Parameterized mixed graph coloring

Citation for the original published paper (version of record):
Damaschke, P. (2019). Parameterized mixed graph coloring. Journal of Combinatorial Optimization, 38(2): 362-374. http://dx.doi.org/10.1007/s10878-019-00388-z

Journal of Combinatorial Optimization (2019) 38:362–374
https://doi.org/10.1007/s10878-019-00388-z

Parameterized Mixed Graph Coloring

Peter Damaschke[1]

Published online: 7 February 2019

Abstract
Coloring of mixed graphs that contain both directed arcs and undirected edges is relevant for scheduling of unit-length jobs with precedence constraints and conflicts. The classic GHRV theorem (attributed to Gallai, Hasse, Roy, and Vitaver) relates graph coloring to longest paths. It can be extended to mixed graphs. In the present paper we further extend the GHRV theorem to weighted mixed graphs. As a byproduct this yields a kernel and a parameterized algorithm (with the number of undirected edges as parameter) that is slightly faster than the brute-force algorithm. The parameter is natural since the directed version is polynomial whereas the undirected version is NP-complete. Furthermore we point out a new polynomial case where the edges form a clique.

Keywords Graph coloring · Mixed graph · Scheduling · Longest path · Parameterized algorithm

1 Introduction

1.1 Mixed Graph Coloring

A *mixed graph* $G = (V, A \cup E)$ consists of a set V of n vertices, a set A of k directed arcs, and a set E of m undirected edges. We always use the terms *arc* and *edge* in this sense and omit the attribute *[un]directed.* We denote edges by uv and arcs by $u \to v$, if not said otherwise. By a *path* or *cycle* we always mean a *directed* path or cycle consisting of arcs (not edges!). Moreover, it is understood that paths and cycles are simple, that is, no vertex is visited more than once. The *length* of a path is the number of its arcs. A directed graph is *acyclic* if it has no (directed) cycles. An *orientation* of a mixed graph replaces every edge with an arc that joins the same vertices, i.e., it

✉ Peter Damaschke
ptr@chalmers.se

[1] Department of Computer Science and Engineering, Chalmers University, 41296 Göteborg, Sweden

decides on some direction of every edge. Already existing arcs are not changed. An orientation is called acyclic if the resulting directed graph is acyclic.

A *coloring* of a mixed graph is a function c assigning to every vertex v a color $c(v)$ from the set of the first χ positive integers $\{1, \ldots, \chi\}$, such that $c(u) \neq c(v)$ holds for all edges $uv \in E$, and $c(u) < c(v)$ holds for all arcs $u \to v \in A$. The MIXED GRAPH COLORING problem asks for a coloring that uses the smallest possible number χ of colors, called the *chromatic number* of the mixed graph.

We remark that the following statements are equivalent for any mixed graph $G = (V, A \cup E)$: G admits a coloring; G admits an acyclic orientation; (V, A) is acyclic.

One well known motivation of MIXED GRAPH COLORING is makespan minimization in batch scheduling of unit-length jobs: some pairs of jobs have precedence constraints and some others have conflicts. For instance, requesting access to the same resource forbids the simultaneous execution of certain jobs, whereas their order is arbitrary. Jobs are modelled as vertices, time slots as colors, precedence constraints as arcs, and conflicts as edges.

1.2 Related literature and complexity status

The MIXED GRAPH COLORING problem dates back to Sotskov and Tanaev (1976) and has been introduced again in Hansen et al. (1997) in the present form where every directed edge implies a strong inequality of colors. The latter paper provides some general bounds on the chromatic number and an $O(n^2)$ time algorithm for trees. Further work has been devoted to heuristics for general mixed graphs and their applications, fast algorithms on special graph classes, as well as several hardness results (Al-Anzi et al. 2006; Furmanczyk et al. 2007; Gendron et al. 2007; Kouider et al. 2017; Ries and de Werra 2008; Sotskov et al. 2001). To mention one example, MIXED GRAPH COLORING restricted to bipartite graphs is NP-complete even with 3 colors and further restrictions, and it is solvable in polynomial time when only 2 colors are permitted (Ries 2007, 2010).

It is also known that colorings of mixed graphs are nicely related to longest paths. The classic Gallai–Hasse–Roy–Vitaver (GHRV) theorem, named after the authors that rediscovered it several times, states the following, first only for directed graphs.

Theorem 1 (GHRV theorem) *In any undirected graph, consider an acyclic orientation that minimizes the maximum length ℓ of all paths. For this value of ℓ, the chromatic number of the graph is $\chi = \ell + 1$.*

The GHRV theorem can be extended literally to mixed graphs, and still the proof is simple (Hansen et al. 1997).

