Fair Short Paths in Vertex-Colored Graphs

Matthias Bentert, Leon Kellerhals, Rolf Niedermeier

Technische Universität Berlin, Algorithmics and Computational Complexity
{matthias.bentert, leon.kellerhals}@tu-berlin.de

Abstract

The computation of short paths in graphs with arc lengths is a pillar of graph algorithmics and network science. In a more diverse world, however, not every short path is equally valuable. For the setting where each vertex is assigned to a group (color), we provide a framework to model multiple natural fairness aspects. We seek to find short paths in which the number of occurrences of each color is within some given lower and upper bounds. Among other results, we prove the introduced problems to be computationally intractable (NP-hard and parameterized hard with respect to the number of colors) even in very restricted settings (such as each color should appear with exactly the same frequency), while also presenting an encouraging algorithmic result ("fixed-parameter tractability") related to the length of the sought solution path for the general problem.

Introduction

Travel agency WhataWonderfulWorld offers adventure bus trips from New Orleans to New York, making stops at exciting country sites for impressive day trips throughout the journey. To address environmental demands, the agency wishes to minimize the overall travel distance, while at the same time striving to maximize the variety and balance of impressions gathered at the day trips. Clearly, such a sustainable travel from a starting point s to an endpoint t can be modeled as finding an s-t-path in a graph with positive arc lengths. To model maximum variety and balance, the vertices—the places to visit—are colored according to agency-chosen categories, say blue vertices are interesting for bathing, green vertices for hiking, et cetera. The quest is to find a short travel path in which the colors are *fairly* represented. Herein, what is to be considered *fair* may depend on the setting. In an idealistic (and hypothetical) setting, one may consider it fair to have the same amount of beaches, hiking spots, and places of any other type on the path. We call such paths balance-fair; an example for such a path is depicted in Figure 1. Possibly, one has asked the travelers beforehand about their preferences. With this, one gains a better understanding on how many places of each type should be visited and one may be able to give lower and upper bounds for each type. The latter (more realistic) fairness constraint indeed generalizes upon multiple fairness concepts introduced in the literature. We discuss these later in this section.

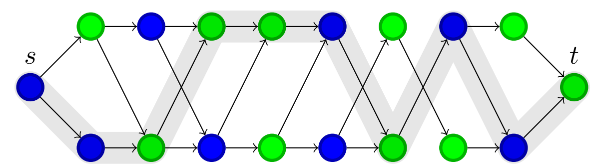

Figure 1: A graph with colored vertices (blue and green), unit-length arcs, and two vertices s and t is depicted. The highlighted path is a shortest path between s and t and contains five blue and five green vertices. Thus, it is balance-fair.

Our contributions. We introduce and study two natural fairness scenarios for one of the back-bone problems in network algorithmics: finding short paths. The problems in consideration are the following.

BALANCE-FAIR SHORTEST PATH

Input: A directed graph $G = (V, A)$, a vertex coloring $\chi\colon V \to [c]$, an arc-length function $w\colon A \to \mathbb{N}$, and two vertices $s, t \in V$.

Question: Is one of the shortest s-t-paths *balance-fair*, that is, it visits the same number of vertices of each color?

SHORT PATH WITH LOWER AND UPPER BOUNDS

Input: A directed graph $G = (V, A)$, a vertex coloring $\chi\colon V \to [c]$, an arc-length function $w\colon A \to \mathbb{N}$, two vertices $s, t \in V$, and integers $\ell, \alpha_1, \alpha_2, \ldots, \alpha_c, \beta_1, \beta_2, \ldots, \beta_c$.

Question: Is there an s-t-path P of length at most ℓ such that the number of vertices in P of color i is at least α_i and at most β_i for each $i \in [c]$?

Simply put, SHORT PATH WITH LOWER AND UPPER BOUNDS allows for very general fairness constraints, while BALANCE-FAIR SHORTEST PATH focuses on a seemingly more simple (and possibly idealistic) constraint and only allows for shortest paths. Note that BALANCE-FAIR SHORTEST PATH is indeed a special case of SHORT PATH WITH LOWER AND UPPER BOUNDS by setting $\ell = \operatorname{dist}(s, t)$ and fixing all lower and upper bounds to $(\ell - 1)/c$. Already the special case BALANCE-FAIR SHORTEST PATH turns out to be NP-hard in general. We cope with this computational intractability by investigating the parameterized complexity of these two problems with respect to the two perhaps most natural parameters: the number of different vertex colors in the input graph and the length of the sought-after solution path. Despite the fact that BALANCE-FAIR SHORTEST PATH seems much more restricted than SHORT PATH WITH LOWER AND UPPER BOUNDS at first glance, we only identify very minor differences in their (parameterized) complexity, that is, we show most of our hardness results for BALANCE-FAIR SHORTEST PATH and most of our positive results for SHORT PATH WITH LOWER AND UPPER BOUNDS. Our results suggest that the difference in complexity is due to asking for a shortest path versus asking for any path, but not due to the more general fairness criterion. We prove that finding a balance-fair shortest path is W[1]-hard for the parameter number of colors and that such paths can be found in polynomial time for any constant number of colors. If, however, we do not require the solution to be a shortest path, then the problem becomes NP-hard for only two colors. Indeed, both hardness results hold even when all arcs have the same lengths. Lastly, we show that the more general variant SHORT PATH WITH LOWER AND UPPER BOUNDS is fixed-parameter tractable with respect to the length of the path. The algorithm is based on representative families.

Results marked with ★ are deferred to the full version (Bentert, Kellerhals, and Niedermeier 2022a).

Related work. Path finding in vertex-colored graphs has been a subject of broad and intensive study. Here, we only point to algorithmically motivated work that seems particularly close to our scenario. First of all, a colorful path (sometimes also called a rainbow path or a multicolored path) is a path containing each color at most once and finding colorful paths is an important algorithmic topic, both in static and in temporal graphs (Alon, Yuster, and Zwick 1995; Dondi and Hosseinzadeh 2021). Cohen et al. (2021) analyze the complexity of finding tropical paths, that is, paths containing at least one vertex of each color. They provide both tractability and intractability results.

Another close (and also broad) area is that of finding resource-constrained shortest paths, where, roughly speaking, the desired path shall have minimum cost and only a limited consumption of resources. Generally, the problem is NP-hard (Handler and Zang 1980) and it has been extensively studied over the years (Ford et al. 2022; Irnich and Desaulniers 2005; Pugliese and Guerriero 2013). Given its intractability, several heuristics have been proposed recently (Ahmadi et al. 2021). These models however do not have fairness aspects in mind. Hanaka et al. (2022) are somewhat closer to fairness aspects in path finding. They study shortest paths under diversity aspects, meaning that they search for multiple shortest paths that are maximally different from each other; this fits into the recent trend of finding diverse sets of solutions to optimization problems (Baste et al. 2022; Petit and Trapp 2019; Kellerhals, Renken, and Zschoche 2021).

