

Ranking and Drawing in Subexponential Time

Henning Fernau¹ Fedor V. Fomin² Daniel Lokshtanov² Matthias Mnich³
Geevarghese Philip⁴ Saket Saurabh⁴

¹ Universität Trier FB 4—Abteilung Informatik, 54286 Trier, Germany.
fernau@uni-trier.de

² Department of Informatics, University of Bergen, 5020 Bergen, Norway.
{fomin|daniello}@ii.uib.no

³ Eindhoven University of Technology, Eindhoven, The Netherlands.
m.mnich@tue.nl

⁴ The Institute of Mathematical Sciences, Taramani, Chennai 600 113, India.
{gphilip|saket}@imsc.res.in

Abstract. In this paper we obtain parameterized subexponential-time algorithms for p -KEMENY AGGREGATION (p -KAGG) — a problem in social choice theory — and for p -ONE-SIDED CROSSING MINIMIZATION (p -OSCM) — a problem in graph drawing (see the introduction for definitions). These algorithms run in time $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k} \log k)})$, where k is the parameter, and significantly improve the previous best algorithms with running times $\mathcal{O}^*(1.403^k)$ and $\mathcal{O}^*(1.4656^k)$, respectively. We also study natural “above-guarantee” versions of these problems and show them to be fixed parameter tractable. In fact, we show that the above-guarantee versions are equivalent to a weighted variant of p -DIRECTED FEEDBACK ARC SET. Our results for the above-guarantee version of p -KAGG reveal an interesting contrast. We show that when the number of “votes” in the input to p -KAGG is *odd* the above guarantee version can still be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k} \log k)})$, while if it is *even* then the problem cannot have a subexponential time algorithm unless the exponential time hypothesis fails (equivalently, unless $\text{FPT}=\text{M}[1]$).

1 Introduction

In this paper we study problems from two different areas of algorithmics: p -KEMENY AGGREGATION (p -KAGG) — a problem in computational social choice theory — and p -ONE-SIDED CROSSING MINIMIZATION (p -OSCM) — a problem in graph drawing — in the realm of parameterized complexity.

Kemeny Aggregation: Preference lists are extensively used in social science surveys and voting systems to capture information about choice. In many such scenarios there arises the need to combine the data represented by many lists into a single list which reflects the opinion of the surveyed group as much as possible. The p -KAGG problem was introduced by Kemeny [24, 25] to abstract out the problem of combining many preference lists into one. This problem appears in a variety of applications, such as in breeding problems in agronomy [21].

In p -KAGG we are given a set of permutations (also called votes) over a set of alternatives (also called candidates), and a positive integer k , and are asked for a permutation π of the set of the set of candidates, called an *optimal aggregation*, such that the sum of the Kendall-Tau distances (KT -distances) of π from all the votes is at most k . The KT -distance between two permutations π_1 and π_2 is the number of pairs of candidates that are ordered differently in the two permutations and is denoted by $KT\text{-}dist(\pi_1, \pi_2)$. The problem is known to be \mathcal{NP} -complete [5] and admits polynomial time approximation schemes (PTASs) [26]. Betzler et al. [6] considered this problem from the point of view of parameterized algorithms and obtained an algorithm that runs in time $\mathcal{O}^*(1.53^k)$ ⁵. More recently Simjour [30] obtained an algorithm for the problem that runs in time $\mathcal{O}^*(1.403^k)$.

One Sided Crossing Minimization: The graph drawing problem that we are interested in is the p -OSCM problem, which is a key ingredient of the well-known ‘‘Sugiyama approach’’ to layered graph drawing [31]. An input to this problem consists of a bipartite graph $G = (V_1, V_2, E)$, a permutation π of V_1 , and a positive integer k . The vertices of V_1 are placed on a line, also called a *layer*, in the order induced by π . The objective is to check whether there is a permutation π_m of V_2 such that, when the vertices of V_2 are placed on a second layer parallel to the first one in the order induced by π_m , then drawing a straight-line segment for each edge in E will introduce no more than k pairwise edge crossings. This seemingly simple problem is \mathcal{NP} -complete [18], even on sparse graphs [28].

The study of the parameterized algorithmics of graph drawing problems was initiated by Dujmović et al [12], and several new generic results were later obtained by Dujmović and others [13]. Dujmović and Whitesides [16] investigated the p -OSCM problem and obtained an algorithm for this problem which runs in time $\mathcal{O}^*(1.6182^k)$. This was later improved to $\mathcal{O}^*(1.4656^k)$ by Dujmović et al. [15]. There has been a similar race to obtain better approximation algorithms for the problem. To the best of our knowledge, the current best approximation factor for p -OSCM is 1.4664, due to Nagamochi [29].

