Abstract. In the Mixed Chinese Postman Problem (MCPP), given a weighted mixed graph $G$ ($G$ may have both edges and arcs), our aim is to find a minimum weight closed walk traversing each edge and arc at least once. The MCPP parameterized by the number of edges in $G$ or the number of arcs in $G$ is fixed-parameter tractable as proved by van Bevern et al. (in press) and Gutin, Jones and Sheng (ESA 2014), respectively. Solving an open question of van Bevern et al. (in press), we show that unexpectedly the MCPP parameterized by the treewidth of $G$ is $W[1]$-hard. In fact, we prove that even the MCPP parameterized by the pathwidth of $G$ is $W[1]$-hard.

1 Introduction

A mixed graph is a graph that may contain both edges and arcs (i.e., directed edges). A mixed graph $G$ is strongly connected if for each ordered pair $x, y$ of vertices in $G$ there is a path from $x$ to $y$ that traverses each arc in its direction.

In this paper, we study the following well-known problem.

| **MIXED CHINESE POSTMAN PROBLEM (MCPP)** |
| **Instance:** | A strongly connected mixed graph $G = (V, E \cup A)$, with vertex set $V$, set $E$ of edges and set $A$ of arcs; a weight function $w : E \cup A \rightarrow \mathbb{N}_0$. |
| **Output:** | A closed walk of $G$ that traverses each edge and arc at least once, of minimum weight. |

There is numerous literature on various algorithms and heuristics for MCPP; for informative surveys, see [2, 4, 11, 15]. When $A = \emptyset$, we call the problem the Undirected Chinese Postman Problem (UCPP), and when $E = \emptyset$, we call the problem the Directed Chinese Postman Problem (DCPP). It is well-known that UCPP is polynomial-time solvable [6] and so is DCPP [13, 6], but MCPP is NP-complete, even when $G$ is planar with each vertex having total degree 3 and all edges and arcs having weight 1 [13]. It is therefore reasonable to believe that MCPP may become easier the closer it gets to UCPP or DCPP and indeed when parameterized by the number of edges in $G$ or the number of arcs in $G$, MCPP is proved to be fixed-parameter tractable by van Bevern et al.

1 For an excellent introduction to parameterized algorithms and complexity, see [5].
al. [15] and Gutin, Jones and Sheng [12], respectively. van Bevern et al. [15] noted that Fernandes, Lee and Wakabayashi [10] proved that MCPP parameterized by the treewidth of $G$ is in XP (when all edges and arcs have weight 1), and asked whether this parameterisation of MCPP is fixed-parameter tractable.

It is well-known that very many graph problems are fixed-parameter tractable when parameterized by the treewidth of the input graph (there are only a few graph problems which are W[1]-hard, see, e.g., [4,8,11]). In this paper, we show that unexpectedly the MCPP parameterized by treewidth belongs to the minority of problems, i.e., it is W[1]-hard. In fact, we prove a stronger result by (i) replacing treewidth with pathwidth, and (ii) assuming that all edges and arcs have weight 1.

A directed multigraph is called balanced if the in-degree of each vertex coincides with its out-degree. To show the W[1]-hardness of MCPP parameterized by pathwidth, we introduce an intermediate problem, MINIMUM BALANCED SUBGRAPH WITH DOUBLE ARCS (MBSDA). This is a problem in which, given a directed multigraph $D$ with integer-weighted arcs, the aim is to find a balanced subgraph of minimum weight. The complication is that we will include ‘double arcs’, which are pairs of arcs with the same start and end vertices, such that if we use one arc of the pair in our subgraph we must use the other.

In the next section we show that even a quite restricted subproblem of MBSDA is W[1]-hard when parameterized by the pathwidth of $D$. Our reduction is from the well-known $k$-MULTICOLORED CLIQUE problem [5]. In Section 3 we reduce the restricted subproblem of MBSDA into MCPP parameterized by pathwidth. We conclude the paper with Section 4.

## 2 Minimum Balanced Subgraph with Double Arcs

**Minimum Balanced Subgraph with Double Arcs (MBSDA)**

*Instance:* A directed multigraph $D = (V, A)$, a weight function $w : A \to \mathbb{Z}$, and a set $X = \{(a_1, a'_1), \ldots, (a_r, a'_r)\}$ of disjoint pairs of arcs (called double arcs), such that $a_i, a'_i \in A$ and $a_i, a'_i$ have the same start vertex and end vertex, for each $(a_i, a'_i) \in X$.

*Output:* A balanced subgraph $D'$ of $D$ such that $|A(D') \cap \{a_i, a'_i\}| \neq 1$ for each $(a_i, a'_i) \in X$, of minimum weight.

Note that we allow arcs to have negative weights or weight 0. As a graph with no arcs is balanced, MBSDA always has a solution of weight at most 0.

We will say that a subgraph $D'$ of $D$ is properly balanced if $D'$ is balanced (i.e. every vertex has equal in- and out-degree) and $|A(D) \cap \{a_i, a'_i\}| \neq 1$ for every double arc $(a_i, a'_i)$.

### 2.1 Gadgets for MBSDA

We now describe some simple gadget graphs (for now we do not assign weights; we will do this later). Each gadget will have some number of input and output arcs.
Later, we will combine these gadgets by joining the input and output arcs of different gadgets together.

A Duplication gadget has one input arc and \( t \) output arcs, for some positive integer \( t \). The vertex set consists of vertices \( x, y, u_i, v_i \) for each \( i \in [t] \). The arcs form a cycle \( xyu_1v_1 \ldots u_tv_x \). The input arc is the arc \( xy \), and the output arcs are the arcs \( u_iv_i \) for each \( i \in [t] \).

As any balanced subgraph of a duplication gadget is either an empty graph or the whole gadget, we have the following proposition:

**Proposition 1.** Let \( D \) be a Duplication gadget with \( t \) output arcs, and \( D' \) a properly balanced subgraph of \( D \). Then \( D' \) satisfies one of two possibilities:

- \( D' \) contains the input arc and every output arc;
- \( D' \) does not contain the input arc or any output arcs.