Coloring a directed graph is equivalent to checking whether it is acyclic, and if so, computing a longest path. All this can be done in linear time by standard algorithms. By way of contrast, coloring undirected graphs is well known to be NP-hard.

This situation suggests a natural bridge between the two problems: MIXED GRAPH COLORING parameterized by the number m of edges. To our best knowledge, MIXED GRAPH COLORING has not been studied before from the parameterized point of view.

[The oriented graph coloring problem considered in Ganian (2009) is different from it.]

In the rest of the paper let $G = (V, A \cup E)$ be a mixed graph with acyclic directed part (V, A), as our input to the MIXED GRAPH COLORING problem. Remember that acyclicity is necessary and sufficient for a coloring to exist, and that $m = |E|$ and $k = |A|$. We can assume that G has no isolated vertices (participating neither in edges nor in arcs), as they could be trivially colored arbitrarily. Therefore $n = |V|$ is bounded by $O(k + m)$.

We presume that the reader is familiar with fixed-parameter tractability (FPT). Briefly, an FPT algorithm solves a problem with input size n and another input parameter m in $O(f(m) \cdot p(n))$ time, where f is some computable function and p is some polynomial. The time bound can be written as $O^*(f(m))$, suppressing the polynomial factor. A *kernel* is an equivalent problem instance which is computable in polynomial time and whose size is bounded by some function of m. A concept that is perhaps less well known is the notion of *annotated kernels* proposed in Abu-Khzam and Fernau (2006). On a high level, an annotated kernelization transforms any instance of the given problem to an instance (whose size is again bounded by some function of m) of a more general problem. Intuitively, the instance gets "annotations", that is, additional data items. [We refer to Abu-Khzam and Fernau (2006) for the exact technical definition.] In our case, we will go from mixed graphs to mixed graphs with arc lengths.

We will also use some standard facts that can be retrieved in any textbook on algorithms: a *topological order* of a directed graph is an order of the vertex set such that all arcs are oriented in one direction (say "from left to right"). A directed graph is acyclic if and only if it possesses a topological order. Furthermore, one can compute a topological order in linear time (in the number of arcs), as well as the lengths of longest paths from any fixed vertex to all other vertices to the right in the topological order.

Using the aforementioned GHRV theorem generalized to mixed graphs (Hansen et al. 1997), membership of MIXED GRAPH COLORING with parameter m in FPT becomes pretty obvious: we first examine each of the at most 2^m orientations of E, check whether the resulting directed graph is acyclic, and if so, compute a longest path, in $O(k + m)$ time. Finally we take an orientation that minimizes the maximum path length and color the directed graph accordingly. This costs $O((k + m)2^m)$ time in total. As a special case, undirected graphs with m edges can be colored optimally in $O(m \cdot 2^m)$ time.

1.3 Contributions

We give a structural result that can be viewed as a further extension of the classic GHRV theorem, to mixed graphs with integer arc lengths. In a sense, we compress the directed part of the mixed graph, and this yields an annotated kernel for MIXED GRAPH COLORING parameterized by m. The result as such is not difficult (and neither is the original GHRV theorem), but it provides some benefits.

As a byproduct we solve MIXED GRAPH COLORING in $O(m^2 \cdot 2^m + mk)$ time. For the problem parameterized by m (that is, for any fixed m but with arbitrarily large

k), this is faster than the brute-force bound $O((k+m)2^m)$. More important than this slight improvement in the time bound is the structural insight that, informally speaking, the entire difficulty of MIXED GRAPH COLORING is in the undirected part, and the information needed to eventually find the longest path can be squeezed into $O(m^2)$ space. Moreover, the time bound is actually only $O(m^2)$ times the number of acyclic orientations to consider, and the factor 2^m is put here only as the worst case. It remains open whether the exponential term 2^m can be improved, but any further work in this direction needs to consider the annotated kernel only.

We also devise a proper kernel of polynomial size, by constructing an equivalent instance with small arc lengths and simply replacing its weighted arcs with paths of unit-length arcs. A quote from Abu-Khzam and Fernau (2006), referring to a different problem, says: "In that particular example, we could produce a reduction rule that gets rid of the annotation ... So, this gives us an example where a kernel for an annotated version could be used to produce a kernel for the original version of the problem. It would be interesting to see if there are more examples along this venue or if the notion of an annotated kernel should be standing as a notion on its own right." So here is another natural example. Actually we are not aware of uses of annotated kernels since the publication of Abu-Khzam and Fernau (2006).

