Parameterized and Approximation Algorithms for the Maximum Bimodal Subgraph Problem

Walter Didimo[1], Fedor V. Fomin[2], Petr A. Golovach[2], Tanmay Inamdar[2], Stephen Kobourov[3], and Marie Diana Sieper[4](✉)

[1] Department of Engineering, University of Perugia, Perugia, Italy
walter.didimo@unipg.it
[2] Department of Computer Science, University of Bergen, Bergen, Norway
{fedor.fomin,petr.golovach,tanmay.inamdar}@uib.no
[3] Department of Computer Science, University of Arizona, Tucson, USA
kobourov@cs.arizona.edu
[4] Department of Computer Science, University of Würzburg, Würzburg, Germany
marie.sieper@uni-wuerzburg.de

Abstract. A vertex of a plane digraph is *bimodal* if all its incoming edges (and hence all its outgoing edges) are consecutive in the cyclic order around it. A plane digraph is bimodal if all its vertices are bimodal. Bimodality is at the heart of many types of graph layouts, such as upward drawings, level-planar drawings, and L-drawings. If the graph is not bimodal, the *Maximum Bimodal Subgraph (MBS)* problem asks for an embedding-preserving bimodal subgraph with the maximum number of edges. We initiate the study of the MBS problem from the parameterized complexity perspective with two main results: (i) we describe an FPT algorithm parameterized by the branchwidth (and hence by the treewidth) of the graph; (ii) we establish that MBS parameterized by the number of non-bimodal vertices admits a polynomial kernel. As the byproduct of these results, we obtain a subexponential FPT algorithm and an efficient polynomial-time approximation scheme for MBS.

Keywords: bimodal graphs · maximum bimodal subgraph · parameterized complexity · FPT algorithms · polynomial kernel · approximation scheme

Research started at the Dagstuhl Seminar 23162: New Frontiers of Parameterized Complexity in Graph Drawing, April 2023, and partially supported by: (*i*) University of Perugia, Ricerca Base 2021, Proj. "AIDMIX - Artificial Intelligence for Decision Making: Methods for Interpretability and eXplainability"; (*ii*) MUR PRIN Proj. 2022TS4Y3N - "EXPAND: scalable algorithms for EXPloratory Analyses of heterogeneous and dynamic Networked Data", (*iii*) MUR PRIN Proj. 2022ME9Z78 - "NextGRAAL: Next-generation algorithms for constrained GRAph visuALization", (*iv*) the Research Council of Norway project BWCA 314528, (*v*) the European Research Council (ERC) grant LOPPRE 819416, and (*vi*) NSF-CCF 2212130.

M. A. Bekos and M. Chimani (Eds.): GD 2023, LNCS 14466, pp. 189–202, 2023.
https://doi.org/10.1007/978-3-031-49275-4_13

1 Introduction

Let G be a plane digraph, that is, a planar directed graph with a given planar embedding. A vertex v of G is *bimodal* if all its incoming edges (and hence all its outgoing edges) are consecutive in the cyclic order around v. In other words, v is bimodal if the circular list of edges incident at v can be split into at most two linear lists, where all edges in the same list are either all incoming or all outgoing v. Graph G is *bimodal* if all its vertices are bimodal. Bimodality is a key property at heart of many graph drawing styles. In particular, it is a necessary condition for the existence of *level-planar* and, more generally, *upward planar* drawings, where the edges are represented as curves monotonically increasing in the upward direction according to their orientations [12–14, 24]; see Fig. 1a. Bimodality is also a sufficient condition for *quasi-upward planar* drawings, in which edges are allowed to violate the upward monotonicity a finite number of times at points called *bends* [5–7]; see Fig. 1b. It has been shown that bimodality is also a sufficient condition for the existence of *planar L-drawings* of digraphs, in which distinct L-shaped edges may overlap but not cross [1–3]; see Fig. 1c. A generalization of bimodality is k-modality. Given a positive even integer k, a plane digraph is *k-modal* if the edges at each vertex can be grouped into at most k sets of consecutive edges with the same orientation [26]. In particular, it is known that 4-modality is necessary for planar L-drawings [10].

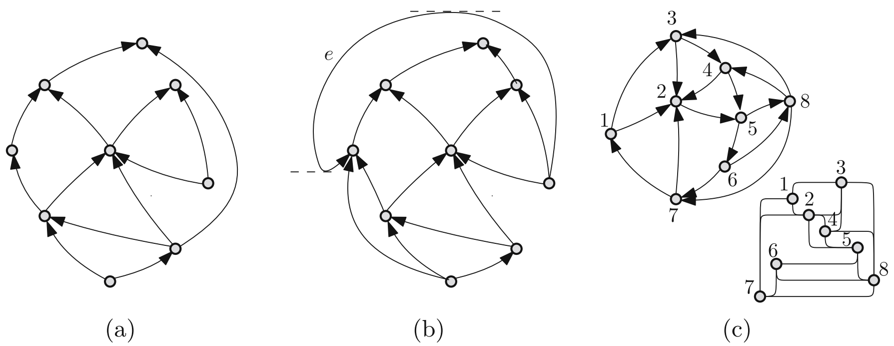

Fig. 1. (a) An upward planar drawing. (b) A quasi-upward planar drawing, where edge e makes two bends (the two horizontal tangent points). (c) A bimodal digraph (above) and a corresponding planar L-drawing (below).

While testing if a digraph G admits a bimodal planar embedding can be done in linear time [5], a natural problem that arises when G does not have such an embedding is to extract from G a subgraph of maximum size (i.e., with the maximum number of edges) that fulfills this property. This problem is NP-hard, even if G has a given planar embedding and we look for an embedding-preserving maximum bimodal subgraph [8]. We address exactly this fixed-embedding version of the problem, and call it the *Maximum Bimodal Subgraph* (MBS) problem.

Contribution. While a heuristic and a branch-and-bound algorithm are given in [8] to solve MBS (and also to find a maximum upward-planar digraph), here we study this problem from the parameterized complexity and approximability perspectives (refer to [11,18] for an introduction to parameterized complexity). More precisely, we consider the following more general version of the problem with weighted edges; it coincides with MBS when we restrict to unit edge weights.