Furthermore, for each possibility there exists a properly balanced subgraph of \( D \) that satisfies it.

A Choice gadget has one input arc \( xy \) and \( t \) output arcs \( u_iv_i : i \in [t] \), for some positive integer \( t \). The vertex set consists of the vertices \( x, y, z, w \) and \( u_i, v_i \) for each \( i \in [t] \). The arcs consist of a path \( wxyz \), and the path \( zu_iv_iw \) for each \( i \in [t] \). (See Figure 1)

**Proposition 2.** Let \( D \) be a Choice gadget with \( t \) output arcs and \( D' \) a (properly) balanced subgraph of \( D \). Then \( D' \) satisfies one of \( t + 1 \) possibilities:

- \( D' \) contains the input arc, the \( i \)th output arc and no other output arcs (for each \( i \in [t] \));
- \( D' \) does not contain the input arc or any output arcs.
Furthermore, for each possibility there exists a properly balanced subgraph of $D$ that satisfies it.

![Diagram of a Checksum gadget](image)

**Fig. 2.** A Checksum gadget with $t_l$ left input arcs and $t_r$ right input arcs. A balanced subgraph will contain the same number of left and right input arcs.

Finally, a Checksum gadget has $t_l$ **left input** arcs $x_iy_i : i \in [t_l]$ for some positive integer $t_l$, and $t_r$ **right input** arcs $u_iv_i : i \in [t_r]$, and no output arcs. The vertex set consists of the vertices $w, z$ together with $x_i, y_i$ for each $i \in [t_l]$ and $u_i, v_i$ for each $i \in [t_r]$. The arc set consists of the path $wx_iy_iz$ for each $i \in [t_l]$, and $zu_iv_iw$ for each $i \in [t_r]$. (See Figure 2)

**Proposition 3.** Let $D$ be a Checksum gadget with $t_l$ left input arcs and $t_r$ right input arcs, and $D'$ a (properly) balanced subgraph of $D$.

Then there exist a set $L$ of left input arcs and $R$ of right input arcs, such that $|L| = |R|$, and $D'$ contains the input arcs of $L$ and $R$ and no others.

Furthermore, for any set $L$ of left input arcs and $R$ of right input arcs such that $|L| = |R|$, there exists a properly balanced subgraph of $D$ that contains the input arcs of $L$ and $R$ and no others.

**Proof.** For a left input arc $x_iy_i$, observe that $x_iy_i \in A(D')$ if and only if $y_i,z \in A(D')$. Similarly, for each right input arc $u_iv_i$, observe that $u_iv_i \in A(D')$ if and only if $zu_i \in A(D')$. As $z$ is balanced in $D'$, the number of left input arcs in $D'$ and the number of right input arcs in $D'$ must be the same.

For the converse, given any set $L$ of left input arcs and $R$ of right input arcs such that $|L| = |R|$, let $D'$ be the graph containing $L$, all arcs from $L$ to $z$, all arcs from $z$ to $R$, $R$, all arcs from $R$ to $w$, and all arcs from $w$ to $L$. Then $D'$ is a properly balanced subgraph of $D'$. 

\[\square\]
Observe that in all of our gadgets, the vertices in input or output arcs all have in-degree and out-degree 1.

We next describe how to combine these gadgets. For two unjoined arcs \( uv \) and \( xy \) (possibly in disjoint graphs), the operation of joining \( uv \) and \( xy \) is as follows: Identify \( u \) and \( x \), and identify \( v \) and \( y \). Keep both \( uv \) and \( xy \), and add \( \{uv, xy\} \) as a double arc.

The following technical lemma will be useful for showing the correctness of our constructions:

**Lemma 1.** Let \( D_1 \) and \( D_2 \) be disjoint graphs. Let \( u_1v_1, \ldots, u_kv_k \) be arcs in \( D_1 \), and let \( x_1y_1, \ldots, x_ly_l \) be arcs in \( D_2 \), such that \( u_i \) and \( v_i \) both have in-degree and out-degree 1 in \( D_1 \), and \( x_i \) and \( y_i \) both have in-degree and out-degree 1 in \( D_2 \), for each \( i \in [t] \). Let \( D \) be the graph formed by joining \( u_iv_i \) and \( x_iy_i \), for each \( i \in [t] \).

Then a subgraph \( D' \) of \( D \) is a properly balanced subgraph of \( D \) if and only if:

1. \( |A(D') \cap \{u_iv_i, x_iy_i\}| \neq 1 \) for each \( i \in [t] \); and
2. \( D' \) restricted to \( D_1 \) is a properly balanced subgraph of \( D_1 \), and \( D' \) restricted to \( D_2 \) is a properly balanced subgraph of \( D_2 \).

**Proof.** Suppose first that \( D' \) is a properly balanced subgraph of \( D \). Then by definition and the fact that \( \{u_iv_i, x_iy_i\} \) is a double arc in \( D \), it must be the case that \( |A(D') \cap \{u_iv_i, x_iy_i\}| \neq 1 \) for each \( i \in [t] \). Now let \( D'_1 \) be \( D' \) restricted to \( D_1 \). We will show that \( D'_1 \) is a properly balanced subgraph of \( D_1 \). First observe that for every double arc \( \{a, a'\} \) in \( D_1 \), \( |A(D'_1) \cap \{a, a'\}| \neq 1 \). Secondly, observe that every vertex in \( V(D_1) \setminus \bigcup_{i \in [t]} \{u_i, v_i\} \) has the same in-degree and out-degree in \( D'_1 \) as in \( D \), and therefore the in- and out-degree of this vertex is the same. It remains to show that \( u_i \) and \( v_i \) have the same in-degree and out-degree in \( D'_1 

For each \( i \in [t] \), the vertex \( v_i \) has two out-arcs in \( D \) (one from \( D_1 \) and one from \( D_2 \)), and two in-arcs, which form the double arc \( \{u_iv_i, x_iy_i\} \). Therefore as \( D' \) is a properly balanced subgraph of \( D \), \( D' \) either contains both of \( u_iv_i, x_iy_i \) and both out-arcs of \( v_i \), or neither of \( u_iv_i, x_iy_i \) and neither of the out-arcs of \( v_i \). It follows that the out-arc of \( v_i \) in \( D_1 \) appears in \( D' \) if and only if the arc \( uv \) appears in \( D' \). Therefore \( D'_1 \) either has in-degree and out-degree both 0 in \( D'_1 \), or both 1. A similar argument holds for \( u_i \). This completes the proof that \( D'_1 \) is a properly balanced subgraph of \( D_1 \). A similar argument shows that \( D' \) restricted to \( D_2 \) is a properly balanced subgraph of \( D_2 \).

