for the graph case. Recently, Höhn et al. [11] initiated a study of Eulerian extension problems applied to sequencing problems. To the best of our knowledge, WEE has not been considered in the literature so far.

EE is closely related to the well-known CHINESE POSTMAN problem [6] and the more general RURAL POSTMAN problem [7, 12]. More specifically, RURAL POSTMAN and WMEE are "equivalent" (see Section 2 for details). With this equivalence, the NP-hardness of WMEE directly follows from the known NP-hardness result for RURAL POSTMAN [12]. Moreover, the fact that RURAL POSTMAN is solvable in polynomial time if the the set of required arcs is connected [10] directly implies that WMEE is solvable in polynomial time if the input is (weakly) connected.

Our Results. Our main achievement is to show that WMEE is fixed-parameter tractable with respect to the parameter "number of extension arcs"[3] denoted by k. The running time is $O(4^k \cdot n^3)$, where n denotes the number of vertices in the input multigraph and k denotes the number of additional arcs. Using the abovementioned equivalence, this implies a first fixed-parameter tractability result for RURAL POSTMAN. In contrast to RURAL POSTMAN, whose unweighted variant is NP-hard [12], we can show that EE and MEE are polynomial-time solvable. Altogether, our work complements and extends known results for WMEE with restricted weight function [11] and RURAL POSTMAN, for which mainly approximation, heuristic, and some polynomial-time algorithms for special cases are known [7, 10].

Due to the lack of space, several technical details are deferred to a full version of the paper.

[2] We assume the weight function to also assign weights to so far nonexistent arcs.

[3] Replacing each weight by a shortest-path-weight, much like in the proof of Lemma 2, decreases the number of arcs needed for an optimal Eulerian extension. It seems possible to extend our results to the corresponding stronger parameter "number of extension arcs after shortest-path-preprocessing". Further considerations in the direction are deferred to a full version of this paper.

2 Preliminaries and Basic Observations

The main focus of this work is on directed (multi-)graphs and, therefore, preliminaries for undirected (multi-)graphs are omitted if they follow trivially from the directed case. In the context of directed (multi-)graphs, connectivity always means weak connectivity, that is, connectivity of the underlying undirected graph. Let $G = (V, A)$ be a directed graph or multigraph (that is, a graph with parallel arcs allowed—we also use the letter M to refer to multigraphs). The set of connected components of G that are not isolated vertices is denoted by $\mathcal{C}_G$. In this work we sometimes apply definitions for graphs to connected components or sets of connected components. For example, we use $V(G)$ to refer to the vertices of the graph G and $V(C)$ to refer to the vertices of the connected component C. For a vertex set $V' \subseteq V$, let $G[V'] := (V', A \cap (V' \times V'))$ denote the directed (multi-)graph that is *induced* by V'. For an arc set $\mathcal{E}$ and some arc a, we abbreviate $\mathcal{E} \cup \{a\}$ to $\mathcal{E} + a$. If G is not a multigraph, then the *complement* $\overline{G}$ of G is the digraph on the vertex set V that contains exactly the arcs that are not in A. An *Eulerian cycle* in a directed (multi-)graph G is a (not necessarily simple) directed cycle that visits all arcs of G exactly once. If such a cycle exists, then we call G *Eulerian.* We call a (multi-)set $\mathcal{E} \subseteq V \times V$ an *Eulerian extension* for G if $(V, A \cup \mathcal{E})$ is Eulerian. Furthermore, $\mathcal{E}$ is called *optimal* if there is no Eulerian extension of less total weight for G. A *walk* W in G is a sequence of arcs of A such that each arc starts in the end vertex of the previous arc. Walks may also be considered as multisets of arcs. For a vertex v of a directed (multi-)graph G, the *outdegree* (*indegree*) of v, denoted by $\operatorname{outdeg}(v)$ ($\operatorname{indeg}(v)$), is the number of arcs in A that are outgoing of (incoming to) v. The *balance* of a vertex v is

$$\operatorname{bal}(v) := \operatorname{indeg}(v) - \operatorname{outdeg}(v)\,.$$