Our Results. We obtain $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k} \log k)})$ -time algorithms for both p -KAGG and p -OSCM. These significantly improve the previous best algorithms with running times $\mathcal{O}^*(1.403^k)$ and $\mathcal{O}^*(1.4656^k)$, respectively. Both of our algorithms are based on modeling these problems as the p -WEIGHTED DIRECTED FEEDBACK ARC SET (p -WDFAS) problem. In p -WDFAS we are given a directed (multi)graph $D = (V, A)$, a weight function $w : A \rightarrow \mathbb{R}^+$ and a positive integer k , and the objective is to find a set of arcs $F \subseteq A$ of total weight at most k such that deleting F from D makes D a directed acyclic graph; such an F is called a *feedback arc set* of D . Both p -KAGG and p -OSCM have been modeled as p -WDFAS in earlier work as well [1, 30, 31]; the novelty in our modeling is that it allows us to work with p -WDFAS on ‘‘tournament-like’’ structures. We call this

⁵ The \mathcal{O}^* notation suppresses polynomial terms in the expression.

specialized problem p -FAST (parameterized Feedback Arc Set on Tournament-like structures). A tournament is a digraph in which between every two vertices there is exactly one arc. By a tournament-like structure, we mean a directed (multi)graph on n vertices that contains a tournament on n vertices as a subgraph. Our modeling allows us to use the chromatic-coding technique recently developed by Alon et al. [3], which they used to obtain the first subexponential time algorithm for p -WDFAS on tournaments.

We also study natural “above-guarantee” versions of these problems and show them to be fixed parameter tractable. We show that the above-guarantee versions of p -KAGG (A - p -KAGG) and p -OSCM (A - p -OSCM) are both equivalent to p -WDFAS and hence both have algorithms that run in time $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ [9]. A finer analysis of A - p -KAGG reveals an interesting contrast in its running time: if the number of votes in the input to p -KAGG is *odd*, then A - p -KAGG can still be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k} \log k)})$, while if it is *even*, then the problem cannot have any subexponential-time algorithm unless the exponential time hypothesis (ETH) is false [22], or equivalently [19], unless $\text{FPT}=\text{M}[1]$.

It is also worth mentioning that our reduction from p -OSCM to p -WDFAS on tournaments implies a PTAS for the graph drawing problem. To summarize, we analyze a common feature of p -KAGG and p -OSCM to provide new insights and findings of interest to both the Graph Drawing community and the Social Choice community.

2 Preliminaries

A *parameterized problem* Π is a subset of $\Gamma^* \times \mathbb{N}$, where Γ is a finite alphabet. An instance of a parameterized problem is a tuple (x, k) , where k is called the parameter. A central notion in parameterized complexity is *fixed-parameter tractability (FPT)* which means, for a given instance (x, k) , decidability in time $\mathcal{O}(f(k) \cdot p(|x|))$, where f is an arbitrary function of k and p is a polynomial in the input size.

Let Π_1, Π_2 be two parameterized problems. A *parameterized reduction* from Π_1 to Π_2 is an algorithm that takes an instance (x, k) of Π_1 as input, runs in time $\mathcal{O}(f(k) \cdot p(|x|))$, and outputs an instance (y, ℓ) of Π_2 such that ℓ is some function of k alone and (x, k) is a YES instance of Π_1 if and only if (y, ℓ) is a YES instance of Π_2 .

A *tournament* is a directed graph in which there is exactly one directed arc between every two vertices. A feedback arc set in a tournament is a set of arcs whose *reversal* results in a DAG. A *tournament-like graph* is a directed (multi)graph on n vertices, for some $n \in \mathbb{N}$, which contains a tournament on n vertices as a subgraph.

3 FPT Algorithms for p -KAGG

Let S be a finite set, and let π_1, π_2 be two permutations of S . For $u, v \in S$, we define

$$d_{\pi_2}^{\pi_1}(u, v) = \begin{cases} 0 & \text{if } \pi_1 \text{ and } \pi_2 \text{ rank } u \text{ and } v \text{ in the same order} \\ 1 & \text{otherwise} \end{cases}$$

The Kendall-Tau distance (*KT-distance*) of π_1 and π_2 is defined as: $KT\text{-dist}(\pi_1, \pi_2) = \sum_{\{u,v\} \subseteq S} d_{\pi_2}^{\pi_1}(u, v)$.