Moreover we study some special case which turn out to be solvable in polynomial time: if the edges form a clique, with the mild additional assumption that no edge is redundant, we can solve the problem in $O(mk)$ time. Although this is a polynomial bound, we make use of the annotated kernel, as the basis of a greedy algorithm.

2 Weighted Mixed Graph Coloring

Let $G = (V, A \cup E)$ be the given mixed graph with acyclic directed part (V, A). We add to V two dummy vertices s and t, called *source* and *sink*, respectively, and we add to A the arcs $s \to v$ and $v \to t$ for all vertices $v \in V$, $v \neq s, t$. In the following, $G = (V, A \cup E)$ refers to this graph extended by a source and a sink.

For formal reasons we must therefore rephrase the MIXED GRAPH COLORING problem as follows. A coloring assigns to every vertex v a color $c(v)$ from $\{0, 1, \ldots, \chi\}$, such that $c(u) \neq c(v)$ holds for all edges $uv \in E$, and $c(u) < c(v)$ holds for all arcs $u \to v \in A$ (as before), and $c(s) = 0$. The chromatic number χ is now the smallest possible value $c(t)$. Obviously this formulation is equivalent, except that the chromatic number increases by 1.

Any two distinct vertices u and v, real or dummy, are said to be in relationship $u \Rightarrow v$ if there exists a path from u to v. (Remember that a "path" always means a directed path of arcs from A.) We can interpret the ordered pairs $u \Rightarrow v$ as another type of arcs. Whenever $u \Rightarrow v$, let $l(u, v)$ denote the length of a longest path from u to v in G. If $u \Rightarrow v$ does not hold, then $l(u, v)$ remains undefined.

We can assume that no two vertices are joined by both an arc and an edge, because the edge is then *redundant* and may be deleted. Similarly, if two vertices u and v with $u \Rightarrow v$ are joined by an edge or arc, we may delete this edge or arc as redundant.

Definition 1 Given a mixed graph $G = (V, A \cup E)$ with acyclic (V, A), we define the *weighted mixed graph* $H = (W, B \cup E, l)$ as follows:

- W is the set of all vertices in V being incident to any edges of E, plus the source s and the sink t.
- For each pair u, v of distinct vertices in W, if there is a path from u to v in G, then we introduce an arc $u \Rightarrow v$ with weight $l(u, v)$, being the length of a longest-path from u to v in G. We define B as the set of all these arcs $u \Rightarrow v$.
- Every edge uv in E gets the length $l(u, v) = l(v, u) = 1$.

Note that the $\Rightarrow$ arcs in H do not form cycles.

Lemma 1 *The weighted mixed graph H in Definition 1 can be computed from the mixed graph G in $O(mk)$ time.*

Proof Since A has k arcs, they can involve at most $2k$ vertices. First we construct a topological order of (V, A) in $O(k)$ time. Next we take every fixed w and compute $l(w, v)$ for all $v \in W$ together. This can be done as follows. By, e.g., breadth-first search we collect all vertices that can be reached from w via arcs in A. Note that these are $O(k)$ vertices. In the graph induced by this vertex set we use the topological order and the standard algorithm for computing longest paths to all vertices, starting in w. Clearly it costs $O(k)$ time, and since we do this for every w in the set W of size $O(m)$, the total running time is $O(mk)$. □

Next we extend the concept of coloring to weighted mixed graphs without directed cycles. This concept will be a central tool for our algorithms.

Definition 2 A coloring of a weighted mixed graph $H = (W, B \cup E, l)$ with source s and sink t is a function c assigning to every vertex $w \in W$ a color $c(w)$, such that all colors are nonnegative integers, $c(s) = 0$, $c(u) \neq c(v)$ for all edges $uv \in E$, and $c(v) - c(u) \geq l(u, v)$ for all arcs $u \Rightarrow v \in B$.

The WEIGHTED MIXED GRAPH COLORING problem takes as input a weighted mixed graph $H = (W, B \cup E, l)$ (i.e., without directed cycles) and asks to find a coloring where $c(t)$ is minimized.

Definition 3 For a fixed coloring c, of either H or G, we denote by $O = O_c$ the orientation obtained by the following rule: every edge $uv \in E$ with $c(u) < c(v)$ becomes an arc $u \to v$. We also say that coloring c *induces* the orientation O_c. Furthermore, let H_O and G_O denote the directed graph obtained in this way from H and G, respectively. When the coloring c is understood from context, we omit the subscript c and denote the orientation simply by O. Note that H_O remains a weighted graph.