MWBS(G, w) (*Maximum Weighted Bimodal Subgraph*). *Given a plane digraph G and an edge-weight function $w : E(G) \to \mathbb{Q}^+$, compute a bimodal subgraph of G of maximum weight, i.e., whose sum of the edge weights is maximum over all bimodal subgraphs of G.*
Our contribution can be summarized as follows.

– *Structural Parameterization.* We show that MWBS is FPT when parameterized by the *branchwidth* of the input digraph G or, equivalently, by the *treewidth* of G (Sect. 3). Our algorithm deviates from a standard dynamic approach for graphs of bounded treewidth. The main difficulty here is that we have to incorporate the "topological" information about the given embedding in the dynamic program. We accomplish this via the sphere-cut decomposition of Dorn et al. [16].

– *Kernelization.* Let b be the number of non-bimodal vertices in an input digraph G. We construct a polynomial kernel for the decision version of MWBS parameterized by b (Sect. 4). Our kernelization algorithm performs in several steps. First we show how to reduce the instance to an equivalent instance whose branchwidth is $\mathcal{O}(\sqrt{b})$. Second, by using specific gadgets, we compress the problem to an instance of another problem whose size is bounded by a polynomial of b. In other words, we provide a polynomial compression for MWBS. Finally, by the standard arguments, [18, Theorem 1.6], based on a polynomial reduction between any NP-complete problems, we obtain a polynomial kernel for MWBS.

By pipelining the crucial step of the kernelization algorithm with the branchwidth algorithm, we obtain a parameterized subexponential algorithm for MWBS of running time $2^{\mathcal{O}(\sqrt{b})} \cdot n^{\mathcal{O}(1)}$. Since $b \leq n$, this also implies an algorithm of running time $2^{\mathcal{O}(\sqrt{n})}$. Note that our algorithms are asymptotically optimal up to the *Exponential Time Hypothesis* (ETH) [21,22]. The NP-hardness result of MBS (and hence of MWBS) given in [8] exploits a reduction from PLANAR-3SAT. The number of non-bimodal vertices in the resulting instance of MBS is linear in the size of the PLANAR-3SAT instance. Using the standard techniques for computational lower bounds for problems on planar graphs [11], we obtain that the existence of an $2^{o(\sqrt{b})} \cdot n^{\mathcal{O}(1)}$-time algorithm for MBWS would contradict ETH.

– *Approximability.* We provide an Efficient Polynomial-Time Approximation Scheme (EPTAS) for MWBS, based on Baker's (or shifting) technique [4]. Namely, using our algorithm for graphs of bounded branchwidth, we give an $(1 + \epsilon)$-approximation algorithm that runs in $2^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1)}$ time.

Full proofs of the results marked with an asterisk (*), as well as additional definitions and technical details, are given in [15].

2 Definitions and Terminology

Let G be a digraph. We denote by $V(G)$ and $E(G)$ the set of vertices and the set of edges of G. Throughout the paper we assume that G is planar and that it comes with a planar embedding; such an embedding fixes, for each vertex $v \in V(G)$, the clockwise order of the edges incident to v. We say that G is a *planar embedded digraph* or simply that G is a *plane digraph*.

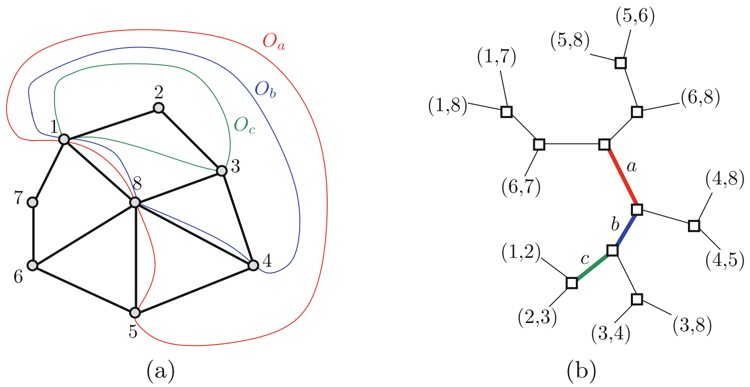

Fig. 2. A plane graph G and a sphere-cut decomposition of G; three nooses are highlighted on G for the arcs a, b, and c of the decomposition tree.

Branch Decomposition and Sphere-Cut Decomposition. A *branch decomposition* of a graph G defines a hierarchical clustering of the edges of G, represented by an unrooted proper binary tree, that is a tree with non-leaf nodes of degree three, whose leaves are in one-to-one correspondence with the edges of G. More precisely, a branch decomposition of G consists of a pair $\langle T, \xi \rangle$, where T is an unrooted proper binary tree and $\xi : \mathcal{L}(T) \leftrightarrow E(G)$ is a bijection between the set $\mathcal{L}(T)$ of the leaves of T and the set $E(G)$ of the edges of G. For each arc a of T, denote by T_1^a and T_2^a the two connected components of $T \setminus \{a\}$, and, for $i = 1, 2$, let G_i^a be the subgraph of G that consists of the edges corresponding to the leaves of T_i^a. The *middle set* $\mathrm{mid}(a) \subseteq V(G)$ is the intersection of the vertex sets of G_1^a and G_2^a, i.e., $\mathrm{mid}(a) := V(G_1^a) \cap V(G_2^a)$. The *width* $\beta(\langle T, \xi \rangle)$ of $\langle T, \xi \rangle$ is the maximum size of the middle sets over all arcs of T, i.e., $\beta(\langle T, \xi \rangle) = \max\{|\mathrm{mid}(a)| : a \in E(T)\}$. An *optimal branch decomposition* of G is a branch decomposition with minimum width; this width is called the *branchwidth* of G and is denoted by $\mathrm{bw}(G)$.

