

# Maximum Reachability Orientation of Mixed Graphs

Florian Hörsch   

CISPA, Saarbrücken, Germany

---

## Abstract

We aim to find orientations of mixed graphs optimizing the total reachability, a problem that has applications in causality and biology. For given a digraph  $D$ , we use  $P(D)$  for the set of ordered pairs of distinct vertices in  $V(D)$  and we define  $\kappa_D : P(D) \rightarrow \{0, 1\}$  by  $\kappa_D(u, v) = 1$  if  $v$  is reachable from  $u$  in  $D$ , and  $\kappa_D(u, v) = 0$ , otherwise. We use  $R(D) = \sum_{(u,v) \in P(D)} \kappa_D(u, v)$ .

Now, given a mixed graph  $G$ , we aim to find an orientation  $\vec{G}$  of  $G$  that maximizes  $R(\vec{G})$ . Hakimi, Schmeichel, and Young proved that the problem can be solved in polynomial time when restricted to undirected inputs. They inquired about the complexity in mixed graphs.

We answer this question by showing that this problem is NP-hard, and, moreover, APX-hard.

We then develop a finer understanding of how quickly the problem becomes difficult when going from undirected to mixed graphs. To this end, we consider the parameterized complexity of the problem with respect to the number  $k$  of preoriented arcs of  $G$ , a poorly studied form of parameterization.

We show that the problem can be solved in time  $n^{O(k)}$  and that a  $(1 - \epsilon)$ -approximation can be computed in time  $f(k, \epsilon)n^{O(1)}$  for any  $\epsilon > 0$ .

**2012 ACM Subject Classification** Mathematics of computing  $\rightarrow$  Combinatorial algorithms; Theory of computation  $\rightarrow$  Fixed parameter tractability; Mathematics of computing  $\rightarrow$  Network optimization; Theory of computation  $\rightarrow$  Approximation algorithms analysis

**Keywords and phrases** orientations, mixed graphs, reachability, parameterized complexity, approximation

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2026.53

**Related Version** *Full Version:* <https://arxiv.org/abs/2506.16171>

**Acknowledgements** I want to thank Charupriya Sharma for making me aware of the connection to causality.

## 1 Introduction

This article deals with the problem of finding an orientation  $\vec{G}$  of a mixed graph  $G$  such that the total number of ordered pairs  $(u, v)$  of vertices in  $V(G)$  such that  $v$  is reachable from  $u$  in  $\vec{G}$  is maximized.

This problem has numerous applications in various sciences. For example, in the context of causality, the vertices of the mixed graph may represent random variables and the edges and the arcs of the mixed graph may represent correlations between these random variables. An arc means that the random variable associated to the tail of the arc is causal for the random variable associated to the head of the arc while an edge means that a correlation between the two random variables can be observed, but it is not known which of them is causal for the other one. Hence a solution of the orientation problem helps to better understand the possible correlations between these random variables. A survey of Vowels, Camgoz, and Bowden [23] contains more details on the connection of causal relationships and mixed graphs.



© Florian Hörsch;

licensed under Creative Commons License CC-BY 4.0

43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026).

Editors: Meena Mahajan, Florin Manea, Annabelle McIver, and Nguyễn Kim Thăng

Article No. 53; pp. 53:1–53:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



One more concrete, very similar application for this problem comes from biology, namely so-called protein-protein interaction. This application has been a driving force for the research on the problem setting considered in the present article. It has fostered significant interaction between the communities of theoretical computer science and biology, see [4, 9, 19]. Concretely, in this application, an interaction between certain pairs of proteins can be observed, but it is often technically more difficult to determine which of the two proteins causes this interaction. It is not difficult to see that the collection of protein interactions can be modelled by a mixed graph again. In Life Sciences, it is important to understand which local interactions of proteins lead to most global interaction, which corresponds to our orientation problem. A better understanding of the current problem may hence help to better understand protein-protein interaction, which may be of interest for example in medicine.

### Problem definition

We now introduce our problem more formally. Basic notation can be found in Section 2. In a mixed graph  $G$ , we say that  $v$  is reachable from  $u$  for some  $u, v \in V(G)$  if  $v$  can be reached from  $u$  by traversing arcs in the given direction and traversing edges in an arbitrary direction. We define  $P(G)$  to be the set of ordered pairs of distinct vertices of  $V(G)$ . We further define the function  $\kappa_G : P(G) \rightarrow \{0, 1\}$  by  $\kappa_G(u, v) = 1$  if  $v$  is reachable from  $u$  in  $G$  and  $\kappa_G(u, v) = 0$ , otherwise, for every  $(u, v) \in P(G)$ . Given a set  $P \subseteq P(G)$ , we use  $R_P(G)$  for  $\sum_{(u,v) \in P} \kappa_G(u, v)$ . Observe that digraphs are in particular mixed graphs, so the above definitions carry over to digraphs. An orientation of a mixed graph is obtained by replacing every edge by an arc with the same two endvertices. In the problem Fixed Pairs Mixed Maximum Reachability Orientation (FPMMRO), the input consists of a mixed graph  $G$  and a set  $P \subseteq P(G)$ , and the objective is to find an orientation  $\vec{G}$  of  $G$  that maximizes  $R_P(\vec{G})$ . We further use Fixed Pairs Undirected Maximum Reachability Orientation (FPUMRO) for the restriction of FPMMRO to undirected inputs. In the following, when speaking about complexity, we need to carefully distinguish between the version where we are searching for a complete solution of the instance and the maximization version. In the complete solution version, we want to decide whether there exists an orientation  $\vec{G}$  of  $G$  with  $R_P(\vec{G}) = |P|$ , while in the maximization version, we assume that the instance comes with an integer  $K$  and we want to decide whether there exists an orientation  $\vec{G}$  of  $G$  with  $R_P(\vec{G}) \geq K$ . Clearly, in all settings, the maximization version is at least as hard as the complete solution version.

### Previous work

For FPUMRO, a simple argument, which was given by Hassin and Meggido [17], shows that the complete solution version can be handled in polynomial time. However, it turns out that the maximization version of FPUMRO is significantly harder. More concretely, it was proved by Elberfeld et al. [10] that the problem is NP-hard. One important paradigm when encountering NP-hard problems is approximation algorithms. In [10], it was proved that FPUMRO on  $n$ -vertex graphs can be approximated within a factor of  $\Omega(\frac{\log \log n}{\log n})$  in polynomial time, improving on an earlier result of Medvedovsky et al. [19]. On the other hand, the authors of [10] show that, unless  $P=NP$ , no polynomial-time approximation within a factor of less than  $\frac{12}{13}$  is available. Another variation of FPUMRO taking into account the lengths of the created directed paths was studied by Blokh, Segev, and Sharan [4]. The study of FPUMRO in the context of parameterized complexity was initialized by Dorn et al. [9]. They studied several problem-specific parameters describing how much the desired paths connecting input pairs interact. Following up on their work, Cygan, Kortsarz, and Nutov [8] proved that the maximization version of FPUMRO is FPT with respect to the objective

function, that is, the number of pairs that are supposed to be satisfied. Interestingly, the existence of a polynomial-time constant-factor approximation algorithm still seems to be open for FPUMRO.

We now turn our attention to FPMMRO, the version which is more general and hence also more powerful for modelling the applications. FPMMRO was first considered by Arkin and Hassin [1]. They show that, in contrast to FPUMRO, the complete solution version of FPMMRO is already NP-hard in general, but can be solved in polynomial time if  $|P| = 2$ . There exists a rich literature for FPMMRO regarding approximation algorithms and parameterized complexity.

For the approximation side, in [11], Elberfeld et al. showed that, unless  $P = NP$ , FPMMRO cannot be approximated in polynomial time within a factor better than  $\frac{7}{8}$ , and, on the other hand, gave a first polynomial-time sublinear approximation algorithm. The approximation ratio was later improved by Gamzu and Medina in [13]. However, a constant-factor approximation algorithm for FPMMRO was proven to be unlikely by Włodarczyk [24], even when allowing algorithms that run in FPT time with respect to the number of pairs. An APX-hardness result in planar mixed graph has been provided by Chitnis, Feldmann, and Suchý [6].

For parameterized complexity, the most studied parameter is the number of pairs. It was proved by Cygan, Kortsarz and Nutov in [8] that the complete solution version of FPMMRO is XP with respect to the number of pairs, improving on the result of Arkin and Hassin. On the other hand, it was proved by Pilipczuk and Wahlström [21] that the problem is  $W[1]$ -hard, which is also implied by the above mentioned result of Włodarczyk. The result of Pilipczuk and Wahlström has been extended to the planar setting by Chitnis, Feldmann, and Suchý [6]. In a recent work, Hanaka et al. [16] dealt with the parameterized complexity of FPMMRO when considering other parameters, for example the number of arcs of  $G$  or the treewidth of its underlying graph.

We now come to the concrete special case of FPMMRO that is dealt with in the present article. A lot of research on FPMMRO so far focused on the case that the set  $P$  of pairs in the instance is small. For example, the above mentioned parameterized complexity results all fall into this category. Here, we deal with a specific, but very natural restriction of FPMMRO, which is in some way the other extreme, where  $P$  is as large as possible. Namely, we consider the case that the set of pairs in consideration is the set of all possible pairs, i.e.  $P = P(G)$ . For simplicity, we use MMRO to denote this restriction and, given a mixed graph  $G$ , we use  $R(G)$  for  $R_{P(G)}(G)$ . For the complete solution version of MMRO, which corresponds to the question of whether a given mixed graph has a strongly connected orientation, a full characterization directly yielding a polynomial-time algorithm was given by Boesch and Tindell [5]. We hence focus on the maximization version.

