A New View on Rural Postman Based on Eulerian Extension and Matching

Manuel Sorge⋆, René van Bevern⋆⋆, Rolf Niedermeier, and Mathias Weller⋆⋆⋆

Institut für Softwaretechnik und Theoretische Informatik, TU Berlin
{manuel.sorge, rene.vanbevern, rolf.niedermeier, mathias.weller}@tu-berlin.de

Abstract We provide a new characterization of the NP-hard arc routing problem Rural Postman in terms of a constrained variant of minimum-weight perfect matching on bipartite graphs. To this end, we employ a parameterized equivalence between Rural Postman and Eulerian Extension, a natural arc addition problem in directed multigraphs. We indicate the NP-hardness of the introduced matching problem. In particular, we use it to make some partial progress towards answering the open question about the parameterized complexity of Rural Postman with respect to the number of weakly connected components in the graph induced by the required arcs. This is a more than thirty years open and long-time neglected question with significant practical relevance.

1 Introduction

The Rural Postman (RP) problem with its special case, the Chinese Postman problem, is a famous arc routing problem in combinatorial optimization. Given a directed, arc-weighted graph G and a subset R of its arcs (called "required arcs"), the task is to find a minimum-cost closed walk in G that visits all arcs of R. The manifold practical applications of RP include snow plowing, garbage collection, and mail delivery [1, 2, 3, 6, 7, 15]. Recently, it has been observed that RP is closely related (more precisely, "parameterized equivalent") to the arc addition problem Eulerian Extension (EE). Here, given a directed and arc-weighted multigraph G, the task is to find a minimum-weight set of arcs to add to G such that the resulting multigraph is Eulerian [10, 4]. RP and EE are NP-hard. In fact, their mentioned parameterized equivalence means that many algorithmic and complexity-theoretic results for one of them transfer to the other. In particular, this gives a new view on RP, perhaps leading to novel approaches to attack its computational hardness. A key issue in both problems is to determine the influence of the number c of connected components[1] on each problem's computational complexity [4, 9, 11, 13, 14]. Indeed, Frederickson [9] observed that RP (and, thus, EE) is polynomial-time solvable when c is constant. However, this left open whether c influences the degree of the polynomial or

⋆ Partially supported by the DFG, project AREG, NI 369/9 and project PABI, NI 369/7.
⋆⋆ Supported by the DFG, project AREG, NI 369/9.
⋆⋆⋆ Supported by the DFG, project DARE, NI 369/11.

[1] More precisely, c refers to the number of weakly connected components in the input for EE and the number of weakly connected components in the graph induced by the required arcs for RP.

whether RP can be solved in $f(c) \cdot n^{O(1)}$ time for some exponential function f. In other words, it remained open whether RP and EE are fixed-parameter tractable[2] with respect to the parameter c [4]. We remark that this parameter is presumably small in a number of applications [4, 7, 9], strongly motivating to attack this seemingly hard open question.

Our Results. In this work, we contribute new insights concerning the seemingly hard open question whether RP (and EE) is fixed-parameter tractable with respect to the parameter "number of components". To this end, our main contribution is a new characterization of RP in terms of a constrained variant of minimum-weight perfect matching on bipartite graphs. Referring to this problem as CONJOINING BIPARTITE MATCHING (CBM), we show its NP-hardness and a parameterized equivalence to RP and EE. Moreover, we show that CBM is fixed-parameter tractable[3] when restricted to bipartite graphs where one partition set has maximum vertex degree two. This implies corresponding fixed-parameter tractability results for relevant special cases of RP and EE which would probably have been harder to formulate and to detect using the definitions of these problems. Indeed, we hope that CBM might help to finally answer the puzzling open question concerning the parameterized complexity of RP with respect to the number of components. In this paper we consider decision problems. However, our results easily transfer to the corresponding optimization problems.

For the sake of notational convenience and justified by the known parameterized equivalence [4], most of our results and proofs refer to EE instead of RP. Due to space constraints, most proofs are deferred to a full version of the paper.

2 Preliminaries and Preparations

Consider a directed multigraph $G = (V, A)$, comprising the vertex set V and the arc multiset A. For notational convenience, we define a *component graph* $\mathbb{C}_G$ as a clique whose vertices one-to-one correspond to the weakly connected components of G. Since we never consider strongly connected components, we omit the adverb "weakly". A *walk* W in G is a sequence of arcs in G such that each arc ends in the same vertex as the next arc starts in. We use $V(W)$ and $A(W)$ to refer to the set of vertices in which arcs of W start or end, and the multiset of arcs of W, respectively. The first vertex in the sequence is called the *initial* vertex of the walk and the last vertex in the sequence is called the *terminal* vertex of the walk. A walk W in G such that $A(W)$ is a submultiset of the multiset $A(G)$ is called a *trail* of G. A trail T in G such that every vertex in G has at most two incident arcs in $A(T)$ is called a *cycle* if the initial and terminal vertices of T are equal, and *path* otherwise. If G is clear from the context, we omit it. We use $\text{balance}(v) := \text{indeg}(v) - \text{outdeg}(v)$ to denote the *balance* of a vertex v in G and I_G^+ and I_G^- to denote the set of all vertices v in G with $\text{balance}(v) > 0$ and $\text{balance}(v) < 0$, respectively. A vertex v is *balanced* if $\text{balance}(v) = 0$.

[2] See Section 2 and the literature [5, 8, 12] for more on parameterized complexity analysis.

