A Sub-Exponential FPT Algorithm and a Polynomial Kernel for Minimum Directed Bisection on Semicomplete Digraphs

Jayakrishnan Madathil
The Institute of Mathematical Sciences, HBNI, Chennai, India
jayakrishnanm@imsc.res.in

Roohani Sharma
The Institute of Mathematical Sciences, HBNI, Chennai, India
roohani@imsc.res.in

Meirav Zehavi
Ben-Gurion University, Beer-Sheva, Israel
meiravze@bgu.ac.il

Abstract

Given an n-vertex digraph D and a non-negative integer k, the Minimum Directed Bisection problem asks if the vertices of D can be partitioned into two parts, say L and R, such that $|L|$ and $|R|$ differ by at most 1 and the number of arcs from R to L is at most k. This problem, in general, is W-hard as it is known to be NP-hard even when $k = 0$. We investigate the parameterized complexity of this problem on semicomplete digraphs. We show that Minimum Directed Bisection on semicomplete digraphs is one of a handful of problems that admit sub-exponential time fixed-parameter tractable algorithms. That is, we show that the problem admits a $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$ time algorithm on semicomplete digraphs. We also show that Minimum Directed Bisection admits a polynomial kernel on semicomplete digraphs. To design the kernel, we use (n, k, k^2)-splitters. To the best of our knowledge, this is the first time such pseudorandom objects have been used in the design of kernels. We believe that the framework of designing kernels using splitters could be applied to more problems that admit sub-exponential time algorithms via chromatic coding. To complement the above mentioned results, we prove that Minimum Directed Bisection is NP-hard on semicomplete digraphs, but polynomial time solvable on tournaments.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases bisection, semicomplete digraph, tournament, fpt algorithm, chromatic coding, polynomial kernel, splitters

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.28

Funding *Meirav Zehavi*: Supported by ISF grant no. 1176/18.

Acknowledgements We thank Daniel Lokshtanov and Saket Saurabh for insightful discussions on bisection in semicomplete digraphs.

1 Introduction

A bisection of a graph is a partition of its vertex set into two (almost) equal parts. In the Minimum Bisection problem, given an undirected graph G and a non-negative integer k, the task to check whether the vertex set of G can be partitioned into two parts, say A and B, such that $||A| - |B|| \leq 1$ and the number of edges with one endpoint in A and the other endpoint in B is at most k. This problem has been studied extensively, resulting in a large volume of literature [7, 11, 12, 19, 20, 27, 28]. In particular, it has been shown that Minimum Bisection is NP-hard [19], but admits a fixed-parameter tractable algorithm (with k as the parameter) [11]. We study the directed counterpart of this problem in the

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 28; pp. 28:1–28:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

framework of parameterized complexity. In the Minimum Directed Bisection problem, the input consists of a digraph D and an integer k, and the question is to determine whether the vertices of D can be partitioned into two parts, say L and R, such that $||L| - |R|| \leq 1$, and there are at most k arcs with their tails in R and heads in L (i.e. arcs directed from R to L). The problem is formally defined below.

Minimum Directed Bisection **Parameter:** k
Input: A digraph D and a non-negative integer k.
Question: Can the vertex set of D be partitioned into two parts, say L and R, such that $||L| - |R|| \leq 1$ and there are at most k arcs (u, v) with $u \in R$ and $v \in L$?

Feige and Yahalom [16] showed that this problem is NP-hard even for $k = 0$, which implies that the Minimum Directed Bisection problem has no polynomial time approximation unless P = NP. Their result also implies that Minimum Directed Bisection is para-NP-hard on general digraphs and hence, it is W-hard when parameterized by k. We study the complexity of this problem on restricted classes of digraphs such as tournaments and semicomplete digraphs.

Our results. Our contribution is threefold.

1. We show that Minimum Directed Bisection is polynomial time solvable on tournaments.
2. On semicomplete digraphs, Minimum Directed Bisection is NP-hard, but admits an algorithm running in time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$.
3. We also show that Minimum Directed Bisection on semicomplete digraphs admits a polynomial kernel.

A *tournament* is a digraph D such that for every distinct pair of vertices u and v of D, *exactly one* of the arcs (u, v) or (v, u) is present in D. A *semicomplete digraph* is a digraph D such that for every pair of distinct vertices u and v of D, *at least one* of the arcs (u, v) or (v, u) is present in D. Note that both the arcs (u, v) and (v, u) may be present in a semicomplete digraph. Tournaments and semicomplete digraphs form two interesting and well-studied classes of digraphs. See, for example, the monograph by Bang-Jensen and Gutin [3] for an overview. In recent years, tournaments and semicomplete digraphs received a great deal of attention, in part, due to the many structural results for these classes that are similar to the theory of minors for undirected graphs. In particular, it has been shown that tournaments are well-quasi ordered under strong minors [23] and strong immersions [9]. Similarly, semicomplete digraphs have been shown to be well-quasi ordered under strong minors, butterfly immersions [22] and strong immersions [5].

Due to the extremely rigid structure of tournaments, a number of problems that are NP-hard on general digraphs become polynomial time solvable on tournaments. Examples include classic problems such as Hamiltonian path [29] and Hamiltonian Cycle [8]. Although semicomplete digraphs are a slight generalization of tournaments, the flexibility in the definition allowing for the presence of anti-parallel arcs makes them distinctly dissimilar to tournaments when it comes to tractability of algorithmic problems. This contrast in behavior is perhaps best illustrated by problems such as Cutwidth and Optimal Linear Arrangement (OLA), which are polynomial time solvable on tournaments, but NP-hard on semicomplete digraphs [4]. Our work reinforces this pattern. We show that while Minimum Directed Bisection is polynomial time solvable on tournaments (Lemma 4), it is NP-hard on semicomplete digraphs (Lemma 16). The polynomial time solvability of Minimum Directed Bisection on tournaments follows from a polynomial time algorithm due to Fradkin [18] for computing the cutwidth of a tournament. Barbero et al. [4] showed that Fradkin's approach in [18] yields a polynomial time algorithm for the Optimal Linear Arrangement problem on tournaments as well. We rely on the analysis of Barbero et al. in [4] to conclude that Fradkin's approach works for Minimum Directed Bisection on tournaments as well. We establish the NP-hardness of Minimum Directed Bisection on semicomplete digraphs by a reduction from the Maximum Bisection problem on directed acyclic graphs.