A sphere-cut decomposition is a special type of branch decomposition (see Fig. 2). Let G be a connected planar graph, topologically drawn on a sphere Σ. A *noose* O of G is a closed simple curve on Σ that intersects G only at vertices and that traverses each face of G at most once. The *length* of O is the number of vertices that O intersects. Note that, O bounds two closed discs Δ_O^1

and Δ_O^2 in Σ; we have $\Delta_O^1 \cap \Delta_O^2 = O$ and $\Delta_O^1 \cup \Delta_O^2 = \Sigma$. Let $\langle T, \xi \rangle$ be a branch decomposition of G. Suppose that for each arc a of T there exists a noose O_a that traverses exactly the vertices of $\text{mid}(a)$ and whose closed discs $\Delta_{O_a}^1$ and $\Delta_{O_a}^2$ enclose the drawings of G_1^a and of G_2^a, respectively. Denote by π_a the circular clockwise order of the vertices in $\text{mid}(a)$ along O_a and let $\Pi = \{\pi_a : a \in E(T)\}$ the set of all circular orders π_a. The triple $\langle T, \xi, \Pi \rangle$ is a *sphere-cut decomposition* of G. We assume that the vertices of $\text{mid}(a) = V(G_1^a) \cap V(G_2^a)$ are enumerated according to π_a. Since a noose O_a traverses each face of G at most once, both graphs G_1^a and G_2^a are connected. Also, the nooses are pairwise non-crossing, i.e., for any pair of nooses O_a and O_b, we have that O_b lies entirely inside $\Delta_{O_a}^1$ or entirely inside $\Delta_{O_a}^2$. For a noose O_a, we define $\text{mid}(O_a) = \text{mid}(a)$, or in general, we define $\text{mid}(\phi)$ to be the vertices cut by ϕ. We rely on the following result on the existence and computation of a sphere-cut decomposition [23] (see also [16]).

Proposition 1 ([23]). *Let G be a connected graph embedded in the sphere with n vertices and branchwidth $\ell \geq 2$. Then there exists a sphere-cut decomposition of G with width ℓ, and it can be computed in $\mathcal{O}(n^3)$ time.*

We remark that the branchwidth $\text{bw}(G)$ and the treewidth $\text{tw}(G)$ of a graph G are within a constant factor: $\text{bw}(G) - 1 \leq \text{tw}(G) \leq \lfloor \frac{3}{2} \text{bw}(G) \rfloor - 1$ (see [25]).

3 FPT Algorithms for MWBS by Branchwidth

In this section we describe an FPT algorithm parameterized by branchwidth. We first introduce configurations, which encode on which side of a closed curve and in what order in a bimodal subgraph for a vertex v the switches between incoming to outgoing edges happen.

Definition 1 ***(Configuration).*** *Let $C = \{(i), (o), (i,o), (o,i), (o,i,o), (i,o,i)\}$. Let G be a graph embedded in the sphere Σ, ϕ be a noose in Σ with a prescribed inside, $v \in \text{mid}\,(\phi)$, and $X \in C$. Let $E^{v,\phi}$ be the set of edges incident to v in ϕ. We say v has* configuration *X in ϕ, if $E^{v,\phi}$ can be partitioned into sets such that:*

1. *For every $x \in X$, there is a (possibly empty) set E_x associated with it.*
2. *Every set associated with an i (o) contains only in- (/out-) edges of v.*
3. *For every set, the edges contained in it are successive around v.*
4. *The sets E_x appear clockwise (seen from v) in the same order in G inside ϕ as the x appear in X.*

For every $v \in \text{mid}\,(\phi)$, let X_v be a configuration of v in ϕ. We say $X_\phi = \{X_v \mid v \in \text{mid}\,(\phi)\}$ is a configuration set *of ϕ.*

If G is bimodal, then for every noose ϕ and every vertex $v \in \text{mid}\,(\phi)$, v must have at least one configuration $X \in C$ in ϕ. Note that configurations and configuration sets are not unique, as seen in Fig. 3(a). A vertex can even have all configurations if it has no incident edges in ϕ. The next definition is needed to encode when configurations can be combined in order to obtain bimodal vertices.

Definition 2 ***(Compatible configurations).*** *Let $X, X', X^* \in C$ be configurations. We say X, X' are* compatible configurations *or short* compatible*, if by concatenating X, X' and deleting consecutive equal letters, the result is a substring of (o, i, o) or (i, o, i). Note that it is not important in which order we concatenate X, X'. See Fig. 3(b). We say X and X' are* compatible with respect to X^* *if by concatenating X, X' (in this order) and deleting consecutive equal letters, the result is a substring of X^*.*

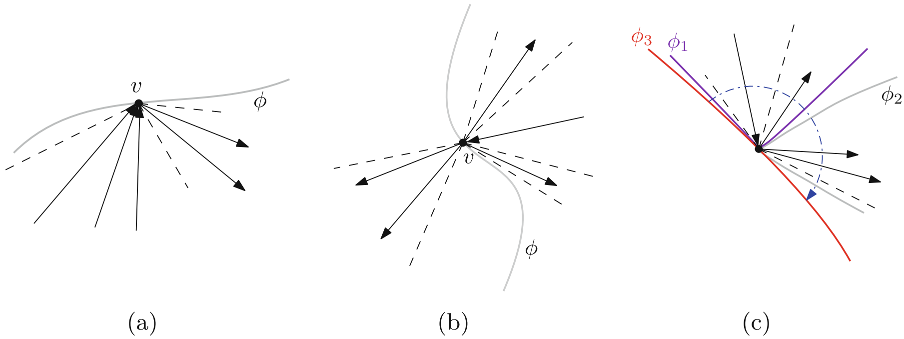

Fig. 3. (a) A vertex with configurations $(o, i), (o, i, o)$ and (i, o, i) in ϕ. The most restricted and thus minimal configuration is (o, i). (b) A vertex with configuration (o, i, o) in ϕ and (o) outside of ϕ. Concatenating (o, i, o) with (o) and deleting consecutive equal letters results in (o, i, o), the result is a substring of (o, i, o), thus (o, i, o) and (o) are compatible. (c) Note that ϕ_3 is composed of ϕ_1 and ϕ_2; the inside of ϕ_1, the inside of ϕ_2 and the outside of ϕ_3 are clockwise in this order around v with configuration (i, o) in ϕ_1 and (o) in ϕ_2. They can be concatenated to configuration (i, o) in ϕ_3, while (i, o) and (o) are compatible w.r.t. (i, o), but not (o, i).

