Available online at www.sciencedirect.com
SCIENCE DIRECT®

Discrete Applied Mathematics 127 (2003) 415–429

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Parameterized complexity of vertex colouring ☆

Leizhen Cai*

Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong, China

Received 19 April 2000; received in revised form 15 October 2001; accepted 5 November 2001

To Derek Corneil for his love of graphs

Abstract

For a family $\mathscr{F}$ of graphs and a nonnegative integer k, $\mathscr{F}+k$e and $\mathscr{F}-k$e, respectively, denote the families of graphs that can be obtained from $\mathscr{F}$ graphs by adding and deleting at most k edges, and $\mathscr{F}+k$v denotes the family of graphs that can be made into $\mathscr{F}$ graphs by deleting at most k vertices.

This paper is mainly concerned with the parameterized complexity of the vertex colouring problem on $\mathscr{F}+k$e, $\mathscr{F}-k$e and $\mathscr{F}+k$v for various families $\mathscr{F}$ of graphs. In particular, it is shown that the vertex colouring problem is fixed-parameter tractable (linear time for each fixed k) for split $+\,k$e graphs and split $-\,k$e graphs, solvable in polynomial time for each fixed k but $W[1]$-hard for split $+\,k$v graphs. Furthermore, the problem is solvable in linear time for bipartite + 1v graphs and bipartite + 2e graphs but, surprisingly, NP-complete for bipartite + 2v graphs and bipartite + 3e graphs. © 2002 Published by Elsevier Science B.V.

MSC: 05C15; 05C85; 68Q17; 68Q25; 68R10

Keywords: Vertex colouring; Fixed-parameter problem; Parameterized complexity; Split graph; Bipartite graph

1. Introduction

Many graph problems are intractable for general graphs but tractable for various special families of graphs. Let Π be an NP-hard problem and $\mathscr{F}$ be a family of graphs for which Π is solvable in polynomial time. Consider the situation that an instance G of Π is not an $\mathscr{F}$ graph, but is "close" to an $\mathscr{F}$ graph in the sense that it can be made

☆ This work was partially supported by an Earmarked Research Grant from the Research Grants Council of Hong Kong.

* Fax: +852-2603-5024.
E-mail address: lcai@cse.cuhk.edu.hk (L. Cai).

0166-218X/03/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0166-218X(02)00242-1

into an $\mathscr{F}$ graph by altering a few vertices or edges. We often need to deal with such a situation in solving practical problems because of incomplete information or errors in data. In most cases, polynomial algorithms for solving Π on $\mathscr{F}$ graphs cannot be applied directly to G and may become totally useless, and the NP-hardness of Π for general graphs does not imply the tractability of Π on G. This calls for a study of the complexity of Π for the families of graphs formed by graphs like G that are nearly $\mathscr{F}$ graphs. Surprisingly, very little attention, if any, has been paid to the complexity of graph problems for such graph families despite the enormous amount of work on their complexity for various special families of graphs.

We now discuss the situation in a formal setting. Let k be a nonnegative integer. We use $\mathscr{F}+k\mathrm{e}$ to denote the family of graphs that can be obtained from $\mathscr{F}$ graphs by adding at most k edges (but no new vertices). Similarly, we use $\mathscr{F}-k\mathrm{e}$ (respectively, $\mathscr{F}-k\mathrm{v}$) to denote the families of graphs that can be obtained from $\mathscr{F}$ graphs by deleting at most k edges (vertices), and $\mathscr{F}+k\mathrm{v}$ to denote the family of graphs that can be made into $\mathscr{F}$ graphs by deleting at most k vertices. Clearly, every graph is an $\mathscr{F}+k\mathrm{e}$ graph for some k whenever $\mathscr{F}$ contains all edgeless graphs. Similarly, every graph is an $\mathscr{F}-k\mathrm{e}$ graph for some k whenever $\mathscr{F}$ contains all complete graphs, and an $\mathscr{F}+k\mathrm{v}$ graph for some k whenever $\mathscr{F}$ contains the single-vertex graph. (However, this kind of statement does not hold for $\mathscr{F}-k\mathrm{v}$ graphs.) Therefore, for most $\mathscr{F}$, $\mathscr{F}+k\mathrm{e}$, $\mathscr{F}-k\mathrm{e}$, and $\mathscr{F}+k\mathrm{v}$ parameterize graphs with respect to $\mathscr{F}$, and will be referred to as *parameterized graph families*.