In [15], Hakimi, Schmeichel, and Young considered a setting which is in some way the intersection of MMRO and FPUMRO, namely the special case of MMRO that the input graph  $G$  is undirected. They gave a complete characterization of the optimal orientations in this case. A polynomial-time algorithm for the maximization version of this restriction of MMRO follows directly from this characterization.

## Our contributions

In [15], the authors also inquired about the complexity of MMRO in general. Surprisingly, despite this question being almost 30 years old, little progress seems to have been made. The first contribution of the present article is to give a negative answer to this question by providing an NP-hardness result. Moreover, the following result is slightly stronger and also shows that the approximability of MMRO has certain limits.

► **Theorem 1.1.** *Unless  $P=NP$ , there exists no polynomial-time  $\frac{581773}{581774}$ -approximation algorithm for MMRO.*

While this inapproximability result may appear marginal, it rules out a polynomial-time approximation scheme, which distinguishes MMRO from a large collection of combinatorial problems. With Theorem 1.1 at hand, it appears interesting to understand special cases and restrictions of MMRO that are tractable. Parameterized complexity is a powerful paradigm to obtain a better understanding of the complexity of NP-hard problems. In the current case, we wish to consider a parameterization which is natural in the light of the concrete input. Namely, we consider the parameterization with respect to the number of arcs of  $G$ . On a high level, we wish to understand the complexity of the problem in the case that the number of arcs is small. It is worth noting that the consideration of this parameter is a relatively new concept. In [14], Gutin et al. studied this parameter for a version of the Chinese Postman Problem in mixed graphs, answering a question of van Bevern et al., see [7]. We are not aware of any previous consideration of this parameter for orientation problems, except for the work of Hanaka et al. [16], which was recently submitted independently of the present article. Before describing our results, we wish to remark that for the natural analogous parameterization by the number of edges of  $G$ , a trivial FPT algorithm exists. Namely, we can compute all possible orientations of  $G$  and compare their respective objective values.

In the following, given an instance  $G$  of MMRO, we use  $k$  for  $|A(G)|$ . We further say that an orientation  $\vec{G}$  of  $G$  is *optimal* if  $R(\vec{G})$  is maximum among all orientations of  $G$ . Our first result is that, for fixed  $k$ , we can indeed compute an optimal orientation for an instance of MMRO in polynomial time. More precisely, we show the following result.

► **Theorem 1.2.** *Given an instance  $G$  of MMRO, we can compute an optimal orientation of  $G$  in time  $n^{O(k)}$ .*

Theorem 1.2 raises the question whether a further improvement in the running time of this algorithm is possible. Namely, it would be desirable to determine whether the problem is in FPT, that is, if there exists an algorithm computing an optimal orientation that runs in time  $f(k)n^{O(1)}$ . While this question remains open, we make some progress in this direction by combining the paradigms of parameterized complexity and approximation. To this end, we say that, for some  $\alpha \in [0, 1]$ , an orientation  $\vec{G}$  of a given mixed graph  $G$  is  $\alpha$ -*optimal* if  $R(\vec{G}) \geq \alpha R(\vec{G}_0)$  holds for every orientation  $\vec{G}_0$  of  $G$ . The following result roughly speaking shows that the problem is 'almost' FPT, that is, it can be arbitrarily well approximated in FPT time.

► **Theorem 1.3.** *Given a constant  $\epsilon > 0$  and an instance  $G$  of MMRO, in time  $f(k, \epsilon)n^{O(1)}$ , we can compute a  $(1 - \epsilon)$ -optimal orientation of  $G$ .*

Due to Theorems 1.1 and 1.3, it becomes evident that for MMRO, algorithms running in FPT time with respect to  $k$  can yield solutions of a higher quality than polynomial time algorithms. Indeed, FPT approximations can yield arbitrarily good approximations by Theorem 1.3, which polynomial time algorithms cannot by Theorem 1.1.

The remainder of this article is structured as follows: in Section 2, we give some more formal definitions, introduce an auxiliary problem and show that MMRO can be reduced to mixed graphs that are acyclic. In Section 3, we prove Theorem 1.1. In Section 4, we prove our algorithmic results, that is, Theorem 1.2 and Theorem 1.3. Finally, in Section 5, we conclude our work.

## 2 Preliminaries

In this section, we give some preliminaries we need for our main proofs. In Section 2.1, we collect the notation we need. In Section 2.2, we introduce a satisfiability problem we need for our reduction in Section 3. Finally, in Section 2.3, we prove the equivalence of MMRO and a closely related problem in acyclic mixed graphs.

### 2.1 Notation

We here give a collection of problem-specific and non-standard definitions. Standard definitions are omitted. If a mixed graph  $G'$  can be obtained from  $G$  by orienting some edges, then we say that  $G'$  is a *partial orientation* of  $G$ . If  $G'$  is a digraph, we speak of an *orientation*. A *mixed cycle* is a mixed graph that can be oriented as a directed cycle. A mixed graph is called *acyclic* if it does not contain a mixed cycle as a mixed subgraph. Given a mixed graph  $G$ , its *underlying graph*  $UG(G)$  is obtained by replacing every arc in  $A(G)$  by an edge with the same two endvertices. For some  $u \in V(G)$ , we use  $Out_G(u)$  for all the vertices in  $V(G)$  which are reachable from  $u$  in  $G$  and  $In_G(u)$  for all the vertices in  $V(G)$  from which  $u$  is reachable in  $G$ . Moreover, given a mixed graph  $G$  and a weight function  $w : V(G) \rightarrow \mathbb{Z}_{\geq 0}$ , we use  $w(X)$  for  $\sum_{x \in X} w(x)$  for some  $X \subseteq V(G)$ , and we use  $w(H)$  for  $w(V(H))$  for some mixed subgraph  $H$  of  $G$ . Further, we use  $R(G, w)$  for  $2 \sum_{v \in V(G)} \binom{w(v)}{2} + \sum_{(u,v) \in P(G)} \kappa_G(u, v) w(u) w(v)$ . Next, we say that an orientation  $\vec{G}$  of  $G$  is  $\alpha$ -*optimal* for  $(G, w)$  if  $R(\vec{G}, w) \geq \alpha R(\vec{G}_0, w)$  holds for every orientation  $\vec{G}_0$  of  $G$ . We use *optimal* for 1-optimal.

Given two mixed graphs  $T$  and  $U$ , we say that  $U$  is a *mixed supergraph* of  $T$  if  $V(T) \subseteq V(U)$  and  $U[V(T)]$  is a partial orientation of  $T$ . Moreover, for a nonnegative integer  $\ell$ , we say that  $U$  is a *mixed  $\ell$ -supergraph* of  $T$  if  $U$  is a mixed supergraph of  $T$  and  $|V(U) \setminus V(T)| \leq \ell$ .

Two digraphs  $D_1$  and  $D_2$  are called *compatible* if  $UG(D_1)[V(D_1) \cap V(D_2)]$  is isomorphic to  $UG(D_2)[V(D_1) \cap V(D_2)]$ . Given two compatible digraphs  $D_1$  and  $D_2$ , we use  $D_1 \langle D_2 \rangle$  for the unique orientation of  $UG(D_1)$  in which an edge  $uv$  has its orientation in  $D_2$  if  $\{u, v\} \subseteq V(D_2)$  and has its orientation in  $D_1$ , otherwise.

### 2.2 A satisfiability problem

An instance of 3-Bounded Max 2-SAT (3BMax2Sat) consists of a set  $X$  of binary variables and a set  $\mathcal{C}$  of clauses each of which contains exactly two literals over  $X$  such that for every  $x \in X$ , there exist at most 3 clauses in  $\mathcal{C}$  that contain a literal of  $x$ . We use the result of Berman and Karpinski [2] that it is NP-hard to approximate the number of clauses that can be satisfied by an assignment within a factor of  $\frac{2012}{2013}$ .