[3] The corresponding parameter "join set size" measures the instance's distance from triviality and translates to the parameter "number of components" in equivalent instances of EE and RP.

Our results are in the context of parameterized complexity [5, 8, 12]. A *parameterized problem* $L \subseteq \Sigma^* \times \mathbb{N}$ is called *fixed-parameter tractable* (*FPT*) with respect to a parameter k if $(x, k) \in L$ is decidable in $f(k) \cdot |x|^{O(1)}$ time, where f is a computable function only depending on k.

We consider two types of parameterized reductions between problems: A *polynomial-parameter polynomial-time many-one reduction* ($\leq_{\mathrm{m}}^{\mathrm{PPP}}$-reduction) from a parameterized problem L to a parameterized problem L' is a polynomial-time computable function g such that $(x, k) \in L \Leftrightarrow (x', k') \in L'$, with $(x', k') := g(x, k)$, and $k' \leq p(k)$, where p is a polynomial only depending on k. If such a reduction exists, we write $L \leq_{\mathrm{m}}^{\mathrm{PPP}} L'$. A *parameterized Turing reduction* ($\leq_{\mathrm{T}}^{\mathrm{FPT}}$-reduction) from a parameterized problem L to a parameterized problem L' is an algorithm that decides $(x, k) \in L$ in $f(k) \cdot |x|^{O(1)}$ time, where queries of the form $(x', g(k)) \in L'$ are assumed to be decidable in O(1) time and f, g are functions solely depending on k. If such a reduction exists, we write $L \leq_{\mathrm{T}}^{\mathrm{FPT}} L'$. If $L \leq_{\mathrm{T}}^{\mathrm{FPT}} L'$ and $L' \leq_{\mathrm{T}}^{\mathrm{FPT}} L$, then we say that L and L' are $\leq_{\mathrm{T}}^{\mathrm{FPT}}$-equivalent. Note that every $\leq_{\mathrm{m}}^{\mathrm{PPP}}$-reduction is a $\leq_{\mathrm{T}}^{\mathrm{FPT}}$-reduction. Also, if $L' \in$ FPT and $L \leq_{\mathrm{T}}^{\mathrm{FPT}} L'$, then $L \in$ FPT.

In this work, we consider the problem of making a given directed multigraph Eulerian by adding arcs. A directed multigraph G is *Eulerian* if it is connected and each vertex is balanced. An *Eulerian extension* E for $G = (V, A)$ is a multiset over $V \times V$ such that $G' = (V, A \cup E)$ is Eulerian.

Eulerian Extension (EE)
Input: A directed multigraph $G = (V, A)$, an integer $\omega_{\max}$, and a weight function $\omega\colon V \times V \to [0, \omega_{\max}] \cup \{\infty\}$.
Question: Is there an Eulerian extension E of G whose weight is at most $\omega_{\max}$?

In the context of EE we speak of *allowed* arcs $a \in V \times V$, if $\omega(a) < \infty$.

2.1 Preprocessing Routines

A polynomial-time preprocessing for EE routine introduced by Dorn et al. [4] ensures that the balance of every vertex is in $\{-1, 0, 1\}$. This simplifies the problem and helps in constructions later on. Dorn et al. [4] showed that the corresponding transformation can be computed in $O(n(n + m))$ time. In the following, we assume that all input instances of EE have been transformed thusly, and hence, we assume that the following observation holds.

Observation 1. *Let v be a vertex in a pre-processed instance of* EE*. Then,* $\mathrm{balance}(v) \in \{-1, 0, 1\}$.

We use a second preprocessing routine to make further observations about trails in Eulerian extensions. This preprocessing is a variant of the algorithm used by Dorn et al. [4] to remove isolated vertices from the input graph. Basically, it replaces the weight of a vertex pair by the weight of the "lightest" path in the graph $(V, V \times V)$ with respect to ω. Note that the resulting weight function respects the triangle inequality. Dorn et al. [4] showed that this transformation can be computed in $O(n^3)$ time. In the following, we assume all input instances of EE to have gone through this transformation, and hence, we assume that the following holds.

Observation 2. *Let ω be a weight-function of a pre-processed instance of* EE. *Then, ω respects the triangle inequality, that is, for each x, y, z, it holds that $\omega(x, z) \leq \omega(x, y) + \omega(y, z)$.*

In the subsequent sections, we use this preprocessing in parameterized algorithms and reductions. To this end, note that both transformations are parameter-preserving, that is, they do not change the number of connected components.

The presented transformations lead to useful observations regarding trails in Eulerian extensions. For instance, we often need the following fact.

Observation 3. *For any Eulerian extension E of G, there is an Eulerian extension E' of at most the same weight such that any path p and any cycle c in E' does not visit a connected component of G twice, except for the initial and terminal vertex of p and c.*

2.2 Advice

Since Eulerian extensions have to balance every vertex, they contain paths starting in vertices with positive balance and ending in vertices with negative balance. These paths together with cycles have to connect all connected components of the input graph. In order to reduce the complexity of the problem, we use advice as additional information on the structure of optimal Eulerian extensions. Advice consists of hints which specify that there must be a path or cycle in an Eulerian extension that visits connected components in a distinct order. Hints however do not specify exactly which vertices these paths or cycles visit. For an example of advice, see Figure 1a.

