Efficient Algorithms for Eulerian Extension and Rural Postman[*]

Frederic Dorn[†] Hannes Moser[‡¶] Rolf Niedermeier[§]
Mathias Weller[§‖]

January 23, 2013

The aim of directed Eulerian extension problems is to make a given directed, possibly arc-weighted, (multi-)graph Eulerian by adding a minimum-cost set of arcs. These problems have natural applications in scheduling and arc 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 problem is fixed-parameter tractable with respect to the number k of extension arcs. For a directed, n-vertex multigraph, the corresponding running time amounts to $O(4^k \cdot n^3)$. This also implies a fixed-parameter tractability result for the “equivalent” RURAL POSTMAN problem parameterized above guarantee. In addition, we present several polynomial-time algorithms for natural Eulerian extension problems, including undirected variants which can be defined analogously to the directed ones.

1 Introduction

Edge modification problems in graphs have many applications and are well-studied in algorithmic graph theory [6, 34]. 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). In this work, as part

[*] A preliminary version of this work appeared under the title “Efficient Algorithms for Eulerian Extension” in *Proceedings of the 36th International Workshop on Graph-Theoretic Concepts in Computer Science (WG’10), Zarós (Crete), Greece, June 2010, volume 6410 in Lecture Notes in Computer Science, pages 232–243, Springer. Significant parts of this work were done while the authors were staying at FSU Jena.*

[†] SINTEF Energy Research, Trondheim, Norway E-mail: *frederic.dorn@sintef.no*

[‡] Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany.

[§] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany. E-mail: {*rolf.niedermeier, mathias.weller*}*@tu-berlin.de*

[¶] Supported by the DFG, project AREG (NI 369/9).

[‖] Corresponding author. Supported by the DFG, project DARE (NI 369/11).

of a larger project on Eulerian graph modification problems, we study the problem of making a digraph Eulerian by arc additions.[1] We focus our introduction on directed graphs and point out differences to undirected graphs whenever necessary.

A digraph is called *Eulerian* if it contains a tour visiting every arc exactly once. An *Eulerian extension* is a set of arcs to add to a digraph so that it becomes Eulerian. The corresponding decision problem reads as follows.

Eulerian Extension (EE)

Input: A digraph G and an integer k.

Question: Is there an Eulerian extension $\mathcal{E}$ for G with $|\mathcal{E}| \leq k$?

Variants of EE include Weighted Eulerian Extension (WEE), where an additional weight function $\omega : V \times V \to \mathbb{N}_0$ is given[2] that assigns weights to non-arcs and the solution is required to have total weight at most $\omega_{\max}$. There are also unweighted and weighted *multigraph* variants (where parallel arcs are allowed in the input and output) referred to as Multigraph Eulerian Extension (MEE) and Weighted Multigraph Eulerian Extension (WMEE), respectively. The main focus of this paper lies with the latter of these problems:

Weighted Multigraph Eulerian Extension (WMEE)

Input: A directed multigraph $M = (V, A)$, a weight function $\omega : V \times V \to \mathbb{N}_0$, and an integer $\omega_{\max}$.

Question: Is there an Eulerian extension $\mathcal{E}$ for M with $\sum_{a \in \mathcal{E}} \omega(a) \leq \omega_{\max}$?

WMEE has applications in scheduling where it generalizes the three-machine flowshop problem with "no-wait" constraint that aims at minimizing the number of interruptions (denoted by F3—nwt—$\mathcal{G}$ in standard scheduling notation [24]). This problem occurs in steel manufacturing [26]. Here, a special case of WMEE with restricted weight function is used. Furthermore, the numerous applications of Rural Postman carry over to WMEE since both problems are equivalent. These applications range from vehicle routing problems like street sweeping, snow plowing, garbage collection, road sanding, or bus routing to parking meter collection and electrical meter reading [13, 17, 28, 29].

Related Problems and Previous Work The unweighted and undirected extension problems for graphs and multigraphs were already discussed in 1977 by Boesch et al. [5], who developed a linear-time algorithm for the multigraph case and a matching-based algorithm for the graph case. Later, Lesniak and Oellermann [30] surveyed undirected Eulerian graphs, including characterizations of Eulerian (multi-)graphs and graphs that can be extended to Eulerian graphs. Recently, Höhn et al. [26] initiated a study of Eulerian extension problems applied to sequencing problems in scheduling. They showed that special cases of WMEE with weight function $\omega : V \times V \to \{0, 1\}$ are NP-complete. In particular, they considered vertex sets $V \subseteq \mathbb{R}_0^+ \times \mathbb{R}_0^+$ and for each $u = (u_x, u_y)$

[1] Here, following previous work [26], we call this "extension" problem. In the graph modification context, this is also known as "completion" or "addition" problem.

[2] We use $\mathbb{N}_0$ to denote $\mathbb{N} \cup \{0\}$.

and $v = (v_x, v_y)$ they set

$$\omega((u,v)) := \begin{cases} 0, & \text{if } u_x \leq v_x \text{ and } u_y \leq v_y \\ 1, & \text{otherwise.} \end{cases}$$

To the best of our knowledge, general WMEE has not been considered in the literature so far.

Regarding other modification operations, Cai and Yang [8] considered Eulerian subgraphs of undirected graphs. They presented a color-coding based randomized algorithm for computing a k-edge Eulerian subgraph of a given graph and stated that the derandomized algorithm runs in $2^{O(k)} nm \log n$ time on an n-vertex, m-edge graph. They also showed that the problem of finding k vertices to delete to make a given graph Eulerian is W[1]-hard with respect to k. In recent work, Fomin and Golovach [20] continued their research, answering open questions about dual parameterizations for parameterized problems introduced by Cai and Yang [8]. Furthermore, they proved that both, the problem of computing a k-edge Eulerian subgraph and the problem of computing a k-vertex Eulerian subgraph are unlikely to admit polynomial-size problem kernels with respect to k. Cygan et al. [11] showed that the problem of deleting k arcs (edges) in order to obtain an Eulerian graph is NP-hard and fixed-parameter tractable with respect to k and is unlikely to admit a problem kernel whose size is polynomial in k. Cechlárová and Schlotter [9] examined a similar graph modification problem. There, the task is to delete a minimum number of arcs of a given digraph such that each strongly connected component of the resulting digraph is Eulerian. Apart from NP-hardness, they showed fixed-parameter tractability and intractability for some application-related parameters. Under the name Min-DESC, this problem was recently considered by Crowston et al. [10] who showed that, on tournaments, it is fixed-parameter tractable with respect to the number of arcs to delete.

Eulerian extension problems are closely related to the well-known Chinese Postman problem [16], where the goal is to find a shortest walk that visits all arcs of a given directed graph, and the more general Rural Postman problem [17, 29] that asks for a shortest walk covering at least a given set of arcs. These given arcs are called “required arcs”. Although Rural Postman can be defined for undirected graphs, we mostly consider its directed version in this work. For Rural Postman, we allow requiring to visit some arcs multiple times, that is, we allow the required arcs to form a multiset. 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 [29] (in fact, Rural Postman is one of the prominent NP-complete problems featured by Garey and Johnson [23]). Moreover, the fact that Rural Postman is solvable in polynomial time if the required arcs form a weakly connected component [22] 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 “minimum number k of extension arcs”. In fact, given a multigraph M on n vertices and a weight function ω, our algorithm finds an Eulerian extension $\mathcal{E}$ with minimum total weight among all Eulerian extensions $\mathcal{E}'$ for M in time $O(4^k \cdot n^3)$. Using the above-mentioned equivalence, our algorithm implies a—to the best of our knowledge—first fixed-parameter tractability result for Rural Postman.

	weighted, (weakly) connected	unweighted
undir. graph	$O(n^3)$ (Theorem 1)	$O(\overline{m}\sqrt{n})$ (Theorem 2)
undir. multigraph	$O(n^3)$ (Corollary 1)	$O(n+m)$ (Proposition 5)
dir. graph	$O(\overline{m}^2 + n\overline{m}\log n)$ (Prop. 2)	$O(\overline{m}(n+\overline{m}))$ (Proposition 4)
dir. multigraph	$O(n^3 \log n)$ (Proposition 3)	$O(n+m)$ (Proposition 5)

Table 1: Polynomial-time solvable Eulerian extension problems. Here, n denotes the number of vertices in the input, m denotes the number of edges (arcs) in the input, and $\overline{m}$ denotes the number of edges (arcs) in the complement of the input (di)graph. The running times are under the assumption that the complement of the input is given in advance. If this is not the case, then running times involving $\overline{m}$ receive an additional summand $O(n^2)$. In general, weighted variants of Eulerian extension problems are NP-hard if the input (multi-)graph is not (weakly) connected [26, 29].