### 2.3 An adapted version of MMRO

Intuitively, it should be possible to restrict MMRO to acyclic instances by contracting vertex sets admitting a strongly connected orientation. However, due to the specific character of the problem, we have to associate a polynomial weight function to the newly created mixed graph. This weight function displays how many vertices of the original mixed graph were contained in the vertex set of the original mixed graph which was contracted into that vertex. Formally, an instance of Weighted Acyclic Mixed Maximum Reachability Orientation (WAMMRO) consists of an acyclic mixed graph  $G$  and a weight function  $w : V(G) \rightarrow \mathbb{Z}_{\geq 0}$ . The objective is to find an orientation  $\vec{G}$  of  $G$  that maximizes  $R(\vec{G}, w)$ . Recall that  $R(\vec{G}, w) = 2 \sum_{v \in V(G)} \binom{w(v)}{2} + \sum_{(u,v) \in P(G)} w(u) w(v) \kappa_{\vec{G}}(u, v)$ . The first term in this expression reflects the contributions to the objective function made by vertices in

the same strongly connected component of the original mixed graph. While this term is clearly the same for all orientations, it may not be neglected when considering approximation algorithms. The second term in the expression reflects the contributions to the objective function made by vertices in distinct strongly connected component of the original mixed graph. We say that an instance  $(G, w)$  of WAMMRO is *connected* if  $G$  is connected. Given two instances  $(G_1, w_1)$  and  $(G_2, w_2)$  of WAMMRO, we say that  $(G_1, w_1)$  *extends*  $(G_2, w_2)$  if  $V(G_2) \subseteq V(G_1)$  and  $w_1|_{V(G_2)} = w_2$ . Most results in Sections 3 and 4 will be proved for WAMMRO, which then implies the corresponding results for MMRO. The remainder of Section 2.3 is dedicated to giving the two statements below showing the equivalence of these problems. Their proofs are omitted. Here, we need to specify that when speaking about an instance  $(G, w)$  of WAMMRO, we suppose that  $w$  is given in unitary encoding. Further, we generally use  $n$  for  $|V(G)| + w(G)$ . When we say that an algorithm runs in polynomial time, we refer to  $n$ . As a side remark, we wish to point out that this specification is crucial. Indeed, as pointed out in [15], the Subset Sum problem can easily be reduced to the one of finding an orientation  $\vec{G}$  maximizing  $R(\vec{G}, w)$  when given an undirected graph  $G$  and a function  $w : V(G) \rightarrow \mathbb{Z}_{\geq 0}$  given in binary encoding.

► **Proposition 2.1.** *Let  $G$  be an instance of MMRO and  $n = |V(G)|$ . Then, in polynomial time, we can compute an instance  $(G_0, w)$  of WAMMRO with  $|V(G_0)| \leq |V(G)|$ ,  $|A(G_0)| \leq |A(G)|$ , and  $w(G_0) = n$  such that for every positive integer  $K$ , there exists an orientation  $\vec{G}_0$  of  $G_0$  with  $R(\vec{G}_0, w) \geq K$  if and only if there exists an orientation  $\vec{G}$  of  $G$  with  $R(\vec{G}) \geq K$ . Moreover, given an orientation  $\vec{G}_0$  of  $G_0$ , we can compute an orientation  $\vec{G}$  of  $G$  with  $R(\vec{G}) \geq R(\vec{G}_0, w)$  in polynomial time.*

► **Proposition 2.2.** *Let  $(G, w)$  be an instance of WAMMRO. Then, in polynomial time, we can compute an instance  $G_0$  of MMRO and an integer  $K_0 \leq n^9$  such that for every positive integer  $K$ , there exists an orientation  $\vec{G}_0$  of  $G_0$  with  $R(\vec{G}_0) \geq Kn^8 + K_0$  if and only if there exists an orientation  $\vec{G}$  of  $G$  with  $R(\vec{G}, w) \geq K$ .*

### 3 Hardness result

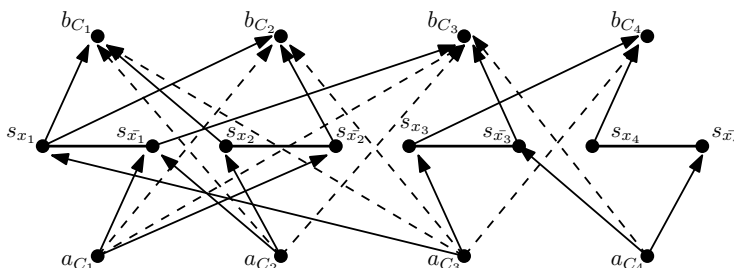
This section is dedicated to the proof of Theorem 1.1. First, in Lemma 3.1, we prove the APX-hardness of WAMMRO. This part is a reduction from 3BMax2Sat and the more challenging part. Our reduction is similar to the one proving the NP-hardness of the complete solution version of FPMMRO in [1], but more care is needed. Roughly speaking, in both reductions, we actually want to only consider the reachabilities among certain pairs corresponding to clauses being satisfied. While in the reduction in [1] the restriction to these pairs can be obtained from the problem definition, for the hardness of WAMMRO, we need to make sure by some additional connections that we can control the contribution of the remaining pairs to the objective function. Further, the restricted number of variable occurrences allows us to make sure that these additional reachabilities do not dominate those obtained by the pairs of vertices we actually want to consider, which is crucial for the APX-hardness.

► **Lemma 3.1.** *Unless  $P = NP$ , there is no polynomial-time algorithm whose input is an instance  $(G, w)$  of WAMMRO and a positive integer  $K \geq \frac{1}{16}n$ , that outputs 'yes' if  $G$  admits an orientation  $\vec{G}$  with  $R(\vec{G}, w) \geq K$ , and that outputs 'no' if  $R(\vec{G}, w) < \frac{34220}{34221}K$  holds for every orientation  $\vec{G}$  of  $G$ .*

**Proof sketch.** Suppose that an algorithm  $A$  with the properties described in Lemma 3.1 exists. In the following, we describe an algorithm based on  $A$  whose input is an instance of 3BMax2Sat and a positive integer  $K$ . We will show that the algorithm has a behavior such that  $P = NP$  follows by the hardness of 3BMax2Sat.

Let  $(X, \mathcal{C})$  be an instance of 3BMax2Sat and  $K$  a positive integer. By easy arguments, we may suppose that  $|\mathcal{C}| \geq \frac{1}{2}|X|$  and  $K > \frac{3}{4}|\mathcal{C}|$ .

In the following, we let  $\mathcal{Y}$  denote the set of subsets of  $\mathcal{C}$  that consist of exactly two distinct clauses  $C, C' \in \mathcal{C}$  that share at least one variable. We now construct an instance  $(G, w)$  of WAMMRO. First, we let  $V(G)$  contain a set  $V_{\mathcal{C}}$  that contains two vertices  $a_C$  and  $b_C$  for every  $C \in \mathcal{C}$ . For every  $\{C, C'\} \in \mathcal{Y}$ , we let  $A(G)$  contain the arcs  $a_C b_{C'}$  and  $a_{C'} b_C$ . Next, we let  $V(G)$  contain a set  $V_X$  that contains two vertices  $s_x$  and  $s_{\bar{x}}$  for every  $x \in X$ . For every  $x \in X$ , we let  $E(G)$  contain an edge linking  $s_x$  and  $s_{\bar{x}}$ . Further, for every  $x \in X$  and every  $C \in \mathcal{C}$  such that  $x \in C$ , we let  $A(G)$  contain the arcs  $a_C s_{\bar{x}}$  and  $s_x b_C$  and for every  $x \in X$  and every  $C \in \mathcal{C}$  such that  $\bar{x} \in C$ , we let  $A(G)$  contain the arcs  $a_C s_x$  and  $s_{\bar{x}} b_C$ . Finally, we set  $w(v) = 1$  for all  $v \in V_{\mathcal{C}}$  and  $w(v) = 0$  for all  $v \in V_X$ . This finishes the description of  $(G, w)$ . Further, we set  $K_0 = 2|\mathcal{Y}| + K$ . For an illustration, see Figure 1.



**Figure 1** An example of the construction in the proof of Lemma 3.1, where  $X = \{x_1, x_2, x_3, x_4\}$  and  $\mathcal{C} = \{C_1 = \{x_1, x_2\}, C_2 = \{x_1, \bar{x}_2\}, C_3 = \{\bar{x}_1, \bar{x}_3\}, C_4 = \{x_3, x_4\}\}$ . For the sake of better readability, the arcs with both endvertices in  $V_{\mathcal{C}}$  are dashed while the arcs with one endvertex in  $V_{\mathcal{C}}$  and one endvertex in  $V_X$  are solid.

We now apply  $A$  to  $(G, w)$  and  $K_0$  and output the output of  $A$ . This finishes the description of our algorithm.

The crucial connection between the original instance  $(X, \mathcal{C})$  of 3BMax2Sat and the newly created instance  $(G, w)$  of WAMMRO is a correspondence between the truth assignments for  $(X, \mathcal{C})$  and the orientations of  $G$ . Namely, for every  $x \in X$ , orienting the edge linking  $s_x$  and  $s_{\bar{x}}$  from  $s_{\bar{x}}$  to  $s_x$  corresponds to assigning  $x$  to True and orienting the edge linking  $s_x$  and  $s_{\bar{x}}$  from  $s_x$  to  $s_{\bar{x}}$  corresponds to assigning  $x$  to False. We then obtain that a clause  $C \in \mathcal{C}$  is satisfied by a certain truth assignment if and only if  $b_C$  is reachable from  $a_C$  in the corresponding orientation of  $G$ . Moreover, for all pairs  $(u, v) \in P(G)$  which are not of the form  $(a_C, b_C)$  for some  $C \in \mathcal{C}$ , we either have  $\min\{w(u), w(v)\} = 0$  or  $\kappa_{\vec{G}}(u, v)$  is the same for all orientations  $\vec{G}$  of  $G$ . The details are omitted.  $\blacktriangleleft$

We then conclude Theorem 1.1 from Lemma 3.1 and Proposition 2.2. The proof does not contain any conceptual difficulties and is omitted.

## 4 Algorithmic results

The objective of this section is to prove Theorems 1.2 and 1.3. We now give an overview of these proofs. We mainly explain the proof strategy for Theorem 1.2 and later point out how to adapt it for the algorithm of Theorem 1.3.