Formally, a *hint* for a directed multigraph $G = (V, A)$ is an undirected path or cycle t of length at least one in the component graph $\mathbb{C}_G$ together with a flag determining whether t is a cycle or a path.[4] Depending on this flag, we call the hints *cycle hints* and *path hints*, respectively. We say that a set of hints H is an *advice* for the graph G if the hints are edge-disjoint.[5] For a trail t in G, $\mathbb{C}_G(t)$ is the trail in $\mathbb{C}_G$ that is obtained by making t undirected and, for every connected component C of G, substituting every maximum length subtrail t' of t with $V(t') \subseteq C$ by the vertex in $\mathbb{C}_G$ corresponding to C. We say that a path p in the graph $(V, V \times V)$ *realizes* a path hint h if $\mathbb{C}_G(p) = h$ and the initial vertex of p has positive balance and the terminal vertex has negative balance in G. We say that a cycle c in the graph $(V, V \times V)$ realizes a cycle hint h if $\mathbb{C}_G(c) = h$. We say that an Eulerian extension E *heeds the advice* H if it can be decomposed into a set of paths and cycles that realize all hints in H.

A topic in this work is how having an advice helps in solving an instance of Eulerian extension. In order to discuss this, we introduce the following version of EE.

Eulerian Extension with Advice (EEA)
Input: A directed multigraph $G = (V, A)$, an integer $\omega_{\max}$, a weight function $\omega\colon V \times V \to [0, \omega_{\max}] \cup \{\infty\}$, and advice H.
Question: Is there an Eulerian extension E of G that is of weight at most $\omega_{\max}$ and heeds the advice H?

[4] The flag is necessary because a hint to a path in $\mathbb{C}_G$ may correspond to a cycle in G.

[5] Note that there is a difference between advice in our sense and the notion of advice in computational complexity theory. There, an advice applies to every instance of a specific length.

We will see that the hard part of computing an Eulerian extension that heeds a given advice H is to choose initial and terminal vertices for path hints in H. In fact, it is possible to compute optimal realizations for all cycle hints in any given advice in $O(n^3)$ time.

Observation 4 ([16]). *Let $(G, \omega_{max}, \omega, H)$ be an instance of* EEA*. In $O(n^3)$ time we can compute an equivalent instance $(G', \omega_{max}, \omega, H')$ such that H' does not contain a cycle hint. Furthermore, the number of components at most decreases.*

In this regard we note that we can compute an optimal realization of a path hint for given endpoints in the corresponding directed multigraph. This is possible in quadratic time, mainly using Observation 2, forbidding arcs contained in one connected component, and Dijkstra's algorithm.

Observation 5. *Let $(G, \omega_{max}, \omega, H)$ be an instance of* EE*, let $h \in H$ be a path hint and let C_i, C_t be the connected components of G that correspond to the endpoints of h. Furthermore, let $(u, v) \in (C_i \times C_t) \cap (I_G^+ \times I_G^-)$. Then, we can compute a minimum-weight realization of h with initial vertex u and terminal vertex v in $O(n^2)$ time.*

Since we want to derive Eulerian extensions from an advice and every Eulerian extension for a graph connects all of the graphs connected components, we are mainly interested in "connecting" advice. We say that an advice for a directed multigraph G is *connecting*, if its hints connect all vertices in $\mathbb{C}_G$. Furthermore, if there is no connecting advice H' with $H' \subset H$, then H is called *minimal connecting* advice. We consider the following restricted version of EEA that allows only minimal connecting advice (note that, by Observation 4, we can assume the given advice to be cycle-free).

Eulerian Extension with Cycle-free Minimal Connecting Advice (EE∅CA)
Input: A directed multigraph $G = (V, A)$, an integer $\omega_{\max}$, a weight function $\omega\colon V \times V \to [0, \omega_{\max}] \cup \{\infty\}$, and minimal connecting cycle-free advice H.
Question: Is there an Eulerian extension E of G with weight at most $\omega_{\max}$ and heeding the advice H?

We can show that each minimal connecting cycle-free advice can be obtained from a forest in $\mathbb{C}_G$. Enumerating these forests allows us to generate all such advices for a given graph G with c connected components in $f(c) \cdot |G|^{O(1)}$ time, where f is some function only depending on c. Deferring the presentation of details to a long version of this work, we state that EE is parameterized Turing reducible to EE∅CA [16].

Lemma 1. EE *is $\leq_{\mathrm{T}}^{\mathrm{FPT}}$-reducible to* EE∅CA *in $16^{c \log(c)}|G|^{O(1)}$ time.*

3 Eulerian Extension and Conjoining Bipartite Matching

This section shows that Rural Postman (RP) is parameterized equivalent to a matching problem. By the parameterized equivalence of RP and Eulerian Extension (EE) given by Dorn et al. [4], we may concentrate on the equivalence of EE and matching instead.

First we introduce a variant of perfect bipartite matching. Let G be a bipartite graph, M be a matching of the vertices in G, and let P be a vertex partition with the cells $C_1, \ldots, C_k$. We call an unordered pair $\{i, j\}$ of integers $1 \leq i < j \leq k$ a *join* and a set J of such pairs a *join set* with respect to G and P. We say that a join $\{i, j\} \in J$ is *satisfied* by the matching M of G if there is at least one edge $e \in M$ with $e \cap C_i \neq \emptyset$ and $e \cap C_j \neq \emptyset$. We say that a matching M of G is J*-conjoining* with respect to a join set J if all joins in J are satisfied by M. If the join set is clear from the context, we simply say that M is conjoining.