The observations in the previous paragraph imply that for most graph families $\mathscr{F}$, if k is part of input, problem Π is NP-hard for $\mathscr{F}+k\mathrm{e}$, $\mathscr{F}-k\mathrm{e}$ and $\mathscr{F}+k\mathrm{v}$. However, its complexity status becomes elusive when k is not part of input but a fixed constant, i.e., when we regard problem Π for $\mathscr{F}+k\mathrm{e}$ (similarly, for other parameterized graph families) as a fixed-parameter problem. Is Π solvable in polynomial time for each fixed k? If so, is Π fixed-parameter tractable, i.e., is it solvable in polynomial time (as a function of instance size) with the degree of the polynomial independent of k? If not, does the complexity of Π jump from polynomial to NP-hard when some k increases to $k+1$? In this paper, we try to answer these questions for the following classical Vertex Colouring problem ([GT4] in [11]):

Instance: Graph $G=(V,E)$, positive integer $t \leqslant |V|$.

Question: Is G t-colourable, i.e., is there a function $f: V \to \{1,2,\ldots,t\}$ such that for every edge uv of G, $f(u) \neq f(v)$?

The problem is NP-complete even for $t=3$ [12], but polynomial-time solvable for many special families of graphs, such as bipartite graphs, split graphs, interval graphs and partial k-trees [19].

The research in this paper is mainly inspired by the parameterized complexity theory of Downey and Fellows [9], and also motivated by the polynomial-time solvability of many NP-hard problems on partial k-trees [1–4,7]. An algorithm for a fixed-parameter problem (I,k), where I is an instance and k is the parameter, is *uniformly polynomial* if it runs in time $\mathrm{O}(f(k)|I|^c)$, where $|I|$ is the size of I, for an arbitrary function $f(k)$ and a constant c independent of k. A fixed-parameter problem is *fixed-parameter tractable* if it admits a uniformly polynomial algorithm. This notion of fixed-parameter tractability attempts to distinguish tractable and intractable fixed-parameter problems, which is very akin to the notion of polynomial algorithms in distinguishing tractable and intractable problems. Downey and Fellows also defined a W-hierarchy, which corresponds to NP-completeness, to capture intractable fixed-parameter problems: a fixed-parameter problem that is hard for any level of the hierarchy is unlikely to be fixed-parameter tractable. The reader is referred to their monograph [9] for the theory of parameterized complexity.

In this paper, we study the complexity of Vertex Colouring from the parameterized complexity point of view. Usually, we parameterize a problem by a parameter associated with solutions, such as t in Vertex Colouring. However, this approach does not make the parameterized complexity theory applicable to Vertex Colouring as the problem is NP-complete for every fixed $t \geqslant 3$. In this paper we parameterize Vertex Colouring by a parameter linked to the input graph instead, and demonstrate through interesting results that this new way of parameterizing problems adds a new dimension to the applicability of the parameterized complexity theory.

We prove that Vertex Colouring is linear-time solvable for each fixed k for split$+k$e graphs and split $-\,k$e graphs, polynomial-time solvable for each fixed k but $W[1]$-hard for split $+\,k$v graphs (Section 4). Furthermore, we show that it is linear-time solvable for bipartite $+$ 1v graphs and bipartite $+$ 2e graphs but, surprisingly, NP-complete for bipartite $+$ 2v graphs and bipartite $+$ 3e graphs (Section 5). We also give some general results regarding the colouring problem for parameterized graphs in Section 3, and discuss future research directions in Section 6. To set the stage for our discussion, we establish notation and definitions and give some elementary results in Section 2.

2. Preliminaries

All graphs in this paper are undirected simple graphs. We follow standard notations in graph theory (see [22], for instance) with the convention that m and n, respectively, denote the number of edges and number of vertices of the input graph. A graph is a *split graph* if its vertex set can be partitioned into an independent set and a clique. Split graphs are precisely the family of graphs that contain no induced subgraph isomorphic to $2K_2$, C_4 or C_5 [10].

For two nonadjacent vertices u and v of G, we use $G(u=v)$ to denote the graph obtained from G by identifying u with v, i.e., replacing vertices u and v by a new vertex and connecting all vertices adjacent to either u or v to the new vertex. For an edge e in G, $G \cdot e$ denotes the graph obtained from G by contracting edge e, i.e., deleting e and identifying its two ends.