Specifically, let $\mathcal{I}_G^+$ ($\mathcal{I}_G^-$) denote the set of vertices v of G for which $\operatorname{bal}(v) > 0$ ($b(v) < 0$), that is, $\operatorname{indeg}(v) > \operatorname{outdeg}(v)$ ($\operatorname{indeg}(v) < \operatorname{outdeg}(v)$). In an undirected graph, we define the balance $\operatorname{bal}(v)$ of a vertex v to be one if the number of its neighbors is odd and zero otherwise. For both directed and undirected (multi-)graphs G, vertices v of G with $\operatorname{bal}(v) = 0$ are called *balanced*, while all other vertices of G are called *imbalanced*, with $\mathcal{I}_G$ denoting the set of imbalanced vertices of G. With the concept of vertex balance, we can state a well-known fact about Eulerian graphs and multigraphs.

Lemma 1 (Folklore). *A (directed) (multi-)graph is Eulerian if and only if all edges (arcs) are in the same connected component and all vertices are balanced.*

Eulerian extension and Related Problems. In the most general problem that we study, we have weights and allow the input and output to be multigraphs.

> Weighted Multigraph Eulerian Extension (WMEE)
> **Input**: A directed multigraph $M = (V, A)$, a weight function $\omega : V \times V \to \mathbb{N}$, and positive integers k and $\omega_{\max}$.
> **Question**: Is there an arc multiset $\mathcal{E}$ with $|\mathcal{E}| \leq k$ and total weight at most $\omega_{\max}$ such that $(V, A \cup \mathcal{E})$ is an Eulerian multigraph?

Since multigraphs allow the presence of parallel arcs, we may also add arcs that are already present in the input. If we restrict the problem to digraphs, that is, we prohibit parallel arcs in both the input and the resulting digraph, then we arrive at the Weighted Eulerian Extension problem (WEE). Both WMEE and WEE are also considered in their unweighted versions, where all arcs have weight one (and, hence, the extension set $\mathcal{E}$ may contain at most $\omega_{\max}$ arcs). Since being Eulerian is defined for both directed and undirected graphs, all presented variants of Eulerian Extension also have an undirected version.

Eulerian extensions are closely related to arc routing. An important role in this relation plays the following problem:

> Rural Postman (RP)
> **Input**: A digraph $G = (V, A)$, a nonempty set $R \subseteq A$ of "required" arcs, a weight function $\omega : A \to \mathbb{N}$, and integers q and $\omega_{\max} \geq 0$.
> **Question**: Is there a closed walk W in G such that W visits all arcs in R and contains at most $q + |R|$ arcs whose total weight is at most $\omega_{\max}$?

If $R = A$, then RP degenerates to the also well-known Chinese Postman problem.

Parameterized Complexity. Our results are in the context of parameterized complexity, which is a two-dimensional framework for studying computational complexity [5, 9, 15]. One dimension is the input size n, and the other one is the *parameter* (usually a positive integer). A problem is called *fixed-parameter tractable* (fpt) with respect to a parameter k if it can be solved in $f(k) \cdot n^{O(1)}$ time, where f is a computable function only depending on k. A parameterized problem P_1 is *parameterized reducible* to a parameterized problem P_2 if P_1 can be reduced to P_2 in "fpt-time" such that the new parameter exclusively depends on the old parameter. If P_1 is parameterized reducible to P_2 and vice versa, then P_1 and P_2 are *parameterized equivalent*. If used as parameterized problems, all variants of EE are parameterized by the number k of allowed arcs in a solution and RP is parameterized by the number q of allowed *additional* arcs, that is, the number of arcs outside of R that are visited by the walk W. Note that for RP, q is a "stronger" parameter than the number of arcs in W, because it is always smaller. Since all solutions guarantee to contain R, choosing q can be considered an above-guarantee parameterization of RP.

Helpful Observations. We present observations that help us prove our results and give insights into the structure of the considered problems. First, observe that, over all vertices of a graph, the balance always adds up nicely, that is, for each "missing" incoming arc, there is also a "missing" outgoing arc.