CONJOINING BIPARTITE MATCHING (CBM)
Input: A bipartite graph $G = (V_1 \uplus V_2, E)$, an integer $\omega_{\max}$, a weight function $\omega \colon E \to [0, \omega_{\max}]$, a partition $P = \{C_1, \ldots, C_k\}$ of the vertices in G, and a join set J.
Question: Is there a matching M of the vertices of G such that M is perfect, M is conjoining and M has weight at most $\omega_{\max}$?

CBM can be interpreted as a job assignment problem with additional constraints: an assignment of workers to tasks is sought such that each worker is busy and each task is being processed. Furthermore, every worker must be qualified for his or her assigned task. Both the workers and the tasks are grouped and the additional constraints are of the form “At least one worker from group A must be assigned a task in group B”. An assignment that satisfies such additional constraints may be favorable in settings where the workers are assigned to projects and the projects demand at least one worker with additional qualifications.

Over the course of the following subsections, we prove the following theorem.

Theorem 1. CONJOINING BIPARTITE MATCHING *and* EULERIAN EXTENSION *are* $\leq_{\mathrm{T}}^{\mathrm{FPT}}$*-equivalent with respect to the parameters “join set size” and “number of connected components in the input graph.”*

The proof of Theorem 1 consists of four reductions, one of which is a parameterized Turing reduction. The other three reductions are polynomial-time polynomial-parameter many-one reductions.

It is easy to see that the equivalence of EE and RP given by Dorn et al. [4] also holds for the parameters “number of components” and “number of components in the graph induced by the required arcs.” Thus, we obtain the following from Theorem 1.

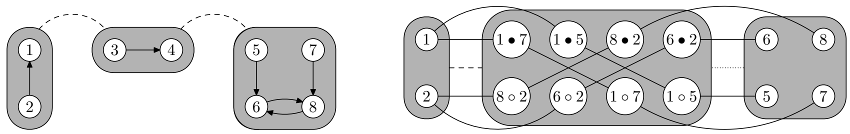
Theorem 2. CONJOINING BIPARTITE MATCHING *and* RURAL POSTMAN *are* $\leq_{\mathrm{T}}^{\mathrm{FPT}}$*-equivalent with respect to the parameters “number of components in the graph induced by the required arcs” and “join set size.”*

3.1 From Eulerian Extension to Matching

In this section we sketch a reduction from EE∅CA to CBM. By Lemma 1 this reduction leads to the following theorem.

Theorem 3. EULERIAN EXTENSION *is* $\leq_{\mathrm{m}}^{\mathrm{PPP}}$*-reducible to* CONJOINING BIPARTITE MATCHING *with respect to the parameters “number of components” and “join set size.”*

Outline of the Reduction. The basic idea of our reduction is to use vertices of positive balance and negative balance in an instance of EE∅CA as the two cells of the graph bipartition in a designated instance of CBM. Edges between vertices in the new instances



(a) EE∅CA instance (b) Long-hint gadget in CBM instance

Figure 1: Example of the long-hint gadget. In (a) an EE∅A-instance is shown, consisting of a graph with three connected components and an advice that contains a single path hint h (dashed lines). In (b) a part of an instance of CBM is shown, comprising the cells that correspond to the initial and terminal vertices of h and a gadget to model h. The gadget consists of new vertices put into a new cell which is connected by two joins (dashed and dotted lines) to the cells corresponding to the initial and terminal vertices of h.

represent shortest paths between these vertices that consist of allowed extension arcs in the original instance. Every connected component in the original instance is represented by a cell in the partition in the matching instance and hints are basically modeled by joins.

Description of the Reduction. For the description of the reduction, we need the following definition.

Definition 1. *Let $C_1, \dots, C_c$ be the connected components of a directed multigraph G, and let H be a cycle-free advice for G. For every $h \in H$ define* $\text{connect}(h) := \{i, j\}$, *where C_i, C_j are the components corresponding to the initial and terminal vertices of h.*

First, consider an EE∅CA-instance $(G, \omega_{\max}, \omega, H)$ such that H is a cycle-free minimal connecting advice that contains only hints of length one. We will deal with longer hints later. We create an instance I_{CBM} of CBM by first defining $B_0 = (I_G^+ \uplus I_G^-, E_0)$ as a bipartite graph. Here, the set E_0 consists of all edges $\{u, v\}$ such that $u \in I_G^+$, $v \in I_G^-$, and $\omega(u, v) < \infty$.[6] Second, we derive a vertex partition $\{V_1', \dots, V_c'\}$ of B_0 by intersecting the connected components of G with $(I_G^+ \uplus I_G^-)$. The vertex partition obviously models the connected components in the input graph, and the need for connecting them according to the advice H is modeled by an appropriate join-set J_0, defined as $\{\text{connect}(h) : h \in H\}$. Finally, we make sure that matchings also correspond to Eulerian extensions weight-wise, by defining the weight function $\omega'(\{u, v\})$ for every $u \in I_G^+, v \in I_G^-$ as $\omega(u, v)$ with $\omega'_{\max} = \omega_{\max}$.

By Observation 3 we may assume that every hint in H of length one is realized by a single arc. Since the advice connects all connected components, by the same observation, we may assume that all other trails in a valid Eulerian extension have length one. Finally, by Observation 1, we may assume that every vertex has at most one incident incoming or outgoing arc in the extension and, hence, we get an intuitive correspondence between conjoining matchings and Eulerian extensions.