As a warm-up, we present a number of polynomial-time solvable variants of Eulerian extension problems. In particular, we consider inputs that are connected or unweighted (see Table 1). This stands in contrast to Rural Postman, whose unweighted variant is NP-hard [29]. The results translate to RP in the following way. Unweighted inputs correspond to instances of RP in which all edges (arcs) are present. (Weakly) connected inputs correspond to instances of RP in which the required edges (arcs) form a (weakly) connected component. Disallowing multisets as inputs corresponds to disallowing using an edge (arc) more than once in a solution for RP.

Altogether, our work complements and extends known results for special cases of WMEE where the weight function only assigns values zero and one [26] (for these variants, only NP-hardness was shown so far; we provide a first algorithmic result for this case) and Rural Postman, for which mainly approximation, heuristic, and some polynomial-time algorithms for special cases were known [17, 22].

Organization of the Paper In the next section, we fix the notation and present basic observations that will help us prove some of our results. Section 3 contains polynomial-time algorithms for special cases of Eulerian extension problems. The main result, a dynamic programming algorithm for WMEE, is presented and discussed in Section 4. Section 5 concludes with a review on open questions and future work.

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 C_G. Let a be an arc of G. We denote the result of removing a from G with $G - a$. If $G - a$ has more connected components than G, then we say that a is a *bridge* of G. For an arc set $A' \subseteq V \times V$, let $G + A' := (V, A \cup A')$ and $G - A' := (V, A \setminus A')$. 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 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 tour* in a directed (multi-)graph G is a tour that visits all arcs of G exactly once (possibly visiting each vertex more than once). If such a tour 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, where the total weight of $\mathcal{E}$ is $\sum_{a \in \mathcal{E}} \omega(a)$ for a weight function $\omega : V \times V \to \mathbb{N}_0$. Note that, in this context, the sum is over all elements of the multiset $\mathcal{E}$, including identical elements. Hence, if arc a occurs twice in $\mathcal{E}$, then the weight of a is counted twice in the sum. Likewise, the cardinality $|\mathcal{E}| := \sum_{a \in \mathcal{E}} 1$ counts identical elements individually. For sets or multisets A of arcs, we abbreviate $\omega(A) := \sum_{a \in A} \omega(a)$. 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. Since each arc can occur multiple times in a walk, we also consider walks as multisets of arcs. If the sequence W starts in the same vertex as it ends, then we call W *closed*.

For a vertex v of a directed (multi-)graph G, the *out-degree* of v, denoted by $\mathrm{d}^+(v)$, is the number of arcs in A that are outgoing of v. Likewise, the *in-degree* of v, denoted by $\mathrm{d}^-(v)$, is the number of arcs in A that are incoming to v. The *imbalance* of a vertex v is

$$\mathrm{imbal}(v) := \mathrm{d}^-(v) - \mathrm{d}^+(v).$$

Specifically, let $\mathcal{I}_G^-$ ($\mathcal{I}_G^+$) denote the multiset of vertices v of G for which $\mathrm{imbal}(v) > 0$ ($\mathrm{imbal}(v) < 0$). Herein, each vertex v is contained $|\mathrm{imbal}(v)|$ times in the multiset. In an undirected graph, we define the imbalance $\mathrm{imbal}(v)$ 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 $\mathrm{imbal}(v) = 0$ are called *balanced*, while all other vertices of G are called *imbalanced*, with $\mathcal{I}_G := \mathcal{I}_G^- \uplus \mathcal{I}_G^+$ denoting the multiset of imbalanced vertices of G. Connected components in C_G that do not contain imbalanced vertices are called *balanced*. We refer to the set of all balanced components of G by $C_G^{\mathrm{bal}} \subseteq C_G$. With the concept of vertex balance, we can state a well-known characterization of Eulerian graphs and multigraphs that helps us prove a (multi-)graph to be Eulerian.

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. 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.

Eulerian extension problems are closely related to arc routing problems (for more details about arc routing, refer to Eiselt et al. [16, 17], Dror [13], and van Bevern et al. [3]). An important role in this relation plays the Rural Postman problem [17, 29]:

> Rural Postman (RP)
>
> **Input:** A digraph $G = (V, A)$, a nonempty multiset R of "required" arcs of A, a weight function $\omega : A \to \mathbb{N}_0$, and an integer $\omega_{\max} \geq 0$.
>
> **Question:** Is there a closed walk W in G such that W visits all arcs in R and the total weight of W is at most $\omega_{\max}$?

If $R = A$, then RP degenerates to the Chinese Postman problem [7, 14, 16, 33].

Parameterized Complexity Our results are in the context of parameterized complexity, which is a two-dimensional framework for studying computational complexity [12, 19, 35]. 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*. When considered as parameterized problems, in this article, all variants of EE are parameterized by the minimum cardinality k of all Eulerian extensions with weight at most $\omega_{\max}$. More formally,

$$k := \min\{|\mathcal{E}| : \mathcal{E} \text{ is an Eulerian extension for } G \text{ and } \omega(\mathcal{E}) \leq \omega_{\max}\}. \tag{1}$$

RP is parameterized by q, the minimum number of "additional" arcs over all solutions. More formally,

$$q := \min\{|W| - |R| : W \text{ is a walk in } G \text{ visiting all arcs in } R\}. \tag{2}$$

Note that for RP, q is a "stronger" parameter than the number of arcs in W, because $q \leq |W|$. This implies that all positive (algorithmic) results for RP parameterized by q also hold for RP parameterized by $|W|$. Since all solutions guarantee to contain R, choosing q can be considered an above-guarantee parametrization [31, 32] 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 imbalance always adds up to zero, that is, for each "missing" incoming arc, there is also a "missing" outgoing arc.

Observation 1. *Let G be a directed (multi-)graph. Then $\sum_{v \in V(G)} \operatorname{imbal}(v) = 0$ and, equivalently, $|\mathcal{I}_G^-| = |\mathcal{I}_G^+|$.*

In undirected graphs and multigraphs, the sum over all imbalances is even. Observation 1 also applies to connected components. Next, observe that an undirected Eulerian graph cannot have

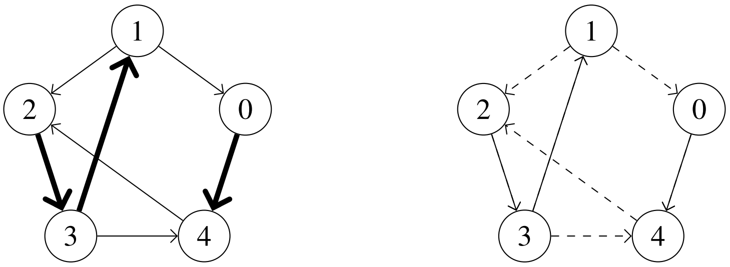

Figure 1: An illustration of equivalent instances of Rural Postman and Weighted Multigraph Eulerian Extension, respectively. Left: Instance of Rural Postman with required arc drawn in bold. Right, a corresponding instance of Weighted Multigraph Eulerian Extension, where the dashed arcs are not present in the graph but have finite weight. Arcs that are not drawn have infinite weight. An exemplary postman tour visiting all required arcs would be $(2, 3, 1, 0, 4, 2)$. The corresponding Eulerian extension is $\{(1, 0), (4, 2)\}$. Note that the weight of the Eulerian extension equals the weight of the postman tour minus the weight of the required arcs.

bridges since a walk visiting all edges would have to cross the bridge, effectively cutting off its way back to the starting point.