Conversely, suppose that \( |A(D') \cap \{uv, xy\}| \neq 1 \), \( D' \) restricted to \( D_1 \) is a properly balanced subgraph of \( D_1 \), and \( D' \) restricted to \( D_2 \) is a properly balanced subgraph of \( D_2 \). Then by construction \( |A(D) \cap \{a, a'\}| \neq 1 \) for every double arc \( \{a, a'\} \) in \( D \). As \( D_1, D_2 \) partition the arcs of \( D \), and \( D' \) is balanced when restricted to either of these graphs, we have that \( D' \) is balanced. Thus, \( D' \) is a properly balanced subgraph of \( D \), as required.

The next lemma will be useful for bounding the pathwidth of our constructions:
Lemma 2. Let $D', D_1, D_2, \ldots, D_l$ be disjoint graphs, let $a_1, \ldots, a_l$ be distinct arcs in $D'$, and let $D$ be a graph formed by joining the arc $a_i$ to an arc in $D_i$, for each $i \in [l]$. Then $pw(D) \leq pw(D') + \max_i(pw(D_i)) + 1$.

Proof. Consider a minimum width path decomposition of $D'$. For each $i \in [l]$, let $x_i$ be the bag in the path decomposition of $D'$ that contains both vertices of $a_i$ (if there is a choice of bags, let $x_i$ be the bag of smallest size). Now replace $x_i$ with two identical bags $x'_i, x''_i$, and in between $x'_i$ and $x''_i$ add a sequence of bags formed by taking a minimum width path decomposition of $D_i$ and adding all the vertices of $x_i$ to each bag. Do this for each $i \in [l]$. The resulting decomposition is a path decomposition of $D$. By construction and by choice of $x_i$, the width of this decomposition is at most $pw(D') + \max_i(pw(D_i)) + 1$. \qed

Observe that a Duplication gadget can be turned into a path by the removal of one vertex, and Choice and Checksum gadgets can be turned into disjoint unions of paths by the removal of two vertices. Therefore Duplication gadgets have pathwidth 2, and Choice and Checksum gadgets have pathwidth 3.

2.2 W[1]-hardness of MBSDA

We will show that MBSDA is W[1]-hard, by a reduction from the following problem:

| $k$-MULTICOLORED CLIQUE |
|---------------------------|
| **Instance:** A graph $G = (V_1 \cup V_2 \cdots \cup V_k, E)$, such that for each $i \in [k]$, $V_i$ forms an independent set. |
| **Parameter:** $k$ |
| **Question:** Does $G$ contain a clique with $k$ vertices? |

Theorem 1. [9] $k$-MULTICOLORED CLIQUE is W[1]-hard.

Our reduction is similar in structure to that of Dom et al. [4], although the details are rather different. We produce a large graph of constant pathwidth that represents a choice of one vertex $v_i$ from each class $V_i$. In order to ensure that these vertices form a clique, the graph also requires that for each chosen vertex $v_i \in V_i$, we choose an edge between $v_i$ and $V_j$, for each $j \neq i$. Finally, we add $O(k^2)$ vertices to check that for each $i \neq j$, the choice of edge between $v_i$ and $V_j$ is the same as the choice of edge between $v_j$ and $V_i$.

Theorem 2. MBSDA is W[1]-hard parameterized by pathwidth, even under the following restrictions:

- There exists a single arc $a^*$ of weight −1;
- $a^*$ is not part of a double arc;
- All other arcs have weight 0.
In the rest of this section, we will prove Theorem 2. We give a reduction from $k$-MULTICOLORED CLIQUE.

Given an instance $G = (V_1 \cup V_2 \cdots \cup V_k, E)$, let $e_1, \ldots, e_m$ be an arbitrary numbering of the edges of $E$. For each unordered pair $\{i, j\} \subseteq [k]$ with $i \neq j$, let $E_{\{i,j\}}$ be the subset of edges in $E$ with one vertex in $V_i$ and the other in $V_j$. Note that any $k$-clique in $G$ will have exactly one edge from $E_{\{i,j\}}$ for each choice of $i, j$.

The structure of our MBSDA instance will force us to choose a vertex $v_i$ from each class $V_i$, corresponding to the vertices of a $k$-clique. In addition, for each chosen vertex $v_i$ and each $j \neq i$, we choose an edge $e_{i \to j}$ between $v_i$ and $V_j$. A set of $O(k^2)$ Checksum gadgets will ensure that for each $i \neq j$, the chosen edges $e_{i \to j}$ and $e_{j \to i}$ must be the same. This ensures that $v_i$ and $v_j$ are adjacent for each $i \neq j$, and that therefore the vertices $v_1, \ldots, v_k$ form a clique.

We build our MBSDA instance $(D, w, X)$ out of Duplication, Choice and Checksum gadgets, as follows.

Let START be a Duplication gadget with input arc $a^*$, and $k$ output arcs. Label each output arc with a different integer $i$ from $[k]$.

For each $i \in [k]$, let CHOOSEVERTEX($i$) be a Choice gadget with $|V_i|$ output arcs. Label each output arc with a different vertex $v$ from $V_i$. Join the input arc of CHOOSEVERTEX($i$) to the output arc of START with label $i$.