To model hints of length at least two, we utilize gadgets similar to the one shown in Figure 1. The gadget comprises two vertices ($u \circ v$ and $u \bullet v$) for every pair (u, v) of vertices with one vertex in the component the hint starts and one in the component the hint ends. The vertices $u \circ v$ and $u \bullet v$ are adjacent and each of these two vertices is connected with

[6] This serves the purpose of modeling the structure of “allowed” arcs in the matching instance.

one vertex of the pair it represents. The edge $\{u \bullet v, u\}$ is weighted with the cost it takes to connect u, v with a path that realizes h. This cost is computed using Observation 5. The edges $\{u \bullet v, u \circ v\}$ and $\{u \circ v, v\}$ have weight 0. Intuitively these three edges in the gadget represent one concrete realization of h. If $u \circ v$ and $u \bullet v$ are matched, this means that this specific path does not occur in a designated Eulerian extension. However, by adding the vertices of the gadget as cell to the vertex partition and by extending the join set to the gadget, we enforce that there is at least one outgoing edge that is matched. If a perfect matching matches $u \circ v$ with u, then it also matches $u \bullet v$ with v and vice versa. This introduces an edge to the matching that has weight corresponding to a path that realizes h.

3.2 From Matching to Eulerian Extension with Advice

We reduce Conjoining Bipartite Matching to Eulerian Extension with Advice:

Theorem 4. Conjoining Bipartite Matching *is* $\leq_{\mathrm{m}}^{\mathrm{PPP}}$*-reducible to* Eulerian Extension with Advice *with respect to the parameters "join set size" and "connected components in the input graph".*

To reduce CBM to EEA we first observe that for every instance of CBM there is an equivalent instance such that every cell in the input vertex partition contains equal numbers of vertices from both cells of the graph bipartition. This observation enables us to model cells as connected components and vertices in the bipartite graph as unbalanced vertices in the designated instance of EEA.

Lemma 2. *For every instance of* CBM *there is an equivalent instance comprising the bipartite graph $G = (V_1 \uplus V_2, E)$, the vertex partition $P = \{C_1, \ldots, C_k\}$ and the join set J, such that*

(i) for every $1 \leq i \leq k$ it holds that $|V_1 \cap C_i| = |V_2 \cap C_i|$, and

(ii) the graph $(P, \{\{C_i, C_j\} : \{i, j\} \in J\})$ is connected.

This equivalent instance contains at most one cell more than the original instance.

Description of the Reduction. To reduce instances of CBM that conform to Lemma 2 to instances of EEA we use the simple idea of modeling every cell as connected component, vertices in V_1 as vertices with balance -1, vertices in V_2 as vertices with balance 1, and joins as hints.

Construction 1. Let the bipartite graph $B = (V_1 \uplus V_2, E)$, the weight function $\omega\colon E \to [0, \omega_{\max}]$, the vertex partition $P = \{C_1, \ldots, C_k\}$ and the join set J constitute an instance I_{CBM} of CBM that corresponds to Lemma 2. Let $v_1^1, v_1^2, \ldots, v_{n/2}^1, v_{n/2}^2$ be a sequence of all vertices chosen alternatingly from V_1 and V_2. Let the graph $G = (V, A) := (V_1 \cup V_2, A_1 \cup A_2)$ where the arc sets A_1 and A_2 are as follows: $A_1 := \{(v_i^1, v_i^2) : 1 \leq i \leq n/2\}$. For every $1 \leq j \leq k$ let $C_j = \{v_1, \ldots, v_{j_k}\}$, and let

$$A_2^j := \{(v_i, v_{i+1}) : 1 \leq i \leq j_k - 1\} \cup \{(v_{j_k}, v_1)\}$$

and define $A_2 := \bigcup_{j=1}^{k} A_2^j$. Define a new weight function ω' for every pair of vertices $(u, v) \in V \times V$ by

$$\omega'(u, v) := \begin{cases} \omega(\{u, v\}), & u \in V_2, v \in V_1, \{u, v\} \in E \\ \infty, & \text{otherwise.} \end{cases}$$

Finally, derive an advice H for G by adding a length-one hint h to H for every join $\{o, p\} \in J$ such that h consists of the edge that connects vertices in $\mathbb{C}_G$ that correspond to the connected components C_o, and C_p. The graph G, the weight function ω', the maximum weight $\omega_{\max}$ and the advice H constitute an instance of EEA.

Theorem 4 follows, since Construction 1 is a $\leq_m^{\mathrm{PPP}}$-reduction.

3.3 Removing Advice

For Theorem 1, it remains to show, that the advice in an instance of EEA created by Construction 1 can be removed. That is, it remains to show the following theorem.

Theorem 5. Eulerian Extension with Advice *is* $\leq_m^{\mathrm{PPP}}$*-reducible to* Eulerian Extension *with respect to the parameter "number of components in the input graph."*

The basic ideas for proving Theorem 5 are as follows. First, we remove every cycle-hint using Observation 4. We use the fact that every Eulerian extension has to connect all connected components of the input graph. Thus, for each hint h, we introduce a new connected component C_h. Let the components C_s, C_t correspond to the endpoints of hint h. To enforce that hint h is realized, we use the weight function to allow an arc from every vertex with balance 1 in C_s to a number of distinct vertices in C_h. This number is the number of vertices with balance -1 in C_t. That is, for every pair of unbalanced vertices in $C_s \times C_t$, we have an associated vertex in C_h. Then, for every inner vertex v on h, we copy C_h and connect it to the component corresponding to v. From one copy to another, using the weight function, we allow only arcs that start and end in vertices corresponding to the same pair of unbalanced vertices in $C_s \times C_t$. This enforces that every hint is realized and connects every component it visits. Using the weight function and Observation 5 we can ensure that the arcs corresponding to a realization of a hint have the weight of an optimal realization with the same endpoints. Using this construction, Theorem 5 can be proven which concludes the proof of Theorem 1.