Observation 2. *Let G be a directed or undirected Eulerian (multi-)graph. Then there is no bridge in G.*

Next, consider an Eulerian extension $\mathcal{E}$ of some directed (multi-)graph G. Clearly, for each occurrence of a vertex v in $\mathcal{I}_G^-$, there is an arc outgoing of v in $\mathcal{E}$. Moreover, for each balanced connected component of G, there is at least one arc in $\mathcal{E}$ that leaves this component. Considering that $|\mathcal{I}_G| = 2|\mathcal{I}_G^-|$, we can make the following observation.

Observation 3. *Let G be a directed or undirected (multi-)graph that is not Eulerian and let $\mathcal{E}$ be an Eulerian extension of G. Then $|\mathcal{I}_G|/2 + |C_G^{\text{bal}}| \leq |\mathcal{E}|$.*

In the remainder of this section, we show an important relation between RP and WMEE. An example illustrating this relation is shown in Figure 1.

Proposition 1. Rural Postman *and* Weighted Multigraph Eulerian Extension *are parameterized equivalent.*

Proof. The idea to show the claim is to identify the required arcs of the RP-instance with the arcs of the input multigraph for WMEE. Furthermore, identify non-arcs in the RP-instance with non-arcs of weight ∞ in the WMEE-instance. Note that parallel arcs carried into the RP instance from the WMEE-instance can be handled by subdividing them.

First, we show that RP is parameterized reducible to WMEE. To this end, we construct an instance $(M, \omega', \omega'_{\max})$ of WMEE with parameter k from an instance $(G = (V, A), R, \omega, \omega_{\max})$ of RP with parameter q by setting $M := (V, R)$, $\omega'_{\max} = \omega_{\max} - \omega(R)$, $k := q$, and

$$\omega'(a) := \begin{cases} \omega(a) & \text{if } a \in A, \\ \infty & \text{otherwise.} \end{cases}$$

Since the weight functions are basically equal, "the weight" of some arc set may refer to either ω or ω'. Let $\mathcal{E}$ denote a solution for $(M, \omega', \omega'_{\max})$. Then, by definition of Eulerian graphs, $W := \mathcal{E} \cup R$ is a walk of weight at most $\omega'_{\max} + \omega'(R)$ that visits all arcs in R. Furthermore, if we have a solution W for $(G, R, \omega, \omega_{\max})$, then $\mathcal{E} := W \setminus R$ is an Eulerian extension of weight at most $\omega_{\max} - \omega(R) = \omega'_{\max}$ for M with $|\mathcal{E}| \leq q = k$.

Second, we show that WMEE is parameterized reducible to RP. Given an instance $(M = (V, A'), \omega', \omega'_{\max})$ of WMEE with parameter k, we construct an instance $(G, R, \omega, \omega_{\max})$ of RP with parameter q by setting $G := (V, V \times V)$, $R := A'$, $\omega := \omega'$, $\omega_{\max} := \omega'_{\max} + \omega'(R)$, and $q := k$. Note that parallel arcs in M can be handled by subdivision, that is, if the arc (u, v) appears twice in A' then we introduce a new vertex w into M and replace one of the arcs (u, v) with the two arcs (u, w) and (w, v) with $\omega((u, w)) = 0$ and $w((w, v)) = w((u, v))$.

Analogously to the first part of the proof, a solution $\mathcal{E}$ for $(M, \omega', \omega'_{\max})$ implies a solution $W = \mathcal{E} \cup R$ for $(G, R, \omega, \omega_{\max})$ and vice versa. □ □

3 Polynomial-Time Solvable 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 the number n of vertices in the input, the number m of edges (arcs) in the input, and the number $\overline{m}$ of edges (arcs) in the complement of the input graph. To allow focus on the algorithmic results, we assume that the complement of the input is given in advance. If this is not the case, then running times involving $\overline{m}$ receive an additional summand $O(n^2)$. We refer to Table 1 for an overview of the results of this section.

In this section, we present results for all polynomially solvable combinations of the properties connected and weighted. First, we consider weighted variants of Eulerian extension problems if the input (multi-)graph is connected (Section 3.1). Then, we consider the unweighted variant and allow disconnected (multi-)graphs (Section 3.2).

3.1 Algorithms for Connected Weighted Variants

Keeping in mind that the disconnected versions of WEE and WMEE are NP-hard [26, 29], we provide polynomial-time algorithms for both problems in case of connected inputs. When considering directed inputs, "connected" always means "weakly connected".

The first result states that Weighted Eulerian Extension on connected, undirected graphs can be solved in polynomial time. This can be derived from a result by Edmonds and Johnson [15], who introduced the notion of "T-joins". For a graph G and a vertex-set T, a *T-join* is a set J of edges of G such that each vertex v of G is incident to an odd number of edges in J if and only if $v \in T$. Edmonds and Johnson [15] showed that the following problem can be solved in polynomial time.

Minimum-weight T-Join

Input: An undirected graph $G = (V, E)$, a set $T \subseteq V$, a weight function $\omega : E \to \mathbb{N}$, and an integer k.

Question: Is there a T-join of weight at most k for G?

Theorem 1 ([15, 27])**.** Weighted Eulerian Extension *on connected undirected graphs can be solved in $O(n^3)$ time.*

In order to apply the result of Edmonds and Johnson [15] to multigraphs, we show that there is a minimum-weight Eulerian extension in a connected undirected multigraph that does not use any edge more than once.

Lemma 2. *Let $M = (V, E)$ be a connected undirected multigraph, let $\omega : \binom{V}{2} \to \mathbb{N}_0$ be an edge-weight function and let $\mathcal{I}_M$ denote the imbalanced vertices of M. Let G be a clique on the vertex set V. Then, a minimum-weight $\mathcal{I}_M$-join in G is also a minimum-weight Eulerian extension $\mathcal{E}$ for M.*

Proof. Clearly, each $\mathcal{I}_M$-join in G is an Eulerian extension for M. However, to show that there is no Eulerian extension of less weight, we prove that, for each minimum-weight Eulerian extension that uses some edge twice, we can find a minimum-weight Eulerian extension in which less edges occur twice. By iterating this argument, we can find a minimum-weight Eulerian extension that does not use any edge more than once and, hence, this Eulerian extension also constitutes a minimum-weight $\mathcal{I}_M$-join in G.

Let $\mathcal{E}$ be a minimum-weight Eulerian extension in M that contains an edge e at least twice and let $\mathcal{E}'$ denote the result of removing two occurrences of e from $\mathcal{E}$. Clearly, $M + \mathcal{E}'$ is connected and, since $M + \mathcal{E}$ is balanced, $M + \mathcal{E}'$ is balanced. Thus, by Lemma 1, $\mathcal{E}'$ is an Eulerian extension for M. □ □

With Lemma 2, we can now state Theorem 1 for connected undirected multigraphs.

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

It remains to show polynomial-time solvability for directed graphs and multigraphs. First, we present an algorithm for digraphs, which is then modified to work for directed multigraphs. The algorithm is based on computing network flows or matchings (refer to Ahuja et al. [2] for flow notations).

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 $(G, \omega, k, \omega_{\max})$ of WEE, where G is a connected digraph, and a function imbal : $V(G) \to \mathbb{Z}$ measuring the imbalance of each vertex (see Section 2). Consider the flow network $\overline{G}$ (the complement graph of G) with supply determined by the function imbal (negative supply indicates demand), arc capacity one for each arc, and arc-costs determined by ω. Let f be an integral flow of value $1/2 \cdot \sum_{v \in V} |\text{imbal}(v)|$ in this network. Then, the set of arcs carrying f 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.[3] □ □

[3] See Exercise 10.17 of Ahuja et al. [2], a solution to which can be found on the web [1].