Roughly speaking, the main strategy of the algorithm is to cut a given instance into small parts which are highly structured, then compute optimal solutions for these parts and merge them. We mainly deal with a certain substructure we call undirected components. Namely, given an instance  $(G, w)$  of WAMMRO, an *undirected component* of  $G$  is a component of  $G \setminus A(G)$ .

Consider an undirected component  $T$  of  $G$ . Observe that  $T$  is a tree as  $G$  is acyclic. Our objective is to find a set of orientations  $\mathcal{T}$  of  $T$  such that for any orientation  $\vec{G}$  of  $G$ , there exists some  $\vec{T} \in \mathcal{T}$  with the property that when reorienting all edges in  $E(T)$  according to  $\vec{T}$ , we obtain an orientation that is at least as good as  $\vec{G}$ . Formally, given an instance  $(G, w)$  of WAMMRO and an undirected component  $T$  of  $G$ , we say that a set  $\mathcal{T}$  of orientations of  $T$  is an *optimal replacement set* for  $((G, w), T)$  if for every orientation  $\vec{G}$  of  $G$ , there exists some  $\vec{T} \in \mathcal{T}$  such that  $R(\vec{G}\langle\vec{T}\rangle, w) \geq R(\vec{G}, w)$ . Here, recall that  $\vec{G}\langle\vec{T}\rangle$  is the unique orientation of  $G$  in which all edges in  $E(T)$  have the orientation they have in  $\vec{T}$  and all other edges in  $E(G)$  have the orientation they have in  $\vec{G}$ .

The importance of optimal replacement sets can be summarized in the following result.

► **Lemma 4.1.** *Let  $(G, w)$  be an instance of WAMMRO, let  $T_1, \dots, T_q$  be the undirected components of  $G$  and for  $i \in [q]$ , let  $\mathcal{T}_i$  be an optimal replacement set for  $((G, w), T_i)$ . Then there exists an optimal orientation  $\vec{G}$  for  $(G, w)$  such that  $\vec{G}[V(T_i)] \in \mathcal{T}_i$  for  $i \in [q]$ .*

The proof of Lemma 4.1 is not difficult. Observe that Lemma 4.1 suggests an algorithm for WAMMRO: given an instance  $(G, w)$  of WAMMRO, find an optimal replacement set of  $T$  of appropriate size for all undirected components  $T$  of  $G$  and then try all possible orientations of  $G$  such that the inherited orientations of the undirected components come from the optimal replacement sets. Keep the best orientation found during this procedure. It is easy to prove that this yields an optimal orientation for  $(G, w)$  indeed.

The difficulty now is to prove that optimal replacement sets of appropriate size exist for all undirected components of  $G$  and can be computed sufficiently fast. We first restrict the structure of the instances in consideration by a preprocessing step.

Roughly speaking, we consider instances in which the interaction of every undirected component with the remaining part of the mixed graph is limited. Formally, we say that an instance  $(G, w)$  of WAMMRO is *dismembered* if every undirected component of  $G$  either contains at most one vertex of  $V(A(G))$  or contains exactly two vertices of  $V(A(G))$  and each of them is incident to exactly one arc of  $A(G)$ . We show that it suffices to prove Theorem 1.2 for dismembered instances. Moreover, we make the simple observation that we can restrict ourselves to connected instances, where an instance  $(G, w)$  is *connected* if  $UG(G)$  is connected. In order to be able to reuse the result later for the proof of Theorem 1.3, the following strong version is given. It also handles approximate solutions.

► **Lemma 4.2.** *Let  $g : \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0} \times [0, 1] \rightarrow \mathbb{Z}_{\geq 0}$  be a fixed function and suppose that there exists an algorithm  $A$  that computes a  $(1 - \epsilon)$ -optimal orientation for every connected, dismembered instance of WAMMRO and every  $\epsilon \geq 0$  in time  $g(n, k, \epsilon)$ . Then, for every  $\epsilon \geq 0$ , a  $(1 - \epsilon)$ -optimal orientation for every instance of WAMMRO can be computed in time  $f(k)g(n, k', \epsilon)n^{O(1)}$  for some  $k' = O(k)$ .*

It turns out that for dismembered instances, the situation is much brighter and we can indeed efficiently compute optimal replacement sets of polynomial size. The following is the main lemma for the proof of Theorem 1.2.

► **Lemma 4.3.** *Let  $(G, w)$  be a connected, dismembered instance of WAMMRO and let  $T$  be an undirected component of  $G$ . Then, in polynomial time, we can compute an optimal replacement set  $\mathcal{T}$  for  $((G, w), T)$  of size  $O(n^3)$ .*

Lemma 4.3, Lemma 4.1, and Lemma 4.2 will imply Theorem 1.2 with Proposition 2.1.

We now give an overview of the proof of Lemma 4.3. Consider an undirected component  $T$  of a dismembered, connected instance  $(G, w)$  of WAMMRO. Our strategy is to compute a collection of instances of WAMMRO extending  $(T, w|V(T))$  such that a collection of

orientations of  $T$  consisting of the one inherited from an optimal solution to each of these instances forms an optimal replacement set for  $((G, w), T)$ . Informally, the idea is that it suffices to consider all orientations of  $E(G) \setminus E(T)$  and find the best way to extend these orientations to  $T$ . However, of course, we cannot afford to enumerate all orientations of  $E(G) \setminus E(T)$ . Therefore, we aim to find a small set of instances that, roughly speaking, simulates all possible orientations of  $E(G) \setminus E(T)$ . Formally, a *simulation set* for  $((G, w), T)$  is a set  $\mathcal{U}$  of mixed supergraphs of  $T$  which are compatible with  $G$  such that for every orientation  $\vec{G}$  of  $G$ , there exists some  $U \in \mathcal{U}$  and a weight function  $w' : V(U) \rightarrow [n]_0$  with  $|w'| \leq |w|$  that is consistent with  $w$  and with the property that for every orientation  $\vec{U}$  of  $U$ , we have that  $\vec{G} \langle \vec{U} \rangle$  is an orientation of  $G$ ,  $\vec{U} \langle \vec{G} \rangle$  is an orientation of  $U$  and  $R(\vec{G} \langle \vec{U} \rangle, w) - R(\vec{G}, w) \geq R(\vec{U}, w') - R(\vec{U} \langle \vec{G} \rangle, w')$  holds.

Once we have a good simulation set  $\mathcal{U}$  for  $((G, w), T)$ , we want to compute an optimal replacement set  $\mathcal{T}$  in the following way. For every  $U \in \mathcal{U}$  and every appropriate weight function  $w'$  on  $V(U)$  consistent with  $w$ , we compute an optimal orientation for  $(U, w')$  and add the inherited orientation  $\vec{T}$  of  $T$  to  $\mathcal{T}$ . It is not difficult to see that  $\mathcal{T}$  is an optimal replacement set indeed.

However, in order to obtain an efficient algorithm from this observation, we need to take care of several things. First, we need to make sure that the number of mixed graphs in  $\mathcal{U}$  is small enough. Next, we need to make sure that the mixed graphs in  $\mathcal{U}$  contain only slightly more vertices than  $T$  in order to be able to enumerate all weight functions extending  $w$ . Finally, we need to make sure that for every  $U \in \mathcal{U}$  and for every weight function  $w'$  on  $V(U)$ , we can compute an optimal orientation for  $(U, w')$  sufficiently fast. To this end, we need some more technical definitions. First, for some positive integer  $\ell$ , an  $\ell$ -*simulation set* for  $((G, w), T)$  is a simulation set for  $((G, w), T)$  such that every  $U \in \mathcal{U}$  is a mixed  $\ell$ -supergraph of  $T$ . We further need to define a class of tree-like mixed graphs. Namely, we say that a mixed graph  $U$  is *arboresque* if  $A(G)$  contains an arc  $rs$  such that the underlying graph of  $G \setminus \{rs\}$  is a tree and one of  $d_G^+(r) = d_{UG(G)}(r) = 2$  and  $d_G^-(s) = d_{UG(G)}(s) = 2$  holds. We say that a collection  $\mathcal{U}$  of mixed graphs is *arboresque* if every element of  $\mathcal{U}$  is arboresque.

We are now ready to formulate our main result on simulation sets.

► **Lemma 4.4.** *Let  $(G, w)$  be a dismembered instance of WAMMRO and  $T$  an undirected component of  $G$  with  $|V(T) \cap V(A(G))| \geq 1$ . Then, in polynomial time, we can compute an arboresque 3-simulation set of size at most 2 for  $((G, w), T)$ .*

In order to make use of Lemma 4.4, we need to handle arboresque mixed graphs, which are a rather minor extension of mixed graphs whose underlying graphs are trees. We use this fact to show that an optimal orientation for an instance  $(U, w)$  of WAMMRO such that  $U$  is arboresque can be computed in polynomial time, using a dynamic programming approach.

► **Lemma 4.5.** *Let  $(G, w)$  be an arboresque instance of WAMMRO. Then an optimal orientation  $\vec{G}$  for  $(G, w)$  can be computed in polynomial time.*

We wish to remark that Lemma 4.5 is a strengthening of the main result of [15]. Lemma 4.5 and Lemma 4.4 imply Lemma 4.3. This completes the overview of the proof Theorem 1.2.