Finally, we only mention in passing that fairness aspects are currently investigated in all kinds of optimization problems (particularly graph-based ones), including topics such as graph-based data clustering (Ahmadian et al. 2020a,b; Friggstad and Mousavi 2021; Froese, Kellerhals, and Niedermeier 2022), influence maximization (Khajehnejad et al. 2020), matching (Chierichetti et al. 2019), and graph mining (Dong et al. 2022; Kang and Tong 2021).

Fairness measures. Our fairness concept—ensuring that the number of occurrences of each color in the solution is within a given range—is closely related to that introduced by Celis, Straszak, and Vishnoi (2018) to find fair rankings. When considering paths, our constraints generalize finding rainbow and tropical graphs. It also generalizes upon many recently introduced fairness variants. We list three examples.

Max-min fairness. Defined as the difference or quotient between the number of occurrences of the most frequent and the least frequent color in the solution, this concept generalizes our balance-fairness (for which the difference is 0 and the quotient is 1). This variant has been used in many works, including the seminal work on fair clustering by Chierichetti et al. (2017). We can find solutions with a max-min fairness of k by guessing[1] the number α of occurrences of the least frequent colors. All lower bounds are then set to α and all upper bounds are set to $\alpha + k$ or αk.

Proportional fairness. The goal of this variant is to make each color appear roughly with the same frequency in the solution as in the input. This is a standard axiom in the area of fair division and was used in graph-based clustering (Froese, Kellerhals, and Niedermeier 2022) and principal component analysis (Samadi et al. 2018) to ensure a balanced error or distortion among the colors. We can model this fairness variant in the same way as max-min fairness.

Margin-of-victory fairness. Herein, one minimizes the difference in occurrences between the first and second most frequent color in the solution. This more relaxed fairness notion prevents a color from becoming a dominating majority in the solution. It was incorporated into the problem of finding fair many-to-one matchings (Stoica et al. 2020; Boehmer and Koana 2022). To model this variant, we guess the two most frequent colors as well as their number of occurrences. The bounds for each color can then be trivially obtained.

Preliminaries

We denote by $\mathbb{N}$ the set of all positive integers and by $\mathbb{N}_0$ the set of all non-negative integers. For an integer $n \in \mathbb{N}$, let $[n] \coloneqq \{1, 2, \ldots, n\}$.

Graphs. We use standard graph-theoretic terminology. All graphs are directed if not explicitly stated otherwise. For a directed graph $G = (V, A)$, we set $n \coloneqq |V|$ and $m \coloneqq |A|$. For a vertex $v \in V$, we denote by $N^{\text{in}}(v)$ the set of all vertices u such that $(u, v) \in A$. A *path P on ℓ vertices* is a graph with vertex set $\{v_1, v_2, \ldots, v_\ell\}$ and arc set $\{(v_i, v_{i+1}) \mid i \in [\ell-1]\}$. The vertices v_1 and v_ℓ are called *endpoints*. We denote by $V(P)$ the set $\{v_1, v_2, \ldots, v_\ell\}$ of

[1]Whenever we pretend to guess something, we iterate over all possibilities and consider the correct iteration.

vertices in P. Let $G = (V, A)$ be a graph with two vertices s and t and $w\colon A \to \mathbb{N}$ be an arc-length function. An s-t-path P is a subgraph of G which is a path and whose endpoints are s and t. The *length* $w(P)$ of the path is the sum of its arc lengths. We denote by $\mathrm{dist}_G(s,t)$ the length of a shortest s-t-path in G. Whenever clear from context, we may drop the subscript G. If not stated otherwise, we assume arc lengths to be positive.

For a graph $G = (V, A)$, a vertex coloring $\chi\colon V \to [c]$, and a color $i \in [c]$, we denote by $\chi^i := \{v \in V \mid \chi(v) = i\}$ the set of vertices of color i. For a subgraph H of G and a color i, we denote by χ_H the coloring χ restricted to the vertices of H, and by χ^i_H the set of vertices of color i in H.

Matroids. A pair $M = (U, \mathcal{I})$, where U is called *ground set* and $\mathcal{I}$ is a family of subsets (called *independent sets*) of U, is a *matroid* if (i) $\emptyset \in \mathcal{I}$, (ii) if $A' \subseteq A$ and $A \in \mathcal{I}$, then $A' \in \mathcal{I}$ (hereditary property), and (iii) if $A, B \in \mathcal{I}$ and $|A| < |B|$, then there is an $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$ (exchange property). An inclusion-wise maximal independent set is a *basis* of M. It follows from the exchange property that all bases of M have the same size. This size is called the *rank* of M. Let A be a matrix over a finite field $\mathbb{F}$, and let U be the set of columns of A. We associate a matroid $M = (U, \mathcal{I})$ with A as follows. A set $X \subseteq U$ is independent (i.e., $X \in \mathcal{I}$) if the columns in X are linearly independent over $\mathbb{F}$. We say that the matroid is *linear* and that A represents M.

Gammoids are a family of matroids defined as follows. Given a directed graph $G = (V, A)$ with vertex subsets $S, T \subseteq V$, we say that $X \subseteq T$ is *linked* to S if there exist $|X|$ vertex-disjoint paths (possibly of length 0) going from S to X. The *gammoid* $M = (T, \mathcal{I})$ corresponding to G, S has $\mathcal{I} = \{X \subseteq T \mid X \text{ is linked to } S\}$ as its family of independent sets.

Theorem 1 ((Kratsch and Wahlström 2014))**.** *Let $\varepsilon > 0$, let $G = (V, A)$ be a directed graph, let $S, T \subseteq V$, and let $M = (T, \mathcal{I})$ be the gammoid corresponding to G, S, and T. Then, one can compute in polynomial time an $|S| \times |T|$ matrix A over the rationals which represents a matroid $\widetilde{M} = (T, \widetilde{\mathcal{I}})$ such that for any $X \subseteq T$, we have $X \notin \widetilde{\mathcal{I}}$ whenever $X \notin \mathcal{I}$ and $\Pr[X \in \widetilde{\mathcal{I}}] \geq 1 - \varepsilon$ whenever $X \in \mathcal{I}$. Moreover, the entries in A are of bit-length $\mathcal{O}(\min\{|T|, |S| \log|T|\} + \log(1/\varepsilon) + \log|V|)$.*

Parameterized complexity. A parameterized problem is *fixed-parameter tractable* if there exists an algorithm solving any instance (I, ρ) (I is the input instance and ρ is some parameter) in $f(\rho) \cdot |I|^{\mathcal{O}(1)}$ time, where f is a (computable) function solely depending on ρ. To show that a parameterized problem L' is presumably not fixed-parameter tractable, one may use a *parameterized reduction* from a W[1]-hard problem to L. A parameterized reduction from a parameterized problem L to another parameterized problem L' is a function satisfying the following. There are two computable functions f and g, such that given an instance (I, ρ) of L, the reduction computes in $f(\rho) \cdot |I|^{\mathcal{O}(1)}$ time an instance (I', ρ') of L' such that $\rho' \leq g(\rho)$ and (I, ρ) is a yes-instance of L if and only if (I', ρ') is a yes-instance of L'.