A configuration X can have several compatible configurations, for example $(i, o) \in C$ is compatible with (o), (i) and (o, i). From these (o, i) is in some sense maximal, meaning that configurations (o) and (i) are substrings of (o, i). Given a configuration X, a *maximal compatible configuration* X' of X is a configuration that is compatible with X, and all other compatible configurations of X are substrings of X'. Observe that every configuration has a unique maximal compatible configuration, they are pairwise: $(i) - (i, o, i)$, $(o) - (o, i, o)$ and $(o, i) - (i, o)$.

We say a noose ϕ_3 is *composed* of the nooses ϕ_1 and ϕ_2, if the edges of G in ϕ_3 are partitioned by ϕ_1 and ϕ_2. If a noose ϕ_3 is composed of nooses ϕ_1 and ϕ_2, and there exists a vertex $v \in \mathrm{mid}(\phi_1) \cap \mathrm{mid}(\phi_2) \cap \mathrm{mid}(\phi_3)$, such that in ϕ_3 around v, all adjacent edges of v in ϕ_1 are clockwise before all adjacent edges of v in ϕ_2. If X, X' and X^* are nooses and X and X' are compatible with respect to X^*, and v has configuration X in ϕ_1 and configuration X' in ϕ_2, then it has configuration X^* in ϕ_3. See Fig. 3(c).

If a curve ϕ contains only one edge on its inside, finding maximal subgraphs for a configuration inside ϕ is easy.

Lemma 1 (*). *Let G be a graph embedded in the sphere Σ, let $e = \{u, v\}$ be an edge and let ϕ be a noose that cuts G only in u and v, such that e is in ϕ and all other edges are on the outside of ϕ. Let X_u, X_v be prescribed configurations. Then we can compute in $\mathcal{O}(1)$ time the maximum subgraph G' of G such that u, v have configuration X_u respectively X_v in ϕ in G'.*

We will now see how we can compute optimal subgraphs bottom-up.

Lemma 2 (*). *Let G be a graph embedded in the sphere Σ, let ϕ_1, ϕ_2, ϕ_3 be nooses with length at most ℓ each, and let $E_{\phi_1}, E_{\phi_2}, E_{\phi_3}$ be the sets of edges contained inside the respective noose with E_{ϕ_1}, E_{ϕ_2} being a partition of E_{ϕ_3}. Let X_{ϕ_3} be a configuration set for ϕ_3. Let further for every configuration set X_{ϕ_1} (X_{ϕ_2}) of ϕ_1 (ϕ_2), the maximum subgraph that has configuration set X_{ϕ_1} (X_{ϕ_2}) and is bimodal in ϕ_1 (ϕ_2) be known. Then a maximum subgraph G' of G that has configuration set X_{ϕ_3} and is bimodal in ϕ_3 can be computed in $\mathcal{O}(6^{2\ell}) \cdot n^{\mathcal{O}(1)}$ time.*

If a noose ϕ contains only $e \in E$, we have only two options in ϕ: delete e or do not. Testing which is optimal can be done in constant time, this leads to Lemma 1. Now let ϕ_3 be a noose that contains more than one edge, let ϕ_1, ϕ_2 be two nooses that partition the inside of ϕ_3, and let X_{ϕ_3} be a given configuration set. If we already know optimal solutions for any given configuration set in ϕ_1 (ϕ_2) (which we already computed when traversing the sphere-cut decomposition bottom up), we can guess for some optimal solution for ϕ_3 for every $v \in \mathrm{mid}(\phi_1) \cap \mathrm{mid}(\phi_2)$ the configuration it has in ϕ_1 and in ϕ_2. This gives us configuration sets X_{ϕ_1} and X_{ϕ_2} for ϕ_1 and ϕ_2, respectively (for every $v \in \mathrm{mid}(\phi_1) \setminus \mathrm{mid}(\phi_2)$ we take its configuration in X_{ϕ_3}). We obtain the corresponding solution G' that coincides with the optimal solution for ϕ_1 (ϕ_2) in ϕ_1 (ϕ_2) respecting X_{ϕ_1} (X_{ϕ_2}) and that coincides with G outside of ϕ_3. Since $|\mathrm{mid}(\phi_1) \cap \mathrm{mid}(\phi_2)| \leq \ell$, we achieve the same by enumerating all possible configurations for $\mathrm{mid}(\phi_1) \cap \mathrm{mid}(\phi_2)$, compute the corresponding solutions and take the maximum in $\mathcal{O}(6^{2\ell}) \cdot n^{\mathcal{O}(1)}$ time, leading to Lemma 2. We now obtain the following theorem.

Theorem 1 (*). *There is an algorithm that solves* MWBS(G, w) *in* $2^{\mathcal{O}(\mathrm{bw}(G))} \cdot n^{\mathcal{O}(1)}$ *time. In particular,* MWBS *is FPT when parameterized by branchwidth.*

Proof (Sketch). Assume that G is connected (otherwise process every connected component independently). If $\mathrm{bw}(G) = 1$, G is a star and we can compute an optimal solution in polynomial time. Otherwise, according to Proposition 1 we can compute a sphere-cut decomposition $\langle T, \xi, \Pi \rangle$ for G with optimal width ℓ. We pick any leaf of T to be the root r of T. For every noose O corresponding to an arc of T let X_O be a configuration set for O. Then we define $E_{(O,X_O)}$ to be edge set of minimum weight, such that $G \setminus E_{(O,X_O)}$ is bimodal inside of O and has configuration set X_O in O. We now compute the $E_{(O,X_O)}$ bottom-up. For a noose O corresponding to a leaf-arc in T, Lemma 1 shows that we can compute all possible values of $E_{(O,X_O)}$ in linear time. For a noose O corresponding to a non-leaf arc in T, Lemma 2 shows that we can compute E_{O,X_O} for a given X_O in $\mathcal{O}(6^{2\ell}) \cdot n^{\mathcal{O}(1)}$ time, and thus all entries for O in $\mathcal{O}(6^{3\ell}) \cdot n^{\mathcal{O}(1)}$ time. Let $e \in E$

be the edge associated with r. We have only two options left, delete e or do not. In both cases we obtain the optimal solution for the rest of G from the values $E_{(O,X_O)}$. The overall running time is $2^{\mathcal{O}(\ell)} \cdot n^{\mathcal{O}(1)}$. □

Since our input graphs are planar, we immediately obtain a subexponential algorithm for MWBS because for a planar graph G, $\mathrm{bw}(G) = \mathcal{O}(\sqrt{n})$ [19].