Observation 1 *Let G be a directed (multi-)graph. Then,* $\sum_{v \in V(G)} \operatorname{bal}(v) = 0$.

In undirected graphs and multigraphs we can observe that the sum over all balances is even. Observation 1 can also be applied to connected components. Next, we note the relation between RP and WMEE.

	unweighted	weighted, connected
undir. graph	$O(\overline{m}\sqrt{n})$ (Theorem 1)	$O(n^3 \log n)$ (Corollary 2)
undir. multigraph	$O(n+m)$ (Proposition 4)	$O(n^3 \log n)$ (Corollary 2)
dir. graph	$O(\overline{m}^2 + n\overline{m}\log n))$ (Prop. 3)	$O(\overline{m}^2 + n\overline{m}\log n))$ (Prop. 2)
dir. multigraph	$O(n+m)$ (Proposition 4)	$O(n^3 \log n)$ (Corollary 1)

Table 1: Polynomial-time solvable Eulerian extension problems. Here, n,m, and $\overline{m}$ are defined as in Section 3. In general, weighted variants of Eulerian extension problems are NP-hard if the input (multi-)graph is not connected [11, 12].

Proposition 1. RP *is parameterized equivalent to* WMEE*.*

This implies that the NP-hardness of RP for disconnected arc sets R carries over to WMEE for disconnected inputs. The basic idea is to let R be the arc set in the WMEE instance and identify an Eulerian extension with the set of additional arcs in a walk that visits all arcs in R.

3 Polynomial-Time Cases of Eulerian Extension

In this section, we present polynomial-time algorithms for various variants of Eulerian extension problems and their weighted versions. All running times are given as functions in n (the number of vertices in the input), m (the number of arcs (edges) in the input), and $\overline{m}$ (the number of arcs (edges) in the complement of the input). We refer to Table 1 for an overview of the results of this section. So far, the following result was known.

Theorem 1 ([3, 13]). Eulerian Extension *on undirected graphs can be solved in $O(\overline{m}\sqrt{n})$ time.*

In the following, we present polynomial-time algorithms for weighted variants of Eulerian extension problems if the input (multi-)graph is connected. Then, we consider the unweighted variant and allow disconnected (multi-)graphs.

Algorithms for Connected Weighted Variants. Keeping in mind that the disconnected versions of WEE and WMEE are NP-hard [11] (see also [12]), we provide polynomial-time algorithms for both problems in case of connected inputs. Most algorithms are based on computing flows or matchings. First, we present an algorithm for digraphs, which is then modified to work for directed multigraphs and undirected graphs as well.

Proposition 2. Weighted Eulerian Extension *on connected digraphs can be solved in $O(\overline{m}^2 + n\overline{m}\log n))$ time.*

Proof. Consider an instance $(G, \omega, \omega_{\max})$ for WEE, where G is a connected digraph, and a function $b : V(G) \to \mathbb{Z}$ measuring the balance of each vertex (see Section 2). Consider the flow network $\overline{G}$ with supply determined by b (negative supply indicates demand), arc capacity one for each arc, and arc-costs determined by ω. It is easy to see that a flow of value $\frac{1}{2}\sum_{v\in V}|\operatorname{bal}(v)|$ in this network corresponds to an Eulerian extension for G and, thus, the minimum cost of such a flow is also the minimum cost of an Eulerian extension for G. Such a flow can be computed in $O(\overline{m}^2 + n\overline{m}\log n))$ time.[4] $\square$

Next, for a directed multigraph M let G_M be the complete digraph (containing all possible arcs) on the vertex set of M. Analogously to the proof of Proposition 2, we can use a min-cost flow algorithm on G_M with arc capacities ∞ and weights according to ω to solve WEE on connected directed multigraphs M. The uncapacitated version of the min-cost flow algorithm (running in $O(n^3\log n)$ time [2]) suffices in this case.