Let C be a set of candidates and V a set of votes over C . For any permutation r of C , the *Kemeny Score of r with respect to V* is defined as: $KS(r, V) = \sum_{\pi \in V} KT\text{-dist}(r, \pi)$. Observe that

$$KS(r, V) = \sum_{\pi \in V} KT\text{-dist}(r, \pi) = \sum_{\pi \in V} \sum_{\{u,v\} \subseteq C} d_{\pi}^r(u, v) = \sum_{\{u,v\} \subseteq C} \sum_{\pi \in V} d_{\pi}^r(u, v) \quad (1)$$

3.1 Parameterized Reduction from p -KAGG to p -WDFAS

We now describe a parameterized reduction from p -KAGG to p -WDFAS, briefly mentioned by Betzler et al. [6], which runs in *polynomial* time and takes the parameter from k to k . Let (C, V, k) be an instance of p -KAGG. In what follows, we assume without loss of generality that $|V| \geq 1$. We construct a digraph G such that (C, V, k) is a YES instance of p -KAGG if and only if G has a feedback arc set of weight at most k .

We set the vertex set of G to be the set C of candidates. For each vote $\pi_i \in V$ and for each pair of vertices (u, v) of G , we add a new arc with weight 1 from u to v in G if and only if u appears before v in π_i (equivalently, when u is preferred over v by π_i). This completes the construction; the parameter is k .

Fix a vote $\pi_i \in V$. For each pair of candidates $u, v \in C$, π_i prefers exactly one of these candidates over the other. Thus, for any two vertices u, v of G , each vote contributes exactly one arc between u and v in G . As a consequence, we have:

Observation 1 *Let G be the digraph constructed from an instance (C, V, k) of p -KAGG as described above. For any two vertices u, v of G , let i be the number of arcs in G from u to v , and j the number of arcs from v to u . Then $i + j = |V|$.*

The next two claims show that the reduction is sound.

Claim 1. Let (C, V, k) be a YES instance of p -KAGG, and let G be the digraph constructed from (C, V, k) as described above. Then G has a feedback arc set of weight at most k .

Proof. Since (C, V, k) is a YES instance of p -KAGG, there exists a permutation r of the set C such that $\sum_{\pi \in V} KT\text{-dist}(r, \pi) \leq k$. For $u, v \in V(G)$, let $\overline{r_{uv}}$ be the set of arcs in G between u and v that are oriented *contrary* to the direction implied by r . That is, if u appears before v in r , then $\overline{r_{uv}}$ consists of all arcs from v to u in G ; if u appears after v in r , then $\overline{r_{uv}}$ consists of all arcs from u to v in G . Using Equation 1, we get $\sum_{\{u,v\} \subseteq C} \sum_{\pi \in V} d_{\pi}^r(u, v) \leq k$. By construction, this implies $\sum_{\{u,v\} \subseteq V(G)} |\overline{r_{uv}}| \leq k$.

That is, there are at most k arcs in G , each of weight exactly 1, that are oriented contrary to the directions implied by r . Reversing these arcs, we get a digraph G' in which every arc is oriented according to the direction implied by r . Since r is a permutation of $V(G) = V(G')$, it follows that G' is acyclic. \square

Claim 2. Let G be the digraph constructed from an instance (C, V, k) of p -KAGG as described above. If G has a feedback arc set S of weight at most k , then (C, V, k) is a YES instance of p -KAGG.

Proof. Note that since each arc in G has weight exactly 1, S contains exactly k arcs. Consider the DAG G' obtained from G by reversing the arcs in S . Note that this operation preserves the number of arcs between any pair of vertices. From Observation 1, and since G' is a DAG, between each pair u, v of vertices of G' there are exactly $|V|$ arcs, all of which are in the same direction. The arcs of G' thus define a permutation r of C , where for any $u, v \in C$, u appears before v in r if and only if there is an arc (in fact, $|V|$ arcs) from u to v in G' . For $u, v \in V(G)$, let $\overline{r_{uv}}$ be the set of arcs between u and v in G that are oriented *contrary* to the direction implied by r . Then $\cup_{\{u,v\} \subseteq V(G)} \overline{r_{uv}} = S$, $\sum_{\{u,v\} \subseteq V(G)} |\overline{r_{uv}}| = |S| \leq k$, and from this and the construction we get $\sum_{\{u,v\} \subseteq C} \sum_{\pi \in V} d_{\pi}^r(u, v) \leq k$. From Equation 1 it follows that $KS(r, V) \leq k$, and so (C, V, k) is a YES instance of p -KAGG. \square