The Parameter Number of Colors

In this section, we study the computational complexity of our two problems parameterized by the number of colors. Recall that we assume that all arc lengths are positive. We justify this assumption by showing that, if arc lengths may be zero, BALANCE-FAIR SHORTEST PATH with two colors becomes NP-hard.

Observation 2. BALANCE-FAIR SHORTEST PATH *is* NP-*hard for two colors when zero-length arcs are allowed.*

Proof. We reduce from the well-known problem DIRECTED HAMILTONIAN s-t-PATH: where one is asked whether a given a directed graph with two vertices s and t, contains an s-t-path visiting all vertices. The problem is known to be NP-hard (Garey and Johnson 1979). Given an instance $I = (G = (V, A), s, t)$, we construct an equivalent instance $I' = (G' = (V', A'), \chi, w, s', t')$ of BALANCE-FAIR SHORTEST PATH as follows. The graph G' consists of G plus a $|V|$-vertex path Q attached to t. The endpoint of Q is t'. We identify s' with s. All vertices except for those in Q are colored with the first color. All arcs in G' have length zero. Now any s'-t'-path contains all $|V|$ vertices in Q, which have the second color. Hence, it is balance-fair if and only if it contains an s-t-subpath that visits all vertices in V. ☐

Next, we look at the more general SHORT PATH WITH LOWER AND UPPER BOUNDS. Note that the problem coincides with DIRECTED HAMILTONIAN s-t-PATH when there is one color, the arcs have unit length, and the lower bound is equal to the number of vertices.

Observation 3. SHORT PATH WITH LOWER AND UPPER BOUNDS *is* NP-*hard even with unit arc lengths and only one vertex color.*

The reduction in Observation 3 makes use of the fact that one may take detours to satisfy the lower bounds. Hence, it does not work for the special case BALANCE-FAIR SHORTEST PATH or if we require that $\ell = \mathrm{dist}(s, t)$. When enforcing the solution to be a shortest s-t-path, we can show the following.

Theorem 4. BALANCE-FAIR SHORTEST PATH *is solvable in $\mathcal{O}((2n+1)^{c-1} \cdot m)$ time, where c is the number of colors.*

Proof. Let $(G = (V, E), \chi, w, s, t)$ be an instance of BALANCE-FAIR SHORTEST PATH with c colors. We devise a dynamic program with a Boolean table $T\colon V \times [-n, n]^{c-1} \to \{0, 1\}$ storing for each vertex v and each tuple $(x_1, x_2, \ldots, x_{c-1})$ whether there is a shortest s-v-path in G in which the difference between the number of vertices of colors c and i is exactly x_i for all $i \in [c-1]$. We say that such paths *respect* the tuple $(x_1, x_2, \ldots, x_{c-1})$. The table is computed for all vertices in order of their distances from s. Note that if all arc lengths are positive, then for each arc (u, v) in any shortest s-t-path P, the vertex u is always closer to s than v and $\mathrm{dist}(s, v) = \mathrm{dist}(s, u) + w((u, v))$ holds. Any arc not satisfying this equality cannot be part of a shortest s-t-path and can therefore be deleted. Denote by

$$A' := \{(u,v) \in A \mid \mathrm{dist}_G(s,v) = \mathrm{dist}_G(s,u) + w((u,v))\}$$

the set of remaining arcs. To compute an entry $T[v, x_1, x_2, \ldots, x_{c-1}]$, we distinguish whether the color i of v is c or not. If $i \neq c$, then we iterate over all incoming arcs $(u,v) \in A'$ of v and compute

$$T[v, x_1, x_2, \ldots, x_{c-1}] = \bigvee_{(u,v)\in A'} T[u, x'_1, x'_2, \ldots, x'_{c-1}],$$

wherein $x'_i = x_i - 1$ and $x'_j = x_j$ for all $j \neq i$. That is, $T[v, x_1, x_2, \ldots, x_{c-1}]$ is set to true if and only if there is a vertex $u \in N^{\mathrm{in}}(v)$ for which $T[u, x'_1, x'_2, \ldots, x'_{c-1}]$ is true. If $T[v, x_1, \ldots, x_{c-1}]$ is set to true, then there is a shortest s-u-path respecting $(x'_1, x'_2, \ldots, x'_{c-1})$. Appending v to this path results in a path respecting $(x_1, x_2, \ldots, x_{c-1})$ since $x_i = x'_i + 1$. By the definition of A', this path is also a shortest s-v-path. In the other direction, assume that there is a shortest s-v-path P respecting $(x_1, x_2, \ldots, x_{c-1})$. Consider the penultimate vertex u in P and the subpath P' from s to u. Note that since P and P' only differ in v, it holds that P' is a shortest s-u-path respecting $(x'_1, x'_2, \ldots, x'_{c-1})$, where $x'_i = x_i - 1$ and $x'_j = x_j$ for all colors $j \neq i$. Thus, $T[u, x'_1, x'_2, \ldots, x'_{c-1}]$ is set to true and thus by construction also $T[v, x_1, x_2, \ldots, x_{c-1}]$.

If v has color c, then we compute

$$T[v, x_1, x_2, \ldots, x_{c-1}] = \bigvee_{(u,v)\in A'} T[u, x'_1, x'_2, \ldots, x'_{c-1}],$$

where $x'_j = x_j - 1$ for all $j \in [c-1]$. If such an arc (u,v) exists, then there is a shortest s-u-path respecting $(x'_1, x'_2, \ldots, x'_{c-1})$. Appending v to this path results in a path that has one additional vertex of color c and hence this path respects the tuple $(x_1, x_2, \ldots, x_{c-1})$. Again, this is also a shortest s-v-path and thus $T[v, x_1, x_2, \ldots, x_{c-1}]$ is correctly computed in this case. For the other direction, if a shortest s-v-path P respecting $(x_1, x_2, \ldots, x_{c-1})$ exists, then the subpath P' from s to the penultimate vertex u in P shows that $T[v, x_1, x_2, \ldots, x_{c-1}]$ is correctly set to true in this case.

Once the whole table is computed, we can check whether there is a balance-fair shortest s-t-path in G by checking whether $T[t, 0, 0, \ldots, 0]$ is set to true. If so, then there is a balance-fair shortest s-t-path in G since the path respecting the tuple $(0, 0, \ldots, 0)$ contains the same number of occurrences of each color. If not, then there is no balance-fair shortest s-t-path in G.

Lastly, we analyze the running time. Observe that there are $(2n+1)^{c-1}$ table entries for each vertex v and computing one table entry takes $\mathcal{O}(|N^{\mathrm{in}}(v)|)$ time. Thus, the overall running time is in $\mathcal{O}(m \cdot (2n+1)^{c-1})$. □

A very simliar dynamic program to the one in the proof of Theorem 4 can also be used to solve SHORT PATH WITH LOWER AND UPPER BOUNDS in $O(m \cdot n^c)$ time if we require $\ell = \mathrm{dist}(s,t)$. Hence, Theorem 4 also holds if one asks for a *shortest* path that respects given lower and upper bounds per color. Thus, when parameterized by the number c of colors, SHORT PATH WITH LOWER AND UPPER BOUNDS is computationally harder than BALANCE-FAIR SHORTEST PATH only due to not requiring a solution to be a shortest path.