Corollary 1. Weighted Multigraph Eulerian Extension *on connected directed multigraphs can be solved in $O(n^3\log n)$ time.*

To handle undirected multigraphs, we replace the min-cost flow in the auxiliary graph G_M with a min-cost perfect matching in the complete undirected graph on the vertex set $\mathcal{I}_M$ with the weight of each edge $\{u, v\}$ equal to the weight of a minimum weight path between u and v in M. These paths are computed by an all-pairs shortest path algorithm. For each edge in the perfect matching, all edges of the corresponding shortest path are added to the extension set. With some effort we can show that the same algorithm can be used for WEE (assuming connected inputs).

Corollary 2. Weighted (Multigraph) Eulerian Extension *on connected undirected graphs and multigraphs can be solved in $O(n^3\log n)$ time.*

Algorithms for General Unweighted Variants. Since EE is a special case of WEE, we can solve EE for connected digraphs using the algorithm from the proof of Proposition 2 with a unit-weight version of the min-cost flow algorithm running in $O(\overline{m}^2)$ time.[5]

Corollary 3. Eulerian Extension *on connected directed graphs can be solved in $O(\overline{m}^2)$ time.*

This algorithm cannot handle multiple components. A more general algorithm that also allows to solve the problem on disconnected digraphs (at the cost of increased running time) will be presented in the full paper.

Proposition 3. Eulerian Extension *on disconnected digraphs can be solved in $O(\overline{m}^2 + n\overline{m}\log n))$ time.*

This stands in contrast with RP being NP-hard for unweighted digraphs [12], which seems to be due to the possibility to prohibit arcs by choosing the input

[4] See Exercise 10.17 of [2], a solution to which can be found in [1].

[5] Combine the solution found in [1] for Exercise 10.17 in [2] with breadth-first search as shortest path algorithm, which is valid in case of unit weights.

digraph. In fact, the subsequent Proposition 4 implies that unweighted RP is solvable in polynomial time if the input digraph is complete. More precisely, we can solve MEE for directed inputs by a straightforward greedy strategy much like the algorithm known for undirected multigraphs [3].

Proposition 4 (See [3]). Multigraph Eulerian Extension *on directed and undirected multigraphs can be solved in $O(n+m)$ time.*

4 Weighted Eulerian Extension on Directed Multigraphs

We prove that WMEE is fpt with respect to the size k of a solution by describing a dynamic programming algorithm to solve WMEE. More precisely, our algorithm computes the solution with smallest total weight over all solutions of size at most k. First, we modify the input, to obtain an equivalent but simpler instance. This preprocessing is described in the first paragraph. Next, we transform the preprocessed instance into an instance of a modified problem called Black/Gray Weighted Multigraph Eulerian Extension (BGWMEE). This problem has the advantage that a corresponding Eulerian extension has a particularly simple structure to be exploited by a dynamic programming algorithm. In the last paragraph, we present such an algorithm for BGWMEE.

Preprocessing the Input. We present two preprocessing algorithms that compute an equivalent instance in which (a) the balance of each vertex v is in $\{-1, 0, 1\}$, and (b) there are no isolated vertices. To achieve (a), we repeatedly find a vertex v with $|\operatorname{bal}(v)| > 1$ and split v into two vertices: one vertex v' with $|\operatorname{bal}(v')| = 1$ and one vertex v'' with $|\operatorname{bal}(v'')| < |b(v)|$. To achieve (b), we replace the weight of a direct connection between two vertices u and v with the weight of the cheapest path of potential arcs from u to v that visits only isolated vertices.

Lemma 2. *Let $(M, \omega, \omega_{max})$ be an instance of* Weighted Multigraph Eulerian Extension *and let V_I denote the set of isolated vertices in M. Then, in $O(n^3)$ time, we can compute a weight function ω' such that $(M - V_I, \omega', \omega_{max})$ is equivalent to $(M, \omega, \omega_{max})$.*

Lemma 3. *Let $(M, \omega, \omega_{max})$ be an instance of* Weighted Multigraph Eulerian Extension *and let $\mathcal{C}_M$ be the set of connected components of M. One can modify $(M, \omega, \omega_{max})$ in $O(k(n+m)+k^2)$ time to obtain an equivalent instance $(M', \omega', \omega_{max})$ such that $|\operatorname{bal}(u)| \leq 1$ for each vertex u in M'.*

Lemma 2 and Lemma 3 imply a preprocessing algorithm that removes all isolated vertices and assures $|\operatorname{bal}(v)| \leq 1$ for all vertices v in $O(n^3)$ time. In the following, we assume all inputs to be preprocessed in this way.