Tournaments and semicomplete digraphs found favor with the parameterized complexity community mainly because of the possibility that several problems might admit sub-exponential time parameterized algorithms on these classes of digraphs. In the field of parameterized complexity, problems that admit sub-exponential time algorithms (except those on planar graphs, using the bidimensionality approach) are something of a rarity. In fact, the first such sub-exponential time algorithm was designed by Alon, Lokshtanov and Saurabh [1] for the Feedback Arc Set problem on tournaments (FAST). They developed the technique of chromatic coding to design a $2^{\mathcal{O}(\sqrt{k}\log^2 k)}$ algorithm for this problem. This running time has since been improved, independently by Feige [15], Karpinski and Schudy [21], and Fomin and Pilipczuk [17], to $2^{\mathcal{O}(\sqrt{k})}$. Fomin and Pilipczuk's [17] algorithm for FAST was in fact a by-product of their work on the Cutwidth and Optimal Linear Arrangement (OLA) problems on semicomplete digraphs. They showed that both Cutwidth and OLA admit sub-exponential FPT algorithms on semicomplete digraphs. Later, Barbero et al. [4] showed that these problems are indeed NP-hard on semicomplete digraphs. We show that Minimum Directed Bisection on semicomplete digraphs admits a sub-exponential time algorithm that runs in time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$ (Theorem 21).

Both our algorithm and kernel for Minimum Directed Bisection build on a crucial observation that in any bisection (L, R) of a semicomplete digraph, every vertex v has a "preferred position" (either L or R), which we call the "canonical position" of v (denoted by $\mathtt{can}(v)$). In particular, if (L, R) is a bisection of size at most k, then we show that in (L, R), no more than $2k$ vertices can deviate from their canonical positions. Note that with just this information one can easily design an $n^{\mathcal{O}(k)}$ algorithm for the problem. In order to achieve sub-exponential FPT running time, we use the technique of chromatic coding to identify a small set of vertices that contain these $2k$ vertices that can deviate from their canonical positions. More precisely, we color the vertex set with $\mathcal{O}(\sqrt{k})$ colors such that all the k arcs in a k-sized bisection (if it exists) are properly colored. Then, to identify the desired set of vertices, we exploit the fact that no arc within a color class can be directed from R to L. The crucial point in chromatic coding is that it is sufficient to try out $2^{\mathcal{O}(\sqrt{k}\log k)}\log n$ different coloring functions so as to ensure that we find a coloring with the required property.

The kernelization complexity of problems on tournaments and semicomplete digraphs has proved to be even more diverse, requiring a wide array of techniques specific to individual problems. For instance, it is now textbook knowledge that the Feedback Arc Set problem on tournaments admits a simple kernel with $\mathcal{O}(k^2)$ vertices [10, 14]. There has been a line of work on this front that improved the size of the kernel, the current best being a linear kernel by Bessy et al. [6]. Barbero et al. [4] showed that while Cutwidth on semicomplete digraphs does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP}/\text{poly}$, OLA does admit a kernel with $2k$ vertices. In addition, by using the submodularity property of directed cuts, Barbero et al. [4] showed that Cutwidth admits a Turing kernel on semicomplete digraphs.

We employ a family of hash functions called (n, k, k^2)-splitters, introduced by Naor et al. [26], to design a polynomial kernel for Minimum Directed Bisection on semicomplete digraphs (Theorem 22). Splitters are a well-known combinatorial tool often used to derandomize algorithms. To the best of our knowledge, this is the first occurrence of splitters' being used to design polynomial kernels. Our kernelization algorithm for Minimum Directed Bisection on semicomplete digraphs has two main steps. In the first step, we reduce the given instance (D, k) of the Minimum Directed Bisection problem to $k^{\mathcal{O}(1)}$ many instances of a weighted variant of the problem, with the guarantee that the sizes of the reduced instances is $k^{\mathcal{O}(1)}$ and at least one of the reduced instances is equivalent to the original instance (D, k). Then, in the second step, we apply a polynomial time reduction from the reduced instances to Minimum Directed Bisection to obtain the desired kernel. To elaborate a little about the first step, like in the design of our algorithm, we want a coloring function that properly colors the k arcs of the bisection. But unlike in the algorithm where we have to use $\mathcal{O}(\sqrt{k})$ colors to ensure sub-exponential runtime, for the kernel, it is enough to have such a coloring using $\mathcal{O}(k^2)$ colors. This increase in the number of colors from $\mathcal{O}(\sqrt{k})$ to $\mathcal{O}(k^2)$ allows us to bring down the size of the coloring family that guarantees the existence of the good coloring that we want, to a polynomial in k. This is where we use (n, k, k^2)-splitters: we construct a reduced instance for each of the $k^{\mathcal{O}(1)}$ coloring functions. The one corresponding to the good coloring function is the one that is always a yes-instance if the input instance is a yes-instance.

Related work on Minimum Bisection. As mentioned earlier, the Minimum Bisection problem, (the undirected counterpart of our problem) is an extensively-studied problem in algorithmic graph theory. Being a natural variant of the well-known Min Cut problem, Minimum Bisection has already been known to be NP-hard since the 1970s [19]. However, the parameterized complexity of the problem had been open for a long time, until it was finally settled by Cygan et al. [11] in 2014, who showed that the problem is fixed-parameter tractable. Before that, Jansen et al. [20] had shown that the problem is fixed-parameter tractable when parameterized by the treewidth of the input graph. The best known approximation algorithm for Minimum Bisection, to the best of our knowledge, is due to Räcke [28], with an approximation factor of $\mathcal{O}(\log n)$. While Minimum Bisection on planar graphs admits a single-exponential parameterized algorithm [7], it remains open whether the problem is NP-hard. On the related (but incomparable) class of unit disk graphs, Minimum Bisection is known to be NP-hard [12], but was recently shown to admit a single-exponential parameterized algorithm [27]. The Minimum Vertex Bisection problem (the vertex-deletion variant of Minimum Bisection) asks the following question. Given an undirected graph G and a non-negative integer k, is it possible to partition the vertex set of G into three parts A, S and B such that $|S| \leq k$, $|A| = |B|$ and there are no edges between A and B? This problem is W[1]-hard: as observed in [11], the W[1]-hardness of Minimum Vertex Bisection follows from a more general result due to Marx [25].

2 Preliminaries

For a natural number $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, \ldots, n\}$. For $z \in \mathbb{N}$, $|z|$ denotes the number of bits in the binary representation of z.

Splitters. Consider two finite sets A and B and a function $f : A \to B$. For a subset $S \subseteq A$, we say that f *splits S evenly* if for every $b_1, b_2 \in B$, $|f^{-1}(b_1) \cap S|$ and $|f^{-1}(b_2) \cap S|$ differ by at most 1. Notice that if $|B| > |S|$, then f splits S evenly if and only if f is injective on S.