We next show that there is little hope for a significantly better algorithm for BALANCE-FAIR SHORTEST PATH parameterized by the number c of colors. We prove that BALANCE-FAIR SHORTEST PATH is W[1]-hard with respect to c and cannot be solved in $n^{o(c/\log c)}$ time unless the Exponential Time Hypothesis (ETH) fails. The ETH states that 3-SAT cannot be solved in subexponential time (Impagliazzo and Paturi 1999). Both results are shown via reductions from MULTICOLORED CLIQUE: Given a k-partite undirected graph $G = (V, E)$ with partitions $V_1, V_2, \ldots, V_k$, does G contain a clique on k vertices? MULTICOLORED CLIQUE is known to be W[1]-complete (Pietrzak 2003).

Theorem 5 (★). BALANCE-FAIR SHORTEST PATH *is* W[1]*-hard when parameterized by the number c of colors.*

Proof. We derive a parameterized reduction from MULTICOLORED CLIQUE parameterized by solution size k. To this end, let $G = (V, E)$ be the input graph of our instance of MULTICOLORED CLIQUE with partitions $V_1, V_2, \ldots, V_k$. For the sake of simplicity, we will assume that each V_i has the same number $\eta := |V|/k$ of vertices (this is no restriction since we can simply add isolated vertices) and that $V_i = \{v_1^i, v_2^i, \ldots, v_\eta^i\}$.

In the following, we will construct a graph H with two vertices s and t that contains a balance-fair shortest s-t-path if and only if G contains a clique of size k. Let us first give an intuitive description of the different pieces. The graph H will be made mostly from two parts: a vertex-selection gadget for each partition V_i and an edge-verification gadget for each pair $V_i \neq V_j$ of partitions. The former (broadly speaking) decides for each $i \in [k]$ which vertex of V_i is supposed to be in the clique. The latter verifies that there is an edge between the two chosen vertices of the two respective partitions. We need $2k(k-1)+1$ colors: two colors $r_{i,j}$ and $q_{i,j}$ for each $i \in [k]$ and each $j \in [k] \setminus \{i\}$ and a special color p.

We start by introducing $K := k + \binom{k}{2} + 1$ vertices $u_1, u_2, \ldots, u_K$ of color p. The vertex-selection gadget for each partition V_i consists of η vertex-disjoint paths, the endpoints of which are adjacent to u_i and u_{i+1}, respectively. Each of these paths represents one vertex v_a^i and contains a vertices of color $r_{i,j}$ and $\eta - a$ vertices of color $q_{i,j}$ for each $j \in [k] \setminus \{i\}$. The edge-verification gadget for each pair of partitions $V_i \neq V_j$ also consists of vertex-disjoint paths, the endpoints of which are adjacent to two vertices u_d and u_{d+1} such that each gadget uses a different "slot" d. Each of these vertex-disjoint paths represents one edge $\{v_a^i, v_b^j\}$ and contains $\eta - a$ vertices of color $r_{i,j}$, a vertices of color $q_{i,j}$, $\eta - b$ vertices of color $r_{j,i}$, and b vertices of color $q_{j,i}$. To complete the construction, we add $2\eta - K$ vertices of color p if $2\eta \geq K$ and $K - 2\eta$ vertices of each color $r_{i,j}$ and $q_{i,j}$ for each $i \neq j \in [k]$ if $2\eta < K$. All of these new vertices form a directed path. We add an arc from u_K to the start of this path and call the last vertex t. Finally, we rename u_1 to s and set all arc lengths to one.

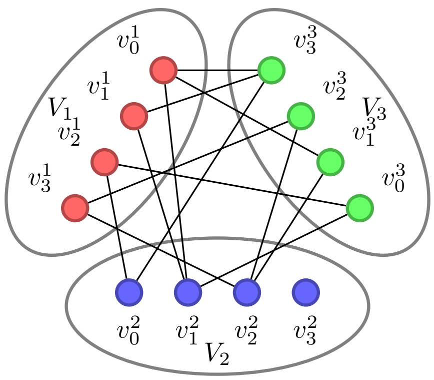

Figure 2: An example instance of MULTICOLORED CLIQUE with $k = 3$ colors and $\eta = 4$ vertices of each color.

It is easy to see that the reduction takes polynomial time. We defer the proof of the correctness to the full version of this paper. □

We continue with the lower bound based on the Exponential Time Hypothesis (ETH). If the hypothesis is true, then MULTICOLORED CLIQUE parameterized by solution size k cannot be solved in $f(k) \cdot n^{o(k)}$ time for any computable function f (Lokshtanov, Marx, and Saurabh 2018).

Theorem 6. BALANCE-FAIR SHORTEST PATH *parameterized by the number c of colors cannot be solved in $f(c) \cdot n^{o(c/\log c)}$ time for any computable function f unless the ETH breaks.*

Proof. We reduce from MULTICOLORED CLIQUE parameterized by solution size k. To this end, let $G = (V, E)$ be the input graph of our instance of MULTICOLORED CLIQUE with partitions $V_1, V_2, \ldots, V_k$. For the sake of simplicity, we will again assume that each V_i has the same number $\eta := n/k$ of vertices. Let $V_i = \{v_0^i, v_1^i, \ldots, v_{\eta-1}^i\}$. Figure 2 gives an example used throughout this proof.

In the following, we construct a graph H with two vertices s and t that contains a balance-fair shortest s-t-path if and only if G contains a clique of size k. Again, H contains a vertex-selection gadget for each partition V_i and an edge-verification gadget for each pair $V_i \neq V_j$ of partitions. Our reduction will use the colors r_i and q_i for each $i \in [k]$ and a special color p. The main idea behind the reduction is to represent each vertex v_a^i with $k^{\mathcal{O}(\log n)}$ vertices of color r_i and/or q_i using a binary encoding of a. In the vertex-selection gadgets, we will represent vertices of G by paths of vertices such that if this path is part of the final balance-fair path P, then P contains exactly $k - 1$ paths in edge-verification gadgets that correspond to edges with v_a as an endpoint.