For each $i \in [k], v \in V_i$, let ASSIGNVERTEX($i, v$) be a Duplication gadget with $k - 1$ output arcs. Label the output arcs with the integers from $[k] \setminus \{i\}$. Join the input arc of ASSIGNVERTEX($i, v$) to the output arc of CHOOSEVERTEX($i$) with label $v$.

For each $i \in [k], v \in V_i, j \in [k] \setminus \{i\}$ let CHOOSEEDGE($i, v, \rightarrow j$) be a Choice gadget with $|N(v) \cap V_j|$ output arcs. Label each output arc with a different edge $e_{r}$ between $v$ and $V_j$. Join the input arc of CHOOSEEDGE($i, v, \rightarrow j$) to the output arc of ASSIGNVERTEX($i, v$) with label $j$.

For each $i \in [k], v \in V_i, j \in [k] \setminus \{i\}$ and edge $e_r$ between $v_i$ and $V_j$, let ASSIGNEDGE($i, v, \rightarrow j, e_r$) be a Duplication gadget with $r$ output arcs. Label this whole set of output arcs as OUTPUT($i, v, \rightarrow j, e_r$). Join the input arc of ASSIGNEDGE($i, v, \rightarrow j, e_r$) to the output arc of CHOOSEEDGE($i, v, \rightarrow j$) with label $e_r$.

Finally, for each $i, j \in [k]$ with $i < j$, let CHECKEDGE($i, j$) be a Checksum gadget with $\sum\{r : e_r \in E_{\{i,j\}}\}$ left input arcs and $\sum\{r : e_r \in E_{\{i,j\}}\}$ right input arcs. Partition the left and right input arcs of CHECKEDGE($i, j$) as follows. For each $e_r \in E_{\{i,j\}}$, let INPUT($i, v, \rightarrow j, e_r$) be a set of $r$ left input arcs, where $v$ is the endpoint of $e_r$ in $V_i$. Similarly, let INPUT($j, u, \rightarrow i, e_r$) be a set of $r$ right input arcs, where $u$ is the endpoint of $e_r$ in $V_j$. Now, join each set of arcs of the form INPUT($i, v, \rightarrow j, e_r$) to the set of arcs of the form OUTPUT($i, v, \rightarrow j, e_r$) from the gadget ASSIGNEDGE($i, v, \rightarrow j, e_r$).

Finally, we assign weights. Let $a^*$ have weight $-1$ and let all other arcs have weight $0$. This concludes the construction of $D$. Observe that every output arc is joined to an input arc, and every input arc except $a^*$ is joined to an output arc.
Correctness: We now show that $D$ has a properly balanced subgraph of negative weight if and only if $G$ has a clique with $k$ vertices.

Observe that by repeated use of Lemma 1 a subgraph $D'$ of $D$ is a properly balanced subgraph if and only if

- $D'$ restricted to any gadget $\text{START}$, $\text{CHOOSEVERTEX}(i)$, $\text{ASSIGNVERTEX}(i, v)$, $\text{CHOOSEEDGE}(i, v, \rightarrow j)$, $\text{ASSIGNEDGE}(i, v, \rightarrow j, e_r)$ or $\text{CHECKEDGE}(i, j)$ is a properly balanced subgraph; and
- for each output arc $a$ that is joined to an input arc $a'$, $a$ is in $D'$ if and only if $a'$ is in $D'$.

First suppose $G$ has a clique on $k$ vertices. By definition of $G$, this clique must have exactly one vertex from each class $V_i$, $i \in [k]$. For each $i \in [k]$, let $v_i$ be the vertex of $V_i$ that is in the clique. For each $i \neq j$, let $r(i, j)$ be the index such that $e_r(i, j)$ is the edge between $v_i$ and $v_j$.

We will now describe a graph $D'$ by describing its restriction to each gadget. The construction will be such that an output arc is in $D'$ if and only if the input arc it is joined to is also in $D'$.

Let $D'$ restricted to $\text{START}$ be a properly balanced subgraph containing $a^*$ and each of the output arcs of $\text{START}$. By Proposition 1 such a subgraph exists.

For each $i \in [k]$, let $D'$ restricted to $\text{CHOOSEVERTEX}(i)$ be a properly balanced subgraph containing the input arc and the output arc labelled with $v_i$, and no other output arcs. Such a subgraph exists by Proposition 2.

For each $i \in [k], v \in V_i$, if $v = v_i$ then let $D'$ restricted to $\text{ASSIGNVERTEX}(i, v)$ be a properly balanced subgraph containing the input arc and all of the output arcs. Otherwise, let $D'$ restricted to $\text{ASSIGNVERTEX}(i, v)$ be a properly balanced subgraph that does not contain the input arc or any output arcs. In either case, such a subgraph exists by Proposition 1.

For each $i \in [k], v \in V_i, j \in [k] \setminus \{i\}$, if $v = v_i$ let $D'$ restricted to $\text{CHOOSEEDGE}(i, v, \rightarrow j)$ be a properly balanced subgraph containing the input arc and the output arc labelled with $e_{r(i, j)}$, and no other output arcs. Otherwise, let $D'$ restricted to $\text{CHOOSEEDGE}(i, v, \rightarrow j)$ be a properly balanced subgraph that does not contain the input arc or any output arcs. In either case, such a subgraph exists by Proposition 2.

For each $i \in [k], v \in V_i, j \in [k] \setminus \{i\}$ and edge $e_r$ between $v_i$ and $V_j$, if $v = v_i$ and $r = r(i, j)$, let $D'$ restricted to $\text{ASSIGNEDGE}(i, v, \rightarrow j, e_r)$ be a properly balanced subgraph containing the input arc and all output arcs. Otherwise, let $D'$ restricted to $\text{ASSIGNEDGE}(i, v, \rightarrow j, e_r)$ be a properly balanced subgraph that does not contain the input arc or any output arcs. In either case, such a subgraph exists by Proposition 2.