The above reduction can clearly be done in polynomial time, and the number of vertices in the reduced instance (G, k) is equal to the number of candidates $|C|$ in the input instance (C, V, k) . Further, the reduced instance has at least one arc (in fact, exactly $|V|$ arcs) between every pair of vertices. Let H be the edge-weighted digraph obtained from G by replacing parallel arcs with single weighted arcs in the natural way. That is, if there are $i > 0$ arcs from u to v in G , then H contains a single arc of weight i from u to v . It is easy to verify that H has a feedback arc set of weight at most k if and only if G has a feedback arc set of weight at most k . Hence from Claims 1 and 2 we have

Lemma 1. *Given an instance (C, V, k) of p -KAGG, we can construct, in polynomial time, an equivalent instance (G, k) of p -WDFAS where G is a tournament-like graph and $|V(G)| = |C|$.*

3.2 A Subexponential FPT Algorithm for p -KAGG

Our algorithm is based on the observation that the algorithm of Alon et al. [3] for p -WDFAS on tournaments also works for tournament-like graphs. The algorithm

presented in [3] starts by preprocessing the instance and obtains an equivalent instance with at most $\mathcal{O}(k^2)$ vertices in polynomial time. That is, given a tournament T and a positive integer k , in polynomial time the preprocessing algorithm either concludes that T does not have a feedback arc set of weight at most k or finds a new tournament T' with $\mathcal{O}(k^2)$ vertices and $k' \leq k$ such that the original tournament T has a feedback arc set of weight at most k , if and only if T' has a feedback arc set of weight at most k' . This preprocessing allows them to assume that the instance where they actually apply the subexponential time algorithm is of size $\mathcal{O}(k^2)$ only, which is integral to their time analysis. Their preprocessing can also be applied to tournament-like graphs by allowing both directed cycles of length two and triangles in the reduction rules proposed in [3, Lemma 1]. So we always first apply these preprocessing rules and obtain a tournament-like graph on $\mathcal{O}(k^2)$ vertices. Let the preprocessed tournament-like graph be $T = (V, A)$.

To obtain our algorithm we also use *universal coloring families* introduced in [3]. For integers m, k and r , a family \mathcal{F} of functions from $[m]$ to $[r]$ is called a universal (m, k, r) -coloring family if for any graph G on the set of vertices $[m]$ with at most k edges, there exists an $f \in \mathcal{F}$ which is a proper vertex coloring of G . The following result gives a bound on the size of universal coloring families.

Proposition 1. [3] *For any $n > 10k^2$ there exists an explicit universal $(n, k, \mathcal{O}(\sqrt{k}))$ -coloring family \mathcal{F} of size $|\mathcal{F}| \leq 2^{\mathcal{O}(\sqrt{k} \log k)} \log n$.*

We enumerate each function in the universal coloring family and then color the vertices of T with these functions. Observe that since the number of arcs possible in the solution is at most k , there exists a function $f \in \mathcal{F}$ such that no end-points of the arc in the solution is colored with same color, that is, no arc of the solution is monochromatic. Now using the dynamic programming algorithm proposed in [3, Lemma 3] we can find a feedback arc set of weight at most k of T , if there exists one, in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{k} \log k)})$. This yields the following theorem.

Theorem 1. *The p -KEMENY AGGREGATION problem with n candidates can be solved in $2^{\mathcal{O}(\sqrt{k} \log k)} + n^{\mathcal{O}(1)}$ time.*

This is a significant improvement over the previous best known algorithm for p -KEMENY AGGREGATION which runs in $\mathcal{O}^*(1.403^k)$ time [30].

3.3 FPT Algorithms for A - p -KAGG

Consider an instance of the p -KAGG problem. Let π be *any* permutation of the candidate set C , let V be the set of all votes and let $KS(\pi, V)$ denote the sum of the KT -distances of π from all the votes in the set V . Suppose A and B are two candidates in the input, and let i votes prefer A over B and j votes prefer B over A . Clearly, the pair $\{A, B\}$ contributes at least $\min(i, j)$ to $KS(\pi, V)$. For $\{u, v\} \subseteq C$, let $I(u, v)$ (respectively $J(u, v)$) be the number of votes that rank u before v (respectively v before u), and let $g = \sum_{\{u, v\} \subseteq C} \min\{I(u, v), J(u, v)\}$. Then $KS(\pi, V) \geq g$, and so in the natural above-guarantee version of p -KAGG,

which we call A - p -KAGG, we ask for a permutation π of C such that $KS(\pi, V) \leq g + k$.