4 Conjoining Bipartite Matching: Properties and Special Cases

This section investigates the properties of CBM introduced in Section 3. As discussed before, CBM might eventually help us derive a fixed-parameter algorithm for EE with respect to the parameter number of connected components. Section 4.1 first shows that also CBM is NP-complete. Section 4.2 then establishes tractability of the problem on restricted graph classes and translates this tractability result into the world of EE and RP.

4.1 NP-Hardness

NP-Hardness for Conjoining Bipartite Matching (CBM) does not follow from the parameterized equivalence to Eulerian Extension (EE) we gave in Section 3, since the reduction from EE we gave is a *parameterized* Turing reduction. To show that CBM is NP-hard, we polynomial-time many-one reduce from the well-known 3SAT, where a Boolean formula ϕ in 3-conjunctive normal form (3-CNF) is given and it is asked whether there is an assignment to the variables of ϕ that satisfies ϕ. Herein, a formula ϕ in 3-CNF is a conjunction of disjunctions of three literals each, where each literal is either x or $\neg x$ and x is a variable of ϕ. In the following, we represent each clause as three-element-set $\gamma \subseteq X \times \{+, -\}$, where $(x, +) \in \gamma$ means that x is contained in the clause represented by γ and $(x, -) \in \gamma$ means that $\neg x$ is contained in the clause represented by γ.

Construction 2. Let ϕ be a Boolean formula in 3-CNF with the variables $X := \{x_1, \dots, x_n\}$ and the clauses $\gamma_1, \dots, \gamma_m \subseteq X \times \{+, -\}$. We translate ϕ into an instance of CBM that is a yes-instance if and only if ϕ is satisfiable. To this end, for every variable x_i, introduce a cycle with $4m$ edges consisting of the vertex set $V_i := \{v_i^j : 1 \leq j \leq 4m\}$ and the edge set $E_i := \{e_i^k := \{v_i^k, v_i^{k+1}\} \subseteq V_i\} \cup \{e_i^{4m} := \{v_i^1, v_i^{4m}\}\}$. Let $G := (\bigcup_{i=1}^n V_i, \bigcup_{i=1}^n E_i)$, and let $\omega(e) := 0, e \in E_i$ for any $1 \leq i \leq n$, and define $\omega_{\max} := 1$. To construct an instance of CBM, it remains to find a suitable partition of the vertices of G and a join set.

Inductively define the vertex partition P_m of $V(G)$ and the join set J_m as follows: Let $J_0 = \emptyset$, and let $P_0 := \emptyset$. For every clause γ_j introduce the cell

$$C_j := \{v_i^{4j-1} : (x_i, +) \in \gamma_j \vee (x_i, -) \in \gamma_j\} \cup \{v_i^{4j-2} : (x_i, +) \in \gamma_j\} \cup \{v_i^{4j} : (x_i, -) \in \gamma_j\}.$$

Define $P_j := P_{j-1} \cup \{C_j\}$ and $J_j := J_{j-1} \cup \{\{0, j\}\}$.

Finally, define $C_0 := V(G) \setminus (\bigcup_{j=1}^m C_j)$. The graph G, the weight function ω, the vertex partition $P_m \cup \{C_0\}$ and the join set J_m constitute an instance of CBM.

Using this construction, we can prove the following theorem.

Theorem 6. CBM *is NP-complete, even in the unweighted case and when the input graph $G = (V \uplus W, E)$ has maximum degree two, and for every cell C_i in the given vertex partition of G it holds that $|C_i \cap V| = |C_i \cap W|$.*

Proof. CBM is contained in NP, because a perfect conjoining matching of weight at most $\omega_{\max}$ is a certificate for a yes-instance.

We prove that Construction 2 is a polynomial-time many-one reduction from 3SAT to CBM. Notice that in instances created by Construction 2 any matching has weight lower than $\omega_{\max}$ and, thus, the soundness of the reduction implies that CBM is hard even without the additional weight constraint. Also, since the cells in the instances of CBM are disjoint unions of edges, every cell in the partition P_m contains the same number of vertices from each cell of the graph bipartition.

It is easy to check that Construction 2 is polynomial-time computable. For the correctness we first need the following definition: For every variable $x_i \in X$ let

$$M_i^{\text{true}} := \{e_i^k \in E_i : k \text{ odd}\} \text{ and}$$
$$M_i^{\text{false}} := E_i \setminus M_i^{\text{true}} = \{e_i^k \in E_i : k \text{ even}\}.$$

Observe that all perfect matchings in G are of the form $\bigcup_{i=1}^{n} M_i^{\nu(x_i)}$, where ν is an assignment of truth values to variables in X. We show that the matching $\bigcup_{i=1}^{n} M_i^{\nu(x_i)}$ is a conjoining matching for G with respect to the join set J_m if and only if ν is satisfies ϕ. For this, it suffices to show that for every variable $x_i \in X$ it holds that

$$\{j : (x_i, +) \in \gamma_j\} = \{j : M_i^{\text{true}} \text{ satisfies the join } \{0, j\}\}, \text{ and} \quad (1)$$

$$\{j : (x_i, -) \in \gamma_j\} = \{j : M_i^{\text{false}} \text{ satisfies the join } \{0, j\}\}. \quad (2)$$