Note at this point that $D'$ contains all arcs of the form $\text{OUTPUT}(i, v, \rightarrow j, e_r)$ if $v = v_i$ and $r = r(i, j)$, and otherwise $D'$ contains no arcs from $\text{OUTPUT}(i, v, \rightarrow j, e_r)$. Finally, for each $i, j \in [k]$ with $i < j$, let $D'$ restricted to $\text{CHECKEDGE}(i, j)$ be a properly balanced subgraph containing the left input arcs from $\text{INPUT}(i, v_i, \rightarrow j, e_r(i, j))$, the right input arcs from $\text{INPUT}(j, v_j, \rightarrow i, e_r(j, i))$, and no other input arcs. As $r(i, j) = r(j, i)$, such a subgraph exists by Proposition 3.
This concludes the construction of $D'$. As $D'$ restricted to each gadget is a properly balanced subgraph, and an output arc is in $D'$ if and only if the input arc it is joined to is in $D'$, we have that $D'$ is a properly balanced subgraph of $D$. As $D'$ contains the arc $a^*$ of weight $-1$ and all other arcs have weight $0$, $D'$ is a properly balanced subgraph with negative weight, as required.

Now for the converse, suppose that $D$ has a properly balanced subgraph $D'$ of negative weight. Then $D'$ must contain $a^*$, the input arc of $START$ with weight $-1$. By Proposition $1$ $D'$ must contain all of the output arcs of $START$.

It follows that for each $i \in [k]$, $D'$ contains the input arc of $CHOOSE_VERTEX(i)$. By Proposition $2$ $D'$ contains exactly one output arc of $CHOOSE_VERTEX(i)$. So for each $i \in [k]$, let $v_i \in V_i$ be the unique vertex in $G$ such that $D'$ restricted to $CHOOSE_VERTEX(i)$ contains the output arc labelled with $v_i$.

It now follows that for each $i \in [k], v \in V_i$, $D'$ contains the input arc of $ASSIGN_VERTEX(i,v)$ if and only if $v = v_i$. Then by Proposition $1$ if $v = v_i$ then $D'$ contains the all the output arcs of $ASSIGN_VERTEX(i,v)$, and otherwise $D'$ contains none of the output arcs of $ASSIGN_VERTEX(i,v)$.

It follows that for each $i \in [k], v \in V_i, j \in [k] \setminus \{i\}$, $D'$ contains the input arc of $CHOOSE_EDGE(i,v,\rightarrow j)$ if and only if $v = v_i$. If $v \neq v_i$ then by Proposition $2$ $D'$ contains none of the output arcs of $CHOOSE_EDGE(i,v,\rightarrow j)$.

So for each $i \in [k], v \in V_i, j \in [k] \setminus \{i\}$, let $r(i \rightarrow j)$ be the index such that $D'$ contains the output arc of $CHOOSE_EDGE(i,v_i,\rightarrow j)$ labelled with $e_{r(i\rightarrow j)}$. (Later we will show that $r(i \rightarrow j) = r(j \rightarrow i)$, implying that $v_i$ and $v_j$ are adjacent.)

It now follows that for each $i \in [k], v \in V_i, j \in [k] \setminus \{i\}$ and edge $e_r$ between $v_i$ and $V_j$, $D'$ contains the input arc of $ASSIGN_EDGE(i,v,\rightarrow j, e_r)$ if and only if $v = v_i$ and $r = r(i \rightarrow j)$. Furthermore by Proposition $1$ $D'$ contains the set of output arcs $OUTPUT(i,v,\rightarrow j, e_r)$ if $v = v_i$ and $r = r(i \rightarrow j)$, and otherwise $D'$ contains none of the arcs from $OUTPUT(i,v,\rightarrow j, e_r)$.

We now have that for each $i, j \in [k]$ with $i < j$, the left input arcs of $CHECK_EDGE(i,j)$ in $D'$ are exactly those in $INPUT(i,v_i,\rightarrow j, e_{r(i\rightarrow j)})$, and the right input arcs of $CHECK_EDGE(i,j)$ in $D'$ are exactly those in $INPUT(j,v_j,\rightarrow i, e_{r(j\rightarrow i)})$. By Proposition $3$ we have that $|INPUT(i,v_i,\rightarrow j, e_{r(i\rightarrow j)})| = |INPUT(j,v_j,\rightarrow i, e_{r(j\rightarrow i)})|$ and so $r(i \rightarrow j) = r(j \rightarrow i)$. It follows that $e_{r(i\rightarrow j)}$ and $e_{r(j\rightarrow i)}$ are the same edge, and that therefore this is an edge in $G$ between $v_i$ and $v_j$.

Thus we have that $v_1, \ldots, v_k$ form a clique in $G$, as required.

**Structure of the constructed graph** Having showed that $D$ represents the instance of $k$-MULTICOLORED CLIQUE, it remains to show that $D$ satisfies the specified properties, that $pw(D)$ is bounded by a function of $k$, and that it can be constructed in fixed-parameter time.

It is clear that there exists a single arc $a^*$ of weight $-1$, that $a^*$ is not part of a double arc, and that all other arcs have weight $0$. To see that pathwidth is bounded, let $D^*$ be the graph derived from $D$ by deleting all arcs and the two
vertices that are not in left or right input arcs from each CheckEdge gadget. (That is, $D^*$ is the graph we had before adding CheckEdge gadgets in the construction of $D$.) We constructed $D^*$ by joining arcs in Start to the input arcs of the ChooseVertex($i$) gadgets, then joining arcs of the resulting graph to the input arcs of the AssignVertex($i, v$) gadgets, then joining arcs of the resulting graph to the input arcs of the ChooseEdge($i, v, \rightarrow j$) gadgets, then joining arcs of the resulting graph to the input arcs of the AssignEdge($i, v, \rightarrow j, e_r$) gadgets. Then by repeated use of Lemma 2 and the fact that a Duplication gadget has pathwidth 2 and a Choice gadget has pathwidth 3, $D^*$ has pathwidth at most $(((2 + 3 + 1) + 2 + 1) + 3 + 1) + 2 + 1) = 16$.