We now describe a reduction from A - p -KAGG to p -WDFAS, originally due to Dwork et al. [17]. When the number of votes in the input instance is odd (A - p -KAGG(*odd*)), the reduced instance is a tournament with positive integral edge weights. When the number of votes is even (A - p -KAGG(*even*)), the reduced instance is not necessarily a tournament. In both cases, the parameter goes from k to k . That is, the reduction takes A - p -KAGG(*odd*) to p -WDFAS on tournaments, and A - p -KAGG(*even*) to p -WDFAS in general digraphs, in both cases preserving the parameter. Together with the subexponential FPT algorithm of Alon et al. [3] for p -WDFAS on tournaments, this implies a subexponential FPT algorithm for A - p -KAGG(*odd*). In the next subsection we describe a parameterized reduction from p -WDFAS to A - p -KAGG(*even*) in which the parameter goes from k to $2k$. This implies that A - p -KAGG(*even*) does not have a subexponential FPT algorithm unless the exponential time hypothesis is false.

Let (C, V, k) be an instance of A - p -KAGG. We construct an instance (H, k) of p -WDFAS in two stages, as follows.

Stage 1. We construct a digraph G exactly as in the previous reduction. We set the vertex set of G to be the set C of candidates. For each vote $\pi_i \in V$ and for each pair of vertices (u, v) of G , we add a new arc of weight 1 from u to v in G if and only if u appears before v in π_i (equivalently, when u is preferred over v by π_i).

Stage 2. We now prune the “above-guarantee” arcs of G . We process every two-vertex subset $\{u, v\}$ of G as follows: Let there be a total of i arcs from u to v and j arcs from v to u in H . Assume without loss of generality that $i \geq j$. We replace all the arcs between u and v by a single arc of weight $i - j$ from u to v . If $i - j = 0$, then we just remove all the arcs between u and v , and do not add any arc to replace them. We repeat this for every 2-subset of vertices of G to obtain a digraph H with integer-weighted arcs. (H, k) is the desired instance of p -WDFAS.

Suppose the number $|V|$ of votes in the input instance (C, V, k) is odd. Then, with the same notation as above, $i + j = |V|$ is odd for each 2-subset $\{u, v\}$ of G (Observation 1), and so $i > j$. Thus there is exactly one arc between every two vertices of H , and so H is a tournament. If $|V|$ is even, then it is possible that $i = j$ for some $\{u, v\} \subseteq V(G)$, and so in H there will not be any arc between u and v . Hence when $|V|$ is odd, H is not necessarily a tournament or a tournament-like graph.

Dwork et al. [17] show that the above reduction is sound; see also Mahajan et al. [27]:

Lemma 2. [17, 27] *Let (H, k) be the instance of p -WDFAS obtained from an instance (C, V, k) of A - p -KAGG as described above. Then (H, k) is a YES instance of p -WDFAS if and only if (C, V, k) is a YES instance of A - p -KAGG.*

The fastest known FPT algorithm for p -WDFAS runs in $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ time [9], and the fastest known FPT algorithm for p -WDFAS on tournaments runs in $2^{\mathcal{O}(\sqrt{k} \log k)} + n^{\mathcal{O}(1)}$ time [3]. Hence from Lemma 2 we get

Theorem 2. *The A - p -KAGG problem with n candidates can be solved in $2^{\mathcal{O}(\sqrt{k} \log k)} + n^{\mathcal{O}(1)}$ time when the number of votes is odd, and in $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ time when the number of votes is even.*

3.4 A Lower Bound for A - p -KAGG(*even*)

We now argue that the A - p -KAGG(*even*) problem does not have a subexponential FPT algorithm unless the exponential time hypothesis (ETH) is false. To see this, consider the following sequence of two reductions:

$$\text{VERTEX COVER} \rightarrow \text{DIRECTED FEEDBACK ARC SET} \rightarrow A\text{-}p\text{-KAGG}$$