We only show that (1) holds; (2) can be proven analogously. Assume that $(x_i, +) \in \gamma_j$. By Construction 2 $v_i^{4j-2} \in C_j, v_i^{4j-3} \in C_0$ and thus, since

$$\{v_i^{4j-2}, v_i^{4j-3}\} = e^{4j-3} \in M_i^{\text{true}},$$

the matching M_i^{true} satisfies the join $\{0, j\}$. Now assume that $(x_i, +) \notin \gamma_j$, that is, either (1) $(x_i, \pm) \notin \gamma_j$ or (2) $(x_i, -) \in \gamma_j$. If $(x_i, \pm) \notin \gamma_j$, then V_i and C_j are disjoint and, thus, no matching in $G[V_i]$ can satisfy the join $\{0, j\}$. If $(x_i, -) \in \gamma_j$, then the only edges in E_i that can satisfy the join $\{0, j\}$ are e_i^{4j-2} and e_i^{4j}. Both edges are not in M_i^{true} and, thus, this matching cannot satisfy the join $\{0, j\}$. ◻

4.2 Tractability on Restricted Graph Classes

This section presents data reduction rules and employs them to sketch an algorithm for CBM on a restricted graph class, leading to the following theorem:

Theorem 7. Conjoining Bipartite Matching *can be solved in* $O(2^{j(j+1)}n + n^3)$ *time, where j is the size of the join set, provided that in the bipartite input graph $G = (V_1 \uplus V_2, E)$ each vertex in V_1 has maximum degree two.*

In this section, let $(G, \omega_{\max}, \omega, P = \{C_1, \ldots, C_c\}, J)$ be an instance of CBM, where in G is as in Theorem 7. The following lemma plays a central role in the proof of Theorem 7. It implies that, in a yes-instance, every component of G consists of an even-length cycle with a collection of pairwise vertex-disjoint paths incident to it.

Lemma 3. *If G has a perfect matching, then every connected component of G contains at most one cycle as subgraph.*

Proof. We show that if G contains a connected component that contains two cycles c_1, c_2 as subgraphs, then G does not have a perfect matching. First assume that c_1, c_2 are vertex-disjoint. Then, there is a path p from a vertex $v \in V(c_1)$ to a vertex $w \in V(c_2)$ such that p is vertex-disjoint from c_1 and c_2 except for v, w. It is clear that both $v, w \in V_2$ because they have degree three. Consider the vertices $V_1^{\text{cp}} := (V(c_1) \cup V(p) \cup V(c_2)) \cap V_1$ and the set $V_2^{\text{cp}} := (V(c_1) \cup V(p) \cup V(c_2)) \cap V_2$. The set V_2^{cp} is the set of neighbors of vertices in V_1^{cp}, because they have degree two and thus have neighbors only within p, c_1, and c_2. It is $|V_1^{\text{cp}}| = (|E(c_1)| + |E(p)| + |E(c_2)|)/2$ since neither of these paths and cycles overlap in a vertex in V_1. However, it is $|V_2^{\text{cp}}| = |V_1^{\text{cp}}| - 1$ because c_1 and p overlap in v and c_2 and p overlap in w. This is a violation of Hall's condition and thus G does not have a perfect matching.

The case where c_1 and c_2 share vertices can be proven analogously. (Observe that then there is a subpath of c_2 that is vertex-disjoint from c_1 and contains an even number of edges.) □

We now present four polynomial-time executable data reduction rules for CBM. The correctness of the first three rules is easy to verify, while the correctness of the fourth one is more technical and omitted. We note that all rules can be applied exhaustively in $O(n^3)$ time.

Reduction Rule 1 removes paths incident to the cycles of a graph G in a yes-instance. As a side-result, Reduction Rule 1 solves CBM in linear time on forests.

Reduction Rule 1. *If there is an edge* $\{v, w\} \in E(G)$ *such that* $\deg(v) = 1$*, then remove both* v *and* w *from* G*, and remove all joins* $\{i, j\}$ *from* J*, with* $v \in C_i, w \in C_j$*. Decrease* ω_{max} *by* $\omega(\{v, w\})$*.*

If exhaustively applying Reduction Rule 1 to G does not transform G such that each connected component is a cycle, which is checkable in linear time, then, by Lemma 3, G does not have a perfect matching and we can return “NO”. Hence, in the following, assume that each connected component of G is a cycle. Reduction Rule 2 now deletes connected components that cannot satisfy joins.

Reduction Rule 2. *If there is a connected component* D *of* G *such that it contains no edge that could satisfy any join in* J*, then compute a minimum-weight perfect matching* M *in* $G[D]$*, remove* D *from* G *and decrease* ω_{max} *by* $\omega(M)$*.*

After exhaustively applying Reduction Rule 2, we may assume that each connected component of G contains an edge that could satisfy a join. We next present a data reduction rule that removes joins that are always satisfied. To this end, we need the following definition.