There are $\left(\frac{k}{2}\right) = \frac{k^2 - k}{2}$ CheckEdge gadgets, and therefore we can remove $k^2 - k$ vertices from $D$ to get $D^*$. It follows that $D$ has pathwidth at most $k^2 - k + 16$ (as we can add the $k^2 - k$ extra vertices to every bag in a path decomposition of $D^*$).

**Running time** Let $n$ be the number of vertices and $m$ the number of edges in $G$. In our construction of $D$, we first construct the gadget Start, then $k$ ChooseVertex gadgets, then $n$ AssignVertex gadgets, then $(k-1)n$ ChooseEdge gadgets, then $2m$ AssignEdge gadgets, and finally $\frac{k^2 - k}{2}$ CheckEdge gadgets.

Each CheckEdge gadget has at most $6m^2$ arcs, each AssignEdge gadget has at most $2m + 2$ arcs, and assuming that $m \geq n \geq k$, each other gadget has at most $3m + 3$ arcs. Therefore the construction of $D$ takes $O(k^2m^2)$ time.

We have now provided a fixed-parameter time reduction that reduces any instance of $k$-Multicolor Clique to any instance of MBSDA with the required properties and with pathwidth bounded by a function of $k$. This concludes the proof.

### 3 Reducing MBSDA to MCPP

Let $(D = (V, A), \{(a_i, a'_i) : i \in [t]\})$ be an instance of MBSDA. By Theorem 2 we may assume that $D$ has a single arc $a^*$ of weight $-1$, which is not in a pair of joined arcs, and all other arcs have weight 0. We will produce an instance $G$ of MCPP and an integer $W$, such that $G$ has a solution of weight $W$, and $G$ has a solution of weight less than $W$ if and only if our instance of MBSDA has a solution with negative weight. All edges and arcs in our MCPP instance will have weight 1.

In what follows, we will talk about introducing paths of length $M$ between two vertices. Here $M$ is a sufficiently large number, to be defined later. For such paths, we may assume that the internal vertices have no other neighbors in the graph. It follows that for any minimal optimal solution, each arc or edge in the path will be traversed the same number of times (if one edge in the path is traversed more time than its neighbor, it must be traversed at least three times, including at least once in each direction, and so the solution is not minimal). We will choose $M$ to be such that for any reasonable solution, each edge/arc in
the path is traversed only once (and therefore, exactly once). This allows us to impose a lot of structure on the possible solutions.

For each pair of vertices \( u, v \) such that there is an arc from \( u \) to \( v \) in \( D \), we produce a gadget \( \text{Gadget}(u, v) \) that contains \( u, v \) and new vertices appearing in \( \text{Gadget}(u, v) \). The graph \( G \) will then be the union of these gadgets. The structure of \( \text{Gadget}(u, v) \) will depend on whether there is a double arc between \( u \) and \( v \), and what the cost of the arc is.

For each \( \text{Gadget}(u, v) \), we will consider the restriction of an MCPP solution to this gadget. Within this local solution, the vertices \( u, v \) need not be balanced (as they have edges/arcs in other gadgets), but all other vertices must be balanced. Under the assumption that each long path is traversed exactly once, a gadget will have exactly two feasible solutions, which we call the active and neutral solutions. The active solution will correspond to using the arc \( uv \) (or double arc from \( u \) to \( v \)) in a properly balanced subgraph of \( D \).

Given a directed graph \( H \) (corresponding to part of a solution to a MCPP instance) and a vertex \( v \), the imbalance of \( v \) is \( d_H^+(v) - d_H^-(v) \).

3.1 Gadgets

For an arc \( uv \) of weight 0 that is not part of a double arc:

Construct \( \text{Gadget}(u, v) \) as follows. Add a new vertex \( z_{uv} \), with an arc from \( z_{uv} \) to \( u \) and an arc from \( z_{uv} \) to \( v \). In addition, create two directed paths of length \( M \) from \( u \) to \( z_{uv} \), with the internal vertices all new vertices, and create a directed path of length \( M \) from \( v \) to \( z_{uv} \). (See Figure 3.)

For any solution in which each long path is traversed exactly once, \( z_{uv} \) has in-degree exactly 3. The remaining arcs are the two out-arcs of \( z_{uv} \), which must be traversed exactly three times between them. There are therefore two possible choices:

- **Neutral Solution**: Traverse \( z_{uv}u \) twice and \( z_{uv}v \) once. In this solution, every vertex is balanced within \( \text{Gadget}(u, v) \) and the cost is \( 3M + 3 \).

- **Active Solution**: Traverse \( z_{uv}u \) once and \( z_{uv}v \) twice. In this solution, every vertex is balanced except for \( u \), which has imbalance 1, and \( v \), which has imbalance \(-1\). The cost of this solution is also \( 3M + 3 \).

Observe that the difference between the weight of the neutral and active solutions is 0, and the imbalance at \( u \) and \( v \) for the active solution is the same as in an arc from \( u \) to \( v \).

The total weight of \( \text{Gadget}(u, v) \) is \( 3M + 2 \).

For an arc \( uv \) of weight \(-1\) that is not part of a double arc:

Construct \( \text{Gadget}(u, v) \) as follows. Add two new vertices \( w_{uv} \) and \( z_{uv} \), with arcs \( z_{uv}u, z_{uv}w_{uv} \) and \( v_{w_{uv}} \). In addition, create two paths of length \( M \) from \( u \) to \( z_{uv} \), a path of length \( M \) from \( w_{uv} \) to \( z_{uv} \), and two paths of length \( M \) from \( w_{uv} \) to \( v \). (See Figure 4)
For any solution in which each long path is traversed exactly once, $z_{uv}$ will have exactly 3 in-arcs, and $w_{uv}$ will have exactly 3 out-arcs. It remains to decide how many times to use the arcs out of $z_{uv}$ and into $w_{uv}$.