Note also that H_O contains two types of arcs: the oriented edges from E and the arcs from B. (We stress that B is unchanged, and still defined by the paths from A only.) We still distinguish the two types of arcs, i.e., the oriented edges from E and the arcs from B, by symbols $\to$ and $\Rightarrow$, respectively.

So every coloring yields an acyclic orientation as specified in Definition 3. Next we also do the converse step: we take any acyclic orientation of our weighted mixed graph and construct a function c that will turn out to be a coloring.

Definition 4 Let $G = (V, A \cup E, l)$ and $H = (W, B \cup E, l)$ be defined as above. Let O be any orientation such that H_O remains acyclic. We construct a function $c = c_O$ on V (including s, t) as follows:

- For $w \in W$, let $c(w)$ be the length of a longest path from s to w in H_O.
- For $v \notin W$, let $c(v) := \max_{w|\, w \in W, w \Rightarrow v}(c(w) + l(w, v))$.

(Similarly as before, when the orientation O is clear from context, we simply write c for c_O.)

Note that $c(v)$ is well defined for every vertex: H_O is acyclic, and since $s \Rightarrow v$ and $s \in W$, there exists some $w \in W$ satisfying $w \Rightarrow v$.

The following theorem can be viewed as an extension of the GHRV theorem beyond the mixed-graph case (Hansen et al. 1997) and might therefore be of independent interest.

Theorem 2 *For any mixed graph $G = (V, A \cup E)$ with acyclic (V, A), let $H = (W, B \cup E, l)$ be the weighted mixed graph. Let O be any orientation of E such that H_O is still acyclic. Then $c = c_O$ is a coloring of G_O.*
Morever, the coloring c_O restricted to W is minimal in the following sense: for every vertex $w \in W$, the color $c_O(w)$ is the smallest possible color of w, in any valid coloring of H_O.
Similarly, the coloring c_O is minimal in the following sense: for every vertex $v \in V$, the color $c_O(v)$ is the smallest possible color of v, in any valid coloring of G_O.

Proof Consider any arc $u \to v$ or $u \Rightarrow v$ in H_O; recall that $u, v \in W$. Let P be any path in H_O from s to u, of length $c(u)$. Such a path exists due to Definition 4. If v occured already on P, then this would create a cycle, contradicting the acyclicity of H_O. Thus, P extended by the arc from u to v is a path from s to v, of some length larger than $c(u)$. This shows that $c(u) < c(v)$ holds for every arc in H_O.

Next consider any arc $u \to v$ from A. Let P be any path from any vertex $w \in W$ to u, such that P has length $l(w, u)$, and $c(u) = c(w) + l(w, u)$. Such a path P exists: if $u \in W$, then we choose $w = u$, therefore, trivially $c(u) = c(w) + l(w, u) = c(w) + 0$ holds true, and P is chosen as the empty path. If $u \notin W$, then P exists due to Definition 4 again, since we can take some $w \in W$ with $w \Rightarrow u$ that maximizes $c(w) + l(w, u)$.

Similarly as above we conclude further: if v occured already on P, then this would create a cycle, contradicting the acyclicity of (V, A). Thus, P extended by $u \to v$ is a path from w to v of length $l(w, u) + 1$. By Definition 4 this shows $c(v) \geq c(w) + l(w, u) + 1 = c(u) + 1$. Now we have established $c(u) < c(v)$ also for every arc in A.

Altogether this shows that the condition of a coloring of a directed graph is satisfied on all arcs of G_O, that is, c is a coloring of G_O. The last two assertions about minimality are seen as follows.

Let c' be any other coloring. For every vertex $w \in W$, the length of any path from s to w in H_O is a lower bound on $c'(w)$ (since the colors on a path form an ascending sequence of integers). Hence $c(w) \leq c'(w)$, that is, all $c(w)$ are minimal colors. Similarly, for every vertex $v \notin W$, every sum $c(w) + l(w, v)$, $w \in W$, $w \Rightarrow v$, is a lower bound on $c'(v)$. Hence all colors $c(v)$ are minimal, too. □

3 A parameterized algorithm

Now we use these concepts to devise a parameterized algorithm running faster than the simple $O((k+m)2^m)$-time algorithm if the parameter m is small. We establish a reduction from Mixed Graph Coloring to Weighted Mixed Graph Coloring. Recall that G is acyclic and has a source s and a sink t.