As in the proof of Theorem 5, we place $K := k + \binom{k}{2} + 1$ vertices $u_1, u_2, \ldots, u_K$ of color p on a line and place the different gadgets in between. For each partition V_i, we create a vertex-selection gadget as follows (see Figure 3 for an illustration). The gadget consists of η vertex-disjoint paths from u_i to u_{i+1}. Each of these paths represents one vertex v_j^i

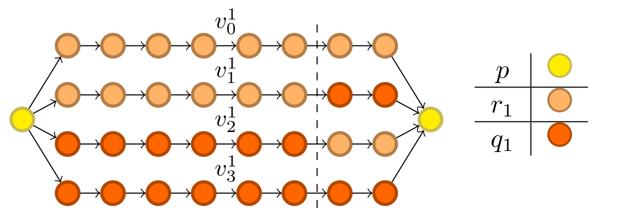

Figure 3: The vertex-selection gadget for V_1 in Figure 2 and a legend providing the names of the colors. Each path represents a vertex in V_1 and the respective vertex is listed above each path. Vertices right of the dashed line belong to the first layer while vertices to the left belong to the second layer.

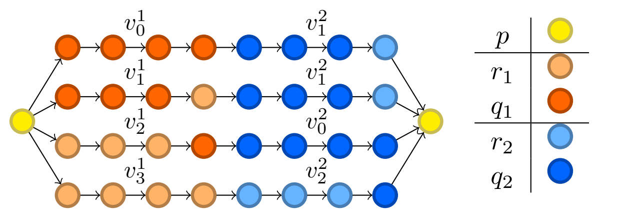

Figure 4: The edge-verification gadget for V_1 and V_2 in Figure 2 and a legend providing the names for each color. Each path represents an edge and both incident vertices are listed above the respective path.

and is constructed using the binary encoding of j. To this end, let $\tau := \lceil \log_2 \eta \rceil$ be the number of digits required to encode η. For each $\ell \in [\tau]$ let d_ℓ be the ℓ-th least significant bit in the binary encoding of j. Then, we add $(k-1) \cdot k^{\ell-1}$ vertices and color them with r_i if $d_\ell = 0$ and with q_i if $d_\ell = 1$. We call the set of vertices added for a specific ℓ the *ℓ-th level* of this gadget. Denote by $x := (k-1) \cdot \sum_{\ell=1}^{\tau} k^{\ell-1} = k^\tau - 1$ the number of vertices in any of the above vertex-disjoint paths.

The edge-verification gadget for a pair V_i, V_j of partitions is again a collection of vertex-disjoint parallel paths whose endpoints are adjacent to two vertices of color p (see Figure 4 for an illustration). There is one path for each edge $\{v_a^i, v_b^j\}$ in G. For each $\ell \in [\tau]$, let d_ℓ be the ℓ-th least significant bit in the binary encoding of a and let d'_ℓ be the ℓ-th least significant bit in the binary encoding of b. We then add $k^{\ell-1}$ vertices and color them with q_i if $d_\ell = 0$ and with color r_i if $d_\ell = 1$. Analogously, we add $k^{\ell-1}$ vertices and color them with q_j if $d'_\ell = 0$ and with r_j if $d'_\ell = 1$. We again call the set of vertices added for a specific ℓ the ℓ-th level of this gadget and denote by $y := 2 \cdot \sum_{\ell=1}^{\tau} k^{\ell-1}$ the number of vertices in any of the above vertex-disjoint paths.

Observe that the number of vertices not of color p in each shortest path through all the different gadgets is exactly $k \cdot x + \binom{k}{2} y$, which is equal to

$$k(k-1)\sum_{\ell=1}^{\tau} k^{\ell-1} + 2\binom{k}{2}\sum_{\ell=1}^{\tau} k^{\ell-1} = 2k(k-1)\sum_{\ell=1}^{\tau} k^{\ell-1}.$$

Hence, in order for such a path to contain the same amount of vertices of each color, each of the $2k$ colors different from p have to appear exactly $(k-1) \cdot \sum_{\ell=1}^{\tau} k^{\ell-1}$ times. Moreover, every path passing through all gadgets contains exactly K vertices of color p. To complete the construction, we add a path on $(k-1) \cdot \sum_{\ell=1}^{\tau} k^{\ell-1} - K$ vertices of color p to H, call the first vertex in it s, add an arc from its last vertex to u_1, and rename the vertex u_K to t. Observe that $(k-1) \cdot \sum_{\ell=1}^{\tau} k^{\ell-1} - K \geq 0$ for all $\tau \geq 2$ and $k \geq 3$.

It remains to show that the original instance is a yes-instance if and only if the constructed instance is a yes-instance and that a running time in $f(k) \cdot n^{o(k/\log k)}$ for some computable function f would refute the ETH. The main idea in there is to analyze the layers separately and for each layer analyze the number of vertices of colors r_i and q_i in the respective layer in all (vertex-selection and edge-verification) gadgets combined.

Claim 7 (★)**.** *The original instance is a* yes*-instance if and only if the constructed instance is a* yes*-instance.*

In order to show the running-time lower bound, assume that there are computable functions f and g with $g(x) \in o(x/\log x)$ such that BALANCE-FAIR SHORTEST PATH parameterized by the number c of colors can be solved in $f(c) \cdot n^{g(c)}$ time. Let $f'(x) := f(2x+1)$, $g^*(x) := g(x) \cdot \log(x)$, and $g'(x) := g^*(2x+1)$. As $g^*(x) \in o(x)$ we have $g'(x) \in o(x)$. It remains to show that MULTICOLORED CLIQUE can be solved in $\mathcal{O}(f'(k) \cdot n^{g'(k)})$ time. Since

$$\sum_{\ell=1}^{\tau} k^{\ell-1} = \sum_{\ell=1}^{\lceil \log \eta \rceil} k^{\ell-1} = \sum_{\ell=0}^{\lfloor \log \eta \rfloor} k^{\ell} \leq k^{1+\log \eta},$$

the number of vertices in the constructed instance as well as the running time for the reduction are in $k^{\mathcal{O}(\log n)} = 2^{\mathcal{O}(\log n \cdot \log k)} = n^{\mathcal{O}(\log k)}$. Moreover, observe that $c = 2k+1$. Thus, first computing the described reduction and then solving the resulting instance of BALANCE-FAIR SHORTEST PATH in $f(c) \cdot n^{g(c)}$ time results in an overall running time of $n^{\mathcal{O}(\log k)} + f(2k+1) \cdot n^{g(2k+1)}$. □

We remark that there is still a gap between the running time upper bound of $n^{\mathcal{O}(c)}$ (Theorem 4) and the lower bound of $n^{o(c/\log c)}$ (Theorem 6).

The Parameter Path Length

This section is devoted to proving that SHORT PATH WITH LOWER AND UPPER BOUNDS is fixed-parameter tractable with respect to the length ℓ of the path. More specifically, the running time of our algorithm depends on the number k of vertices that are visited by the sought solution path. Recall that we assume that each edge length is at least one; thus we have $k \leq \ell+1$. This is a reasonable assumption as otherwise the problem is NP-hard even if $\ell = 0$ (Observation 2).

Throughout this section, we will make use of the fact that we can model the balance constraints as a matroid. Then, making use of the representative-families framework by Fomin et al. (2016), we obtain the algorithm.