▶ **Definition 1** ([26]). *For positive integers n, k and ℓ, an (n, k, ℓ)-*splitter *is a family $\mathcal{F}$ of functions from $[n]$ to $[\ell]$ such that for every subset $S \subseteq [n]$ of size at most k, there exists a function $f \in \mathcal{F}$ that splits S evenly.*

▶ **Proposition 2** ([2, 26]). *There is an algorithm that, given positive integers n and k, runs in time $k^{\mathcal{O}(1)} n \log n$ and constructs an (n, k, k^2)-splitter of size $k^{\mathcal{O}(1)} \log n$.*

Digraphs. For a digraph D, we denote by $V(D)$ and $A(D)$ the vertex set and arc set of D, respectively. For sets $X, Y \subseteq V(D)$, $A(X, Y) = \{(x, y) \in A(D) \mid x \in X \text{ and } y \in Y\}$. (Whenever dealing with multiple digraphs on the same vertex set, we may write $A_D(X, Y)$ instead of $A(X, Y)$ to emphasize that we mean the set of arcs from X to Y *in* D.) For a vertex $v \in V(D)$, $A(v)$ denotes the set of arcs incident with v. For an arc $a \in A(D)$, $V(a)$ denotes the set of endpoints of a. That is, if $a = (u, v) \in A(D)$, then $V(a) = \{u, v\}$. More generally, for a set of arcs $A \subseteq A(D)$, $V(A) = \cup_{a \in A} V(a)$. The *complement* of D, denoted by $\overline{D}$, is the digraph defined as follows: $V(\overline{D}) = V(D)$ and $A(\overline{D}) = (V^2 \setminus \{(u, u) \mid u \in V(D)\}) \setminus A(D)$. (We assume that D has no self-loops.) Note that for vertices $u, v \in V(D)$, if $(u, v) \notin A(D)$, then $(u, v), (v, u) \in A(\overline{D})$. For a digraph D, $\texttt{rev}(D)$ denotes the digraph obtained by reversing all arcs of D, i.e. $V(\texttt{rev}(D)) = V(D)$ and $A(\texttt{rev}(D)) = \{(u, v) \mid (v, u) \in A(D)\}$.

Consider a digraph D. Throughout this paper, we use (L_D, R_D) to denote a bipartition of D. That is $V(D) = L_D \cup R_D$ and $L_D \cap R_D = \emptyset$. We may drop the subscript D and simply write (L, R) when D is clear from the context. A bipartition (L_D, R_D) is a *bisection* of D if $||L_D| - |R_D|| \leq 1$; we refer to the value $|A(R_D, L_D)|$ as the *size of the bipartition* (L_D, R_D). Moreover, we say that (L_D, R_D) is a *one way partition* if $A(R_D, L_D) = \emptyset$, i.e. if (L_D, R_D) is a bipartition of size 0.

3 Some Observations and Simple Lemmas

Throughout this paper, whenever dealing with an instance (D, k) of Minimum Directed Bisection, we assume that $|V(D)|$ is even. This assumption about $V(D)$ is made without loss of generality since if $|V(D)|$ is odd, then we may construct two instances (D_1, k) and (D_2, k) of Minimum Directed Bisection such that at least one of them will be equivalent to (D, k) and $|V(D_1)| = |V(D_2)| = |V(D)| + 1$. The following lemma shows that it is safe to make this assumption.

▶ **Lemma 3** ($\star$[1]). *Given an instance (D, k) of* Minimum Directed Bisection *where $|V(D)|$ is odd, it is possible to construct two instances (D_1, k) and (D_2, k) of* Minimum Directed Bisection *in polynomial time such that* (i) *$|V(D_1)|$ and $|V(D_2)|$ are even,* (ii) *D_1 and D_2 are semicomplete digraphs (tournaments) if D is a semicomplete digraph (tournament), and* (iii) *(D, k) is a yes-instance if and only if either (D_1, k) or (D_2, k) is a yes-instance.*

The following lemma deals with Minimum Directed Bisection on tournaments.

▶ **Lemma 4** ($\star$). Minimum Directed Bisection *is polynomial time solvable on tournaments.*

Proof Sketch. The algorithm for Minimum Directed Bisection on tournaments works as follows. Given a tournament T on n vertices, sort the vertices in non-decreasing order by their in-degrees. Take L_T to be the set of the first $n/2$ vertices in the sorted list, and $R_T = V(T) \setminus L_T$. Then, our claim is that (L_T, R_T) is a minimum bisection of T, i.e. $|A(R_T, L_T)| \leq |A(Y, X)|$ for every bisection (X, Y) of T. The correctness of this algorithm follows from Lemma 1 in [4]. ◀

[1] Due to paucity of space, proofs of statements marked with a $\star$ have been omitted.

We now state two observations about semicomplete digraphs. These observations are considered to be folklore, so we omit their proofs. Specifically, the first observation is immediate, and the second can be found in [13].

▶ **Observation 5.** *Let D be a semicomplete digraph on n vertices, where n is even. For every $v \in V(D)$, we have $d^+(v) \geq n/2$ or $d^-(v) \geq n/2$.*

▶ **Observation 6.** *Every semicomplete digraph contains a Hamiltonian path. Moreover, there is a polynomial time algorithm that, given a semicomplete digraph D as input, finds a Hamiltonian path in D.*

We now develop some preliminary results that will be used in the design and analysis of our algorithm and kernel.

Canonical Position. A crucial idea behind our algorithm and kernel is that every vertex has a "preferred position" in a bisection (either L or R). In particular, if a vertex v deviates from its preferred position, then at least one of the arcs incident with v would have to belong to $A(R, L)$. Therefore, in a bisection of size at most k, no more than $\mathcal{O}(k)$ vertices can deviate from their preferred positions. Next, we formalize this idea.

▶ **Definition 7.** *Let D be a semicomplete digraph on n vertices, where n is even. The partition $(L^{\mathrm{c}}_D, R^{\mathrm{c}}_D)$ (not necessarily a bisection) of $V(D)$ defined as $L^{\mathrm{c}}_D = \{v \in V(D) \mid d^+(v) \geq n/2\}$ and $R^{\mathrm{c}}_D = V(D) \setminus L^{\mathrm{c}}_D$ is called the* canonical partition *of D. For a vertex $v \in L^{\mathrm{c}}_D$, we say that L^{c}_D is the* canonical position of v *and write $\mathtt{can}(v) = L^{\mathrm{c}}_D$. Similarly, for a vertex $v \in R^{\mathrm{c}}_D$, we say that R^{c}_D is the* canonical position of v *and write $\mathtt{can}(v) = R^{\mathrm{c}}_D$.*

▶ **Definition 8.** *Let D be a semicomplete digraph with a bipartition (L, R). For a vertex $v \in V(D)$, we say that (L, R)* respects the canonical position of v *if $v \in L$ and $\mathtt{can}(v) = L^{\mathrm{c}}_D$ or if $v \in R$ and $\mathtt{can}(c) = R^{\mathrm{c}}_D$. Otherwise, we say that (L, R)* violates the canonical position of *v.*

The following lemma follows directly from the definition of the canonical position of a vertex. It shows that if a bisection (L, R) violates the canonical position of a vertex v, then at least one of the arcs incident with v must belong to $A(R, L)$.