Theorem 2. *$MWBS(G = (V, E), w)$ can be solved in $2^{\mathcal{O}(\sqrt{n})}$ time.*

4 Compression for MWBS by b

Throughout this section we assume that (i) the weights are rational, that is, for (G, w), $w\colon V(G) \to \mathbb{Q}^+$ and (ii) we consider the decision version of MWBS, that is, additionally to (G, w), we are given a target value $W \in \mathbb{Q}^+$ and the task is to decide whether G has a bimodal subgraph G^* with $w(E(G^*)) \geq W$.

Further Definitions. For simplicity, we say that a bimodal vertex of G is a *good* vertex, and that a non-bimodal vertex is a *bad* vertex. We denote by $\mathcal{G}(G)$ and $\mathcal{B}(G)$ the sets of good and bad vertices of G, respectively. Given a vertex $v \in V(G)$, an *in-wedge* (resp. *out-wedge*) of v is a maximal circular sequence of consecutive incoming (resp. outgoing) edges of v. Clearly, if v is bimodal it has at most one in-wedge and at most one out-wedge. Given a vertex $v \in \mathcal{B}(G)$, a *good edge-section* of v is a maximal consecutive sequence of in- and out- wedges of v, such that no edge is incident to another bad vertex.

Observation 1. Let (G, w) be an instance of MWBS with b bad vertices, and let $v \in \mathcal{B}(G)$. Then v can have at most $b - 1$ good edge-sections.

We introduce a generalization of MWBS called Cut-MWBS$(G, w, \mathcal{E})$ (*maximum weighted bimodal subgraph with prescribed cuts*). Given a plane digraph G, an edge-weight function $w : E(G) \to \mathbb{Q}^+$, and a partition $\mathcal{E}$ of $E(G)$, compute a bimodal subgraph G' of G of maximum weight, i.e., whose sum of the edge weights is maximum over all bimodal subgraphs of G, under the condition that for every set $E_i \in \mathcal{E}$, either all $e \in E_i$ are still present in G' or none of them are. We can see that every instance (G, w) of MWBS is equivalent to the instance $(G, w, \{\{e\} \mid e \in E(G)\})$ of Cut-MWBS, and thus Cut-MWBS is NP-hard. Also, the decision variant of the problem is NP-complete.

We now give reduction rules for the MWBS to Cut-MWBS compression, and prove that each of them is *sound*, i.e., it can be performed in polynomial time and the reduced instance is solvable if and only if the starting instance is solvable.

Reduction Rule 1. Let (G, w) be an instance of MWBS, and $v \in V(G)$ be an isolated vertex. Then, let (G', w) be the new instance, where $V(G') = V(G) \setminus \{v\}$.

Reduction Rule 2. Let (G, w) be an instance of MWBS with the target value W, and $u, v \in \mathcal{G}(G)$ be such that (u, v) is an edge. Then, the resulting instance is (G', w), where $G' = G - (u, v)$, and the new target value is $W' = W - w(u, v)$.

Reduction Rule 3. Let (G, w) be an instance of MWBS and $v \in \mathcal{G}(G)$ of degree ≥ 2. Let (G', w) be the new instance, where in G' we replace each edge $e = (u, v)$ (resp. $e = (v, u)$) where $u \in \mathcal{G}(G)$ with another edge $e' = (u, x_{uv})$ (resp. $e' = (x_{uv}, u)$), where x_{uv}'s are distinct vertices created for each such edge, and each e' is embedded within the embedding of e, where $w(e') = w(e)$ (see Fig. 4).

Claim 1 **(*).** Reduction rules 1, 2 and 3 are sound.

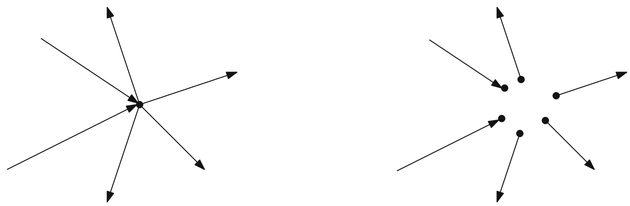

Fig. 4. A bimodal vertex (a) before and (b) after Reduction rule 3 is applied.

By applying Reductions 1, 2 and 3 exhaustively, we get Lemma 3, which is already enough to give a subexponential FPT algorithm by b (Theorem 3).

Lemma 3 (*). *Given an instance (G, w) of MWBS, there exists a polynomial-time algorithm to obtain an equivalent instance (G', w) with G' being a subgraph of G, such that (i) $|\mathcal{B}(G')| \leq |\mathcal{B}(G)|$, (ii) $\mathcal{G}(G')$ is an independent set in G', and (iii) for all $v \in \mathcal{G}(G')$, $\deg(v) = 1$ in the underlying graph of G'.*

Theorem 3. *There exists an algorithm that solves MWBS(G, w) with b bad vertices in $2^{\mathcal{O}(\sqrt{b})} \cdot n^{\mathcal{O}(1)}$ time.*

Proof. By Lemma 3, (G, w) is equivalent to (G', w) with at most b vertices of degree > 1, which we can compute in polynomial time. This implies $\mathrm{bw}(G') = \mathcal{O}(\mathrm{tw}(G')) = \mathcal{O}(\sqrt{b})$, and we can apply Theorem 1 to obtain an algorithm that computes a solution for (G', w) in $2^{\mathcal{O}(\mathrm{bw}(G'))}|V(G')|^{\mathcal{O}(1)}$ time. □

We now describe how we can partition, for a given input, all good-edge sections into edge sets in such a way that there exists an optimal solution in which every set is either contained or deleted completely, and the total number of sets is bounded in a function of b. We will then show how we can replace the sets with edge sets of size at most two. The main difficulty will be to ensure that sets that exclude each other continue to do so in the reduced instance.