We model our lower-bound and upper-bound constraints as a matroid $M = (V, \mathcal{I})$. Its bases are exactly those sets $B \subseteq V$ with $|B| = k$ and $\alpha_i \leq |B \cap \chi^i| \leq \beta_i$. Thus its family of independent sets is

$$\mathcal{I} = \{X \subseteq V \mid |X \cap \chi^i| \leq \beta_i \text{ and } |X| + g_X \leq k,\ i \in [c]\}.$$

Here, $g_X = \sum_{i=1}^{c} \max\{0, \alpha_i - |X \cap \chi^i|\}$, and $|X| + g_X \leq k$ ensures that there is still enough "room" to add vertices of colors whose lower bounds are not yet met.

By definition of M, any path P on exactly k vertices fulfills the lower and upper bounds if and only if $V(P)$ is a basis of M. We call a vertex set $X \subseteq V$ M-*extendable*[2] if X is independent in M, that is, $X \in \mathcal{I}$. Similarly, a path P is M-extendable if $V(P) \in \mathcal{I}$.

Define for each $v \in V$ and $p \in [k]$ the family

$$\mathcal{P}_v^p := \big\{X \subseteq V \;\big|\; |X| = p,\ X \in \mathcal{I}, \text{ and there is an } s\text{-}v\text{-path } P \text{ with } V(P) = X\big\}.$$

We now may find an s-t-path fulfilling the lower and upper bounds with a simple Dijkstra-like dynamic programming approach. This would not yield the promised running time, as each set $\mathcal{P}_v^p$ contains up to $\binom{n}{p}$ subsets. The crucial observation here is that $\mathcal{P}_v^p$ stores *too much*: Consider a set $Y \subseteq V$. If there is a set $X \subseteq V$ with $X \cap Y = \emptyset$ such that $X \cup Y$ is a basis of M, then we say that Y *extends* X, and that X is *extendable by* Y. Now, for each set Y, we only need to remember *one* set X that is extendable by Y. Hence, if Y extends multiple sets in our family, we only need to remember one of them as a *representative* for the remaining sets. We consider a weighted variant of this concept, defined as follows.

Definition 8 (min-q-representative family)**.** Given a matroid $M = (U, \mathcal{I})$, a family $\mathcal{S}$ of subsets of U of size p, and a weight function $\rho\colon \mathcal{S} \to \mathbb{N}$, a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ is a *min-q-representative* for $\mathcal{S}$ ($\widehat{\mathcal{S}} \subseteq_{\text{rep}}^{q} \mathcal{S}$) if the following holds: for every set $Y \subseteq E$ of size at most q, if there is a set $X \in \mathcal{S}$ disjoint from Y with $X \cup Y \in \mathcal{I}$, then there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ disjoint from Y with (1) $\widehat{X} \cup Y \in \mathcal{I}$ and (2) $\rho(\widehat{X}) \leq \rho(X)$.

We remark that representative families are transitive: If $\mathcal{A} \subseteq_{\text{rep}}^{q} \mathcal{B}$ and $\mathcal{B} \subseteq_{\text{rep}}^{q} \mathcal{C}$, then $\mathcal{A} \subseteq_{\text{rep}}^{q} \mathcal{C}$. Apart from proving their transitivity, Fomin et al. (2016) presented an efficient algorithm to compute such representative families. We denote by $\omega < 2.373$ the matrix multiplication constant (Alman and Vassilevska Williams 2021).

Theorem 9 (Fomin et al. (2016))**.** *Let $M = (U, \mathcal{I})$ be a linear matroid of rank $p+q = k$ given together with its representation matrix A over a field $\mathbb{F}$. Let $\mathcal{A} = \{A_1, \ldots, A_t\}$ be a family of independent sets of size p and let $w\colon \mathcal{A} \to \mathbb{N}_0$ be a weight function. Then, a min-q-representative family $\widehat{\mathcal{A}}$ for $\mathcal{A}$ with at most $\binom{p+q}{p}$ sets can be found in $\mathcal{O}\big(\binom{p+q}{p} tp^{\omega} + t\binom{p+q}{q}^{\omega-1}\big)$ operations over $\mathbb{F}$.*

To be able to make use of this theorem with our matroid M, we prove the following.

Lemma 10 (★)**.** *M is a gammoid of rank k.*

We obtain a linear representation in polynomial time from Theorem 1, which may be errorneous with low probability. Indeed, the representation is actually the representation of *another* matroid $\widetilde{M} = (V, \widetilde{\mathcal{I}})$ such that for each $X \subseteq V$ we

[2] We drop the prefix if there is no ambiguity.

have $X \notin \widetilde{\mathcal{I}}$ if $X \notin \mathcal{I}$ and we have $\Pr[X \in \widetilde{\mathcal{I}}] \geq (1-\varepsilon)$ if $X \in \mathcal{I}$. As we will compute min-q-representatives corresponding to $\widetilde{M}$, it is convenient to introduce a corresponding adaptation for $\mathcal{P}_v^p$. For each $v \in V$ and $p \in [k]$, let

$$\widetilde{\mathcal{P}}_v^p \coloneqq \big\{X \subseteq V \;\big|\; |X| = p,\, X \in \widetilde{\mathcal{I}} \text{ and there is an } s\text{-}v\text{-path } P \text{ with } V(P) = X\big\}. \tag{1}$$

Our algorithm for Short Path with Lower and Upper Bounds then builds representative families of $\widetilde{\mathcal{P}}_v^p$ inductively. For each family $\mathcal{P}_v^p$, we also keep a weight function ρ_v^p which is intended to store for each set $X \in \mathcal{P}_v^p$ the weight of the shortest s-v-path with vertex set X.

Algorithm 1. Set $\widehat{\mathcal{P}}_s^1 \coloneqq \{\{s\}\}$, $\rho(\{s\}) \coloneqq 0$, $\widehat{\mathcal{P}}_v^1 \coloneqq \emptyset$ for each $v \in V \setminus \{s\}$, and $\rho(X) \coloneqq \infty$ for each $\{s\} \neq X \subseteq V$. For $p = 2, \ldots, k$, compute for each $v \in V$

$$\mathcal{N}_v^p = \bigcup_{(u,v)\in E} \left\{ X \cup \{v\} \;\middle|\; \begin{array}{c} v \notin X,\, X \in \widehat{\mathcal{P}}_u^{p-1}, \\ \text{and } X \cup \{v\} \in \widetilde{\mathcal{I}} \end{array} \right\}, \tag{2}$$

and for each set $X' = X \cup \{v\}$ to add to $\mathcal{N}_v^p$, set $\rho_v^p(X') \coloneqq \min\{\rho_v^p(X'), \rho_u^{p-1}(X) + w((u,v))\}$. Afterwards, compute $\widehat{\mathcal{P}}_v^p \subseteq_{\text{rep}}^{k-p} \mathcal{N}_v^p$ for each $v \in V$. Finally, return `yes` if there is an $X^* \in \widehat{\mathcal{P}}_t^k$ with $\rho_t^k(X^*) \leq \ell$.

Let us show that Algorithm 1 is correct.