▶ **Lemma 9** ($\star$). *Consider a semicomplete digraph D on n vertices, where n is even. Let (L, R) be a bisection of D. Let $v \in V(D)$ be such that (L, R) violates the canonical position of v. Then, $A(R, L) \cap A(v) \neq \emptyset$.*

An immediate consequence of Lemma 9 is that any bisection of size at most k of a semicomplete digraph can violate the canonical positions of at most $2k$ vertices.

▶ **Corollary 10.** *Let (D, k) be an yes-instance of* Minimum Directed Bisection, *and let (L, R) be a bisection of D of size at most k. Let $X = \{v \in V(D) \mid (L, R)$ violates the canonical position of $v\}$. Then, $|X| \leq 2k$.*

Proof. As Lemma 9 shows, for every $v \in X$, $A(R, L)$ contains an arc incident with v. Since each arc in $A(R, L)$ can be incident with at most two vertices in X, and $|A(R, L)| \leq k$, it follows that $|X| \leq 2k$. ◀

Consider a bisection (L, R) of D, and a subset of vertices $Z \subseteq V(D)$ such that no arc of $D[Z]$ belongs to $A(R, L)$. The following lemma identifies the vertices in Z whose canonical positions can possibly be violated by (L, R). Moreover, it shows that there can be at most $4k$ such vertices in Z, and that (L, R) must respect the canonical position of every other vertex.

▶ **Lemma 11.** *Let D be a semicomplete digraph with a bisection (L,R) of size at most k. Let $Z \subseteq V(D)$ be such that no arc of $D[Z]$ belongs to $A_D(R,L)$. That is, $(L \cap Z, R \cap Z)$ is a one way partition of Z. Let P be a Hamiltonian path in $D[Z]$ from a vertex α to a vertex β for some $\alpha, \beta \in Z$. Let $X = \{x \in Z \mid \mathtt{can}(x) = R^c_D\}$ and $Y = \{y \in Z \mid \mathtt{can}(y) = L^c_D\}$. For $1 \leq j \leq |X|$, let x_j be the j^{th} vertex of P that belongs to X as we traverse P from α to β, and for $1 \leq p \leq |Y|$, let y_p be the p^{th} vertex of P that belongs to Y as we traverse $\mathtt{rev}(P^i)$ from β to α. Then, if $|X| > 2k$, then for every $x_j \in X$ with $j > 2k$, $x_j \in R$. Similarly, if $|Y| > 2k$, then for every $y_p \in Y$ with $p > 2k$, $y_p \in L$. (In other words, (L,R) respects the canonical positions of x_j and y_p for $j > 2k$ and $p > 2k$.)*

Proof. Note first that for any arc $(z,z') \in A(P)$, it cannot be the case that $z \in R$ and $z' \in L$, as $(L \cap Z, R \cap Z)$ is a one way partition of Z. More generally, this property holds for any two distinct vertices $z, z' \in Z$ such that z appears before z' as we traverse P from α to β. To see this, suppose by way of contradiction that $z \in R$ and $z' \in L$. Then, consider the subpath of P from z to z'. This subpath must have an arc from a vertex in R to a vertex in L, which we have already argued to be impossible.

Now, suppose that $|X| > 2k$ and consider a vertex $x_j \in X$ with $j > 2k$. Then, $\mathtt{can}(x_j) = R^c_D$. We need to show that $x_j \in R$. Assume for contradiction that $x_j \in L$, i.e. (L,R) violates the canonical position of x_j. Then, for every $j' \leq j$, we must have $x_{j'} \in L$ as well, as otherwise, we obtain a pair $\{x_{j'}, x_j\}$ such that $x_{j'}$ appears before x_j as we traverse P from α to β while $x_{j'} \in R$ and $x_j \in L$, which by the preceding argument, is not possible. Having $x_{j'} \in L$ for every $j' \leq j$ means that (L,R) violates the canonical position of $x_{j'}$ for every $j' \leq j$. Thus, (L,R) violates the canonical positions of more than $2k$ vertices. However, as (L,R) is a bisection of D of size at most k, by Corollary 10, (L,R) can violate the canonical positions of at most $2k$ vertices.

Using symmetric arguments, we also derive that $y_p \in L$ for every $p > 2k$. ◀

▶ Remark 12. In the above proof, it is actually shown that for any $j \leq |X|$ such that (L,R) violates the canonical position of $x_j \in X$, (L,R) must violate the canonical position of $x_{j'}$ for every $j' < j$ as well. Similarly, for any $p \leq |Y|$ such that (L,R) violates the canonical position of $y_p \in Y$, (L,R) must violate the canonical position of $y_{p'}$ for every $p' < p$ as well.

Tools for Our Kernel. We now develop some tools that will be used to design our kernel. Consider a vertex-weighted digraph D, where the weights are given by a function $w : V(D) \to \mathbb{N}$. For a subset $X \subseteq V(D)$ of vertices, the weight of X is defined as $w(X) = \sum_{x \in X} w(x)$. We say that a partition (L,R) of $V(D)$ is a *w-bisection* of D (or simply a *bisection* when w is clear from the context) if $|w(L) - w(R)| \leq 1$, and $|A(R,L)|$ is called the size of the bisection. So, given (D,w,k), the Weighted Minimum Bisection (WMB) problem asks whether D has a w-bisection of size at most k. We now define a "composition" of this problem, where the input consists of multiple pairs (D,w), and at least one is required to have a bisection of size at most k.

At Least One Weighted Minimum Bisection (ALO-WMB)
Input: A collection $D_1, D_2, \ldots, D_r$ of vertex-weighted semicomplete digraphs, a weight function $w_i : V(D_i) \to \mathbb{N}$ for every $i \in [r]$, and a non-negative integer k.
Question: Does there exist $i \in [r]$ such that D_i has a w_i-bisection of size at most k?

It is not difficult to see that this problem belongs to the class NP: a polynomial-sized certificate for the problem would be a partition (L,R) of $V(D_i)$ for some i, and it can be verified whether (L,R) is indeed a bisection of size at most k in time polynomial in $\sum_{j \in [r]}(|V(D_j)| + \sum_{v \in V(D_j)} |w_j(v)|)$. For future reference, we record this observation below.

▶ **Observation 13.** ALO-WMB *belongs to the class NP.*

$\mathcal{P}$-Contraction of a Digraph. Consider a digraph D. Let $\mathcal{P} = \{V_1, V_2, \ldots, V_p\}$ be a partition of $V(D)$ (i.e. $V_i \cap V_j = \emptyset$ for every distinct $i, j \in [p]$ and $\bigcup_{i\in[p]} V_i = V(D)$). The *$\mathcal{P}$-contraction* of D is the digraph $\mathcal{P}(D)$ obtained by "contracting" every part V_i into a single vertex, formally defined as follows. The digraph $\mathcal{P}(D)$ has p vertices, one corresponding to every part $V_i \in \mathcal{P}$. For each $i \in [p]$, let z_i deonte the vertex of $\mathcal{P}(D)$ corresponding to the part V_i. For distinct $i, j \in [p]$, the arc (z_i, z_j) is present in $\mathcal{P}(D)$ if and only if there exist $v_i \in V_i$ and $v_j \in V_j$ such that $(v_i, v_j) \in A(D)$. (Note that $\mathcal{P}(D)$ has no self-loops.) For a vertex $z_i \in V(\mathcal{P}(D))$, we refer to the part V_i as the *parent-part of* z_i (or simply *part of* z_i), and write $\texttt{part}(z_i) = V_i$. Moreover, we define the *$\mathcal{P}$-weighted-contraction* of D to be the digraph $\mathcal{P}(D)$, where weights are assigned to its vertices by a weight function $w_{\mathcal{P}}$, which is given by $w_{\mathcal{P}}(z_i) = |V_i|$ for every vertex z_i of $\mathcal{P}(D)$.