Transformation To BGWMEE. The following observation helps to picture Eulerian extensions as collections of paths between imbalanced vertices which is fundamental for the algorithm. The observation is based on the fact that for each balanced vertex u, each Eulerian extension contains as many arcs outgoing of u as arcs incoming to u.

Observation 2 *Let M be a directed multigraph and let $\mathcal{E}$ be an Eulerian extension of M. Then, $\mathcal{E}$ can be decomposed into a collection of paths that start at a vertex in $\mathcal{I}_M^+$ and end at a vertex in $\mathcal{I}_M^-$ or start and end at a balanced vertex.*

Observation 2 implies that an Eulerian extension $\mathcal{E}$ can be decomposed into paths. Our idea to attack WMEE is to use dynamic programming to construct such paths arc by arc. There are, however, a few obstacles to this approach. Assuming that each path visits a component of the input multigraph at most once, that is, no path contains two vertices of the same component, proved helpful in overcoming these obstacles. Since this is not always the case, we modify the input multigraph in order to use it in a slightly different (unfortunately more technical) problem, for which this assumption is valid.

> Black/Gray WMEE (BGWMEE)
> **Input:** A directed multigraph $M = (V, A_{\text{black}} \cup A_{\text{gray}})$ where each connected component of (V, A_{black}) has either no imbalanced vertex or exactly two imbalanced vertices (one in $\mathcal{I}_M^-$ and one in $\mathcal{I}_M^+$), a weight function $\omega : V \times V \to \mathbb{N}$, and an integer $\omega_{\max} \geq 0$.
> **Question:** Is there an Eulerian extension $\mathcal{E}'$ of total weight at most $\omega_{\max}$ for M such that in each component C_{black} of (V, A_{black}) there is exactly one start vertex of an arc in $\mathcal{E}'$ and exactly one end vertex of an arc in $\mathcal{E}'$ (that is, $|(V(C_{\text{black}}) \times V) \cap \mathcal{E}'| = 1$ and $|(V \times V(C_{\text{black}})) \cap \mathcal{E}'| = 1$)?

Again, we can decompose a black/gray Eulerian extension into paths analogously to Observation 2. The advantage of BGWMEE is that each black component is visited exactly once by such a path. The gray arcs (arcs in A_{gray}) are used to model the connectivity constraints given by the original WMEE instance. We first describe how WMEE can be solved using an algorithm for BGWMEE and then present such an algorithm in the next paragraph. The main idea is to transform an instance $(M, \omega, \omega_{\max})$ of WMEE into an instance $(M', \omega', \omega'_{\max})$ of BGWMEE by duplicating each component C of M as many times as it is visited by paths of a solution for $(M, \omega, \omega_{\max})$. To model that the copies of C originate from one connected component of M, the copies of C are connected by adding *gray* arcs. In the following, we describe the exact transformation algorithm.

First, find pairs of imbalanced vertices sharing a component. By Observation 1 and Lemma 3, there is a bijection $m : \mathcal{I}_M^- \to \mathcal{I}_M^+$ such that for all $v \in \mathcal{I}_M^-$, the vertices v and $m(v)$ are in the same component. We use an arbitrary bijection that respects this condition. Second, for a fix solution $\mathcal{E}$ of $(M, \omega, \omega_{\max})$ and all arcs $(u, v) \in \mathcal{E}$, make a copy of the component of M that contains u. In the following, we denote the number of copies of C by $\#(C)$. Since $\#(C)$ depends on $\mathcal{E}$, we do not know it in advance. Hence, we will try all possibilities. However, not all functions $\#$ are feasible: The total number of copies cannot exceed $|\mathcal{E}|$ $(= k)$ and since each copy has at most two imbalanced vertices, each component C must have at least $|\mathcal{I}_C|/2$ copies. Thus, we need only consider functions of the form $\# : \mathcal{C}_M \to \mathbb{N}^+$ with $\sum_{C \in \mathcal{C}_M} \#(C) \leq k$ and $\#(C) \geq |\mathcal{I}_C|/2$ for all $C \in \mathcal{C}_M$. It can be shown that there are at most 2^k such functions. Third, for each component C of M, assign a copy C' of C to each pair $(v, m(v))$ of