Lemma 11. *For each $v \in V$ and $p \in [k]$, $\widehat{\mathcal{P}}_v^p$ contains at most $\binom{k}{p}$ sets and is a min-$(k-p)$-representative for $\widetilde{\mathcal{P}}_v^p$. For each $X \in \widehat{\mathcal{P}}_v^p$, the weight $\rho_v^p(X)$ equals the length of a shortest $\widetilde{M}$-extendable s-v-path with vertex set X.*

Proof. Our proof is by induction. The statement is trivially correct for $p = 1$. So assume that for some $p > 1$, $\widehat{\mathcal{P}}_v^{p-1} \subseteq_{\text{rep}}^{k-(p-1)} \widetilde{\mathcal{P}}_v^{p-1}$, $|\widehat{\mathcal{P}}_v^{p-1}| \leq \binom{k}{p}$, and for each $X \in \widehat{\mathcal{P}}_v^{p-1}$, the weight $\rho_v^{p-1}(X)$ equals the length of the shortest $\widetilde{M}$-extendable s-v-path on $p-1$ vertices.

Let $Y \subseteq V$ be a set of size at most $k-p$ and let $v \in V$. Suppose that there exists a set $X \in \widetilde{\mathcal{P}}_v^p$ such that Y $\widetilde{M}$-extends X. Let P be the path corresponding to X, and let $w(P) = \rho_v^p(X)$. Consider the subpath P' of P going from s to the predecessor u of v. Then P' contains $p-1$ vertices and is of length $w(P') = w(P) - w((u,v)) \geq \rho_u^{p-1}(X')$. Let $X' \coloneqq V(P)$ and let $Y' \coloneqq Y \cup \{v\}$. By the hereditary property of $\widetilde{M}$, the path P' is also $\widetilde{M}$-extendable. Further, as $X \cup Y \in \widetilde{I}$ and $X \cap Y = \emptyset$, we also have that Y' $\widetilde{M}$-extends X' (note that $X' \cap Y' = \emptyset$). Hence, by the induction hypothesis, there a set $\widehat{X}' \in \widehat{\mathcal{P}}_u^{p-1}$ with $\rho_u^{p-1}(\widehat{X}') \leq \rho_u^{p-1}(X')$ such that Y $\widetilde{M}$-extends X. Let $\widehat{P}'$ be the s-u-path of length $\rho_u^{p-1}(\widehat{X}')$ corresponding to $\widehat{X}'$ and let $\widehat{X} \coloneqq \widehat{X}' \cup \{v\}$. By Equation (2) we have $\widehat{X} \in \mathcal{N}_v^p$ and $\rho_v^p(\widehat{X}) \leq \rho_u^{p-1}(\widehat{X}') + w((u,v)) \leq \rho_v^p(X)$. Thus, $\mathcal{N}_v^p \subseteq_{\text{rep}}^{k-p} \widetilde{\mathcal{P}}_v^p$. Now, as $\widehat{\mathcal{P}}_v^p \subseteq_{\text{rep}}^{k-p} \mathcal{N}_v^p$, we obtain $\widehat{\mathcal{P}}_v^p \subseteq_{\text{rep}}^{k-p} \widetilde{\mathcal{P}}_v^p$ by transitivity of representatives (Fomin et al. (2016)). The upper bound on $|\widehat{\mathcal{P}}_v^p|$ follows from Theorem 9. □

We next analyze the running time of Algorithm 1.

Lemma 12 (★). *Algorithm 1 runs in $2^{\omega k} n^{\mathcal{O}(1)} \log(1/\varepsilon)$ time, where ε is the error probability used to compute $\widetilde{M}$.*

Our main theorem now follows from Lemmas 11 and 12.

Theorem 13 (★). *Let $\varepsilon > 0$. Then there is a randomized algorithm without false positives and success probability at least $(1-\varepsilon)$ that solves* Short Path with Lower and Upper Bounds *in $5.181^\ell \log(1/\varepsilon) n^{\mathcal{O}(1)}$ time.*

Conclusion

Our study provides insight into a very general fairness model injected into the problem of finding short paths in graphs. We indicate that even very simple cases of the considered problem turn out to be computationally hard, but, if the sought path is sufficiently short, then the problem can be efficiently solved. For parameterizing Balance-Fair Shortest Path by the number c of colors, our results leave a gap between the running time upper bound of $n^{\mathcal{O}(c)}$ and the lower bound of $n^{o(c/\log c)}$. In a follow-up work, we studied the parameterized complexity of Balance-Fair Shortest Path with respect to structural parameterizations (Bentert, Kellerhals, and Niedermeier 2022b).

Our fairness model generalizes upon many established variants. Another benefit of this fairness concept is that it can be modeled by a matroid. We make use of this to obtain an efficient (randomized) parameterized algorithm. The pervasiveness of matroids lets us believe that our fairness concept may be easily applicable also to other problems. Indeed, it can also be applied to a problem studied by Chierichetti et al. (2017), which we will call Balanced Matroid: Given a matroid $Q = (U, \mathcal{I}_Q)$ of rank ℓ for which one can decide in polynomial time whether a set X is independent in Q or not, a coloring $\chi\colon U \to [c]$, and two rationals $0 \leq \alpha \leq \beta \leq 1$, the task is to find the independent set X of largest cardinality such that for each color i, the fraction of elements of color i in X lies between α and β. They prove that if c is constant, then this problem can be solved in polynomial time (Chierichetti et al. 2019, Lem. 3). Indeed, this problem is polynomial-time solvable even if c is not a constant. To this end, one first guesses the cardinality $p \leq \ell$, then one uses the matroid M devised in the previous section, choosing $\alpha_i \coloneqq \lceil \alpha \cdot p \rceil$ and $\beta_i \coloneqq \lfloor \beta \cdot p \rfloor$ for each color i. Using the matroid intersection algorithm (Edmonds 1970), one can compute in polynomial time the maximum-cardinality set which is independent in both Q and M. If the set has cardinality p, then the guess was correct.

Corollary 14. Balanced Matroid *is polynomial-time solvable.*

There are also some variants which do not fit into our model. An intriguing model variation is to enforce fairness not only to the path as a whole, but also to each (sufficiently long) subpath of the solution. This problem seems computationally easier to tackle as the constraints are more local. A natural extension that feels harder to tackle is the case in which a vertex may hold multiple colors (e.g., a place for both biking and hiking).

Acknowledgments

This work was initiated at the 2021 Research Retreat of the Algorithmics and Computational Complexity group, Technische Universität Berlin. We thank an anonymous reviewer for an observation that lead to the ETH-based lower bound and a significant simplification of the proof of Theorem 5.

We dedicate this paper to Rolf, who tragically passed away recently. We are deeply affected by this loss of our co-author, colleague, and advisor. Rolf contributed tremendously to computer science and, in particular, to parameterized algorithmics, and should have continued doing so for a long time. The computer science community shall build on the foundations he has laid.