Lemma 2 *G admits a coloring where $c(t) = \chi$ if and only if H does. Furthermore, the orientation induced by any χ-coloring of G is also induced by some χ-coloring of H, and vice versa.*

Proof Let c be any coloring of G with χ as the largest color. Let O_c be the induced orientation of G according to Definition 3. Since c is a coloring, this orientation is acyclic. In $H = (W, B \cup E, l)$ we apply the same orientation to the edges of E. It turns H into a weighted directed graph H' which is acyclic as well. In H' we define $c'(w)$ for each $w \in W$ to be the length of a longest path from s to w.

In every directed acyclic graph with a source, this longest-path function is a coloring. In particular, c' is a coloring of H', hence a coloring of H.

For every coloring of G we observe that the color of t is at least the length of a longest path from s to t (since the colors on that path form an ascending sequence of integers). Now the definitions of H and c' imply that $\chi = c(t) \geq c'(t)$. Thus we have found a coloring of H with colors no larger than χ and inducing the same orientation.

Conversely, let c' be any coloring of H with χ as the largest color. Let $O = O_{c'}$ be the induced orientation of H according to Definition 3. Again, since c' is a coloring, H_O (from Definition 3) is acyclic. From Theorem 2 we get that $c = c_O$ is a coloring of G_O, hence a coloring of G.

By the conditions of weighted coloring (in Definition 2) applied to H_O, in every coloring of H_O, the color of t is at least the length of a longest path from s to t. Since c' is a coloring of H_O, too, it follows that χ is at least the length of a longest path from s to t.

As $t \in W$, by the construction of $c = c_O$ in Definition 4, we have that $c(t)$ equals the length of a longest path from s to t. The last two statements imply $c(t) \leq \chi$. Thus we have found a coloring of G with colors no larger than χ and inducing the same orientation. □

Algorithm WMGC
Input: a mixed graph $G = (V, A \cup E)$ where (V, A) is acyclic.
Method:

- Construct the weighted mixed graph H as in Definition 1.
- For each orientation O of E, check whether H_O is acyclic, and if so, compute $c_O(t)$ in H_O.
- Fix one orientation O^* that minimizes $c_{O^*}(t)$.
- Compute the coloring c_{O^*} of G as in Definition 4.

Theorem 3 *Algorithm WMGC solves* Mixed Graph Coloring *and runs in $O(m^2 \cdot 2^m + mk)$ time, provided that $m < k$.*

Proof Due to Lemma 2, the chromatic numbers of G and H are equal. Thus we correctly obtain the chromatic number of G by computing the chromatic number of H. Since every coloring of H induces an orientation O of E, and Theorem 2 gives that $c_O(t)$ is the smallest possible color of t in H_O, we conclude that $c_{O^*}(t)$ is the chromatic number of H. By Lemma 2 we have that the same orientation O^* is also induced by some mimimal coloring of G. Finally, c_{O^*} induces O^*, and by Theorem 2, $c_{O^*}(t)$ is the smallest possible color of t in G_{O^*}. This implies that c_{O^*} is a minimal coloring of G.

Graph H is obtained in $O(mk)$ time due to Lemma 1. Since H_0 has $O(m^2)$ edges and all values $l(u, v)$ for $w, v \in W$ are precomputed, the necessary longest-path calculations need only $O(m^2)$ time for each of the 2^m orientations O. The coloring c_{O^*} of G is computed from the values $c(w)$, $w \in W$, and from the values $l(w, v)$, as specified in Definition 4. Note that $|V|$ is bounded by $O(k + m)$ which in turn is bounded by $O(k)$ (since $m < k$), and $|W|$ is bounded by $O(m)$. Hence this final step costs $O(mk)$ time. □

4 A kernel for Mixed Graph Coloring

The weighted mixed graph H from Definition 1 can be seen as an annotated kernel for Mixed Graph Coloring, with $O(m)$ vertices, $O(m)$ edges, and $O(m^2)$ weighted arcs. Remember from Lemma 1 that we have computed H in $O(mk)$ time. However, arc lengths do not appear in the original problem and are added as extra information to this kernel. The question of the existence of a "proper" polynomial kernel remains natural, too. We address it now.

Theorem 4 Mixed Graph Coloring *has a kernel with $O(m)$ edges, and $O(m^4)$ vertices and arcs, which can be constructed in $O(mk + m^4)$ time.*

Proof Given a mixed graph $G = (V, A \cup E)$ with soruce and sink, we first ignore E and define an initial coloring c_I of (V, A) by letting $c_I(v)$ be the length of a longest path from s to v, for all vertices $v \in V$.