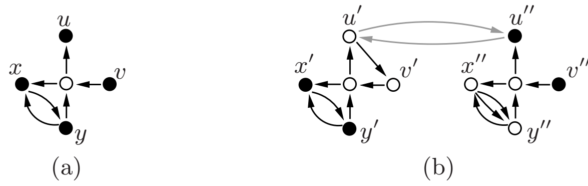

Fig. 1: (a): A component C of a directed multigraph. With $\#(C) = 2$, $m(v) = u$ and $m(y) = x$, the transformation $\mathrm{tr}^m_\#$ transforms (a) into (b). Here, white vertices are balanced, black vertices are imbalanced. Note that, (v', u') is fixed in the first copy and (y'', x'') is fixed in the second copy. Furthermore, u' and u'' are connected by gray arcs, according to the last step of the transformation.

imbalanced vertices of C and "fix" C', that is, add an arc from the copy of $m(u)$ to the copy of u in C' for all $u \in \mathcal{I}^-_C - v$. This assures that each copy of C contains at most one pair of imbalanced vertices, and each pair of imbalanced vertices (paired by m) is represented in one copy. All copies that have not been assigned to an imbalanced pair are balanced completely in the above mentioned way. Fourth, for each component C of M, its copies are pairwisely connected by adding gray arcs. To this end, select a vertex v of each component of M and add all possible arcs between all copies of v. Note that only copies of the same component of M are connected by gray arcs. We denote the transformed instance by $(M', \omega', \omega_{\max}) := \mathrm{tr}^m_\#(M, \omega, \omega_{\max})$. See Figure 1 for an example of the described transformation.

An Algorithm for BGWMEE. Having transformed an instance of WMEE to an instance of BGWMEE using the algorithm presented in the previous paragraph, we can now exploit the simple structure of BGWMEE in a dynamic programming algorithm. The main idea in this algorithm is to construct an Eulerian extension arc by arc while maintaining a set of connected components of the input multigraph that have already been visited.

In the following, we describe a dynamic programming algorithm that solves BGWMEE. Let $(M, \omega, \omega_{\max})$ be an instance of BGWMEE and let $\mathcal{C}^{\mathrm{black}}_M$ be the set of black connected components of M. For each subset $S \subseteq \mathcal{C}^{\mathrm{black}}_M$ and each pair of vertices $u, v \in V(S)$, our algorithm computes an entry $\omega(S, u, v)$ with

$$\omega(S,u,v) = \begin{array}{l}\text{minimum weight } \omega(\mathcal{E}) \text{ of an arc set } \mathcal{E} \text{ such that } \mathcal{E} + (v,u) \\ \text{is a black/gray Eulerian extension for } M[V(S)].\end{array} \qquad (1)$$

If no black/gray Eulerian extension is possible with S, u, and v, then the entry $\omega(S, u, v)$ is assigned "∞". The set S represents a subgraph of M and the two vertices correspond to the endpoints of a (possibly "unfinished") path of an Eulerian extension (see Observation 2). The dynamic programming starts with computing the entries for sets S that contain exactly one component and augments S step by step, finally computing the entries for $S = \mathcal{C}^{\mathrm{black}}_M$, which are used to derive a minimum weight black/gray Eulerian extension for M with respect to ω. In the following, we describe the update process for the entries.