The first reduction is due to Karp [23], and the second is due to Dwork et al. [17, Theorem 14]. This sequence of reductions take an input instance (G, k) of VERTEX COVER where G is a graph on n vertices and m edges and $k \leq n$ is a positive integer, and outputs an instance $(C, V, 2k)$ of A - p -KAGG(*even*) where $|C| = 3n + 2m$, $|V| = 4$, and the guarantee is $g = 2\binom{2n}{2} + \binom{n+2m}{2} + n + 2m$; see the references for details. Suppose A - p -KAGG(*even*) has an algorithm that runs in time $\mathcal{O}^*(2^{\mathcal{O}(k)})$. Since $k = \mathcal{O}(n)$ throughout the reduction, we can then use this algorithm to solve VERTEX COVER in $\mathcal{O}^*(2^{\mathcal{O}(n)})$ time: We first apply the above sequence of reductions and then apply the supposed subexponential FPT algorithm for A - p -KAGG(*even*) to the resulting instance. This would in turn imply that ETH is false [22], and so we have

Theorem 3. *The A - p -KAGG problem with an even number of votes cannot be solved in $\mathcal{O}^*(2^{\mathcal{O}(k)})$ time unless ETH is false.*

4 FPT Algorithms for p -OSCM

Let $(G = (V_1, V_2, E), \pi, k)$ be an instance of p -OSCM. In what follows, we assume without loss of generality that in G , every vertex in V_2 has at least one neighbor in V_1 .

4.1 Parameterized Reduction from p -OSCM to p -WDFAS

We now describe a parameterized reduction from p -OSCM to p -WDFAS which runs in *polynomial* time and takes the parameter from k to k . Let $(G = (V_1, V_2, E), \pi, k)$ be an instance of p -OSCM. We construct a digraph H as follows: H has one vertex for each vertex of V_2 . For $\{u, v\} \subseteq V_2$, we draw the arc uv with weight C_{uv} if $C_{uv} > 0$.

Claim 3. Let $(G = (V_1, V_2, E), \pi, k)$ be an instance of p -OSCM, and let H be the digraph obtained from this instance as described above. $(G = (V_1, V_2, E), \pi, k)$ is a YES instance of p -OSCM if and only if H has a feedback arc set of weight at most k .

Proof. Suppose $(G = (V_1, V_2, E), \pi, k)$ is a YES instance of p -OSCM, and let π_m be a permutation of V_2 that witnesses this fact. Place the vertices of H on a line in the order induced by π_m : u is to the left of v if and only if u comes before v in π_m . From the construction it is clear that the sum of the weights of the arcs in H that go from left to right is at most k , and so these arcs together form a feedback arc set of H of weight at most k .

Now suppose S is a *minimal* feedback arc set of H of weight at most k . Let π' be the unique permutation of V_2 such that if we place the vertices of H on a line in the order specified by π' , then the arcs that go from left to right are exactly the arcs in S . It is easily verified that if the vertices of V_2 are placed on the second layer in the order specified by π' , then the number of crossings will be at most k . \square

The above reduction can clearly be done in polynomial time, and the graph H in the reduced instance (H, k) has $|V_2|$ vertices, where the p -OSCM instance is $(G = (V_1, V_2, E), \pi, k)$. Further, it is not difficult to see that the reduced instance has at least one arc between every pair of vertices. Hence from Claim 3 we have

Lemma 3. *Given an instance $(G = (V_1, V_2, E), \pi, k)$ of p -OSCM, we can construct, in polynomial time, an equivalent instance (H, k) of p -WDFAS where H is a tournament-like graph and $|V(H)| = |V_2|$.*

4.2 A Subexponential FPT Algorithm for p -OSCM

From Lemma 3, and using the same argument as in Section 3.2, we get

Theorem 4. *The p -ONE-SIDED CROSSING MINIMIZATION problem can be solved in $2^{\mathcal{O}(\sqrt{k} \log k)} + n^{\mathcal{O}(1)}$ time, where n is the number of vertices in the layer that is not fixed.*

4.3 Lower and Upper Bounds for A - p -OSCM

Let $(G = (V_1, V_2, E), \pi, k)$ be an instance of p -OSCM. For two vertices $u, v \in V_2$, let C_{uv} denote the number of crossings of edges incident to u with edges incident to v , when u appears before v in the second layer. It is known [16] that for a given graph G and a fixed ordering π of the vertices of V_1 , C_{uv} is a constant and can be computed in polynomial time. It is also known that the minimum possible number of crossings is $g = \sum_{\{u,v\} \subseteq V_2} \min(C_{uv}, C_{vu})$ [16]. So in the natural above-guarantee version of p -OSCM, which we call A - p -OSCM, we ask for a permutation π of V_2 such that the number of crossings induced by π is at most $g + k$.