There are two possible solutions:

- **Neutral Solution**: Traverse $z_{uv}u$ twice, $z_{uv}w_{uv}$ once and $v_{w_{uv}}$ twice. In this solution, every vertex is balanced within $\text{GADGET}(u,v)$ and the cost is $5M + 5$.

- **Active Solution**: Traverse $z_{uv}$ once, $z_{uv}w_{uv}$ twice and $v_{w_{uv}}u$ once. In this solution, every vertex is balanced except for $u$, which has imbalance 1, and $v$, which has imbalance $-1$. The cost of this solution is also $5M + 4$.

Observe that the active solution costs 1 less than the neutral solution, and the imbalance at $u$ and $v$ for the active solution is again the same as in an arc from $u$ to $v$. 

**Fig. 3.** The gadget $\text{GADGET}(u,v)$ when $uv$ is an arc of weight 0 that is not part of a double arc. Dashed lines represent paths of length $M$. 
The gadget $\text{Gadget}(u, v)$ when $uv$ is an arc of weight $-1$ that is not part of a double arc. Dashed lines represent paths of length $M$.

The weight of this gadget is $5M + 3$.

**For a double arc from $u$ to $v$:**

Construct $\text{Gadget}(u, v)$ as follows. Add a directed path of length $M$ from $u$ to $v$. In addition, add an undirected path of length $M$ between $u$ and $v$. (See Figure 5) Assuming a solution traverses each long path exactly once, the only thing to decide is in which direction to traverse the undirected path. Thus there are two possible solutions:

- **Neutral Solution:** Traverse the undirected path from $v$ to $u$. In this solution, every vertex is balanced within $\text{Gadget}(u, v)$ and the cost is $2M$.
- **Active Solution:** Traverse the undirected path from $u$ to $v$. In this solution, every vertex is balanced except for $u$, which has imbalance $2$, and $v$, which has imbalance $-2$. The cost of this solution is also $2M$. 

![Diagram](image-url)
(a) A double arc from $u$ to $v$

(b) $\text{Gadget}(u, v)$

(c) Neutral solution

(d) Active solution

Fig. 5. The gadget $\text{Gadget}(u, v)$ when there is a double arc from $u$ to $v$. Dashed lines represent directed paths of length $M$. The dotted line represents an undirected path of length $M$.

Observe that the difference between the weight of the neutral and active solutions is 0, and the imbalance at $u$ and $v$ for the active solution is the same as in a double arc from $u$ to $v$.

The weight of this gadget is $2M$.

3.2 Combining gadgets

Let $G$ be the graph in which every arc / double arc in $D$ is replaced with its corresponding gadget.

Let $m_1$ be the number of arcs of weight 0 not in a double arc in $D$. By Theorem 2, we may assume there is only one arc of weight $-1$ not in a double arc in $D$. Let $m_2$ be the number of double arcs in $D$.

If we use the neutral solution for every gadget, then every vertex is balanced, every arc is covered, and the total cost is $m_1(3M + 3) + 5M + 5 + m_2(2M) = (3m_1 + 2m_2 + 5)M + 3m_1 + 5$. Therefore this is an upper bound on the weight of an optimal solution.

The total weight of the graph is $m_1(3M + 2) + 5M + 3 + m_2(2M) = (3m_1 + 2m_2 + 5)M + 2m_1 + 3$. Therefore, any minimal solution that does not traverse each long path exactly once will have weight at least $(3m_1 + 2m_2 + 6)M + 2m_1 + 3$. This is $M - m_1 - 2$ greater than the solution in which we use the neutral solution for every gadget. So by setting $M$ to be $m_1 + 3$, we may assume that the optimal solution traverses each long path exactly once. Therefore we may assume that we use either the active or neutral solution for each gadget.

3.3 Correctness, Pathwidth and Running Time

Let $W = (3m_1 + 2m_2 + 5)M + 3m_1 + 5$, the cost of using the neutral solution for each gadget. We now show that $G$ has a solution of weight less than $W$ if and only if $D$ has a solution of negative weight.

Suppose first that $G$ has a solution of weight less than $W$. As discussed above, we may assume that every gadget is either given the active or neutral solution. Let $u^*, v^*$ be the vertices such that $u^*v^*$ is the only arc in $D$ of weight $-1$. Then $\text{Gadget}(u^*, v^*)$ is the only gadget for which one solution weighs less than the other. Therefore we may assume $\text{Gadget}(u^*, v^*)$ is given the active solution.
Let $D'$ be the subgraph of $D$ whose arc set is the set of all arcs whose corresponding gadget in $G$ gets the active solution (with both arcs in a double arc included if their gadget has the active solution, and neither if it has the neutral solution). Then $D'$ contains $u^*v^*$ and so $D$ has negative weight. By construction, $|A(D') \cap \{a, a'\}| \neq 1$ for any arc pair $\{a, a'\}$ in $D$. It remains to show that $D'$ is balanced.

Suppose for a contradiction that $D'$ is unbalanced at some vertex $v$. Each out-arc of $v$ in $D'$ corresponds to a gadget of $G$ in which $v$ has one extra out-arc in our MCPP solution. Each in-arc of $v$ in $D'$ corresponds to a gadget of $G$ in which $v$ has one extra in-arc in our MCPP solution. Each arc incident to $v$ that is not used in $D'$ corresponds to a gadget of $G$ in which $v$ is balanced. These gadgets cover all the arcs incident to $v$ in our MCPP solution. Therefore if the number of in-arcs and out-arcs of $v$ are different in $D'$, $v$ is unbalanced in our MCPP solution. But this is a contradiction. So $D'$ is balanced, as required.