For each $C \in \mathcal{C}_M^{\text{black}}$ not containing imbalanced vertices and each $u, v \in V(C)$, set

$$\omega(\{C\}, u, v) := \begin{cases} 0, & \text{if } u = v, \\ \infty, & \text{otherwise.} \end{cases}$$

This assignment is correct, that is, it satisfies (1) by setting $\mathcal{E} := \emptyset$ (which has obviously minimum weight) because adding an arc to a balanced component can only keep the component balanced if the added arc is a loop. Thus $\mathcal{E} + (v, u)$ is an Eulerian extension for $M[V(C)]$. Moreover, $\mathcal{E} + (v, u)$ is also a black/gray Eulerian extension for $M[V(C)]$ since the only connected component has exactly one incoming arc as well as one outgoing arc in $\mathcal{E}$ (in this case, the incoming and outgoing arc is (v, u)).

For each $C \in \mathcal{C}_M^{\text{black}}$ containing two imbalanced vertices $x \in \mathcal{I}_M^-$ and $y \in \mathcal{I}_M^+$, and each $u, v \in V(C)$, set

$$\omega(\{C\}, u, v) := \begin{cases} 0, & \text{if } u = x \text{ and } v = y, \\ \infty, & \text{otherwise.} \end{cases}$$

This assignment satisfies (1) since, by definition of black/gray Eulerian extension, x and y are the only imbalanced vertices of C and both are balanced adding (y, x) (that is, by using $\mathcal{E} = \emptyset$). For the same reasons as above, $\mathcal{E} + (v, u)$ is also a black/gray Eulerian extension for $M[V(C)]$.

Next, we describe the computation of the entries for larger sets S. When we compute the entry for a set S, we assume that all the entries for sets S' with $|S'| < |S|$ have already been computed. For a given $S \subseteq \mathcal{C}_M^{\text{black}}$ with $|S| > 1$, and vertices $u, v \in V(S)$, the entry $\omega(S, u, v)$ is computed as follows. Let $C \in S$ denote the black component of M that contains v and let $S' := S \setminus \{C\}$. If C is balanced, then distinguish the following three subcases:

1. If $u = v$ and there is a gray arc between C and S', then set

$$\omega(S, u, v) := \min_{u', v' \in V(S'), u' \neq v'} \omega(S', u', v') + \omega(v', u').$$

2. If $u \in V(S')$, then set

$$\omega(S, u, v) := \min_{w \in V(S')} \{\omega(S', u, w) + \omega(w, v)\}.$$

3. Otherwise, set $\omega(S, u, v) := \infty$.

If C contains two imbalanced vertices $x \in \mathcal{I}_M^-$ and $y \in \mathcal{I}_M^+$, then we distinguish the following three subcases:

1. If $u = x$ and $v = y$, and there is a gray arc between C and S', then set

$$\omega(S, u, v) := \min_{u', v' \in V(S'), u' \neq v'} \omega(S', u', v') + \omega(v', u').$$

2. If $u \in V(S')$ and $v = y$, then set
$$\omega(S, u, v) := \min_{w \in V(S')} \{\omega(S', u, w) + \omega(w, x)\}.$$
3. Otherwise, set $\omega(S, u, v) := \infty$.

Finally, the weight ω_{opt} of an optimal black/gray Eulerian extension for (M', ω) is computed as follows:
$$\omega_{\text{opt}} := \min_{u,v \in V(\mathcal{C}_M^{\text{black}}), u \neq v} \omega(\mathcal{C}_M^{\text{black}}, u, v) + \omega(v, u)$$

This follows immediately from (1). A corresponding black/gray Eulerian extension can be computed by storing each solution $\mathcal{E}$ in addition to its weight in each entry in the dynamic programming algorithm. Altogether, this algorithm takes $O(2^k \cdot n^3)$ time for solving a given instance.