Due to Theorem 3, there exists some orientation O of E (called O^* in Algorithm WMGC) such that c_O is a minimal coloring of G. If O were already known, we could construct c_O also in an incremental fashion as follows (not caring about efficiency). We start from c_I, insert the edges of E one by one, oriented according to O, and update the lengths and colors accordingly. Upon every insertion, the color of each vertex can increase by at most 1. It follows $c_O(v) \leq c_I(v) + m$ for each $v \in V$. Deriving this inequality was the only purpose of this procedure.

Consider the coloring c_I restricted to W. Suppose that two colors i, j with $j - i \geq m + 2$ are assigned to some vertices in W, but no vertex $w \in W$ has $i < c_I[w] < j$. We refer to such a pair of colors i, j as a *gap*.

Consider any gap i, j. We partition W into the sets W_s and W_t of vertices with colors at most i and at least j, respectively. By the above inequality, the colors in c_O are raised by at most m compared to c_I. Therefore the difference between the colors of any vertices in W_t and W_s is still at least 2 also in c_O.

Let F be the set of all arcs in H that go from W_s to W_t. Assume that F contains an arc uv of length 1. Then, a longest path from s to v in H_O cannot have uv as its last arc, because this would imply $c_O(v) = c_O(u) + 1$, contradicting the difference at least 2. Hence uv cannot appear either on a longest path from s to any other vertex. That is, we can delete the arc uv without changing c_O.

Clearly, every path from s to t contains exactly one arc from F. Thus, if we decrease the length of every arc in F by 1, then $c_O(t)$ is also decreased by exactly 1. Note that j is then also decreased by 1.

The above reasoning holds for every orientation O, whereas the gaps are solely determined by c_I.

These definitions and structural properties lead to the following Mixed Graph Coloring kernelization algorithm, followed by the correctness arguments and time analysis.

Algorithm MGCK

Input: a mixed graph $G = (V, A \cup E)$ where (V, A) is acyclic.
Method:

- Construct the weighted mixed graph H as in Definition 1.
- Compute $c_I(w)$ for all $w \in W$.
- $p := 0$
- (Gap elimination) Repeat until no gap remains:
  - Choose some gap i, j and compute W_s, W_t, and F (defined above).
  - Decrease the lengths of all arcs in F by $d := (j - i) - (m + 1)$, and delete all arcs in F that get zero or negative lengths.
  - $p := p + d$
- (Subdivision) Replace every arc, say of length q, with a path of length q, that is, subdivide the arc by $q - 1$ new internal vertices.

Let K denote the graph produced by Algorithm MGCK, and let O be any orientation of E. As seen above, every step of the gap elimination loop decreases $c_O(t)$ by the corresponding value d. Since p is the sum of all these numbers d, we have: $c(t) = \chi$ in G_O if and only if $c(t) = \chi - p$ in K with the same orientation O of E. In particular this yields: G admits a coloring where $c(t) = \chi$ if and only if K admits a coloring where $c(t) = \chi - p$.

Computing H takes $O(mk)$ time by Lemma 1. Within $O(m^2)$ time we can compute all $c_I(w)$, sort these $O(m)$ values, and mark all gaps. Since trivially only $O(m)$ gaps can exist, each having at worst $O(m^2)$ arcs in the corresponding set F, gap elimination takes $O(m^3)$ time. There remains a weighted mixed graph with reduced arc lengths: since $|W|$ is bounded by $O(m)$ and all gaps have disappered, the path lengths and colors in W are now bounded by $O(m^2)$. In particular, every arc has some length in $O(m^2)$. Since H has $O(m^2)$ weighted arcs also in total, the sum of their lengths is $O(m^4)$. Finally, subdivision takes $O(m^4)$ time. □

Some steps in the above analysis are generous, therefore it may be possible to improve the exponents in Theorem 4, but practically this is not too relevant: the main point with Theorem 4 is the mere existence of a proper kernel of polynomial size. For

algorithmic purposes it is better to work with the more natural annotated kernel H that was obtained in $O(mk)$ time.

5 Edges forming a clique

A more rewarding question than optimizing the kernel is whether the $O(2^m)$ term in Theorem 3 can be improved. We must avoid examining all 2^m orientations of E. For dense graphs this is easy to do: let $h := |W| - 2$. Since every acyclic orientation gives rise to a topological order, it suffices to consider the $h!$ permutations of $W \setminus \{s, t\}$. Note that $h! < 2^m$ for $m = \Omega(h \log h)$.