We now give an overview of how to adapt these results to prove Theorem 1.3. We prove approximation analogues of Lemma 4.1 and Lemma 4.3. The analogue of Lemma 4.1 is rather straight forward and can be proved in a similar way. For the analogue of Lemma 4.3, instead of an optimal replacement set  $\mathcal{T}$  for  $((G, w), T)$  whose size is polynomial in  $n$ , we obtain a set  $\mathcal{T}$  of orientations of  $T$  such that replacing with an orientation from this set may yield an orientation that is by a small additive constant worse than the first orientation in

consideration. On the other hand, we have that  $|\mathcal{T}|$  is bounded by  $f(k, \epsilon)$ . This result is obtained from Lemmas 4.4 and 4.5 in a similar way as Lemma 4.3. The only difference is that instead of trying all possible weight functions for the mixed graphs in the simulation sets, we restrict to a smaller set of functions, avoiding similar ones. We then conclude that an orientation that is only a certain additive amount away from an optimal solution can be computed in the desired running time.

However, we need to prove a small multiplicative loss rather than an additive one. To this end, we prove a result showing that for every connected, dismembered instance of WAMMRO, there exists an orientation  $\vec{G}$  of  $G$  such that  $R(\vec{G}, w)$  is above a certain minimum threshold. In order to so, we focus on a largest undirected component of  $G$  and analyze the objective value of an optimal solution as described in [15]. Together with the additive approximation result described above, Theorem 1.3 follows readily.

The rest of Section 4 is structured as follows. First, in Section 4.1, we overview the proof of Lemma 4.2. Next, in Section 4.2, we give the main ideas for the proof of Lemma 4.4. In Section 4.3, we overview the proof of Lemma 4.5. After, in Section 4.4, we deal with our main lemmas, that is, we conclude Lemma 4.3 and give an overview for its approximate analogue we need for Theorem 1.3. In Section 4.5, we prove Lemma 4.1 and conclude Theorem 1.2. Finally, in Section 4.6, we describe the necessary adaptations to prove Theorem 1.3.

#### 4.1 Restriction to dismembered instances

This section is dedicated to justifying that it suffices to prove our main results for connected, dismembered instances of WAMMRO. More concretely, we prove Lemma 4.2.

The strategy for the proof of Lemma 4.2 can be described as follows. Given a connected instance  $(G, w)$  of WAMMRO, we wish to find a sufficiently small collection of edges in  $E(G)$  whose orientation we guess and that has the property that each of the thus obtained partial orientations of  $G$  together with  $w$  forms a dismembered instance of WAMMRO. Formally, given an undirected graph  $G$  and some  $X \subseteq V(G)$ , we say that a set  $F \subseteq E(G)$  is *dismembering* for  $(G, X)$  if every component of  $G \setminus F$  either contains at most one vertex of  $X \cup V(F)$  or it contains no vertex of  $X$  and exactly two vertices of  $V(F)$  each of which is incident to exactly one edge in  $F$ . The following result is the main lemma for our proof.

► **Lemma 4.6.** *Let  $G$  be an undirected graph and let  $F_0 \subseteq E(G)$  be an edge set such that every component of  $G \setminus F_0$  is a tree. Then there exists a dismembering set  $F$  for  $(G \setminus F_0, V(F_0))$  with  $|F| \leq 10|F_0|$ . Moreover, we can compute  $F$  in polynomial time.*

We now give an overview of the proof of Lemma 4.6. We deal separately with the components of  $G \setminus F_0$ . Let  $T$  be a component of  $G \setminus F_0$  and let  $X = V(T) \cap V(F_0)$ . Further, let  $T'$  be obtained from  $T$  by recursively deleting leaves which are not contained in  $X$ . We now add all edges in  $E(T')$  to  $F$  that are incident to a vertex in  $X$  or a vertex that is of degree at least 3 in  $T'$ . We do this for all components of  $G \setminus F_0$  and let  $F$  be the finally obtained set. It turns out that  $F$  is a dismembering set for  $(G \setminus F_0, V(F_0))$  of appropriate size. The details are omitted.

With Lemma 4.6 at hand, it is not difficult to prove Lemma 4.2. Given a connected instance  $(G, w)$ , we use Lemma 4.6 to compute a dismembering set  $F$  for  $(UG(G), V(F_0))$ , where  $F_0$  is the set of edges in  $E(UG(G))$  that correspond to arcs in  $A(G)$ . We then obtain a collection of dismembered instances of WAMMRO by trying all possible orientations of the edges in  $F$ . We solve all obtained instances by the algorithm whose existence we suppose. We output the best orientation  $\vec{G}$  of  $G$  we find. It turns out that  $\vec{G}$  is an optimal orientation for  $(G, w)$ . Finally, we show that the condition of  $G$  being connected can also be omitted. This completes the proof of Lemma 4.2. The details are omitted.

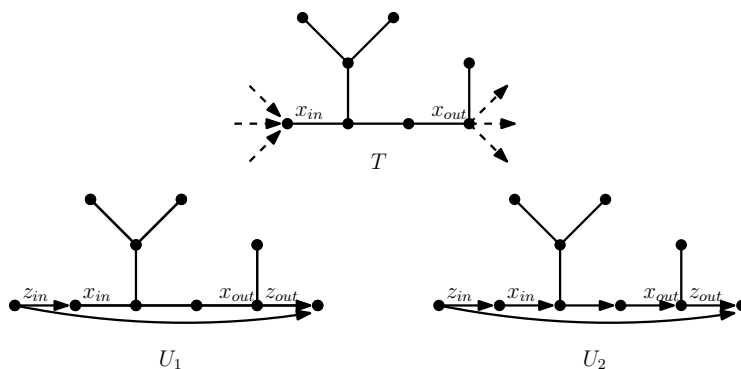
## 4.2 Simulation sets

We here prove Lemma 4.4, omitting some technical parts. We do this by distinguishing two cases depending on the exact interaction of the undirected component  $T$  with  $A(G)$ . In either case, we explicitly construct the simulation set and prove that it has the desired properties.

**Proof sketch of Lemma 4.4.** We use  $\bar{V}$  for  $V(G) \setminus V(T)$ . Recall that  $T$  is a tree as  $T$  is an undirected graph by definition and a mixed subgraph of  $G$ , which is acyclic. We now distinguish two cases depending on the exact interaction of  $T$  with  $A(G)$ . It follows from the fact that  $(G, w)$  is a dismembered instance of WAMMRO that one of the following cases occurs.

► **Case 1.**  $\delta_G^-(V(T)) \subseteq \delta_G^-(x_{in})$  and  $\delta_G^+(V(T)) \subseteq \delta_G^+(x_{out})$  for some  $x_{in}, x_{out} \in V(T)$ .

We first construct a mixed graph  $U_1$  from  $T$  by adding two vertices  $z_{in}$  and  $z_{out}$  and the arcs  $z_{in}x_{in}, x_{out}z_{out}$ , and  $z_{in}z_{out}$ . We further obtain  $U_2$  from  $U_1$  by orienting all edges on the unique  $x_{in}x_{out}$ -path in  $T$  so that we obtain a directed  $x_{in}x_{out}$ -path and we set  $\mathcal{U} = \{U_1, U_2\}$ . An illustration of  $\mathcal{U}$  can be found in Figure 2. Clearly, we have that  $\mathcal{U}$



■ **Figure 2** An example of the construction in the proof of Case 1. On top, there is an example of an undirected component  $T$ , where the dashed halfarcs mark the arcs in  $A(G)$  incident to  $V(T)$ . The corresponding mixed graphs  $U_1$  and  $U_2$  are depicted below.

can be computed in polynomial time and both  $U_1$  and  $U_2$  are compatible with  $G$ . We will show that  $\mathcal{U}$  is an arboresque 3-simulation set for  $((G, w), T)$ . We clearly have that  $U_1$  and  $U_2$  are mixed 3-supergraphs of  $T$ . Next observe that  $d_{UG(U_1)}(z_{in}) = d_{U_1}^+(z_{in}) = 2$  and  $UG(U_1 \setminus \{z_{in}, z_{out}\})$  is a tree. Hence  $U_1$  is arboresque. A similar argument shows that  $U_2$  is also arboresque. For the last property, let  $\vec{G}_1$  be an orientation of  $G$ . We define the weight function  $w' : V(T) \rightarrow [n]_0$  by  $w'(v) = w(v)$  for all  $v \in V(T)$ ,  $w'(z_{in}) = \sum_{v \in \bar{V}} \kappa_{\vec{G}_1}(v, x_{in})w(v)$  and  $w'(z_{out}) = \sum_{v \in \bar{V}} \kappa_{\vec{G}_1}(x_{out}, v)w(v)$ . Observe that  $w'$  is consistent with  $w$  and  $|w'| \leq |w|$ . We now choose  $U = U_1$  if  $\kappa_{\vec{G}_1}(x_{in}, x_{out}) = 0$  and  $U = U_2$  if  $\kappa_{\vec{G}_1}(x_{in}, x_{out}) = 1$ .