Lemma 4. Black/Gray Weighted Multigraph Eulerian Extension *can be solved in $O(2^k \cdot n^3)$ time.*

The Complete Algorithm. The complete algorithm to solve WMEE runs in three steps. First, the input multigraph M is preprocessed in $O(n^3)$ time such that it does not contain isolated vertices or vertices with absolute balance more than one (see Lemma 2 and Lemma 3). Second, a component-respecting bijection $m : \mathcal{I}_M^- \to \mathcal{I}_M^+$ is chosen arbitrarily. Third, for all 2^k possible functions $\# : \mathcal{C}_M \to \mathbb{N}^+$, the instance is transformed and the resulting instance of BGWMEE is solved in $O(2^k \cdot n^3)$ time (see Lemma 4). The correctness of this algorithm follows directly from the correctness of the transformation algorithm and Lemma 4. The overall running time is $O(4^k \cdot n^3)$.

Theorem 2. Weighted Multigraph Eulerian Extension *can be solved in $O(4^k \cdot n^3)$ time.*

Consequently, we can analogously solve Rural Postman parameterized by q (see Proposition 1).

Corollary 4. Rural Postman *can be solved in $O(4^q \cdot n^3)$ time.*

5 Conclusion

We focused on Eulerian extension problems (and due to equivalence, the Rural Postman problem), leaving yet unstudied other Eulerian graph modification problems including the editing version. Eulerian extension problems alone still offer a rich field of challenges for future research in terms of multivariate algorithmics [8, 16]. More specifically, we concentrated on the parameterized complexity with respect to the parameter "number of extension arcs", but there are many natural structural parameters that make sense. For instance, it would be interesting to determine the parameterized complexity with respect to the parameter "number of weakly connected components" in a Weighted Multigraph Eulerian Extension instance. In this context, Orloff [17] observed that "the determining factor in the complexity of the problem seems to be the number (c) of connected components in the required edge set"; Frederickson [10] noted "the existence of an exact recursive algorithm that is exponential only in the number of disconnected components." However, it is doubtful that this meant fixed-parameter tractability with respect to c. In further future work, we also want to study the undirected and non-multigraph versions of WMEE. Here, we conjecture that similar algorithmic approaches may allow for similar results.

References

[1] http://jorlin.scripts.mit.edu/Solution_Manual.html.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[3] F. T. Boesch, C. Suffel, and R. Tindell. The spanning subgraphs of Eulerian graphs. *J. Graph Theory*, 1(1):79–84, 1977.

[4] P. Burzyn, F. Bonomo, and G. Durán. NP-completeness results for edge modification problems. *Discrete Appl. Math.*, 154(13):1824–1844, 2006.

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

[6] H. A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems part I: The chinese postman problem. *Oper. Res.*, 43(2):231–242, 1995.

[7] H. A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems part II: The rural postman problem. *Oper. Res.*, 43(3):399–414, 1995.

[8] M. Fellows. Towards fully multivariate algorithmics: Some new results and directions in parameter ecology. In *Proc. 20th IWOCA*, volume 5874 of *LNCS*, pages 2–10. Springer, 2009.

[9] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[10] G. N. Frederickson. Approximation algorithms for some postman problems. *J. ACM*, 26(3):538–554, 1979.

[11] W. Höhn, T. Jacobs, and N. Megow. On Eulerian extension problems and their application to sequencing problems. Technical Report 008, Combinatorial Optimization and Graph Algorithms, TU Berlin, 2009.

[12] J. K. Lenstra and A. H. G. R. Kan. On general routing problems. *Networks*, 6(3):273–280, 1976.

[13] L. Lesniak and O. R. Oellermann. An Eulerian exposition. *J. Graph Theory*, 10(3):277–297, 1986.

[14] A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Discrete Appl. Math.*, 113:109–128, 2001.

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

[16] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proc. 27th STACS*, volume 5 of *LIPIcs*, pages 17–32. IBFI Dagstuhl, Germany, 2010.

[17] C. S. Orloff. On general routing problems: Comments. *Networks*, 6(3):281–284, 1976.

[18] M. Weller, C. Komusiewicz, R. Niedermeier, and J. Uhlmann. On making directed graphs transitive. In *Proc. 11th WADS*, volume 5664 of *LNCS*, pages 542–553. Springer, 2009.