▶ **Lemma 14** ($\star$)**.** *Let D be an unweighted (not necessarily semicomplete) digraph and (L, R) a bipartition (not necessarily a bisection) of D. Consider a pair of partitions $\mathcal{L} = \{V_1, V_2, \ldots, V_p\}$ and $\mathcal{R} = \{V_{p+1}, V_{p+2}, \ldots, V_q\}$ of L and R, respectively. Let $\mathcal{P}$ be the partition of $V(D)$ that is the union of $\mathcal{L}$ and $\mathcal{R}$, i.e. $\mathcal{P} = \mathcal{L} \cup \mathcal{R}$. Let $H = \mathcal{P}(D)$ be the $\mathcal{P}$-weighted-contraction of D. (Thus, for each $V_i \in \mathcal{P}$, H has a vertex z_i, with weight $w_{\mathcal{P}}(z_i) = |V_i|$.) Let $L_H = \{z_1, z_2, \ldots, z_p\}$, and $R_H = \{z_{p+1}, z_{p+2}, \ldots, z_q\}$. Then, (L, R) is a bisection of D if and only if (L_H, R_H) is a $w_{\mathcal{P}}$-bisection of H.*

4 NP-hardness of Minimum Directed Bisection on Semicomplete Digraphs

In this section, we show that Minimum Directed Bisection is NP-hard on semicomplete digraphs by a reduction from Maximum Directed Bisection on directed acyclic graphs (DAGs). (Recall that in the Maximum Directed Bisection problem, the input consists of a digraph D and an integer k, and the task is to determine whether D admits a bisection (L, R) such that $|A(R, L)| \geq k$.) The NP-hardness of Maximum Directed Bisection on DAGs easily follows from the fact that Directed Max-Cut is NP-hard on DAGs, shown by Lampis et al. [24] (Theorem 4 in [24]).

▶ **Observation 15** ($\star$)**.** Maximum Directed Bisection *is NP-complete on DAGs.*

▶ **Lemma 16** ($\star$)**.** Minimum Directed Bisection *is NP-complete on semicomplete digraphs.*

Proof Sketch. It is easy to see that Minimum Directed Bisection belongs to NP. Now, for the proof of hardness, we give a reduction from the Maximum Directed Bisection problem on DAGs. To this end, consider an instance (D, k) of Maximum Directed Bisection where D is a DAG. Let $|V(D)| = n$. Without loss of generality, assume that n is even. (Add an isolate vertex to D if n is odd.) We construct an instance (D', k') of Minimum Directed Bisection as follows. Take $D' = \overline{D}$ and $k' = n^2/4 - k$. Note first that $\overline{D}$ is a semicomplete digraph, and that D' can be constructed in polynomial time. It can be verified that (D, k) is a yes-instance if and only if (D', k') is a yes-instance. ◀

▶ Remark 17. By Observation 13, the problem At Least One Weighted Minimum Bisection belongs to the class NP. Moreover, we have just shown that Minimum Directed Bisection is NP-complete on semicomplete digraphs. Thus, we can conclude that there is a polynomial time reduction from ALO-WMB to Minimum Directed Bisection on semicomplete digraphs.

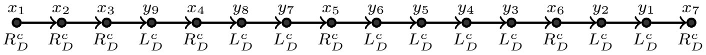

Figure 1 An example of a Hamiltonian path P_i in D_i, with the canonical positions of its vertices. We have $V(D_i) = X_i \cup Y_i$. The vertices of X_i are indexed from left to right, while the vertices of Y_i are indexed from right to left.

5 FPT Algorithm for Minimum Directed Bisection on Semicomplete Digraphs

In this section, we design an algorithm for Minimum Directed Bisection on semicomplete digraphs that runs in time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$. Our algorithm is based on a technique called chromatic coding, introduced by Alon et al. [1]. The basic idea behind this technique is to color the vertices of the input digraph by $\mathcal{O}(\sqrt{k})$ colors so that all the k arcs in a k-sized bisection (if one exists) are properly colored, i.e. have their endpoints in different color classes. As stated below, it was shown that there is a coloring family of size $2^{\mathcal{O}(\sqrt{k}\log k)}\log n$ that does this job.

▶ **Proposition 18** (Chromatic coding [1]). *For positive integers n and k, there exists a family $\mathcal{F}$ of functions from $[n]$ to $[2\lceil\sqrt{k}\rceil]$ with the following property: for every graph G with $V(G) = [n]$ and with at most k edges, there exists a function $f \in \mathcal{F}$ that properly colors $E(G)$ (i.e. for every $uv \in E(G)$, $f(u) \neq f(v)$). Moreover, there is an algorithm that runs in time $2^{\mathcal{O}(\sqrt{k}\log k)}n\log n$ and constructs such a family $\mathcal{F}$ of size $2^{\mathcal{O}(\sqrt{k}\log k)}\log n$.*

The family $\mathcal{F}$ in Proposition 18 is called a *coloring family*, and the elements of $\mathcal{F}$ are called *coloring functions*. For a coloring function $f \in \mathcal{F}$, we refer to the set of all vertices that have been assigned the color i by f as the i^{th} color class.

Informally, our algorithm for Minimum Directed Bisection is as follows. Suppose that (D, k) is a yes-instance, and let (L, R) be a bisection of D of size at most k. We would like to color the vertices of D using $\mathcal{O}(\sqrt{k})$ colors so that all arcs of $A(R, L)$ are properly colored (of course, without knowing (L, R)). Proposition 18 guarantees such a coloring function f. For any color i, let D_i be the subgraph of D induced by vertices in the i^{th} color class. Because all arcs of $A(R, L)$ are properly colored, no arc of D_i can belong to $A(R, L)$. Thus, we "guess" the vertices of D_i whose canonical positions are violated by (L, R), and define a partition of $V(D_i)$ accordingly. Due to Lemma 11 and Remark 12, there are at most $4k$ vertices in D_i whose canonical positions can be violated by (L, R). Thus we would need to consider only $k^{\mathcal{O}(\sqrt{k})} = 2^{\mathcal{O}(\sqrt{k}\log k)}$ guesses, which gives rise to an algorithm with the claimed running time. We present a formal description of our algorithm in Algorithm 1.