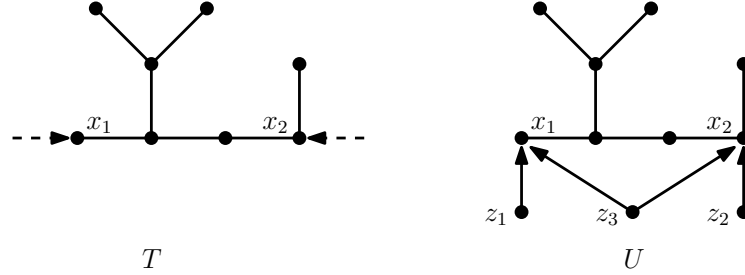
Now let  $\vec{U}$  be an orientation for  $(U, w')$ . It follows directly by construction that  $\vec{G}_1(\vec{U})$  is an orientation of  $G$  and by the definition of  $U$  that  $\vec{U}(\vec{G}_1)$  is an orientation of  $U$ . The details of the proof that  $R(\vec{G}_1(\vec{U}), w) - R(\vec{G}_1, w) \geq R(\vec{U}, w') - R(\vec{U}(\vec{G}_1), w')$  are omitted.

Roughly speaking,  $z_{in}$  represents all the vertices in  $V(G) \setminus V(T)$  from which  $x_{in}$  is reachable and  $z_{out}$  represents all the vertices in  $V(G) \setminus V(T)$  which are reachable from  $x_{out}$ . Moreover, the choice of  $U$  reflects whether there may be directed paths linking distinct vertices in  $V(G) \setminus V(T)$  that contain vertices in  $V(T)$ .

► **Case 2.** One of  $\delta_G^+(V(T))$  and  $\delta_G^-(V(T))$  is empty and  $|V(T) \cap V(A(G))| = 2$ .

By symmetry, we may suppose that  $\delta_G^+(V(T)) = \emptyset$ . Let  $x_1, x_2$  be the two unique vertices in  $V(T)$  incident to an arc of  $\delta_G^-(V(T))$ .

We now construct a mixed graph  $U$  from  $T$  by adding three vertices  $z_1, z_2,$  and  $z_3$  and the arcs  $z_1x_1, z_2x_2, z_3x_1,$  and  $z_3x_2$ . We clearly have that  $U$  is a mixed 3-supergraph of  $T$  and that  $U$  is compatible with  $G$ . Next observe that  $d_{UG(U)}(z_3) = d_U^+(z_3) = 2$  and  $UG(U \setminus \{z_3x_1\})$  is a tree. Hence  $U$  is arboresque. We set  $\mathcal{U} = \{U\}$ . An illustration of  $\mathcal{U}$  can be found in Figure 3. Clearly, we have that  $\mathcal{U}$  can be computed in polynomial time.



■ **Figure 3** An example of the construction in the proof of Case 2. On the left, there is an example of an undirected component  $T$ , where the dashed halfarcs mark the arcs in  $A(G)$  incident to  $V(T)$ . The corresponding mixed graph  $U$  is depicted on the right.

For the last property of simulation sets, let  $\vec{G}_1$  be an orientation of  $G$ . Before defining  $w'$ , we need some further definitions. We let  $\overline{V}_1(\overline{V}_2, \overline{V}_3)$  consist of all  $v \in \overline{V}$  that satisfy  $\kappa_{\vec{G}_1 - \{x_2\}}(v, x_1) = 1$  and  $\kappa_{\vec{G}_1 - \{x_1\}}(v, x_2) = 0$  ( $\kappa_{\vec{G}_1 - \{x_2\}}(v, x_1) = 0$  and  $\kappa_{\vec{G}_1 - \{x_1\}}(v, x_2) = 1$ ,  $\kappa_{\vec{G}_1 - \{x_2\}}(v, x_1) = 1$  and  $\kappa_{\vec{G}_1 - \{x_1\}}(v, x_2) = 1$ ). We now define  $w' : V(U) \rightarrow [n]_0$  by  $w'(z_i) = w(\overline{V}_i)$  for  $i \in [3]$  and  $w'(v) = w(v)$  for all  $v \in V(T)$ .

Now let  $\vec{U}$  be an orientation of  $U$ . It follows directly by construction that  $\vec{G}_1 \langle \vec{U} \rangle$  is an orientation of  $G$  and that  $\vec{U} \langle \vec{G}_1 \rangle$  is an orientation of  $U$ . The details of the proof that  $R(\vec{G}_1 \langle \vec{U} \rangle, w) - R(\vec{G}_1, w) \geq R(\vec{U}, w') - R(\vec{U} \langle \vec{G}_1 \rangle, w')$  are omitted.

Roughly speaking,  $z_1$  represents  $V(G) \setminus V(T)$  which can reach the vertices in  $V(T)$  through  $x_1$ , but not through  $x_2$ ,  $z_2$  represents  $V(G) \setminus V(T)$  which can reach  $V(T)$  through  $x_2$ , but not through  $x_1$ , and  $z_3$  represents the vertices in  $V(G) \setminus V(T)$  which can reach the vertices in  $V(T)$  through either of  $x_1$  and  $x_2$ . The addition of  $z_3$  is important to avoid double counting certain reachabilities. ◀

### 4.3 Arboresque mixed graphs

We here give an overview of the proof of Lemma 4.5.

For instances of WAMMRO containing a mixed graph whose underlying graph is a tree, a rather simple dynamic programming algorithm is available. We root the tree at an arbitrary vertex and use a bottom-up approach, where for every vertex  $v$ , we store whether there exists an orientation of the mixed graph  $T_v$  corresponding to the subtree rooted at  $v$  meeting three conditions: the objective value inside the orientation of  $T_v$  is above a certain threshold, the weight of the vertices reachable from  $v$  is above a certain threshold, and the weight of the vertices from which  $v$  is reachable is above a certain threshold.

For an arboresque mixed graph  $G$ , there exists the additional arc  $rs$ , which forces us to be a bit more careful about the dynamic program. We now give a sketch of the proof of Lemma 4.5.

**Proof sketch of Lemma 4.5.** As  $(G, w)$  is arboresque and by symmetry, we may suppose that  $A(G)$  contains an arc  $rs$  such that  $d_G^+(r) = d_{UG(G)}(r) = 2$  and the underlying graph  $T$  of  $G \setminus \{rs\}$  is a tree. Observe that  $r$  is a leaf of  $T$ . In the following, we consider  $T$  to be rooted at  $s$ . For every  $v \in V(G)$ , we let  $T_v$  be the unique mixed subgraph of  $G$  whose underlying graph is

the subtree of  $T$  rooted at  $v$ . We now start by running a dynamic program. In the following, for the sake of simplicity, we use  $Q$  for  $V(G) \times \{True, False\} \times [n]_0 \times [n]_0 \times [n^2]_0 \times [n]_0$ . For  $v \in V(G)$ , we use  $Q_v$  for the set of all  $q \in Q$  whose first entry is  $v$ .

For every  $q = (v, b, k_{in}, k_{out}, k_{reach}, k_{shared}) \in Q$ , we wish to compute a boolean value  $z[q]$  such that  $z[q] = True$  if and only if there exists an orientation  $\vec{T}_v$  of  $T_v$  that satisfies the following conditions:

- $b = True$  if and only if  $v$  is reachable from  $r$  in  $\vec{T}_v$ ,
- $w(In_{\vec{T}_v}(v)) \geq k_{in}$ ,
- $w(Out_{\vec{T}_v}(v)) \geq k_{out}$ ,
- $R(\vec{T}_v, w) \geq k_{reach}$ ,
- $w(Out_{\vec{T}_v}(v) \cap Out_{\vec{T}_v}(r)) \leq k_{shared}$ .

We now compute  $z[q]$  for all  $q \in Q$  by dynamic programming. First, for some  $v \in V(G) \setminus \{s\}$  that is a leaf of  $T$ , it is straight forward to decide the value of  $z[q]$  for all  $q \in Q_v$ . If  $v$  is not a leaf, for every  $q \in Q_v$ , we run a second dynamic program to compute  $z[q]$ . Here, we start with  $v$  and consider the mixed subgraphs of  $T_v$  constructed by continuously adding  $T_w$  for more and more children  $w$  of  $v$ .

Now suppose that, we have computed  $z[q]$  for all  $q \in Q$ . We now compute the largest  $K$  for which there exists some  $q = (s, b, k_{in}, k_{out}, k_{reach}, k_{shared}) \in Q_s$  with  $z[q] = True$  and  $k_{reach} + w(r)(k_{out} - k_{shared}) \geq K$ . This value is the maximum of  $R(\vec{G}, w)$  over all orientations  $\vec{G}$  of  $G$ , as, due to the choice of  $r$  and  $s$ , for any orientation  $\vec{G}$  of  $G$ , we have

$$\begin{aligned} R(\vec{G}, w) &= R(\vec{G} \setminus \{rs\}, w) + w(r)w(Out_{\vec{G} \setminus \{rs\}}(s) \setminus Out_{\vec{G} \setminus \{rs\}}(r)) \\ &= R(\vec{G} \setminus \{rs\}, w) + w(r)(w(Out_{\vec{G} \setminus \{rs\}}(s)) - w(Out_{\vec{G} \setminus \{rs\}}(s) \cap Out_{\vec{G} \setminus \{rs\}}(r))). \blacktriangleleft \end{aligned}$$

#### 4.4 Main lemmas

This section deals with Lemma 4.3 and its analogue for approximate solutions. We start by giving the proof of Lemma 4.3.