Suppose on the other hand that $D$ has a solution $D'$ of negative weight. Construct a solution to MCPP on $G$ by assigning each gadget the active assignment if the corresponding arc/double-arc appears in $D'$, and the passive assignment otherwise. As $D$ has negative weight, it must use $u^*v^*$ and so $Gadget(u^*, v^*)$ gets the active solution. It follows that the cost of this solution is $W - 1$. It is clear that every arc and edge is traversed at least once. It remains to show that this solution is balanced, and therefore corresponds to a closed walk. In our MCPP solution, the imbalance of a vertex $v$ is equal to the sum of its imbalance in the active gadgets containing it. The imbalance of $v$ in an active gadget is $+1$ if the gadget corresponds to a single out-arc of $v$, $-1$ if the gadget corresponds to a single in-arc of $v$, $+2$ if the gadget corresponds to a double arc starting at $v$, and $-2$ if the gadget corresponds to a double arc ending at $v$. It follows that the imbalance of $v$ in our MCPP solution is equal to its imbalance in $D'$, which is $0$. Therefore our solution is balanced, as required.

We now show that $G$ has pathwidth bounded by a function of $pw(D)$, the pathwidth of $D$. We will use the following technical lemma, whose proof is identical to that of Lemma 3.

**Lemma 3.** Let $H$ be a mixed multigraph, and $G$ the mixed multigraph derived by replacing each arc or double arc from $u$ to $v$ with a gadget $G_{uv}$. Then $pw(G) \leq pw(H) + \max_{uv} pw(G_{uv})$.

Observe that for each gadget $Gadget(u, v)$ in our construction of $G$, $Gadget(u, v)$ can be turned into a disjoint union of paths by the removal of at most 4 vertices, and therefore $Gadget(u, v)$ has pathwidth at most 5. Furthermore, $G$ can be derived from $D$ by each arc or double arc with a corresponding gadgets. It follows from Lemma 3 that $G$ has pathwidth at most $pw(D) + 5$.

Let $m$ be the number of arcs in $D$. Then $G$ is derived from $D$ by introducing at most $m$ gadgets, and each gadget has at most $5M \leq 5m + 15$ arcs. Therefore $G$ can be constructed in $O(m^2)$ time.

We now have that, given an instance $(D, w, X)$ of MBSDA of the type specified in Theorem 2, we can in polynomial time create an instance $G$ of MCPP.
with pathwidth bounded by max(tw(D), 4). We therefore have a parameterized reduction from this restriction of MBSDA, parameterized by pathwidth, to MCPP parameterized by pathwidth. As this restriction of MBSDA is W[1]-hard by Theorem 2, we have the following theorem.

**Theorem 3.** MCPP is W[1]-hard parameterized by pathwidth.

### 4 Discussion

In this paper, we proved that MCPP parameterized by pathwidth is W[1]-hard even if all edges and arcs of input graph $G$ have weight 1. This solves the second open question of van Bevern et al. [15] on parameterizations of MCPP; the first being the parameterization by the number of arcs in $G$, which was proved to be fixed-parameter tractable in [12]. We call a vertex $v$ of $G$ even if the total number of arcs and edges incident to $v$ is even. Another parameterization of MCPP in [15] is motivated by the fact that if each vertex of $G$ is even, then MCPP is polynomial-time solvable [6]. van Bevern et al. [15] ask whether MCPP parameterized by the number of non-even vertices is fixed-parameter tractable.

### References

1. E. J. Beltrami and L. D. Bodin. Networks and vehicle routing for municipal waste collection. *Networks*, 4(1):65–94, 1974.
2. P. Brucker. The chinese postman problem for mixed graphs. In H. Noltemeier, editor, *Graph Theoretic Concepts in Computer Science*, volume 100 of *Lecture Notes in Computer Science*, pages 354–366. Springer Berlin Heidelberg, 1981.
3. N. Christofides. The optimum traversal of a graph. *Omega*, 1(6):719 – 732, 1973.
4. M. Dom, D. Lokshtanov, S. Saurabh, and Y. Villanger. Capacitated domination and covering: A parameterized perspective. In *Proceedings 3rd International Workshop on Parameterized and Exact Computation, IWPEC 2008*, pages 78–91. Springer Verlag, 2008.
5. R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.
6. J. Edmonds and E. L. Johnson. Matching, euler tours and the chinese postman. *Mathematical Programming*, 5(1):88–124, 1973.
7. H. A. Eiselt, M. Gendreau, and G. Laporte. Arc routing problems. i. The Chinese postman problem. *Operations Research*, 43:231–242, 1995.
8. M. Fellows, F. Fomin, D. Lokshtanov, F. Rosamond, S. Saurabh, S. Szeider, and C. Thomassen. On the complexity of some colorful problems parameterized by treewidth. In A. Dress, Y. Xu, and B. Zhu, editors, *Combinatorial Optimization and Applications*, volume 4616 of *Lecture Notes in Computer Science*, pages 366 – 377, 2007.
9. M. R. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410:53–61, 2009.
10. C. G. Fernandes, O. Lee, and Y. Wakabayashi. Minimum cycle cover and chinese postman problems on mixed graphs with bounded tree-width. *Discrete Appl. Math.*, 157(2):272–279, 2009.
11. P. A. Golovach, J. Kratochvil, and O. Suchy. Parameterized complexity of generalized domination problems. *Discrete Applied Mathematics*, 160(6):780 – 792, 2012.

12. G. Gutin, M. Jones, and B. Sheng. Parameterized complexity of the k-arc chinese postman problem. In A. S. Schulz and D. Wagner, editors, *Algorithms - ESA 2014*, volume 8737 of *Lecture Notes in Computer Science*, pages 530–541. Springer Berlin Heidelberg, 2014.

13. C. H. Papadimitriou. On the complexity of edge traversing. *J. ACM*, 23:544–554, 1976.

14. Y. Peng. Approximation algorithms for some postman problems over mixed graphs. *Chinese J. Oper. Res.*, 8:76–80, 1989.

15. R. van Bevern, R. Niedermeier, M. Sorge, and M. Weller. *Arc Routing: Problems, Methods and Applications*, chapter in Complexity of Arc Rooting Problems. SIAM, in press.