In the next two lemmas, we establish the correctness of Algorithm 1, and analyse its runtime.

▶ **Lemma 19** ($\star$). *Algorithm 1 is correct.*

Proof Sketch. Observe first that if Algorithm 1 returns yes, then D has a bisection of size at most k (that is (L^f_t, R^f_t) for some f and t). Therefore, in order to show that Algorithm 1 returns the correct answer, we shall prove the following. If (D, k) is a yes-instance with a k-sized bisection (L, R), then for some $f \in \mathcal{F}$ and for some tuple t defined in Step 2 of the algorithm, we will have $(L, R) = (L^f_t, R^f_t)$ where L^f_t and R^f_t are as defined in Step 2-(ii)-(a) of the algorithm. ◀

▶ **Lemma 20** ($\star$). *Algorithm 1 runs in time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$.*

Algorithm 1 Input: (D, k) where D is a semicomplete digraph on n vertices.

```
1. Run the algorithm in Proposition 18 and construct a coloring family $\mathcal{F}$.
2. For each $f \in \mathcal{F}$, perform the following steps:
   (i). For each $i \in [2\lceil\sqrt{k}\rceil]$, consider $D_i$, the subgraph of $D$ induced by $f^{-1}(i)$, and
        perform the following steps.
        (a). Find a Hamiltonian path $P_i$ in $D_i$. Then, $P_i$ is a path from $\alpha_i$ to $\beta_i$ for some
             $\alpha_i, \beta_i \in V(D_i)$.
        (b). Let $X_i = \{x \in V(D_i) \mid \texttt{can}(x) = R^c_D\}$ and $Y_i = \{y \in V(D_i) \mid \texttt{can}(y) = L^c_D\}$.
             For $1 \leq j \leq |X_i|$, let $x_j$ be the $j^{\text{th}}$ vertex of $P_i$ that belongs to $X_i$ as we
             traverse $P_i$ from $\alpha_i$ to $\beta_i$. For $1 \leq p \leq |Y_i|$, let $y_p$ be the $p$th vertex of $P_i$ that
             belongs to $Y_i$ as we traverse $\texttt{rev}(P_i)$ from $\beta_i$ to $\alpha_i$ (see Figure 1).
        (c). For each pair $(\ell_i, q_i)$, where $0 \leq \ell_i \leq \min\{|X_i|, 2k\}$ and $0 \leq q_i \leq \min\{|Y_i|, 2k\}$,
             construct a partition $(L_{\ell_i,q_i}, R_{\ell_i,q_i})$ of $V(D_i)$ as follows. For every $j \leq \ell_i$, assign
             $x_j$ to $L_{\ell_i,q_i}$, and for every $p \leq q_i$, assign $y_p$ to $R_{\ell_i,q_i}$. For every other vertex
             $z \in V(D_i)$, assign $z$ to $L_{\ell_i,q_i}$ if $\texttt{can}(z) = L^c_D$, and assign $z$ to $R_{\ell_i,q_i}$ otherwise.
   (ii). For each tuple $t = (t_1, t_2, \ldots, t_{2\lceil\sqrt{k}\rceil})$, $t_i = (\ell_i, q_i)$ for some $0 \leq \ell_i \leq \min\{|X_i|, 2k\}$
         and $0 \leq q_i \leq \min\{|Y_i|, 2k\}$ such that $(L_{\ell_i,q_i} R_{\ell_i,q_i})$ is a one way partition of $V(D_i)$,
         perform the following step.
        (a). Construct a partition $(L^f_t, R^f_t)$ of $V(D)$ as follows: $L^f_t = \bigcup_i L_{\ell_i,q_i}$ and $R^f_t =$
             $\bigcup_i R_{\ell_i,q_i}$. If $(L^f_t, R^f_t)$ is a bisection of $D$ of size at most $k$, then return that
             $(D, k)$ is a yes-instance and terminate.
3. Return that $(D, k)$ is a no-instance.
```

We have thus proved the following theorem.

▶ **Theorem 21.** Minimum Directed Bisection *on semicomplete digraphs admits a* $2^{\mathcal{O}(\sqrt{k}\log k)} n^{\mathcal{O}(1)}$ *time algorithm.*

6 Polynomial Kernel for Minimum Directed Bisection on Semicomplete Digraphs

In this section, we design a polynomial kernel for the Minimum Directed Bisection problem on semicomplete digraphs. Specifically, we prove the following theorem.

▶ **Theorem 22.** *There is a polynomial time algorithm that, given an instance* (D, k) *of* Minimum Directed Bisection *where* D *is a semicomplete digraph, produces an equivalent instance* (D', k') *of* Minimum Directed Bisection *such that* $\max\{|V(D')|, k'\} \leq k^{\mathcal{O}(1)}$.

Throughout this section, D denotes a semicomplete digraph on n vertices and k a non-negative integer. Given an instance (D, k) of Minimum Directed Bisection, our kernel proceeds as follows. In the first step, we reduce the instance (D, k) to an equivalent instance of ALO-WMB (defined in Section 2), whose size is bounded by $k^{\mathcal{O}(1)}$. As ALO-WMB belongs to the class NP (see Remark 13), and Minimum Directed Bisection is NP-complete (see Lemma 16), we know that ALO-WMB admits a polynomial time reduction to Minimum Directed Bisection. In the second step, we apply this reduction to the $k^{\mathcal{O}(1)}$-sized ALO-WMB instance to obtain an equivalent instance of Minimum Directed Bisection of size $k^{\mathcal{O}(1)}$. And in order to reduce Minimum Directed Bisection to ALO-WMB in the first step, we use an $(n, 2k, 4k^2)$-splitter. Using such a splitter, we construct $k^{\mathcal{O}(1)}$ many instances of Weighted Minimum Bisection (WMB) such that each one of these instances has size $k^{\mathcal{O}(1)}$, and at least one of them is equivalent to our original instance (D, k).

▶ Remark 23 (Assumption about k). In this section, given an instance (D, k) of Minimum Directed Bisection, we assume that $\log n \leq k$. Otherwise, if $\log n > k$, then we have $n > 2^k > 2^{\sqrt{k}\log k}$. But this implies that Algorithm 1 runs in time polynomial in n. That is, if $\log n > k$, then we can solve Minimum Directed Bisection on semicomplete digraphs in polynomial time.

Terminology. In what follows, we use the following definitions and notations. Consider an instance (D, k) of Minimum Directed Bisection. Let (L, R) be a bisection of D. Let S be the set of endpoints of the arcs in $A(R, L)$, i.e. $S = V(A(R, L))$. For some positive integer $\ell \geq |S|$, let $f : V(D) \to [\ell]$ be a function that splits S evenly. For each $i \in [\ell]$ such that $f^{-1}(i) \neq \emptyset$, let $D_i, P_i, \alpha_i, \beta_i, X_i, Y_i, x_j$ and y_p be as defined in Algorithm 1. Since f splits S evenly, and $\ell \geq |S|$, the function f is injective on S. Therefore, no arc of D_i belongs to $A(R, L)$. Moreover, note that $V(D_i)$ is the disjoint union of X_i and Y_i.