Next, we allow parallel arcs in the input and modify the algorithm described in the proof of Proposition 2. 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 employ a min-cost flow algorithm on G_M with arc capacities ∞ and weights according to ω and supply determined by the function imbal. The uncapacitated version of the min-cost flow algorithm (running in $O(n^3 \log n)$ time [2]) can be used in this case.

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

3.2 Algorithms for General Unweighted Variants

In this section, we provide polynomial-time algorithms for EE on various input (multi-)graphs. First, we state a previously known result for Eulerian Extension and show how algorithms of Section 3.1 help to solve the problem on connected digraphs. We then present an algorithm for disconnected digraphs and an algorithm solving EE on directed and undirected multigraphs.

Eulerian extensions were already considered by Boesch et al. [5], who showed that, on undirected graphs, Eulerian Extension can be solved in polynomial time, even for disconnected graphs.

Theorem 2 ([5, 30])**.** Eulerian Extension *on undirected graphs can be solved in* $O(\overline{m}\sqrt{n})$ *time.*

Next, we extend this result to digraphs. 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 $O(\overline{m}(n + \overline{m}))$ time.[4]

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

This algorithm cannot handle multiple components. In the following, we present a more general algorithm that also allows us to solve the problem on disconnected digraphs (at the cost of increased running time). The algorithm starts with a min-cost max-flow in the complement digraph and locally modifies the arcs that carry flow in order to connect all components of the input digraph. For ease of presentation we split the following lemma concerning digraphs of diameter two[5] from the main correctness proof.

Lemma 3. *Let f be a min-cost max-flow in a unit-weight unit-capacity digraph G. Let F denote the arcs of G that carry flow and let $G-F$ have diameter two. Then there are no three consecutive arcs in F.*

Proof. Assume that F contains three consecutive arcs (u, v), (v, w), and (w, x). Since $G - F$ has diameter two, there is a vertex v' in $G - F$ such that (u, v') and (v', x) are in $G - F$. Hence, replacing (u, v), (v, w), and (w, x) with (u, v') and (v', x) in F yields a max-flow of less cost, contradicting our choice of F. □ □

[4] Combine the solution [1] for Exercise 10.17 of Ahuja et al. [2] with breadth-first search as shortest path algorithm.

[5] In a digraph, the *distance* of u to v is the number of arcs in a shortest path from u to v if such a path exists, and ∞ otherwise. The *diameter* of a digraph G is the maximum distance between any pair of vertices in G.

The following algorithm receives a digraph $G = (V, A)$ and a vertex imbalance function imbal : $V \to \mathbb{Z}$ and computes a minimum-cost Eulerian extension for G, if it exists. The algorithm comprises three steps that modify the input digraph G. In Step i, we denote the current extension set by F_i.

Step 1: Compute a min-cost max-flow f of value $|\mathcal{I}_G|/2$ (recall that $\mathcal{I}_G$ denotes the multiset of imbalanced vertices of G) in the complement graph $\overline{G}$ with supply determined by the function imbal (negative supply indicates demand), arc capacities one and arc weights one for each arc. If there is no such flow, then there is no Eulerian extension for G. Let F_1 denote the set of arcs that carry flow. Note that $G + F_1$ does not contain imbalanced vertices.

Step 2: Let F_2 denote the current extension set after Step 1. In this step, we modify F_2 such that all vertices that are imbalanced in G are in the same connected component of $G + F_2$. If this is already the case, then continue with Step 3. Otherwise, find two different connected components C_1 and C_2 containing vertices that are imbalanced in G. By Observation 2, there are arcs $(u, v) \in (C_1 \times C_1) \cap F_2$ and $(x, y) \in (C_2 \times C_2) \cap F_2$ whose removal does not disconnect C_1 or C_2. Replace (u, v) and (x, y) with (u, y) and (x, v) in F_2 and repeat Step 2. If $G + F_2$ is connected, then return F_2 as an optimal Eulerian extension for G.

Step 3: Let F_3 denote the current extension set after Step 1. Since $G + F_2$ is disconnected, $\overline{G + F_2} = \overline{G} - F_2$ has diameter two and, by Lemma 3, F_3 does not contain paths of length three. Let F_3^1 and F_3^2 denote the sets of paths of arcs in F_3 with length one and two, respectively. Join connected components of $G + F_2$ by first "rerouting" paths in F_3^2 (which can be done by replacing their middle vertex with a vertex in another connected component) if this operation decreases the number of connected components. When this is no longer possible, "split open" an arbitrary path in $G + F_3$ so that it additionally contains a vertex in another connected component.

Proposition 4. Eulerian Extension *on disconnected digraphs can be solved in $O(\overline{m}(n+\overline{m}))$ time.*

Proof. First, we show that the presented algorithm produces an optimal Eulerian extension for G. Let $\mathcal{E}$ denote the set of arcs that are returned by the algorithm above. After Step 1, $G + F_1$ does not contain imbalanced vertices and for each arc (u, v) removed from F_1 in later steps, a path from u to v is added. Hence, $G + \mathcal{E}$ does not contain imbalanced vertices. Since the algorithm only returns connected digraphs, $G + \mathcal{E}$ is also connected and, thus, Eulerian.

If the digraph is connected before splitting paths in Step 3, then $|\mathcal{E}| = |F_2|$, which is optimal because the existence of a smaller Eulerian extension would also imply a lower cost flow in $\overline{G}$, contradicting the fact that the cost of the flow corresponding to F_1 is minimum. Otherwise, for each path p of F_3^2, there is a different connected component C of the input digraph G such that p contains a vertex of C. Hence, at most $|C_G^{\text{bal}}| - |F_3^2|/2$ arcs are added in Step 3, implying $|\mathcal{E}| = |F| + (|C_G^{\text{bal}}| - |F_3^2|/2) = |F_3^1| + |F_3^2|/2 + |C_G^{\text{bal}}|$. By definition of F_3^1 and F_3^2, $2|F_3^1| + |F_3^2| = |\mathcal{I}_G|$. Hence, $|\mathcal{E}| = |\mathcal{I}_G|/2 + |C_G^{\text{bal}}|$, which, by Observation 3, is optimal.

It is not hard to see that computing the min-cost max-flow in in Step 1 dominates the overall running time. This flow can be computed in in $O(\overline{m}(n + \overline{m}))$ time [2]. □ □

Proposition 4 stands in contrast with RP being NP-hard for unweighted digraphs [29], which seems to be due to the fact that the input for RP may prohibit arcs by excluding them from the input digraph. It turns out that, if no arc is forbidden, that is, the input digraph is complete, then RP is solvable in polynomial time. More precisely, we can solve MEE for directed multigraphs M by the following straightforward greedy strategy much like the algorithm known for undirected multigraphs [5].

Step 1: For each connected component C of M, arbitrarily select a pair of vertices $(x_C, y_C) \in \mathcal{I}_C^- \times \mathcal{I}_C^+$ if C is imbalanced, and $(x_C, y_C) \in \{(v, v) : v \in V(C)\}$ if C is balanced.

Step 2: Choose an arbitrary order $(C_1, C_2, \ldots, C_{|\mathcal{C}_M|})$ of the connected components and add the arcs $(y_{C_1}, x_{C_2}), (y_{C_2}, x_{C_3}), \ldots, (y_{C_{|\mathcal{C}_M|}}, x_{C_1})$, thus connecting all connected components in a circular manner.

Step 3: Greedily insert arcs between the remaining imbalanced vertices until all vertices are balanced.

Proposition 5. Multigraph Eulerian Extension *on directed and undirected multigraphs can be solved in* $O(n + m)$ *time.*

Proof. Boesch et al. [5] showed the claim for undirected multigraphs. We extend this to directed multigraphs. If the input multigraph M is connected, then we can skip Step 1 and Step 2. Clearly, after Step 2, the multigraph is connected, and after Step 3 all vertices are balanced. Hence, by Lemma 1, the set $\mathcal{E}$ of all inserted arcs is an Eulerian extension of M.