Given an instance $(G = (V_1, V_2, E), \pi, k)$ of p -OSCM, the well-known *penalty graph* construction of Sugiyama et al. [31] constructs a arc-weighted digraph H with V_2 as the vertex set, and there is an arc in H from u to v with weight $C_{vu} - C_{uv}$ if $C_{uv} < C_{vu}$. It is easy to verify that there is a permutation π_m of V_2 such that the number of crossings induced by π_m is at most $g + k$ if and only if H has a feedback arc set of weight at most k . Thus, using the algorithm in [9] we have

Theorem 5. *The A - p -OSCM problem can be solved in $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ time.*

Muñoz et al. describe a reduction from DIRECTED FEEDBACK ARC SET to p -OSCM that in fact is a parameterized reduction from DIRECTED FEEDBACK ARC SET (where the parameter k is the solution size) to A - p -OSCM which takes the parameter from k to $2k$ [28, Proof of Theorem 1]. Hence by a similar argument as in Section 3.4 we have

Theorem 6. *The A - p -OSCM problem cannot be solved in $\mathcal{O}^*(2^{o(k)})$ time unless ETH is false.*

5 Conclusion and Future Work

In this paper we modeled two problems, from two different domains, as the weighted feedback arc set problem on tournament-like structures. This allowed us to utilize the recently developed technique of chromatic-coding [3] to obtain subexponential-time algorithms, that is, algorithms that run in time $\mathcal{O}^*(c^{\sqrt{k} \log k})$, for p -KEMENY AGGREGATION and p -ONE-SIDED CROSSING MINIMIZATION. The running time of these algorithms might be seen as a breakthrough compared to the hitherto best published algorithms, which had running times of the form roughly $\mathcal{O}^*(1.5^k)$. It is worth mentioning that apart from problems on graphs of bounded genus, only very few problems are known to have running times of the form $\mathcal{O}^*(c^{\sqrt{k}})$ [2, 10, 11].

Our approach also allowed us to show that the above-guarantee versions of these problems are fixed parameter tractable with algorithms having running times of the form $\mathcal{O}^*(c^{k \log k})$. We also show that the above-guarantee versions of these problems cannot have algorithms that run in $\mathcal{O}^*(2^{o(k)})$ time, unless the well known exponential time hypothesis fails.

We believe that our approach will generalize to other related problems considered in the literature. We cite a few concrete examples in the following.

- Çakiroglu et al. [8] considered drawing graphs with edge weights. If two edges cross, then the crossing receives as a weight the product of both edge weights involved. The overall weight of a crossing is then the sum of all respective crossing weights, and the goal is to minimize this weight.
- Forster [20] considered the so-called *constraint variant* where the ordering of some of the vertices of the free layer is already fixed (as part of the input). This can be clearly modeled by the so-called POSITIVE WEIGHTED COMPLETION OF AN ORDERING (PWCO) as studied in [14]. There, also an FPT result was announced, with a running time of $\mathcal{O}^*(1.52^k)$.

- In radial drawings of graphs, also the restricted (\mathcal{NP} -complete) variant called RADIAL ONE-SIDED TWO-LEVEL CROSSING MINIMIZATION has been considered [4].

It also might be interesting to consider the crossing minimization variant of these problems that attempts to minimize the maximum number of crossings per edge as proposed in [7] from the viewpoints of fixed parameter tractability and of approximability.