Lemma 4 (*). *Let (G, w) be an instance of MWBS with n vertices and b bad vertices, such that $\mathcal{G}(G)$ is an independent set in G and $\deg(v) = 1$ for all $v \in \mathcal{G}(G)$. Let further $v \in \mathcal{B}(G)$, and let S be a good edge-section of v. Then S can be partitioned into at most 26 sets $S_1, \ldots, S_{26}$, such that for every optimal* *solution $G' \subseteq G$ of $MWBS(G, w)$, there exists an optimal solution $G^* \subseteq G$ of $MWBS(G, w)$, such that G' and G^* coincides on $G \setminus S$, and for every i, S_i is either contained or removed completely in G^*.*

Further, there exists a partition $P_1, \ldots, P_j$ of $\{S_1, \ldots, S_{26}\}$, such that for all P_i: (1) $|P_i| \leq 2$, (2) the edges in P_i are consecutive in S and (3) if $P_i = \{\mathcal{S}_1, \mathcal{S}_2\}$, then $\mathcal{S}_1$ consists of outgoing edges of v iff $\mathcal{S}_1$ consists of incoming edges of v, and at least one of $\mathcal{S}_1, \mathcal{S}_2$ does not form a set of consecutive edges in S.

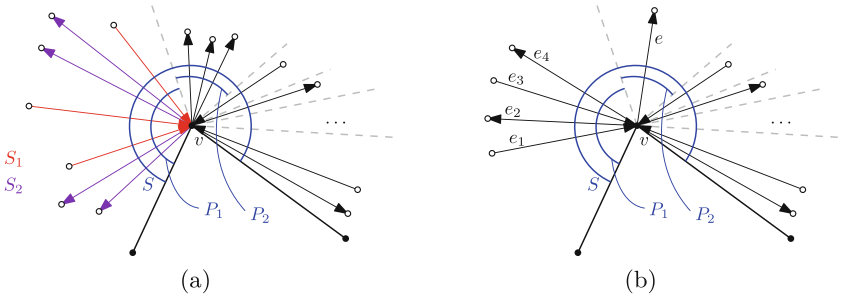

Fig. 5. (a) Illustration for Lemmas 4 and 5. The gray dashed lines correspond to a set of switches between the optimal solution we will choose; they impose the partition $P_1, \ldots, P_{13}$. S_1 (S_2) are the incoming (outgoing) edges of P_1, respectively. (b) The same vertex after transition to Cut-MWBS by Lemma 5, and after Reduction Rule 4 (5) got applied to P_2 (P_1), respectively.

To show this, we enclose S in a curve ϕ, and then compute for every given configuration X the maximal subgraph G' such that v has configuration X in ϕ. This yields a set of at most 12 possible locations for switches between incoming and outgoing edges in S, which gives a partition of S into at most 13 sets (corresponding to $P_1, \ldots, P_j$) that do not contain a switch, and thus at most 26 sets that will not be separated by an optimal solution, corresponding to $S_1, \ldots, S_{26}$. We now describe a parameter-preserving reduction from MWBS to Cut-MWBS.

Lemma 5 (*). *Given an instance (G, w) of MWBS with b bad vertices, we can find in polynomial time an instance $(G', w, \mathcal{E})$ of Cut-MWBS, so that: (i) For every $\mathcal{E}_i \subseteq \mathcal{E}$ with $|\mathcal{E}_i| \geq 2$, there exists a bad vertex $v \in G'$ and a good edge-section S of v, so that $\mathcal{E}_i$ is a subset of S and $\mathcal{E}_i$ contains only outgoing or only incoming edges of v. (ii) $|\mathcal{B}(G')| \leq b$, (iii) $|\mathcal{E}| = \mathcal{O}(b^2)$, (iv) (G, w) and $(G', w, \mathcal{E})$ have the same optimal cost, (v) there exists a partition $P_1, \ldots, P_j$ of $\mathcal{E}$, such that $|P_i| \leq 2$ for all P_i, (vi) if $|P_i| = 1$, then the edges-set contained in P_i is either an edge between two bad vertices, or there exists a bad vertex $v \in G'$ and good edge-section S of v, such that the edges contained in P_i are all consecutive in S, and, (vii) if $|P_i| = 2$ with $P_i = \{\mathcal{E}_1, \mathcal{E}_2\}$, there exists some $v \in \mathcal{B}(G')$ and a good edge-section S of v, such that the edges in P_i are all consecutive in S; and $\mathcal{E}_1$ consists of outgoing edges of v if and only if $\mathcal{E}_1$ consists of incoming edges of v, and at least one of $\mathcal{E}_1, \mathcal{E}_2$ does not form a set of consecutive edges in S.*

See Fig. 5(a) for a visualization. We obtain this transformation by applying Lemma 3 in order to get a simplified equivalent instance G'. Let E_{rest} be all edges incident to two bad vertices. For every bad vertex v and every good edges section S of v, let $\mathcal{S}_{v,S}$ be the partition of S obtained from Lemma 4. We define $\mathcal{E} = \{e \mid e \in E_{\text{rest}}\} \cup \bigcup_{v,S} \mathcal{S}_{v,S}$. This defines the instance $(G', w, \mathcal{E})$ of Cut-MWBS. We will now further reduce the size of $(G, w, \mathcal{E})$.

Reduction Rule 4. Let $(G, w, \mathcal{E})$ be an instance of Cut-MWBS with properties (i) to (vii) of Lemma 5. Let $v \in \mathcal{B}(G)$, let S be a good edge-section of v, and let $\mathcal{E}_i \in \mathcal{E}$ such that $\mathcal{E}_i \subseteq S$ is a *consecutive* set of edges in S. Then let $(G', w', \mathcal{E}')$ be the new instance that is obtained from $(G, w, \mathcal{E})$ by deleting all edges (and their incident good vertices) but one edge e out of $\mathcal{E}_i$, and assigning $w'(e) = w(\mathcal{E}_i)$.