Next, we show that $\mathcal{E}$ is also optimal. Let $\mathcal{E}_2$ and $\mathcal{E}_3$ denote the arcs that are added in Step 2 and Step 3, respectively, and define X_2 and X_3 as the multisets that contain a vertex v as many times as there are arcs in $\mathcal{E}_2$ and $\mathcal{E}_3$, respectively, that are incident to v. Clearly, $|X_2| = 2|\mathcal{E}_2| = 2|\mathcal{C}_M|$ and there are $2|\mathcal{C}_M| - 2|\mathcal{C}_M^{\text{bal}}|$ imbalanced vertices in X_2. Since the result of Step 3 is balanced, $|X_3| = |\mathcal{I}_M| - (2|\mathcal{C}_M| - 2|\mathcal{C}_M^{\text{bal}}|)$. Hence,

$$2|\mathcal{E}| = |X_2| + |X_3| = 2|\mathcal{C}_M| + |\mathcal{I}_M| - (2|\mathcal{C}_M| - 2|\mathcal{C}_M^{\text{bal}}|) = |\mathcal{I}_M| + 2|\mathcal{C}_M^{\text{bal}}|,$$

which, by Observation 3, is optimal.

For the running time note that, in Step 1, we just have to scan each connected component, and in Step 2 and Step 3, each arc addition can be done in constant time. □ □

4 Weighted Eulerian Extension on Directed Multigraphs

In this section, we prove that WMEE is fixed-parameter tractable with respect to the parameter k denoting the cardinality of an Eulerian extension (more thoroughly defined in (1)). We describe a dynamic programming algorithm to solve WMEE. More precisely, given a multigraph M, a weight function ω, and some integer k, our algorithm finds an Eulerian extension $\mathcal{E}$ with minimum total weight over all Eulerian extensions $\mathcal{E}'$ for M with $|\mathcal{E}'| = k$. Then, using a simple self-reduction, we can find an Eulerian extension $\mathcal{E}$ for M with the following properties:

(i) the total weight of $\mathcal{E}$ is at most $\omega_{\max}$,

(ii) of all Eulerian extensions satisfying (i), $\mathcal{E}$ has minimum cardinality, and

(iii) of all Eulerian extensions satisfying (i) and (ii), $\mathcal{E}$ has minimum total weight.

First, in Section 4.1, we preprocess the input to obtain an equivalent yet simpler instance. Next, in Section 4.2, 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. Then, in Section 4.3, we present such an algorithm for BGWMEE. Finally, in Section 4.4, we put everything together to obtain the fixed-parameter algorithm for BGWMEE.

4.1 Preprocessing the Input

In order to simplify the presentation of our algorithm, we first preprocess the input instance $(M, \omega, \omega_{\max})$ such that it does not contain isolated vertices. To this end, we replace the weight of a vertex pair (u, v) of M with the weight of the cheapest u-v-path of non-arcs that visits only isolated vertices. This allows us to remove isolated vertices without increasing the weight of optimal solutions. Having modified the weights between all vertices this way, we remove all isolated vertices from the multigraph. A solution for the modified instance simply inserts an arc for each path visiting isolated vertices. Hence, the minimum number k of arcs in a solution does not increase. This operation can be performed by computing all-pairs shortest paths in $O(n^3)$ time and comparing the weight of each pair of non-isolated vertices with the computed weights in $O(n^2)$ time.

Observation 4. *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 an equivalent instance $(M[V \setminus V_I], \omega', \omega_{max})$ can be computed in $O(n^3)$ time without increasing the parameter k.*

It is important to note that, for each path that causes an update of the weight function, we can remember the vertices it visits. This helps us to reconstruct the actual solution from a solution of the modified multigraph.

With Observation 4, we can use the following observation, which is a direct consequence of Lemma 1, to characterize Eulerian extensions of multigraphs.

Observation 5. *A directed multigraph that does not contain isolated vertices is Eulerian if and only if it is connected and balanced.*

4.2 Transformation to BGWMEE

We introduce the Black/Gray Weighted Multigraph Eulerian Extension (BGWMEE) problem and show how it helps us solve WMEE. For this, we picture Eulerian extensions as collections of paths between imbalanced vertices as stated in the forthcoming observation. This notion is fundamental for the algorithm for BGWMEE that we present in this section. It is based on the fact that for each balanced vertex u, each Eulerian extension contains as many arcs outgoing of u as incoming to u.

Observation 6. *Let M be a directed multigraph and let $\mathcal{E}$ be an Eulerian extension of M. Then $\mathcal{E}$ can be decomposed into paths that start at a vertex in $\mathcal{I}^-_M$ and end at a vertex in $\mathcal{I}^+_M$ or start and end at the same vertex (cycles).*

Our idea to attack WMEE is to use dynamic programming to construct the paths of an optimal Eulerian extension arc by arc. There are, however, a few obstacles to this approach. Assuming that 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 (somewhat technical) problem, for which this assumption is valid.

Black/Gray WMEE (BGWMEE)

Input: Two directed multigraphs $M_{\text{black}} = (V, A_{\text{black}})$ and $M_{\text{gray}} = (V, A_{\text{gray}})$ such that each connected component of M_{black} is either balanced or contains exactly two imbalanced vertices (one with imbalance 1, one with imbalance -1), a weight function $\omega : V \times V \to \mathbb{N}_0$, and an integer $\omega_{\max}$.

Question: Is there an Eulerian extension $\mathcal{E}'$ of total weight at most $\omega_{\max}$ for $M := (V, A_{\text{black}} \cup A_{\text{gray}})$ such that in each component C_{black} of M_{black} there is exactly one tail of an arc in $\mathcal{E}'$ and exactly one head of an arc in $\mathcal{E}'$ (that is, $|(V(C_{\text{black}})\times V) \cap \mathcal{E}'| = |(V \times V(C_{\text{black}})) \cap \mathcal{E}'| = 1$)?

Again, we can decompose a black/gray Eulerian extension $\mathcal{E}'$ into paths analogously to Observation 6. The advantage of BGWMEE is that, in total, these paths visit each black component exactly once. 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 for BGWMEE in Section 4.3.

To solve WMEE using BGWMEE, we transform an instance $(M, \omega, \omega_{\max})$ of WMEE into an instance $(M', \omega', \omega'_{\max})$ of BGWMEE. In this transformation algorithm, we compute a bijection[6] $\mu \subseteq \mathcal{I}^+_M \times \mathcal{I}^-_M$ such that each two vertices related by μ are in the same connected component of M. We call such a bijection *component-respecting*. Since Observation 1 holds for each connected component, there is a component-respecting bijection μ for each instance $(M, \omega, \omega_{\max})$ of WMEE. The transformation receives, additionally to the instance $(M, \omega, \omega_{\max})$, a function $\# : C_M \to \mathbb{N}_0$ indicating for each connected component C of M, the number of times it is visited by the sought solution. We denote the transformed instance by $\text{tr}_\#(M, \omega, \omega_{\max}) := (M', \omega', \omega_{\max})$. See Figure 2 for an example. The following describes the transformation algorithm.

Step 1: Compute a component-respecting bijection μ.

Step 2: Compute a new multigraph M' by creating $\#(C)$ copies of each connected component C of M. Construct ω' such that $\omega'(x', y') = \omega(x, y)$ for all vertices x, y of M and their copies x', y' in M'.

Step 3: For each component C of M, assign a copy C' of C to each pair $(v, w) \in \mu$ of imbalanced vertices of C and *isolate* (v, w) in C', that is, add all arcs in $(\mathcal{I}^+_C \times \mathcal{I}^-_C) \setminus \{(v, w)\}$

[6] We say that μ is bijective if for all submultisets $X \subseteq \mathcal{I}^+_M$, it holds that $|\mu \cap (X \times \mathcal{I}^-_M)| = |X|$ and for all submultisets $X \subseteq \mathcal{I}^-_M$ it holds that $|\mu \cap (\mathcal{I}^+_M \times X)| = |X|$.