Definition 2. *For each connected component* D *(that is, each cycle) in* G*, denote by* $M_1(D)$ *a minimum-weight perfect matching of* D *with respect to* ω *and denote by* $M_2(D) := E(D) \setminus M_1(D)$ *the other perfect matching of* D*.*[7] *Furthermore, denote*

$$\sigma_1(D) := \{j \in J : \exists e \in M_1(D) : e \text{ satisfies } j\},$$
$$\sigma_2(D) := \{j \in J : \exists e \in M_2(D) : e \text{ satisfies } j\},$$

and the signature $\sigma(D)$ *of* D *as* $\{\sigma_1(D), \sigma_2(D)\}$*.*

Reduction Rule 3. *Let* D *be a connected component of* G*. If there is a join* $j \in \sigma_1(D) \cap \sigma_2(D)$*, then remove* j *from* J*.*

A final data reduction rule removes connected components that satisfy the same joins.

Reduction Rule 4. *Let* $S = \{D_1, \ldots, D_j\}$ *be a maximal set of connected components of* G *such that* $\sigma(D_1) = \ldots = \sigma(D_j)$ *and* $j \geq 2$*. Let* $M_1^* = \bigcup_{k=1}^{j} M_1(D_k)$*, let* $D_l \in S$ *such that* $\omega(M_2(D_l)) - \omega(M_1(D_l))$ *is minimum, and let* $M_1^\sim = M_1^* \setminus M_1(D_l)$*.*

[7] Note that in bipartite graphs every cycle is of even length.

(i) If the matching M_1^* is conjoining for the join set $\sigma_1(D_1) \cup \sigma_2(D_1)$, then remove each component in S from G, remove each join in $\sigma_1(D_1) \cup \sigma_2(D_1)$ from the join set J, and reduce ω_{max} by $\omega(M_1^*)$.
(ii) If the matching M_1^* is not conjoining for the join set $\sigma_1(D_1) \cup \sigma_2(D_1)$, then remove each component in $S \setminus \{D_l\}$ from G, remove any join in $\sigma_1(D_1)$ from the join set J, and reduce ω_{max} by $\omega(M_1^{\sim})$.

In either case, update the partition P accordingly.

Observation 6. *If Reduction Rule 4 is not applicable to G, then G contains at most one connected component for each of the $2^{|J|+1}$ possible signatures.*

Now, Theorem 7 follows: Exploiting Observation 6, a search-tree algorithm solving CBM can in $O(n)$ time choose a join $j \in J$ and choose one of the at most $2^{|J|+1}$ connected components of the graph that can satisfy j and match the component accordingly. Then, the algorithm can recurse on how the remaining $|J| - 1$ joins are satisfied.

Analyzing the pre-images that lead to tractable instances of CBM under the reductions we gave in Section 3, Theorem 7 can be translated to a tractability result for EE. A similar tractability result can also be shown for Rural Postman. Due to its length, we only state it for EE here.

Corollary 1. *Let the graph G and the weight function ω constitute an instance I_{EE} of* EE. *Let c be the number of connected components in G.*

(i) If every path or cycle in the set of allowed arcs w.r.t. ω has length at most one,
(ii) if G contains only vertices with balance between -1 and 1,
(iii) if every vertex in I_G^+ (every vertex in I_G^-) has only outgoing allowed arcs (incoming allowed arcs), and
(iv) if in every connected component C of G, either all vertices in $I_G^+ \cap C$ or in $I_G^- \cap C$ have at most two incident allowed arcs,

then it is decidable in $\mathrm{O}(2^{c(c+\log(2c^4))}(n^4 + m))$ time whether I_{EE} is a yes-instance.

5 Conclusion

Clearly, the most important remaining open question is to determine whether Rural Postman is fixed-parameter tractable with respect to the number of connected components of the graph induced by the required arcs. This question also extends to the presumably harder undirected case. The newly introduced Conjoining Bipartite Matching (CBM) problem might also be useful in spotting new, computationally feasible special cases of Rural Postman and Eulerian Extension. The development of polynomial-time approximation algorithms for CBM or the investigation of other (structural) parameterizations for CBM seem worthwhile challenges as well. Finally, we remark that previous work [10, 4] also left open a number of interesting open problems referring to variants of Eulerian Extension. Due to the practical relevance of the considered problems, our work is also meant to further stimulate more research on these challenging combinatorial problems.

References

[1] A. A. Assad and B. L. Golden. Arc routing methods and applications. In *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, pages 375–483. Elsevier B. V., 1995.

[2] E. Benavent, A. Corberán, E. Piñana, I. Plana, and J. M. Sanchis. New heuristic algorithms for the windy rural postman problem. *Comput. Oper. Res.*, 32(12): 3111–3128, 2005.

[3] 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.

[4] F. Dorn, H. Moser, R. Niedermeier, and M. Weller. Efficient algorithms for Eulerian extension. In *Proc. 36th WG*, volume 6410 of *LNCS*, pages 100–111. Springer, 2010.

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

[6] M. Dror. *Arc Routing: Theory, Solutions, and Applications*. Kluwer Academic Publishers, 2000.

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

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

[10] W. Höhn, T. Jacobs, and N. Megow. On Eulerian extensions and their application to no-wait flowshop scheduling. *J. Sched*, 2011. To appear.

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

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

[13] C. S. Orloff. A fundamental problem in vehicle routing. *Networks*, 4(1):35–64, 1974.

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

[15] N. Perrier, A. Langevin, and J. F. Campbell. A survey of models and algorithms for winter road maintenance. Part IV: Vehicle routing and fleet sizing for plowing and snow disposal. *Comput. Oper. Res.*, 34(1):258–294, 2007.

[16] M. Sorge. On making directed graphs Eulerian. Diplomarbeit, Institut für Informatik, Friedrich-Schiller-Universität Jena, 2011. Available electronically. arXiv:1101.4283 [cs.DM].