Fix $i \in [\ell]$. We say that a vertex $z \in V(D_i)$ is *marked* if $z = x_j$ for some $x_j \in X_i$ with $j \leq 2k$, or if $z = y_p$ for some $y_p \in Y_i$ with $p \leq 2k$. Otherwise, we say that z is *unmarked*. That is, the first $2k$ of vertices of P_i that belong to X_i, and the last $2k$ vertices of P_i that belong to Y_i are marked. All other vertices are unmarked. (Here, first and last are defined with respect to the traversal of P_i from α_i to β_i.) Therefore, in light of Lemma 11, if (L, R) is of size at most k, and (L, R) violates the canonical position of a vertex $v \in V(D_i)$, then v is a marked vertex. In other words, every bisection of D of size at most k, if such a bisection exists, respects the canonical position of every unmarked vertex. Let M_i denote the set of all marked vertices in $V(D_i)$.

Let $n_i = |V(D_i)|$. For $r \in [n_i]$, let v_r be the r^{th} vertex of P_i as we traverse P_i from α_i to β_i. That is, $P_i = (\alpha_i =) v_1 v_2 \dots v_{n_i} (= \beta_i)$.

▶ **Observation 24.** *We have $|M_i| = |M_i \cap X_i| + |M_i \cap Y_i| \leq 2k + 2k = 4k$. If $v_r \in X_i \cap M_i$, then for every $v_s \in X_i$ with $s \leq r$, we have $v_s \in M_i$. Similarly, if $v_r \in Y_i \cap M_i$, then for every $v_s \in Y_i$ with $s \geq r$, we have $v_s \in M_i$.*

For $r, s \in [n_i], r \leq s$, let $P_i^{r,s}$ denote the subpath of P_i from vertex v_r to vertex v_s. Also, we say that the subpath $P_i^{r,s}$ is *monochromatic* if for every $r \leq j \leq s$, the vertex v_j is unmarked, and $\texttt{can}(v_j) = \texttt{can}(v_r)$. And we say that a monochromatic subpath $P_i^{r,s}$ is a *maximal monochromatic subpath* if there is no monochromatic subpath of P_i that strictly contains $P_i^{r,s}$. That is, a maximal monochromatic subpath is a maximal subpath of P_i such that all its vertices are unmarked and have the same canonical positions.

▶ **Lemma 25** ($\star$). *Assume that (D, k) is a yes-instance and that (L, R) is a bisection of D of size at most k. Then for every i, the path P_i has at most $4k + 2$ maximal monochromatic subpaths.*

The above lemma shows that $V(D_i) \setminus M_i$ can be partitioned into at most $4k + 2$ maximal monochromatic subpaths. Additionally, $V(D_i)$ contains at most $4k$ marked vertices. Let $\mathcal{P}_{f,i}$ denote this partition of $V(D_i)$ into $4k + 2 + 4k = 8k + 2$ parts. That is, every part in $\mathcal{P}_{f,i}$ is either the set of vertices of a maximal monochromatic subpath of P_i or a singleton set consisting of a marked vertex. We call $\mathcal{P}_{f,i}$ the *monochromatic partition* of $V(D_i)$.

▶ **Lemma 26** ($\star$). *Assume that (D, k) is a yes-instance and that (L, R) is a bisection of D of size at most k. Then, for every part $Z \in \mathcal{P}_{f,i}$, either $Z \subseteq L$ or $Z \subseteq R$.*

Kernel. Our kernelization algorithm works as follows. Given (D, k), we begin by using Proposition 2 to construct an $(n, 2k, 4k^2)$-splitter $\mathcal{F}'$ in time $k^{\mathcal{O}(1)} n \log n$. By Proposition 2 and Remark 23, we get that $|\mathcal{F}'| = k^{\mathcal{O}(1)}$. For each function $f \in \mathcal{F}'$, we do as follows. For each $i \in [4k^2]$ such that $f^{-1}(i) \neq \emptyset$, we find a Hamiltonian path P_i in D_i. We say that $f \in \mathcal{F}$ is a *good function* if for every $i \in [4k^2]$ such that $f^{-1}(i) \neq \emptyset$, the path P_i has at most $4k + 2$ maximal monochromatic subpaths. Let $\mathcal{F} \subseteq \mathcal{F}'$ be the collection of good functions in $\mathcal{F}$. We now compute the monochromatic partition $\mathcal{P}_{f,i}$ of $V(D_i)$. Note that $|\mathcal{P}_{f,i}| \leq 8k+2$. Let $\mathcal{P}_f$ be the partition of $V(D)$ obtained by taking the union of $\mathcal{P}_i$ for all i, that is, $\mathcal{P}_f = \bigcup_{i \in [4k^2]} \mathcal{P}_{f,i}$. We have $|\mathcal{P}_f| \leq 4k^2 \cdot (8k + 2) = k^{\mathcal{O}(1)}$. This completes the first step our kernelizaition algorithm. Observe that this step can be executed in time $k^{\mathcal{O}(1)} n^{\mathcal{O}(1)}$.

Now, construct the $\mathcal{P}_f$-weighted contraction of D, (with weight function $w_{\mathcal{P}_f}$), and denote it by H_f. Note that $|V(H_f)| = |\mathcal{P}_f| = k^{\mathcal{O}(1)}$. As every vertex $v \in V(H_f)$ corresponds to a part of $\mathcal{P}_f$ of size at most n, we have $w_{\mathcal{P}_f}(v) = |\texttt{part}(v)| \leq n$ for every vertex $v \in V(H_f)$. Hence, $|w_{\mathcal{P}_f}(v)| \leq \log n \leq k$ for every $v \in V(H_f)$ (by Remark 23). Thus, H_f is a vertex-weighted semicomplete digraph with $k^{\mathcal{O}(1)}$ vertices and further, the weight of each vertex in $V(H_f)$ can be encoded using $k^{\mathcal{O}(1)}$ bits. This is the second step of our kernel. Observe that this step also runs in time polynomial in n.

Now, consider the instance $(\{(H_f, w_{\mathcal{P}_f}) \mid f \in \mathcal{F}\}, k)$ of ALO-WMB. Note that the size of the ALO-WMB instance $(\{(H_f, w_{\mathcal{P}_f}) \mid f \in \mathcal{F}\}, k)$ is bounded by $k^{\mathcal{O}(1)}$. The following lemma shows the equivalence of this instance to (D, k).

▶ **Lemma 27** ($\star$). *The instance (D, k) is a yes-instance of* Minimum Directed Bisection *if and only if $(\{(H_f, w_{\mathcal{P}_f}) \mid f \in \mathcal{F}\}, k)$ is a yes-instance of* ALO-WMB.