**Proof of Lemma 4.3.** We initialize  $\mathcal{T} = \emptyset$ . By Lemma 4.4, in polynomial time, we can compute an arboresque 3-simulation set  $\mathcal{U}$  for  $((G, w), T)$  of size at most 2. Now consider some  $U \in \mathcal{U}$  and let  $w' : V(U) \rightarrow [n]_0$  be a weight function with  $|w'| \leq |w|$  that is consistent with  $w$ . As  $U \in \mathcal{U}$  is arboresque, by Lemma 4.5, in polynomial time, we can compute an orientation  $\vec{U}_{w'}$  of  $U$  that maximizes  $R(\vec{U}_{w'}, w')$ . Let  $\vec{T}_{U, w'}$  be the inherited orientation of  $T$ . We now add  $\vec{T}_{U, w'}$  to  $\mathcal{T}$ . We do this for every  $U \in \mathcal{U}$  and every weight function  $w' : V(U) \rightarrow [n]_0$  with  $|w'| \leq |w|$  that is consistent with  $w$ . The final assignment of  $\mathcal{T}$  is obtained after the last one of these operations.

Observe that after the last of these operations, we have that  $|\mathcal{T}| = O(n^3)$  as  $|\mathcal{U}| \leq 2$  and there are  $O(n^3)$  choices for  $w'$  as  $w'$  needs to be consistent with  $w$  and  $|w'| \leq |w|$  needs to be satisfied. Moreover, as  $\vec{T}_{U, w'}$  can be computed in polynomial time for every  $U \in \mathcal{U}$  and every considered function  $w'$ ,  $|\mathcal{U}| \leq 2$ , and there are only  $O(n^3)$  choices for  $w'$ , we obtain that  $\mathcal{T}$  can be computed in polynomial time.

We now show that  $\mathcal{T}$  is an optimal replacement set. Consider an orientation  $\vec{G}$  of  $G$ . As  $\mathcal{U}$  is a simulation set for  $((G, w), T)$ , there exists some  $U \in \mathcal{U}$  and a function  $w' : V(U) \rightarrow [n]_0$  that is consistent with  $w$  with  $|w'| \leq |w|$  such that for every orientation  $\vec{U}$  for  $(U, w')$ , we have that  $\vec{G} \langle \vec{U} \rangle$  is an orientation of  $G$ ,  $\vec{U} \langle \vec{G} \rangle$  is an orientation of  $U$  and  $R(\vec{G} \langle \vec{U} \rangle, w) - R(\vec{G}, w) \geq R(\vec{U}, w') - R(\vec{U} \langle \vec{G} \rangle, w')$ . By the optimality of  $\vec{U}_{w'}$ , we obtain  $R(\vec{G} \langle \vec{T}_{U, w'} \rangle, w) = R(\vec{G} \langle \vec{U}_{w'} \rangle, w) \geq R(\vec{G}, w) + R(\vec{U}_{w'}, w') - R(\vec{U}_{w'} \langle \vec{G} \rangle, w') \geq R(\vec{G}, w)$ . As  $\vec{T}_{U, w'} \in \mathcal{T}$  by construction, the statement follows.  $\blacktriangleleft$

In order to state an analogue of Lemma 4.3 for approximate solutions, we need some more definitions. Given some  $\epsilon > 0$ , an instance  $(G, w)$  of WAMMRO and an undirected component  $T$  of  $G$ , we say that a set  $\mathcal{T}$  of orientations of  $T$  is an  $\epsilon$ -optimal replacement set for  $((G, w), T)$  if for every orientation  $\vec{G}$  of  $G$ , there exists some  $\vec{T} \in \mathcal{T}$  such that  $R(\vec{G} \langle \vec{T} \rangle, w) \geq R(\vec{G}, w) - \epsilon|w|^2$ . We are now ready to state this result.

► **Lemma 4.7.** *Let  $\epsilon > 0$ , let  $(G, w)$  be a dismembered instance of WAMMRO and let  $T$  be an undirected component of  $G$ . Then, in polynomial time, we can compute an  $\epsilon$ -optimal replacement set  $\mathcal{T}$  for  $((G, w), T)$  of size  $f(\epsilon)$ .*

The idea is, in comparison to the the proof of Lemma 4.3, not to try all weight functions, but only a small set of weight functions such that every possible weight function is close to one in the set. The proof, which is otherwise similar to the one of Lemma 4.3, is omitted.

## 4.5 XP Algorithm

This section is dedicated to concluding Theorem 1.2. We first prove Lemma 4.1.

**Proof of Lemma 4.1.** Let  $\vec{G}$  be an optimal orientation for  $(G, w)$ . We now define a sequence  $\vec{G}_0, \vec{G}_1, \dots, \vec{G}_q$  of orientations of  $G$ . First, we set  $\vec{G}_0 = \vec{G}$ . We now recursively define  $\vec{G}_i$  for  $i \in [q]$ . Let  $i \in [q]$  and suppose that  $\vec{G}_{i-1}$  has already been defined. As  $\mathcal{T}_i$  is an optimal replacement set for  $T_i$ , there exists some  $\vec{T}_i \in \mathcal{T}_i$  such that  $R(\vec{G}_{i-1} \langle \vec{T}_i \rangle, w) \geq R(\vec{G}_{i-1}, w)$ . Let  $\vec{G}_i = \vec{G}_{i-1} \langle \vec{T}_i \rangle$ . This finishes the definition of  $\vec{G}_0, \vec{G}_1, \dots, \vec{G}_q$ . Observe that by the definition of  $\vec{G}_0, \vec{G}_1, \dots, \vec{G}_q$  and as  $\vec{G}$  is an optimal orientation for  $(G, w)$ , we have that  $\vec{G}_q$  is an optimal orientation for  $(G, w)$ . Further observe that  $\vec{G}_q[V(T_i)] = \vec{G}_i[V(T_i)] = \vec{T}_i$  for  $i \in [q]$ . It follows that  $\vec{G}_q$  has the desired properties. ◀

With Lemma 4.1 at hand, it is easy to conclude a version of Theorem 1.2 for WAMMRO and connected, dismembered instances. Namely, given an instance  $(G, w)$ , we first use Lemma 4.3 to compute optimal replacement sets of polynomial size for all undirected components of  $(G, w)$ . Then, we try all orientations of  $G$  such that the inherited orientations of the undirected components come from these optimal replacement sets. By Lemma 4.1, we find an optimal orientation for  $(G, w)$  during this procedure. Next, as the optimal replacement sets are of polynomial size and there are only  $O(k)$  undirected components of  $G$ , the number of orientations we consider is  $n^{O(k)}$ . As the objective value of every orientation can be computed in polynomial time, the statement follows. Afterwards, Theorem 1.2 follows from Lemma 4.2 and Proposition 2.1. The details are omitted.

## 4.6 FPT approximation scheme

This section is dedicated to giving an overview of the proof of Theorem 1.3. While the remaining parts of the proof are similar to the proof of Theorem 1.2, the main additional difficulty is that applying these techniques, first a solution is obtained whose objective value is an additive amount away from an optimal solution rather than a multiplicative one. This problem is taken care of by the following result.

► **Lemma 4.8.** *Let  $\epsilon > 0$ , let  $(G, w)$  be a connected instance of WAMMRO with  $k \geq 1$  and let  $\vec{G}_0$  be an orientation of  $G$  such that  $R(\vec{G}_0, w) \geq R(\vec{G}, w) - \frac{\epsilon}{196k^2}|w|^2$  holds for all orientations  $\vec{G}$  of  $G$ . Then  $\vec{G}_0$  is a  $(1 - \epsilon)$ -optimal orientation for  $(G, w)$ .*

In order to prove Lemma 4.8, we show that optimal solutions for any dismembered instance of WAMMRO attain a certain minimum value. In order to do so, we first prove a result for undirected graphs, relying on the methods provided in [15]. More precisely, we prove the following result.

► **Lemma 4.9.** *Let  $(G, w)$  be a connected instance of WAMMRO such that  $G$  is undirected with  $|w| \geq 2$ . Then there exists an orientation  $\vec{G}$  of  $G$  such that  $R(\vec{G}, w) \geq \frac{1}{49}|w|^2$ .*

The proof of Lemma 4.9 consists of finding a so-called centroid of the tree  $G$ , which, roughly speaking, is a vertex  $v \in V(G)$  such that for a maximum number of pairs in  $P(G)$ , their unique path in  $G$  goes through that vertex. Then, we can orient  $G$  such that a large number of vertices is reachable from  $v$  and  $v$  is reachable from a large number of vertices. This orientation turns out to have the desired bound, thus proving Lemma 4.9.

It is now not difficult to prove a weaker version of Lemma 4.9 for arbitrary instances  $(G, w)$  of WAMMRO. Namely, we focus on an undirected component of  $G$  of largest weight and apply Lemma 4.9. We then extend the obtained orientation to  $G$  in an arbitrary way. As the connectivities inside the undirected component are clearly maintained, we obtain the same lower bound for the orientation of  $G$ . As there are at most  $O(k)$  undirected components of  $G$ , we obtain a lower bound, which is sufficient to prove Lemma 4.8.

With Lemma 4.8 at hand, we can conclude Theorem 1.3 from Lemma 4.8, Lemma 4.7, Lemma 4.2, and Proposition 2.1. The details are omitted.

## 5 Conclusion

We hope for the results in this article to be the starting point for more research on the complexity of MMRO. There are several interesting follow-up questions that seem worthwhile studying. The perhaps most interesting one, in particular in the light of Theorem 1.1, is whether a constant-factor approximation algorithm is available.