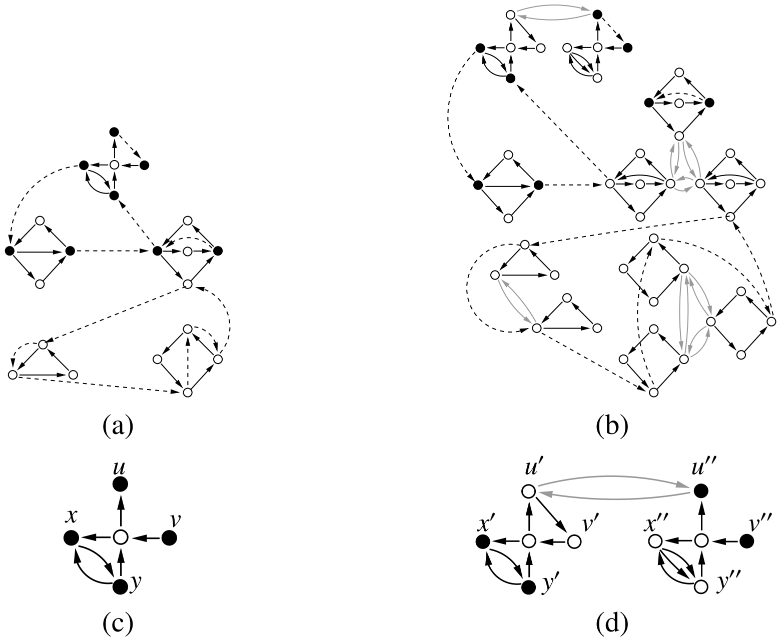

Figure 2: The picture shows a directed multigraph with an Eulerian extension containing eleven (dashed) arcs (a) being transformed to (b) by the transformation $\mathrm{tr}_{\#}$ described in Section 4.2. White vertices are balanced, black vertices are imbalanced. In (c) and (d), we take a closer look at the top connected component C of the graph depicted in (a) and its transformation. Here, $\#(C) = 2$, $\mu(v) = u$ and $\mu(y) = x$. Note that the pair (y'', x'') is isolated in the first copy and the pair (v', u') is isolated in the second copy. Finally, as described in the last step of the transformation, u' and u'' are connected by gray arcs.

to C'. All copies that have not been assigned to an imbalanced pair are balanced completely in the above mentioned way. This assures that each copy of C contains at most one pair (v', w') of vertices and their imbalance is 1 and -1, respectively. Furthermore, each pair of imbalanced vertices in μ is represented in exactly one copy.

Step 4: For each component C of M, its copies are pairwisely connected by adding gray arcs. To this end, select any vertex v of C 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.

In the following, we show that the above transformation is correct. More specifically, it turns out that one can obtain a solution for an instance of Weighted Multigraph Eulerian Extension by transforming the input for all feasible $\# : C_M \to \mathbb{N}_0$ and solving these transformed instances of BGWMEE. Indeed, we can reduce the number of functions # for which $(M, \omega, \omega_{\max})$ is to be transformed by imposing the following restrictions on #. Since a solution $\mathcal{E}$ for BGWMEE visits each copy exactly once (each copy forms a black component), summing up $\#(C)$ over all connected components C of G must not exceed $|\mathcal{E}|$ $(= k)$ and since each copy is assigned to at most one pair of imbalanced vertices, each connected component C must have at least

$|\mathcal{I}_C|/2$ copies. Thus, we need only consider functions of the form

$$\# : \mathcal{C}_M \to \mathbb{N}_0, \text{ with } \sum_{C \in \mathcal{C}_M} \#(C) \leq k \text{ and } \forall C \in \mathcal{C}_M : \#(C) \geq |\mathcal{I}_C|/2\,. \tag{3}$$

To prove the correctness of the described transformation, we show that finding a solution for an instance $(M, \omega, \omega_{\max})$ of WMEE is equivalent to finding a solution for an instance $\mathrm{tr}_\#(M, \omega, \omega_{\max})$ of BGWMEE for any fixed μ.

Lemma 4. *The instance $(M, \omega, \omega_{max})$ is a yes-instance of* Weighted Multigraph Eulerian Extension *if and only if there is a function $\# : \mathcal{C}_M \to \mathbb{N}_0$ complying with* (3) *such that* $\mathrm{tr}_\#(M, \omega, \omega_{max})$ *is a yes-instance of* Black/Gray Weighted Multigraph Eulerian Extension.

Proof. Throughout this proof, let $(M', \omega', \omega'_{\max}) = \mathrm{tr}_\#(M, \omega, \omega_{\max})$. Let $\mathcal{E} \subseteq V(M) \times V(M)$ and $\mathcal{E}' \subseteq V(M') \times V(M')$ such that $(u, v) \in \mathcal{E}$ if and only if there are copies u', v' of u, v in M' with $(u', v') \in \mathcal{E}'$. Under this condition, $M + \mathcal{E}$ is connected if and only if $M' + \mathcal{E}'$ is connected since, by construction, vertices u, v of M are connected if and only if all copies of u and v in M' are connected and the connectedness relation is transitive. We go on to prove the claim of the lemma.

($\Leftarrow$) Suppose that there is a function # such that we have a solution $\mathcal{E}'$ for $(M', \omega', \omega'_{\max})$. Let $\mathcal{E}$ be the arc set that results from replacing each arc $(u', v') \in \mathcal{E}'$ by the corresponding arc (u, v) between the original vertices in M. By construction, the imbalance of a vertex v of M equals the sum of imbalances of its copies in M', implying that $M + \mathcal{E}$ is balanced. Furthermore, since $\mathcal{E}$ and $\mathcal{E}'$ fulfill the conditions of the observation in the first paragraph of this proof, we also know that $M + \mathcal{E}$ is connected. Finally, note that $\mathcal{E}$ has the same weight as $\mathcal{E}'$, implying that $\mathcal{E}$ is a solution for $(M, \omega, \omega_{\max})$.

($\Rightarrow$) Let $(M, \omega, \omega_{\max})$ be a yes-instance of WMEE with solution $\mathcal{E}$. For each connected component C of M, let k_C be the number of arcs in $\mathcal{E}$ that are outgoing of a vertex of C, that is, $k_C := |\mathcal{E} \cap (V(C) \times V(M))|$. Since $|\mathcal{E}| \leq k$, we have $\sum_{C \in \mathcal{C}_M} k_C \leq k$. Moreover, since $\mathcal{E}$ is an Eulerian extension of M, each k_C satisfies $k_C \geq |\mathcal{I}_C^-| = |\mathcal{I}_C|/2$. Thus, the function $\#(C) := k_C$ complies with (3). It remains to show that $(M', \omega', \omega'_{\max})$ is a yes-instance of BGWMEE. To this end, we construct a solution $\mathcal{E}'$ for $(M', \omega', \omega'_{\max})$ as follows. For each path p from u to v in $\mathcal{E}$, we construct a path p' in $\mathcal{E}'$ by starting from a copy of u that is imbalanced in M'. Each time p reaches a vertex $x \neq v$, we select a copy of x whose black connected component is balanced in M' and has not yet been visited by $\mathcal{E}'$. Finally, we select a copy of v that is imbalanced in M' as terminal vertex of p'. Since there are k_C copies of each connected component of M in M', this selection is always possible. Since, by construction, the balance of a vertex v in M equals the sum of balances of its copies in M', we know that $M' + \mathcal{E}'$ is balanced. Furthermore, by the observation in the first paragraph of this proof, $M' + \mathcal{E}'$ is also connected. □ □

4.3 An Algorithm for BGWMEE

Having transformed an instance of WMEE to an instance of BGWMEE using the algorithm presented in Section 4.2, we can now exploit the simpler 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.