References

Ahmadi, S.; Tack, G.; Harabor, D.; and Kilby, P. 2021. A Fast Exact Algorithm for the Resource Constrained Shortest Path Problem. In *Proceedings of the 35th Conference on Artificial Intelligence (AAAI '21)*, 12217–12224. AAAI Press.

Ahmadian, S.; Epasto, A.; Knittel, M.; Kumar, R.; Mahdian, M.; Moseley, B.; Pham, P.; Vassilvitskii, S.; and Wang, Y. 2020a. Fair Hierarchical Clustering. In *Proceedings of the 33rd Annual Coference on Advances in Neural Information Processing Systems (NeurIPS '20)*, 21050–21060.

Ahmadian, S.; Epasto, A.; Kumar, R.; and Mahdian, M. 2020b. Fair Correlation Clustering. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS '20)*, 4195–4205. PMLR.

Alman, J.; and Vassilevska Williams, V. 2021. A Refined Laser Method and Faster Matrix Multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA '21)*, 522–539. SIAM.

Alon, N.; Yuster, R.; and Zwick, U. 1995. Color-Coding. *Journal of the ACM*, 42(4): 844–856.

Baste, J.; Fellows, M. R.; Jaffke, L.; Masarík, T.; de Oliveira Oliveira, M.; Philip, G.; and Rosamond, F. A. 2022. Diversity of solutions: An exploration through the lens of fixed-parameter tractability theory. *Artificial Intelligence*, 303: 103644.

Bentert, M.; Kellerhals, L.; and Niedermeier, R. 2022a. Fair Short Paths in Vertex-Colored Graphs. *CoRR*, abs/2203.17132.

Bentert, M.; Kellerhals, L.; and Niedermeier, R. 2022b. The Structural Complexity Landscape of Finding Balance-Fair Shortest Paths. *Theoretical Computer Science*, 933: 149–162.

Boehmer, N.; and Koana, T. 2022. The Complexity of Finding Fair Many-to-One Matchings. In *Proceedings of the 49th International Colloquium on Automata, Languages, and Programming (ICALP '22)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Celis, L. E.; Straszak, D.; and Vishnoi, N. K. 2018. Ranking with Fairness Constraints. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP '18)*, 28:1–28:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair Clustering Through Fairlets. In *Proceedings of the 30th Annual Coference on Advances in Neural Information Processing Systems (NIPS '17)*, 5029–5037. Curran Associates, Inc.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2019. Matroids, Matchings, and Fairness. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS '19)*, 2212–2220. PMLR.

Cohen, J.; Italiano, G. F.; Manoussakis, Y.; Thang, N. K.; and Pham, H. P. 2021. Tropical paths in vertex-colored graphs. *Journal of Combinatorial Optimization*, 42(3): 476–498.

Dondi, R.; and Hosseinzadeh, M. M. 2021. Finding Colorful Paths in Temporal Graphs. In *Proceedings of the 10th International Conference on Complex Networks and Their Applications (COMPLEX NETWORKS '21)*, 553–565. Springer.

Dong, Y.; Ma, J.; Chen, C.; and Li, J. 2022. Fairness in Graph Mining: A Survey. *CoRR*, abs/2204.09888.

Edmonds, J. 1970. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications (Proceedings of the 1969 Calgary Conference)*, 69–87. Gordon and Breach.

Fomin, F. V.; Lokshtanov, D.; Panolan, F.; and Saurabh, S. 2016. Efficient Computation of Representative Families with Applications in Parameterized and Exact Algorithms. *Journal of the ACM*, 63(4): 29:1–29:60.

Ford, B. T.; Aggarwal, R.; Kumar, M.; Manyam, S. G.; Casbeer, D.; and Grymin, D. 2022. Backtracking Hybrid A* for Resource Constrained Path Planning. In *Proceedings of the AIAA Scitech 2022 Forum (AIAA '22)*, 1592.

Friggstad, Z.; and Mousavi, R. 2021. Fair Correlation Clustering with Global and Local Guarantees. In *Proceedings of the 17th International Symposium on Algorithms and Data Structures (WADS '21)*, 414–427. Springer.

Froese, V.; Kellerhals, L.; and Niedermeier, R. 2022. Modification-Fair Cluster Editing. In *Proceedings of the 36th Conference on Artificial Intelligence (AAAI '22)*, 6631–6638. AAAI Press.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Hanaka, T.; Kobayashi, Y.; Kurita, K.; Lee, S. W.; and Otachi, Y. 2022. Computing Diverse Shortest Paths Efficiently: A Theoretical and Experimental Study. In *Proceedings of the 36th Conference on Artificial Intelligence (AAAI '22)*, 3758–3766. AAAI Press.

Handler, G. Y.; and Zang, I. 1980. A dual algorithm for the constrained shortest path problem. *Networks*, 10(4): 293–309.

Impagliazzo, R.; and Paturi, R. 1999. Complexity of k-SAT. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*, 237–240. IEEE Computer Society.

Irnich, S.; and Desaulniers, G. 2005. Shortest Path Problems with Resource Constraints. In *Column Generation*, 33–65. Springer US.

Kang, J.; and Tong, H. 2021. Fair Graph Mining. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21)*, 4849–4852. ACM.

Kellerhals, L.; Renken, M.; and Zschoche, P. 2021. Parameterized Algorithms for Diverse Multistage Problems. In *Proceedings of the 29th Annual European Symposium on Algorithms (ESA '21)*, 55:1–55:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Khajehnejad, M.; Rezaei, A. A.; Babaei, M.; Hoffmann, J.; Jalili, M.; and Weller, A. 2020. Adversarial Graph Embeddings for Fair Influence Maximization over Social Networks. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI '20)*, 4306–4312. ijcai.org.

Kratsch, S.; and Wahlström, M. 2014. Compression via Matroids: A Randomized Polynomial Kernel for Odd Cycle Transversal. *ACM Transactions on Algorithms*, 10(4): 20:1–20:15.

Lokshtanov, D.; Marx, D.; and Saurabh, S. 2018. Known Algorithms on Graphs of Bounded Treewidth Are Probably Optimal. *ACM Transactions on Algorithms*, 14(2): 13:1–13:30.

Petit, T.; and Trapp, A. C. 2019. Enriching Solutions to Combinatorial Problems via Solution Engineering. *Informs Journal on Computing*, 31(3): 429–444.

Pietrzak, K. 2003. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences*, 67(4): 757–771.

Pugliese, L. D. P.; and Guerriero, F. 2013. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3): 183–200.

Samadi, S.; Tantipongpipat, U.; Morgenstern, J. H.; Singh, M.; and Vempala, S. 2018. The Price of Fair PCA: One Extra dimension. In *Proceedings of the 31st Annual Coference on Advances in Neural Information Processing Systems (NeurIPS '18)*, 10999–11010. Curran Associates, Inc.

Stoica, A.; Chakraborty, A.; Dey, P.; and Gummadi, K. P. 2020. Minimizing Margin of Victory for Fair Political and Educational Districting. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '20)*, 1305–1313. International Foundation for Autonomous Agents and Multiagent Systems.