The extremal case of a dense undirected graph spanned by E is a single clique of h vertices. For this case it is natural to ask whether the $O^*(h!)$ time bound can be further reduced. Below we will give an affirmative answer.

Suppose that E forms the edge set of a clique C, and no edges are redundant (see Sect. 2). That is, for any two vertices, we suppose that $uv \in E$ implies that neither $u \Rightarrow v$ nor $v \Rightarrow u$ holds. This case is rather easy to solve; the following "moving bars' reformulation of the problem might also appear in similar form in other contexts.

Theorem 5 Mixed Graph Coloring *in the case when the edges are not redundant and form a clique can be solved in $O(mk)$ time.*

Proof Our weighted mixed graph H from Definition 1 consists of the said clique C, arcs $s \Rightarrow v$ and $v \Rightarrow t$ for all $v \in C$, with lengths $l(s, v)$ and $l(v, t)$, respectively, and an edge $s \Rightarrow t$ of length $l(s, t)$. Note that $l(s, t) \geq \max_{v \in C}(l(s, v) + l(v, t))$, and strict inequality is possible.

We can work on a horizontal line and think of each path from $v \in C$ to t as a horizontal *bar* of length $l(v, t)$ that has its left end at coordinate $l(s, v)$. (We prefer the name "bar" to "interval", in order to emphasize the analogy of a movable mechanical object.) The problem of computing the chromatic number is then equivalent to moving some bars to the right, such that the left ends of all bars become pairwise distinct integers, and the maximum coordinate of all their right ends is minimized. The chromatic number equals the maximum of these right end coordinates and of the fixed value $l(s, t)$.

Now we can solve this problem by a greedy algorithm: initially let $j := \min_{v \in C} l(s, v)$. If more than one bar has its left end at j, then we move all these bars, except one longest one, by 1 unit to the right. We set $j := j + 1$ and repeat this step until the increasing j has passed all left ends of bars.

The running time is dominated by the time $O(mk)$ for the construction of H, from Lemma 1. Note that $|C|$ is bounded by $O(\sqrt{m})$, and therefore all bars together are moved by $O(\sqrt{m}^2) = O(m)$ steps at worst. The index j is raised $O(k + m)$ times.

Since the left ends of bars to the left of j are always distinct, clearly this algorithm produces a valid solution. Below we prove its optimality by an exchange argument.

Consider the greedy solution Gr and some optimal solution Opt. Let j now denote the first position where Gr deviates from Opt. That is, bars with left ends $i < j$ are identical in both solutions, Gr has a bar b with left end j, and Opt has either a shorter bar b' or no bar with left end j.

In either case, b appears later in Opt, that is, b has its left end at some position $i > j$ in Opt. We can instead place b such that its left end becomes exactly j, and move b' (if it exists) such that its left end becomes i. This exchange operation yields another valid solution and does not move the rightmost right end of the bars further to the right. Iteration of these moves transforms Opt into Gr. □

We had assumed that E only consists of irredundant edges, and amazingly this is essential: suppose that E still forms the edge set of a clique C but may contain redundant edges, that is, $u \Rightarrow v$ may hold for some pairs of vertices $u, v \in C$. It is tempting to modify the above algorithm and keep at j some longest bar v, among all v that are not subject to constraints $u \Rightarrow v$, where u is any further bar with a left end $i \geq j$. However, then the exchange argument breaks down, since b' may be involved in precedence constraints, such that moving b' forces us to move more bars to the right. Therefore we cannot simply extend Theorem 5 to this case, and it remains open whether it is still solvable in polynomial time.

6 Further research

6.1 Jobs of different lengths

The application of Mixed Graph Coloring to scheduling is not restricted to unit-length jobs. Suppose that the jobs have integer durations, time is still divided in unit-length slots, but jobs may be interrupted and resumed, and conflicting jobs cannot share any time slots. In the case of short durations we can afford their unary encoding: every job of length ℓ is then modelled by a path of ℓ vertices, and arcs for precedence constraints only join the first and last vertices of these paths, in an obvious way. For modelling a conflict between two jobs of lengths ℓ and ℓ' we insert $\ell \cdot \ell'$ edges.

However, if interruptions are prohibited, then this transformation does not work. To see this, consider an example with five jobs, each of length 2, whose pairwise conflicts form an undirected cycle (and no precedence constraints prevail). Since the 5-cycle has chromatic number 3, the shortest schedule needs $3 \cdot 2 = 6$ slots. However, one can easily schedule these jobs with some interruptions in only 5 slots. (The more general background is that the fractional chromatic number of the 5-cycle is only 2.5.) That is, the minimal coloring cannot be turned into a schedule of the same length without interruptions. We must therefore modify the Mixed Graph Coloring problem itself.