Reduction Rule 5. Let $(G, w, \mathcal{E})$ be an instance of Cut-MWBS with the properties (i) to (vii) of Lemma 5. Let further $v \in \mathcal{B}(G)$, let S be a good edge-section of v, and let $\mathcal{E}_{\text{in}}, \mathcal{E}_{\text{out}} \in \mathcal{E}$ such that $\mathcal{E}_{\text{in}}, \mathcal{E}_{\text{out}} \subseteq S$, $\mathcal{E}_{\text{in}}$ are all incoming to v, $\mathcal{E}_{\text{out}}$ are all outgoing of v, $\mathcal{E}_{\text{in}} \cup \mathcal{E}_{\text{out}}$ is a consecutive set of edges in S, and at least one of $\mathcal{E}_{\text{in}}$ or $\mathcal{E}_{\text{out}}$ does not form a consecutive set of edges in S. We construct a new edge-set e_1, e_2, e_3, e_4 as follows: e_1, e_3 are incoming for v, e_2, e_4 are outgoing of v, and all of e_1, e_2, e_3, e_4 are incident to a newly inserted (good) vertex v_{e_k} for $k \in \{1, \ldots, 4\}$. We set $w'(e_1) = w'(e_4) = 0$, $w'(e_2) = w(\mathcal{E}_{\text{out}})$ and $w'(e_3) = w(\mathcal{E}_{\text{in}})$. Further we assign $e_1, e_3 \in \mathcal{E}_{\text{in}}$ and $e_2, e_4 \in \mathcal{E}_{\text{out}}$. Let $(G', w', \mathcal{E})$ be the new instance that is obtained from $(G, w, \mathcal{E})$ by replacing the edges in $\mathcal{E}_i \cup \mathcal{E}_j$ with the consecutive sequence e_1, e_2, e_3, e_4.

Claim 2 **(*).** Reductions 4 and 5 are sound.

Lemma 6 (*). *Let $(G, w, \mathcal{E})$ be an instance of Cut-MWBS with b bad vertices and properties (i) till (vii) of Lemma 5. Then we can compute in polynomial time an equivalent instance $(G', w', \mathcal{E}')$ such that $V(G') = \mathcal{O}(b^2)$.*

See Fig. 5(b) for an illustration. We compute $(G', w', \mathcal{E}')$ by applying Reductions 4 and 5 exhaustively. To bound the size of the weights w, we use the approach of Etscheid et al. [17] and the well-known Theorem 4. This yields the compression of MWBS (Theorem 5) and a kernel for MWBS (Theorem 6).

Theorem 4 ([20]). *There is an algorithm that, given a vector $\omega \in \mathbb{Q}^r$ and an integer N, in polynomial time finds a vector $\bar{\omega}$ such that $||\bar{\omega}||_\infty = 2^{\mathcal{O}(r^3)}$ and $sign(\omega \cdot b) = sign(\bar{\omega} \cdot b)$ for all vectors $b \in \mathbb{Z}^r$ with $||b||_1 \leq N - 1$.*

Theorem 5 (*). *There exists a polynomial-time algorithm that, given an instance (G, w) of MWBS with b bad vertices and a target value W, computes an instance $(G', w', \mathcal{E})$ of Cut-MWBS with size $\mathcal{O}(b^8)$, and a new target value W' with size $\mathcal{O}(b^6)$, such that there exists a solution for (G, w) of cost W if and only if there exists a solution for $(G', w', \mathcal{E})$ of cost W'.*

Theorem 6 (*). *The decision version of MWBS parameterized by the number of bad vertices b admits a polynomial kernel.*

5 Efficient PTAS for MWBS and Final Remarks

We sketch our Efficient Polynomial-Time Approximation Scheme (EPTAS) for MWBS, i.e., a $(1-\epsilon)$-approximation that runs in $2^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1)}$ time. We use Baker's technique [4] to design our EPTAS. Our goal is to reduce the problem to (multiple instances of) the problem, where the treewidth (hence, branchwidth) of the graph is bounded by $\mathcal{O}(1/\epsilon)$, at the expense of an ϵ-factor loss in cost. Then, we can use our single-exponential algorithm in the branchwidth to solve each such instance exactly, which implies a $(1-\epsilon)$-approximation.

We sketch the details of this reduction. W.l.o.g. assume that the graph is connected. We perform a breadth-first search starting from an arbitrary vertex $v \in V(G)$, and partition the vertex-set into layers $L_0, L_1, \ldots$, where L_i is the set of vertices at distance *exactly* i from v in the *undirected* version of G. It is known that the treewidth of the subgraph induced by any d consecutive layers is upper bounded by $\mathcal{O}(d)$ – this follows from a result of Bodlaender [9], which states that the treewidth of a planar graph with diameter D is $\mathcal{O}(D)$. Let $t = 1/\epsilon$, and for each $0 \le i \le t$, let $E^{(i,i+1)}$ denote edges uv such that $u \in L_j$, $v \in L_{j+1}$ with j mod $t = i$. By an averaging argument, there exists an index $0 \le i \le t$, such that the total contribution of all the edges from an optimal solution (i.e., the set of edges inducing a maximum-weight bimodal subgraph) that belong to $E^{(i,i+1)}$, is at most $1/t = \epsilon$ times the weight of the optimal solution. Since we do not know this index i, we consider all values of i, and consider the subproblems obtained by deleting the edges. Then, the graph breaks down into multiple connected components, and the treewidth of each component is $\mathcal{O}(1/\epsilon)$. We solve each such subproblem optimally in time $2^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1)}$ using Theorem 1, and combine the solutions for the subproblems to obtain a solution for the original instance. Note that the graph obtained by combining the optimal solutions for the subproblems is bimodal, and for the correct value of i, the weight of the graph is at least $1-\epsilon$ times the optimal cost. That is, the combined solution is a $(1-\epsilon)$-approximation.

Theorem 7 (*). *There exists an algorithm that runs in time $2^{\mathcal{O}(1/\epsilon)} \cdot n^{\mathcal{O}(1)}$ and returns a $(1-\epsilon)$-approximate solution for the given instance of MWBS. That is, MWBS admits an EPTAS.*

We note that Baker's technique can also be used to obtain an EPTAS with the similar running for the *minimization* variant of MWBS. Although the high level idea is similar, the details are more cumbersome.