Note that the first two steps of the kernel – up to the construction of the ALO-WMB instance $(\{(H_f, w_{\mathcal{P}_f}) \mid f \in \mathcal{F}\}, k)$ – runs in polynomial time. Finally, we use the polynomial time reduction in Remark 17 to reduce the instance $(\{(H_f, w_{\mathcal{P}_f}) \mid f \in \mathcal{F}\}, k)$ of ALO-WMB to an equivalent instance (D', k') of Minimum Directed Bisection on semicomplete digraphs. As the size of the instance $(\{(H_f, w_{\mathcal{P}_f}) \mid f \in \mathcal{F}\}, k)$ is bounded by $k^{\mathcal{O}(1)}$, this reduction runs in time polynomial in k. We thus conclude that the size of the instance (D', k') is also bounded by $k^{\mathcal{O}(1)}$. This completes the proof of Theorem 22.

7 Conclusion

We studied the Minimum Directed Bisection problem from the parameterized complexity perspective. In particular, we gave an algorithm with running time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$ and a polynomial kernel for Minimum Directed Bisection on semicomplete digraphs. Some natural questions that arise from this work are: can the $\log k$ dependence on the exponent in the running time be removed? How far can we reach in improving the size of the kernel? In particular, can one prove a lower bound/existence of a linear/quadratic kernel for Minimum Directed Bisection on semi-complete digraphs? We also believe that our technique of employing splitters can be generalized for the design of kernels for many problems, especially the ones that admit algorithms crucially using chromatic coding. At least, for such problems, the use of splitters can "simulate" the effects of chromatic coding that are sufficient for kernelization.

References

1 Noga Alon, Daniel Lokshtanov, and Saket Saurabh. Fast FAST. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, pages 49–58, 2009. doi:10.1007/978-3-642-02927-1_6.

2 Noga Alon, Raphael Yuster, and Uri Zwick. Color-Coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

3 Jørgen Bang-Jensen and Gregory Z. Gutin. *Digraphs - theory, algorithms and applications*. Springer, 2002.

4 Florian Barbero, Christophe Paul, and Michal Pilipczuk. Exploring the Complexity of Layout Parameters in Tournaments and Semicomplete Digraphs. *ACM Trans. Algorithms*, 14(3):38:1–38:31, 2018. doi:10.1145/3196276.

5 Florian Barbero, Christophe Paul, and Michal Pilipczuk. Strong immersion is a well-quasi-ordering for semicomplete digraphs. *Journal of Graph Theory*, 90(4):484–496, 2019. doi:10.1002/jgt.22408.

6 Stéphane Bessy, Fedor V Fomin, Serge Gaspers, Christophe Paul, Anthony Perez, Saket Saurabh, and Stéphan Thomassé. Kernels for feedback arc set in tournaments. *Journal of Computer and System Sciences*, 77(6):1071–1078, 2011.

7 Thang Nguyen Bui and Andrew Peck. Partitioning Planar Graphs. *SIAM J. Comput.*, 21(2):203–215, 1992. doi:10.1137/0221016.

8 Paul Camion. Chemins et circuits hamiltoniens des graphes complets. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, 249(21):2151–2152, 1959.

9 Maria Chudnovsky and Paul D. Seymour. A well-quasi-order for tournaments. *J. Comb. Theory, Ser. B*, 101(1):47–53, 2011. doi:10.1016/j.jctb.2010.10.003.

10 M Cygan, F V Fomin, L Kowalik, D Lokshtanov, D Marx, M Pilipczuk, M Pilipczuk, and S Saurabh. *Parameterized algorithms*. Springer, 2015.

11 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Minimum bisection is fixed parameter tractable. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 323–332, 2014. doi:10.1145/2591796.2591852.

12 Josep Díaz and George B. Mertzios. Minimum Bisection Is NP-hard on Unit Disk Graphs. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, pages 251–262, 2014. doi:10.1007/978-3-662-44465-8_22.

13 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

14 Michael Dom, Jiong Guo, Falk Hüffner, Rolf Niedermeier, and Anke Truß. Fixed-parameter tractability results for feedback set problems in tournaments. *J. Discrete Algorithms*, 8(1):76–86, 2010. doi:10.1016/j.jda.2009.08.001.

15 Uriel Feige. Faster FAST(Feedback Arc Set in Tournaments). *CoRR*, abs/0911.5094, 2009. arXiv:0911.5094.

16 Uriel Feige and Orly Yahalom. On the complexity of finding balanced oneway cuts. *Inf. Process. Lett.*, 87(1):1–5, 2003. doi:10.1016/S0020-0190(03)00251-5.

17 Fedor V. Fomin and Michal Pilipczuk. Subexponential Parameterized Algorithm for Computing the Cutwidth of a Semi-complete Digraph. In *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, pages 505–516, 2013. doi:10.1007/978-3-642-40450-4_43.

18 Alexandra Fradkin. *Forbidden structures and algorithms in graphs and digraphs*. PhD thesis, Princeton University, 2011.

19 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some Simplified NP-Complete Graph Problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976. doi:10.1016/0304-3975(76)90059-1.

20 Klaus Jansen, Marek Karpinski, Andrzej Lingas, and Eike Seidel. Polynomial Time Approximation Schemes for MAX-BISECTION on Planar and Geometric Graphs. *SIAM J. Comput.*, 35(1):110–119, 2005. doi:10.1137/S009753970139567X.

21 Marek Karpinski and Warren Schudy. Faster Algorithms for Feedback Arc Set Tournament, Kemeny Rank Aggregation and Betweenness Tournament. In *Algorithms and Computation - 21st International Symposium, ISAAC 2010, Jeju Island, Korea, December 15-17, 2010, Proceedings, Part I*, pages 3–14, 2010. doi:10.1007/978-3-642-17517-6_3.

22 Ilhee Kim. *On containment relations in directed graphs.* PhD thesis, Princeton University, 2013.

23 Ilhee Kim and Paul D. Seymour. Tournament minors. *J. Comb. Theory, Ser. B*, 112:138–153, 2015. doi:10.1016/j.jctb.2014.12.005.

24 Michael Lampis, Georgia Kaouri, and Valia Mitsou. On the algorithmic effectiveness of digraph decompositions and complexity measures. *Discrete Optimization*, 8(1):129–138, 2011. doi:10.1016/j.disopt.2010.03.010.

25 Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006. doi:10.1016/j.tcs.2005.10.007.

26 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and Near-Optimal Derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 182–191, 1995. doi:10.1109/SFCS.1995.492475.

27 Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Contraction Decomposition in Unit Disk Graphs and Algorithmic Applications in Parameterized Complexity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1035–1054, 2019. doi:10.1137/1.9781611975482.64.

28 Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 255–264, 2008. doi:10.1145/1374376.1374415.

29 L Rédei. Ein kombinatorischer Satz. *Satz. Acta Litt. Szeged*, 7:39–43, 1934.