One possible way to do this is to give each vertex a weight $\ell(v)$ and demand to assign $\ell(v)$ consecutive colors to each vertex v such that the constraints of Mixed Graph Coloring are satisfied for all assigned colors. Alternatively we may model the jobs by arcs of lengths $\ell(v)$ (and their start and end points as vertices) and the conflicts as "edges between arcs", i.e., define some hypergraph with hyperedges of size 4. The colors in these hyperedges must then fulfill some obvious inequalities.

One could also have requirements that executions of conflicting jobs must have certain minimum distances on the time axis. This gives rise to a generalized graph coloring problem in edge-weighted graphs, with arbitrary nonnegative real numbers as edge weights and as vertex "colors": the difference (or absolute value of the difference)

between the colors of any two adjacent vertices must be at least the weight of the arc (edge) joining them, and the goal is to minimize the maximum color.

Direct generalizations of our results should yield GHRV-type theorems and, based on them, FPT algorithms for such problem versions, too.

We remark that the undirected real-valued coloring problem is NP-hard already for the case of cliques where the edge weights satisfy the triangle inequality: then the maximum color is the sum of weights between the neighbored vertices in the order sorted by colors, hence we can reduce TRAVELLING SALESMAN to this problem.

6.2 Other research directions

We have shown that we can almost get rid of the directed part of the mixed graph, but perhaps the main open question is: can MIXED GRAPH COLORING be solved faster than in $O^*(2^m)$ time if the undirected edges form a sparse subgraph, or can one prove lower bounds, e.g., under the Exponential Time Hypothesis? We do not have an answer even in simple cases such as a matching.

MIXED GRAPH COLORING in the special case when the arcs from two disjoint paths can be interpreted as a job shop scheduling problem and be solved in linear time via a shortest path in a certain grid graph (Hardgrave and Nemhauser 1963; Szwarc 1968; Sotskov 1991). We are wondering about generalizations to similar cases.

Acknowledgements The author would like to thank the referees for valuable comments, especially for pointers to the scheduling literature that rubbed out a rediscovery in an earlier manuscript.

References

Abu-Khzam FN, Fernau H (2006) Kernels: annotated, proper and induced. In: Bodlaender HL, Langston MA (eds) IWPEC 2006, LNCS, vol 4169, pp 264–275

Al-Anzi FS, Sotskov YN, Allahverdi A, Andreev GV (2006) Using Mixed Graph Coloring to minimize total completion time in job shop scheduling. Appl Math Comput 182:1137–1148

Furmanczyk H, Kosowski A, Zylinski P (2007) Scheduling with precedence constraints: mixed graph coloring in series-parallel graphs. In: PPAM , LNCS, vol 4967, pp 1001–1008

Ganian R (2009) The parameterized complexity of oriented colouring. In: Hlinený P, Matyás V, Vojnar T (eds) MEMICS, OASICS, vol 13

Gendron B, Hertz A, St-Louis P (2007) On edge orienting methods for graph coloring. J Comb Optim 13:163–178

Hansen P, Kuplinsky J, de Werra D (1997) Mixed Graph Colorings. Math Methods Oper Res 45:145–160

Hardgrave WW, Nemhauser G (1963) A geometric model and graphical algorithm for a sequencing problem. Oper Res 11:889–900

Kouider A, Haddadène HA, Ourari S, Oulamara A (2017) Mixed Graph Coloring for unit-time scheduling. Int J Prod Res 55:1720–1729

Ries B (2007) Coloring some classes of mixed graphs. Discret Appl Math 155:1–6

Ries B (2010) Complexity of two coloring problems in cubic planar bipartite mixed graphs. Discret Appl Math 158:592–596

Ries B, de Werra D (2008) On two coloring problems in mixed graphs. Eur J Comb 29:712–725

Sotskov YN (1991) The complexity of shop-scheduling problems with two or three jobs. Eur J Oper Res 53:326–336

Sotskov YN, Tanaev VS (1976) Chromatic polynomial of a mixed graph. Vestsi Akademii Navuk BSSR, Seryya Fizika-Matematychnykh Navuk 6:20–23 **(in Russian)**

Sotskov YN, Dolgui A, Werner F (2001) Unit-time job-shop scheduling via Mixed Graph Coloring. Int J Math Algorithm 2:289–323

Szwarc W (1968) Solution of the Akers-Friedman scheduling problem. Oper Res 8:782–788