Final Remarks. We conclude by suggesting some open questions. One natural problem is to ask for a maximum k-modal subgraph for any given even integer $k \ge 2$; we believe that our ideas can be extended to this more general setting. Another natural variant of MBS is to limit the number of edges that we can delete to get a bimodal subgraph by an integer h; in this setting, h becomes another parameter in addition to those we have considered. Finally, studying MBS in the variable embedding setting is an interesting future direction.

References

1. Angelini, P., Chaplick, S., Cornelsen, S., Da Lozzo, G.: Planar L-drawings of bimodal graphs. J. Graph Algorithms Appl. **26**(3), 307–334 (2022). https://doi.org/10.7155/jgaa.00596
2. Angelini, P., Chaplick, S., Cornelsen, S., Lozzo, G.D.: On upward-planar L-drawings of graphs. In: Szeider, S., Ganian, R., Silva, A. (eds.) 47th International Symposium on Mathematical Foundations of Computer Science. MFCS 2022, 22–26 August 2022, Vienna, Austria. LIPIcs, vol. 241, pp. 10:1–10:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.MFCS.2022.10
3. Angelini, P., et al.: Algorithms and bounds for L-drawings of directed graphs. Int. J. Found. Comput. Sci. **29**(4), 461–480 (2018). https://doi.org/10.1142/S0129054118410010
4. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. J. Assoc. Comput. Mach. **41**(1), 153–180 (1994)
5. Bertolazzi, P., Di Battista, G., Didimo, W.: Quasi-upward planarity. Algorithmica **32**(3), 474–506 (2002). https://doi.org/10.1007/s00453-001-0083-x
6. Binucci, C., Di Giacomo, E., Liotta, G., Tappini, A.: Quasi-upward planar drawings with minimum curve complexity. In: Purchase, H.C., Rutter, I. (eds.) GD 2021. LNCS, vol. 12868, pp. 195–209. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92931-2_14
7. Binucci, C., Didimo, W.: Computing quasi-upward planar drawings of mixed graphs. Comput. J. **59**(1), 133–150 (2016). https://doi.org/10.1093/comjnl/bxv082
8. Binucci, C., Didimo, W., Giordano, F.: Maximum upward planar subgraphs of embedded planar digraphs. Comput. Geom. **41**(3), 230–246 (2008). https://doi.org/10.1016/j.comgeo.2008.02.001
9. Bodlaender, H.L.: A partial k-arboretum of graphs with bounded treewidth. Theor. Comput. Sci. **209**(1–2), 1–45 (1998). https://doi.org/10.1016/S0304-3975(97)00228-4
10. Chaplick, S., et al.: Planar L-drawings of directed graphs. In: Frati, F., Ma, K.-L. (eds.) GD 2017. LNCS, vol. 10692, pp. 465–478. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73915-1_36
11. Cygan, M., et al.: Parameterized Algorithms (2015). https://doi.org/10.1007/978-3-319-21275-3
12. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice-Hall, Upper Saddle River (1999)
13. Di Battista, G., Nardelli, E.: Hierarchies and planarity theory. IEEE Trans. Syst. Man Cybern. **18**(6), 1035–1046 (1988). https://doi.org/10.1109/21.23105
14. Didimo, W.: Upward graph drawing. In: Encyclopedia of Algorithms, pp. 2308–2312 (2016). https://doi.org/10.1007/978-1-4939-2864-4_653
15. Didimo, W., Fomin, F.V., Golovach, P.A., Inamdar, T., Kobourov, S., Sieper, M.D.: Parameterized and approximation algorithms for the maximum bimodal subgraph problem (2023). https://doi.org/10.48550/arXiv.2308.15635
16. Dorn, F., Penninkx, E., Bodlaender, H.L., Fomin, F.V.: Efficient exact algorithms on planar graphs: exploiting sphere cut decompositions. Algorithmica **58**(3), 790–810 (2010). https://doi.org/10.1007/s00453-009-9296-1
17. Etscheid, M., Kratsch, S., Mnich, M., Röglin, H.: Polynomial kernels for weighted problems. J. Comput. Syst. Sci. **84**, 1–10 (2017). https://doi.org/10.1016/j.jcss.2016.06.004

18. Fomin, F.V., Lokshtanov, D., Saurabh, S., Zehavi, M.: Kernelization. Theory of Parameterized Preprocessing. Cambridge University Press, Cambridge (2019)
19. Fomin, F.V., Thilikos, D.M.: New upper bounds on the decomposability of planar graphs. J. Graph Theory **51**(1), 53–81 (2006). https://doi.org/10.1002/jgt.20121
20. Frank, A., Tardos, É.: An application of simultaneous diophantine approximation in combinatorial optimization. Combinatorica **7**, 49–65 (1987). https://doi.org/10.1007/BF02579200
21. Impagliazzo, R., Paturi, R.: Complexity of k-sat. In: Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, 4–6 May 1999, pp. 237–240. IEEE Computer Society (1999). https://doi.org/10.1109/CCC.1999.766282
22. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. Syst. Sci. **63**(4), 512–530 (2001). https://doi.org/10.1006/jcss.2001.1774
23. Jacob, H., Pilipczuk, M.: Bounding twin-width for bounded-treewidth graphs, planar graphs, and bipartite graphs. In: Bekos, M.A., Kaufmann, M. (eds.) WG 2022. LNCS, vol. 13453, pp. 287–299. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15914-5_21
24. Jünger, M., Leipert, S., Mutzel, P.: Level planarity testing in linear time. In: Whitesides, S.H. (ed.) GD 1998. LNCS, vol. 1547, pp. 224–237. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-37623-2_17
25. Robertson, N., Seymour, P.D.: Graph minors. x. obstructions to tree-decomposition. J. Comb. Theory, Ser. B **52**(2), 153–190 (1991). https://doi.org/10.1016/0095-8956(91)90061-N
26. Vial, J.J.B., Da Lozzo, G., Goodrich, M.T.: Computing k-modal embeddings of planar digraphs. In: Bender, M.A., Svensson, O., Herman, G. (eds.) 27th Annual European Symposium on Algorithms, ESA 2019, 9–11 September 2019, Munich/Garching, Germany. LIPIcs, vol. 144, pp. 19:1–19:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.ESA.2019.19