References

1. N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *STOC '05: Proceedings of the thirty-seventh annual ACM Symposium on Theory of Computing*, pages 684–693. ACM, 2005.
2. J. Alber, H. Fernau, and R. Niedermeier. Parameterized complexity: exponential speedup for planar graph problems. *Journal of Algorithms*, 52:26–56, 2004.
3. N. Alon, D. Lokshtanov, and S. Saurabh. Fast FAST. In S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. E. Nikolettseas, and W. Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Part I*, volume 5555 of *LNCS*, pages 49–58. Springer, 2009.
4. C. Bachmaier. A radial adaptation of the Sugiyama framework for visualizing hierarchical information. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):583–594, 2007.
5. J. Bartholdi, III, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.
6. N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. Fixed-parameter algorithms for Kemeny scores. In R. Fleischer and J. Xu, editors, *Algorithmic Aspects in Information and Management AAIM 2008*, volume 5034 of *LNCS*, pages 60–71. Springer, 2008.
7. T. C. Biedl, F.-J. Brandenburg, and X. Deng. Crossings and permutations. In P. Healy and N. S. Nikolov, editors, *Graph Drawing GD 2005*, volume 3843 of *LNCS*, pages 1–12. Springer, 2006.
8. O. A. Çakiroglu, C. Erten, Ö. Karatas, and M. Sözdinler. Crossing minimization in weighted bipartite graphs. In C. Demetrescu, editor, *Experimental Algorithms, 6th International Workshop, WEA*, volume 4525 of *LNCS*, pages 122–135. Springer, 2007.
9. J. Chen, Y. Liu, S. Lu, B. O’Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. In *STOC '08: Proceedings of the 40th annual ACM Symposium on Theory of Computing*, pages 177–186. ACM, 2008.
10. E. D. Demaine, F. V. Fomin, M. T. Hajiaghayi, and D. M. Thilikos. Subexponential parameterized algorithms on graphs of bounded genus and H -minor-free graphs. In *Symposium on Discrete Algorithms SODA*, pages 823–832. ACM, 2004.
11. F. Dorn, F. V. Fomin, and D. M. Thilikos. Subexponential parameterized algorithms. In L. Arge, C. Cachin, T. Jurdzinski, and A. Tarlecki, editors, *Automata, Languages and Programming ICALP*, volume 4596 of *LNCS*, pages 15–27. Springer, 2007.
12. V. Dujmović, M. R. Fellows, M. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. A. Rosamond, M. Suderman, S. Whitesides, and D. R. Wood. A fixed-parameter approach to 2-layer planarization. *Algorithmica*, 45:159–182, 2006.

13. V. Dujmović, M. R. Fellows, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. A. Rosamond, S. Whitesides, and D. R. Wood. On the parameterized complexity of layered graph drawing. *Algorithmica*, 52(2):267–292, 2008.
14. V. Dujmović, H. Fernau, and M. Kaufmann. Fixed parameter algorithms for one-sided crossing minimization revisited. In G. Liotta, editor, *Graph Drawing, 11th International Symposium GD 2003*, volume 2912 of *LNCS*, pages 332–344. Springer, 2004.
15. V. Dujmović, H. Fernau, and M. Kaufmann. Fixed parameter algorithms for one-sided crossing minimization revisited. *Journal of Discrete Algorithms*, 6:313–323, 2008.
16. V. Dujmović and S. Whitesides. An efficient fixed parameter tractable algorithm for 1-sided crossing minimization. *Algorithmica*, 40:15–32, 2004.
17. C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited. Manuscript available at <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.5118>.
18. P. Eades and N. C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11:379–403, 1994.
19. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, 2006.
20. M. Forster. A fast and simple heuristic for constrained two-level crossing reduction. In J. Pach, editor, *Graph Drawing GD*, volume 3383 of *LNCS*, pages 206–216, 2004.
21. A. Guénoche, B. Vandeputte-Riboud, and J. Denis. Selecting varieties using a series of trials and a combinatorial ordering method. *Agronomie*, 14:363–375, 1994.
22. R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
23. R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Communications*, pages 85–103, 1972.
24. J. Kemeny. Mathematics without numbers. *Daedalus*, 88:571–591, 1959.
25. J. Kemeny and J. Snell. *Mathematical models in the social sciences*. Blaisdell, 1962.
26. C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *STOC '07: Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*, pages 95–103. ACM, 2007.
27. M. Mahajan, V. Raman, and S. Sikdar. Parameterizing above or below guaranteed values. *Journal of Computer and System Sciences*, 75(2):137–153, 2009.
28. X. Muñoz, W. Unger, and I. Vrt'o. One sided crossing minimization is NP-hard for sparse graphs. In P. Mutzel, M. Jünger, and S. Leipert, editors, *9th International Symp. on Graph Drawing GD'01*, volume 2265 of *LNCS*, pages 115–123. Springer, 2002.
29. H. Nagamochi. An improved bound on the one-sided minimum crossing number in two-layered drawings. *Discrete and Computational Geometry*, 33:569–591, 2005.
30. N. Simjour. Improved parameterized algorithms for the Kemeny aggregation problem. In J. Chen and F. V. Fomin, editors, *Parameterized and Exact Computation, 4th International Workshop, IWPEC*, volume 5917 of *LNCS*, pages 312–323. Springer, 2009.
31. K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Systems Man Cybernet.*, 11(2):109–125, 1981.