► **Question 1.** *Is there a constant  $\alpha > 0$  such that there exists a polynomial-time  $\alpha$ -approximation algorithm for MMRO?*

Interestingly, the corresponding question for FPUMRO also seems to be open.

One approach for Question 1 is to determine whether for any instance  $(G, w)$  of WAMMRO, there exists an orientation of  $G$  that maintains a constant fraction of the reachabilities in  $G$  with respect to the given weight function.

► **Question 2.** *Does there exist a constant  $\alpha > 0$  such that for any instance  $(G, w)$  of WAMMRO, there exists an orientation  $\vec{G}$  for  $(G, w)$  such that  $R(\vec{G}, w) \geq \alpha R(G, w)$ ?*

It is easy to see that an affirmative answer to Question 2 implies an affirmative answer to Question 1. Moreover, it is natural to ask about a common strengthening of Theorem 1.2 and Theorem 1.3, namely if MMRO is actually FPT when parameterized by  $k$ .

► **Question 3.** *Is there an algorithm that computes an optimal orientation for any instance  $G$  of MMRO and runs in time  $f(k)n^{O(1)}$ ?*

One way to achieve this would be to improve Lemma 4.3 in the sense that an optimal replacement set of size only dependent on  $k$  is found rather than one of size  $O(n^3)$ . However, this is not possible as the following result shows. Its proof is omitted.

► **Proposition 5.1.** *For any positive integer  $q$ , there exists a connected, dismembered instance  $(G, w)$  of WAMMRO with  $k = 2$  and an undirected component  $T$  of  $G$  such that every optimal replacement set for  $((G, T), w)$  is of size at least  $q + 1$ .*

A further interesting question is how structural parameters influence the difficulty of MMRO. In particular, it would be good to understand how, given an instance  $(G, w)$  of WAMMRO, the treewidth  $tw$  of  $UG(G)$  influences the difficulty of the problem. As every undirected component of  $G$  is a tree, we obtain that  $tw \leq k + 1$ . It follows that an XP algorithm for  $tw$  would imply a qualitative version of Theorem 1.2. While it seems plausible that such an algorithm can be obtained by a dynamic programming approach, we believe that it would need to be significantly more technical than the algorithm presented in Section 4.3. Similarly, an FPT algorithm for  $tw$  would yield an affirmative answer for Question 3.

On a higher level, except for [16], to our best knowledge, this is the first time that the number of arcs has been considered as a parameter for orientation problems of mixed graphs. It would be interesting to consider this parameterization for other problems on mixed graphs. For this question to be meaningful, the problem needs to be solvable in undirected graphs, but hard in mixed graphs.

One candidate is the problem of finding a 2-vertex-connected orientation of a mixed graph  $G$ . This problem can be solved in polynomial time if  $G$  is an undirected graph [22] and is NP-hard in general mixed graphs [18]. It would hence be interesting to see whether this problem is XP or even FPT with respect to the number of arcs of  $G$ .

Another candidate is finding an orientation of a mixed graph that is a well-balanced orientation of its underlying graph. For the definition of a well-balanced orientation, see [12]. The strong orientation theorem of Nash-Williams states that every undirected graph has a well-balanced orientation [20]. However, the above described orientation problem in mixed graphs is NP-hard [3]. Again, the parameterization by the number of arcs could be considered.

---

## References

- 1 Esther M. Arkin and Refael Hassin. A note on orientations of mixed graphs. *Discrete Applied Mathematics*, 116(3):271–278, 2002. doi:10.1016/S0166-218X(01)00228-1.
- 2 Piotr Berman and Marek Karpinski. On some tighter inapproximability results (extended abstract). In Jiri Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming*, pages 200–209, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. doi:10.1007/3-540-48523-6\_17.
- 3 Attila Bernáth and Gwenaél Joret. Well-balanced orientations of mixed graphs. *Information Processing Letters*, 106:149–151, 2008. doi:10.1016/j.ipl.2007.10.014.
- 4 Dima Blokh, Danny Segev, and Roded Sharan. The Approximability of Shortest Path-Based Graph Orientations of Protein–Protein Interaction Networks. *Journal of Computational Biology*, 20(12):945–957, 2013. doi:10.1089/cmb.2013.0064.
- 5 Frank Boesch and Ralph Tindell. Robbins’s Theorem for Mixed Multigraphs. *The American Mathematical Monthly*, 87(9):716–719, 1980. doi:10.1080/00029890.1980.11995131.
- 6 Rajesh Chitnis, Andreas Emil Feldmann, and Ondřej Suchý. A Tight Lower Bound for Planar Steiner Orientation. *Algorithmica*, 81(8):3200–3216, 2019. doi:10.1007/s00453-019-00580-x.
- 7 Ángel Corberán and Gilbert Laporte. *Arc Routing: Problems, Methods, and Applications*. Society for Industrial and Applied Mathematics, USA, 2015.
- 8 Cygan, Kortsarz, and Nutov. Steiner forest orientation problems. *SIAM Journal Discrete on Mathematics*, 27(3), 2013. doi:10.1137/120883931.
- 9 Britta Dorn, Falk Hüffner, Dominikus Krüger, Rolf Niedermeier, and Johannes Uhlmann. Exploiting bounded signal flow for graph orientation based on cause–effect pairs. In Alberto Marchetti-Spaccamela and Michael Segal, editors, *Theory and Practice of Algorithms in (Computer) Systems*, pages 104–115, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. doi:10.1186/1748-7188-6-21.

- 10 Michael Elberfeld, Vineet Bafna, Iftah Gamzu, Alexander Medvedovsky, Danny Segev, Dana Silverbush, Uri Zwick, and Roded Sharan and. On the Approximability of Reachability-Preserving Network Orientations. *Internet Mathematics*, 7(4):209–232, 2011. doi:10.1080/15427951.2011.604554.
- 11 Michael Elberfeld, Danny Segev, Colin R. Davidson, Dana Silverbush, and Roded Sharan. Approximation algorithms for orienting mixed graphs. *Theoretical Computer Science*, 483:96–103, 2013. doi:10.1016/j.tcs.2012.03.044.
- 12 András Frank. *Connections in Combinatorial Optimization*, volume 38. Oxford University Press, NLD, 2012. doi:10.1016/j.dam.2011.09.003.
- 13 Iftah Gamzu and Moti Medina. Improved Approximation for Orienting Mixed Graphs. *Algorithmica*, 74(1):49–64, 2016. doi:10.1007/s00453-014-9932-2.
- 14 Gregory Gutin, Mark Jones, and Bin Sheng. Parameterized complexity of the k-arc Chinese Postman Problem. *Journal of Computer and System Sciences*, 84:107–119, 2017. doi:10.1016/j.jcss.2016.07.006.
- 15 S. Louis Hakimi, Edward F. Schmeichel, and Neal E. Young. Orienting Graphs to Optimize Reachability. *Information Processing Letters*, 63:229–235, 1997. doi:10.1016/S0020-0190(97)00129-4.
- 16 Tesshu Hanaka, Michael Lampis, Nikolaos Melissinos, Edouard Nemery, Hirotaka Ono, and Manolis Vasilakis. Structural Parameters for Steiner Orientation. In Ho-Lin Chen, Wing-Kai Hon, and Meng-Tsung Tsai, editors, *36th International Symposium on Algorithms and Computation (ISAAC 2025)*, volume 359 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:14, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ISAAC.2025.38.
- 17 Rafael Hassin and Nimrod Megiddo. On orientations and shortest paths. *Linear Algebra and its Applications*, 114-115:589–602, 1989. Special Issue Dedicated to Alan J. Hoffman. doi:10.1016/0024-3795(89)90481-3.
- 18 Florian Hörsch and Zoltán Szigeti. The complexity of 2-vertex-connected orientation in mixed graphs. *Discrete Optimization*, 48:100774, 2023. doi:10.1016/j.disopt.2023.100774.
- 19 Alexander Medvedovsky, Vineet Bafna, Uri Zwick, and Roded Sharan. An algorithm for orienting graphs based on cause-effect pairs and its applications to orienting protein networks. In Keith A. Crandall and Jens Lagergren, editors, *Algorithms in Bioinformatics*, pages 222–232, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-540-87361-7\_19.
- 20 Crispin St. John Alvah Nash-Williams. On Orientations, Connectivity and Odd-Vertex-Pairings in Finite Graphs. *Canadian Journal of Mathematics*, 12:555–567, 1960. URL: <https://www.cambridge.org/core/journals/canadian-journal-of-mathematics/article/on-orientations-connectivity-and-oddvertexpairings-in-finite-graphs/2B39912D1666161E4200E61E8F65DE2B>.
- 21 Marcin Pilipczuk and Magnus Wahlström. Directed Multicut is W[1]-hard, Even for Four Terminal Pairs. *ACM Trans. Comput. Theory*, 10(3), 2018. doi:10.1145/3201775.
- 22 Carsten Thomassen. Strongly 2-connected orientations of graphs. *Journal of Combinatorial Theory, Series B*, 110:67–78, 2015. doi:10.1016/j.jctb.2014.07.004.
- 23 Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. D’ya Like DAGs? A Survey on Structure Learning and Causal Discovery. *ACM Computing Surveys*, 55(4), 2022. doi:10.1145/3527154.
- 24 Michał Włodarczyk. Parameterized Inapproximability for Steiner Orientation by Gap Amplification. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 104:1–104:19, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2020.104.