Let $(M, \omega, \omega_{\max})$ be an instance of BGWMEE and let C_M^{black} be the set of black connected components of M. For each subset $S \subseteq C_M^{\text{black}}$ and each pair of vertices $u, v \in V(S)$, our algorithm computes an entry $[S, u, v]$ of a three-dimensional dynamic programming table such that

$$[S, u, v] = \begin{array}{l}\text{minimum weight } \omega(\mathcal{E}) \text{ of an arc set } \mathcal{E} \text{ such that } \mathcal{E} + (v, u) \\ \text{is a black/gray Eulerian extension for } M[V(S)].\end{array} \tag{4}$$

If no black/gray Eulerian extension is possible with S, u, and v, then the entry $[S, u, v]$ is assigned "∞". The set S represents a submultigraph of M and the two vertices correspond to the endpoints of a (possibly "unfinished") path of an Eulerian extension (see Observation 6). The dynamic programming starts with computing the entries for sets S that contain exactly one connected component and augments each S step by step, finally computing the entries for $S = C_M^{\text{black}}$, which are used to derive a minimum-weight black/gray Eulerian extension for M with respect to ω. In the following, we describe the initialization process for the entries.

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

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

This assignment is correct, that is, it satisfies (4) 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 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 $C \in C_M^{\text{black}}$ containing two imbalanced vertices $x \in \mathcal{I}_M^+$ and $y \in \mathcal{I}_M^-$, and each $u, v \in V(C)$, set

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

This assignment satisfies (4) 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 a black/gray Eulerian extension for $M[V(C)]$.

Next, we describe the computation of the entries for larger sets S. We assume that all entries for sets S' with $|S'| < |S|$ have already been computed. For a given $S \subseteq C_M^{\text{black}}$ with $|S| > 1$ and vertices $u, v \in V(S)$, the entry $[S, u, v]$ 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

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

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

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

3. Otherwise, set $[S, u, v] := \infty$.

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

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

2. If $u \in V(S')$ and $v = y$, then set

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

3. Otherwise, set $[S, u, v] := \infty$.

Then, the weight ω_{opt} of an optimal black/gray Eulerian extension for (M', ω) is computed by

$$\omega_{\text{opt}} := \min_{u,v \in V(C_M^{\text{black}})} \{[C_M^{\text{black}}, u, v] + \omega(v, u)\}.$$

This follows immediately from (4). A corresponding black/gray Eulerian extension can be computed by storing each solution $\mathcal{E}$ in addition to its weight in each entry of the dynamic programming table.

Lemma 5. Black/Gray Weighted Multigraph Eulerian Extension *can be solved in $O(2^k \cdot n^3)$ time, where k denotes the size of the solution.*

Proof. In the above description of the dynamic programming algorithm, we already established correctness for $|S| = 1$, that is, the case that S contains a single black component. In the following, consider $|S| > 1$. We prove that the semantics of $[S, u, v]$ (see (4)) is met by the computation. To this end, recall that C denotes the black component of v in M and $S' := S \setminus \{C\}$.

The correctness proofs for the case that C is balanced and the case that C is imbalanced are very similar. Hence, we merge the correctness arguments for both cases in each of the three subcases 1–3 of the algorithm above. For ease of presentation, let $\mathcal{E}(S, u, v)$ be an arc set that corresponds to the entry $[S, u, v]$ (that is, $[S, u, v]$ contains the total weight of $\mathcal{E}(S, u, v)$) as defined in (4).

Subcase 1: In Subcase 1, we assume that u and v are in the same connected component C of S. Being a black/gray Eulerian extension for S, $\mathcal{E}(S, u, v) + (v, u)$ contains only one arc incoming to and one arc outgoing of C. Since both u and v are in C, no arc in $\mathcal{E}(S, u, v)$ starts or ends in C. Hence, C is connected to S' by gray arcs. If C is balanced, then (4) is satisfied only if $u = v$, as otherwise inserting (v, u) would make v imbalanced. If C contains two imbalanced vertices $x \in \mathcal{I}_M^+$ and $y \in \mathcal{I}_M^-$, then (4) is satisfied only if $u = x$ and $v = y$, as otherwise inserting (v, u) does not balance u and v. Thus, according to (4), we only have to ensure that $\mathcal{E}(S, u, v)$ is a black/gray Eulerian extension for S'. To this end, we try all possible $\mathcal{E}(S', u', v')$ and add the arc (v', u'). The resulting arc set of minimum weight is assigned to $\mathcal{E}(S, u, v)$. Note that entries $[S', u', v']$ with $u' = v'$ do not have to be considered, since the arc set $\mathcal{E}(S', u', v') + (v', u') + (v, u)$ would contain the loop (u', v').

Subcase 2: In Subcase 2, we assume that the connected component C is connected to S' by an arc $a \in \mathcal{E}(S, u, v)$. By an argument of Subcase 1, $u \in V(C)$ contradicts the existence of black arcs between C and S'. Hence, $u \in V(S')$. Furthermore, there cannot be an arc from C to S' in $\mathcal{E}(S, u, v)$, since together with the arc (v, u) there would be two arcs that are outgoing of C. Hence, we just have to guess the arc from S' to C in $\mathcal{E}(S, u, v)$. If C is balanced, then we try all possible $\mathcal{E}(S', u, w)$ and add the arc (w, v). The resulting arc set of minimum weight is assigned to $\mathcal{E}(S, u, v)$. If C is imbalanced, that is, it contains two imbalanced vertices $x \in \mathcal{I}_M^+$ and $y \in \mathcal{I}_M^-$, then we try all possible $\mathcal{E}(S', u, w)$ and add the arc (w, x). The resulting arc set of minimum weight is assigned to $\mathcal{E}(S, u, v)$. Observe that this is the only way for $\mathcal{E}(S, u, v)$ to balance x and y, which is necessary to satisfy (4).

Subcase 3: In Subcase 3, the entry $[S, u, v]$ does not correspond to an arc set $\mathcal{E}(S, u, v)$ such that $\mathcal{E}(S, u, v) + (v, u)$ is a black/gray Eulerian extension for S; therefore, we set $[S, u, v] := \infty$. This concludes the correctness proof of the updating process for $[S, u, v]$.

To finish the proof of Lemma 5, we show the running time of the dynamic programming. Clearly, if S contains exactly one connected component, then $[S, u, v]$ can be computed in constant time. Otherwise, since $S \subseteq C_M^{\text{black}}$, $u, v \in V(S)$, and $|C_M^{\text{black}}| \leq k$, there are $O(2^k \cdot n)$ entries of the form $[S, u, u]$, each of which can be computed in $O(n^2)$ time (Subcase 1). Furthermore, there are $O(2^k \cdot n^2)$ entries of the form $[S, u, v]$, each of which can be computed in $O(n)$ time (Subcase 2). We arrive at a total running time of $O(2^k \cdot n^3)$. □ □

4.4 The Complete Algorithm

An important part of the algorithm presented in Section 4.3 is to try all feasible functions # (denoting the number of visits for each connected component, see Section 4.2). We use the following lemma to bound on the number of these functions. Recall that k is the cardinality of a solution for WMEE.

Lemma 6. *There are at most 2^k possible functions $\# : C_M \to \mathbb{N}_0$ as defined in (3), that is, there are at most 2^k possible ways to replace each connected component of M by several copies such that there are at most k copies in total.*

Proof. Let C_M with $c := |C_M|$ denote the set of components of the input multigraph M. Note that $c \leq k$. Since each component in C_M must contribute at least one copy, it remains to distribute at most $k - c$ copies to c components. This is equal to choosing at most $k - c$ from c elements with repetition. There are at most $\sum_{i \leq k-c} \binom{c+i-1}{i} \leq 2^k$ possible ways to do so. □ □

The complete algorithm to solve WMEE runs in three steps.

Step 1: Preprocess the input multigraph M in $O(n^3)$ time such that it does not contain isolated vertices (see Observation 4).

Step 2: Compute an arbitrary component-respecting bijection $\mu : \mathcal{I}_M^+ \to \mathcal{I}_M^-$ in $O(m^2)$ time.

Step 3: For all 2^k possible functions $\# : C_M \to \mathbb{N} \setminus \{0\}$ (see Lemma 6), transform the instance and solve the transformed instance of BGWMEE in $O(2^k \cdot n^3)$ time (see Lemma 5).

The correctness of this algorithm follows directly from the correctness of the transformation algorithm (see Lemma 4) and Lemma 5. The running time is $O(4^k \cdot n^3)$.

The presented algorithm depends heavily on the knowledge of the parameter k. Since we do not know k in advance, we run Step 2 and Step 3 (the preprocessing procedure of Step 1 does not depend on k) for increasing values of k. Using geometric progression, we estimate

$$\sum_{i=0}^{k} O(4^i \cdot n^3) = O(n^3 \cdot \sum_{i=0}^{k} 4^i) = O(4^k \cdot n^3).$$

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

As a consequence of the characterization provided in Proposition 1, we can analogously solve Rural Postman parameterized by q.

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

In the preprocessing procedure described in Section 4.1, the parameter k may decrease since a minimum-size Eulerian extension may use extension paths that visit isolated vertices. These paths are arcs in the corresponding solution for the preprocessed instance, whose cardinality is thus smaller. This implies that our results also hold for the stronger parameter k' denoting the minimum cardinality of an Eulerian extension of weight at most $\omega_{\max}$ for the graph that results from the input by applying the preprocessing procedure described in Section 4.1.

5 Conclusion

We focused on Eulerian extension problems (and, due to parameterized equivalence, the Rural Postman problem), leaving the "editing version", where adding *and* deleting arcs is allowed, yet unstudied. However, even Eulerian extension problems still offer a rich field of challenges for future research in terms of multivariate algorithmics [18, 36]. 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 [37] 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 [21, 22] noted "the existence of an exact recursive algorithm that is exponential only in the number of disconnected components." However, this statement refers to an $n^{f(c)}$-time algorithm, leaving open whether the problem is fixed-parameter tractable with respect to the parameter c. Another interesting open question is whether there is a polynomial-size problem kernel (see [4, 25]) for any of the special cases of WMEE—for example, the special case studied by Höhn et al. [26]. Very recently, there was partial progress on both of these open questions [38, 39, 40]: On the one hand, WMEE was shown fixed-parameter tractable with respect to the combined parameter $(|\mathcal{I}_M|, c)$; on the other hand, for the special case studied by Höhn et al., it was shown that there is no polynomial-size kernel with respect to c (implying the same result for the weaker (that is, larger) parameter k) unless some unexpected complexity-theoretic collapse occurs [38, 39].

Finally, we conjecture that our presented approach can be extended to undirected and non-multigraph versions of Weighted Eulerian Extension.

Acknowledgments

We thank René van Bevern and Manuel Sorge for their thorough proofreading and anonymous referees for their helpful comments (in particular, pointing us to the MINIMUM-WEIGHT T-JOIN problem).

References

[1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, and C. C. Aggarwal. *Solution Manual for "Network Flows: Theory, Algorithms, and Applications"*. 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] R. van Bevern, R. Niedermeier, M. Sorge, and M. Weller. Complexity of arc routing problems. Submitted as a chapter in the book *Arc Routing: Problems, Methods and Applications* by Ángel Corberán and Gilbert Laporte (editors) that is submitted for publication by SIAM.

[4] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *Proc. 4th IWPEC*, volume 5917 of *LNCS*, pages 17–37. Springer, 2009.

[5] F. T. Boesch, C. Suffel, and R. Tindell. The spanning subgraphs of Eulerian graphs. *J. Graph Theory*, 1(1): 79–84, 1977. doi: 10.1002/jgt.3190010115.

[6] P. Burzyn, F. Bonomo, and G. Durán. NP-completeness results for edge modification problems. *Discrete Appl. Math.*, 154(13):1824–1844, 2006. doi: 10.1016/j.dam.2006.03.031.

[7] E. A. Cabral, M. Gendreau, G. Ghiani, and G. Laporte. Solving the hierarchical Chinese postman problem as a rural postman problem. *European J. Oper. Res.*, 155(1):44–50, 2004.

[8] L. Cai and B. Yang. Parameterized complexity of even/odd subgraph problems. *J. Discrete Algorithms*, 9(3): 231–240, 2011. doi: DOI:10.1016/j.jda.2011.03.004.

[9] K. Cechlárová and I. Schlotter. Computing the deficiency of housing markets with duplicate houses. In *Proc. 5th IPEC*, volume 6478 of *LNCS*, pages 72–83. Springer, 2010. ISBN 978-3-642-17492-6.

[10] R. Crowston, G. Gutin, M. Jones, and A. Yeo. Parameterized Eulerian strong component arc deletion problem on tournaments. *Inf. Process. Lett.*, 112(6):249–251, 2012. ISSN 0020-0190. doi: 10.1016/j.ipl.2011.11.014. URL http://dx.doi.org/10.1016/j.ipl.2011.11.014.

[11] M. Cygan, D. Marx, M. Pilipczuk, M. Pilipczuk, and I. Schlotter. Parameterized complexity of Eulerian deletion problems. In *Proc. 37th WG*, volume 6986 of *LNCS*, pages 131–142. Springer, 2011. ISBN 978-3-642-25869-5.

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

[13] M. Dror, editor. *Arc Routing: Theory, Solutions, and Applications*. Kluwer Academic Publishers, 2000. ISBN 9780792378983.

[14] J. Edmonds. The Chinese postman problem. *Oper. Res.*, 13(1):73–77, 1965.

[15] J. Edmonds and E. L. Johnson. Matching, Euler tours and the Chinese postman. *Math. Program.*, 5:88–124, 1973.

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

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

[18] 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.

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

[20] F. V. Fomin and P. A. Golovach. Parameterized complexity of connected even/odd subgraph problems. In *Proc. 29th STACS*, volume 14 of *LIPIcs*, pages 432–440. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2012. ISBN 978-3-939897-35-4. doi: http://dx.doi.org/10.4230/LIPIcs.STACS.2012.432. URL http://drops.dagstuhl.de/opus/volltexte/2012/3398.

[21] G. N. Frederickson. *Approximation algorithms for NP-hard routing problems*. PhD thesis, Faculty of the Graduate School of the University of Maryland, 1977.

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

[23] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[24] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. of Math.*, 5:287–326, 1979.

[25] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1): 31–45, 2007.

[26] W. Höhn, T. Jacobs, and N. Megow. On eulerian extensions and their application to no-wait flowshop scheduling. *Journal of Scheduling*, 15(3):295–309, 2012.

[27] B. H. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer, 5th edition, 2012. ISBN 978-3-642-24487-2.

[28] G. Laporte and I. H. Osman. Routing problems: A bibliography. *Ann. Oper. Res.*, 61:227–262, 1995.

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

[30] L. Lesniak and O. R. Oellermann. An Eulerian exposition. *J. Graph Theory*, 10(3):277–297, 1986. doi: 10.1002/jgt.3190100306. URL http://dx.doi.org/10.1002/jgt.3190100306.

[31] M. Mahajan and V. Raman. Parameterizing above guaranteed values: MaxSat and MaxCut. *J. Algorithms*, 31 (2):335–354, 1999.

[32] M. Mahajan, V. Raman, and S. Sikdar. Parameterizing above or below guaranteed values. *J. Comput. System Sci.*, 75(2):137–153, 2009.

[33] K. Mei-Ko. Graphic programming using odd or even points. *Chinese Math.*, 1:273–277, 1962.

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

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

[36] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proc. 27th STACS*, volume 5 of *LIPIcs*, pages 17–32. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010.

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

[38] M. Sorge. On making directed graphs Eulerian. Master's thesis, Friedrich-Schiller-Universität Jena, Germany, 2011. Available online at http://arxiv.org/abs/1101.4283v2.

[39] M. Sorge, R. van Bevern, R. Niedermeier, and M. Weller. From few components to an Eulerian graph by adding arcs. In *Proc. 37th WG*, volume 6986 of *LNCS*, pages 307–318. Springer, 2011. ISBN 978-3-642-25869-5. doi: 10.1007/978-3-642-25870-1_28. URL http://dx.doi.org/10.1007/978-3-642-25870-1_28.

[40] M. Sorge, R. van Bevern, R. Niedermeier, and M. Weller. A new view on rural postman based on Eulerian extension and matching. *J. Discrete Algorithms*, 16:12–33, 2012.