A family $\mathscr{F}$ of graphs is *hereditary* if for every graph $G \in \mathscr{F}$, all its induced subgraphs are $\mathscr{F}$ graphs; *closed under edge contraction* if for every $G \in \mathscr{F}$ and every edge e of G, $G \cdot e \in \mathscr{F}$; and *closed under identification of nonadjacent vertices* if for every $G \in \mathscr{F}$ and every pair of nonadjacent vertices u and v of G, $G(u=v) \in \mathscr{F}$. Note that for every hereditary family $\mathscr{F}$ and every k, $\mathscr{F} - kv = \mathscr{F}$.

A *modulator* of an $\mathscr{F} + k$e graph G is a subset E_k of at most k edges in G such that $G - E_k \in \mathscr{F}$. Modulators of $\mathscr{F} - k$e and $\mathscr{F} + k$v graphs are defined similarly. It is clear that $\mathscr{F} + k\mathrm{e} \subseteq \mathscr{F} + k'\mathrm{e}$ for every $k \leqslant k'$ and $\mathscr{F} + k\mathrm{e} = (\mathscr{F} + (k-1)\mathrm{e}) + 1\mathrm{e}$ for every $k \geqslant 1$, and similar relations hold for $\mathscr{F} - k\mathrm{e}$ and $\mathscr{F} + k\mathrm{v}$. However we should note that normally $(\mathscr{F} + k\mathrm{e}) - k\mathrm{e} \neq \mathscr{F}$. The following properties of parameterized $\mathscr{F}$ graphs are useful and can be easily proved from definitions.

Lemma 2.1. *Let $\mathscr{F}$ be hereditary. Then for every k, $\mathscr{F} - k\mathrm{e}$, $\mathscr{F} + k\mathrm{e}$ and $\mathscr{F} + k\mathrm{v}$ are all hereditary, and, furthermore, $\mathscr{F} - k\mathrm{e} \subseteq \mathscr{F} + k\mathrm{v}$ and $\mathscr{F} + k\mathrm{e} \subseteq \mathscr{F} + k\mathrm{v}$.*

We note that the set inclusions in the above lemma may not hold when $\mathscr{F}$ is not hereditary. For example, let $\mathscr{C}$ be the family of all cycles. Then $\mathscr{C} - 1\mathrm{e}$ is the family of all paths and cycles, and $\mathscr{C} + 1\mathrm{e}$ is the family of all cycles with at most one chord. Neither family is a subset of $\mathscr{C} + 1\mathrm{v}$.

Lemma 2.2. *If $\mathscr{F}$ is closed under edge contraction then for every k, $\mathscr{F} - k\mathrm{e}$ is also closed under edge contraction.*

Lemma 2.3. *If $\mathscr{F}$ is closed under identification of nonadjacent vertices, then for every k, $\mathscr{F} + k\mathrm{e}$ is also closed under identification of nonadjacent vertices.*

We now turn to vertex colourings. A *t-colouring* of $G = (V, E)$ is a function $f: V \to \{1, 2, \ldots, t\}$ such that for every edge $uv \in E$, $f(u) \neq f(v)$. The *chromatic number* of G, denoted $\chi(G)$, is the least integer t for which G has a t-colouring, and a t-colouring is an *optimal colouring* if $t = \chi(G)$. The *chromatic polynomial* $\chi(G, t)$ of G is a polynomial in t whose value at a given t equals the number of t-colourings of G. The List Colouring problem is to determine, given a graph G and a list $L(v)$ of admissible colours for each vertex v of G, whether there is a colouring f of G such that $f(v) \in L(v)$ for every vertex.

The following fundamental *connection–contraction method* is a useful tool for dealing with a vertex colouring problem Π: Whenever the input graph G is not a complete graph, find two nonadjacent vertices u and v in G, construct two graphs $G_1 = G + uv$ and $G_2 = G(u = v)$, and then recursively solve Π on G_1 and G_2 and combine their solutions to get a solution for G. Note that $\chi(G) = \min\{\chi(G_1), \chi(G_2)\}$, $\chi(G, t) = \chi(G_1, t) + \chi(G_2, t)$, G is t-colourable iff at least one of G_1 and G_2 is t-colourable, and an optimal colouring of G can be obtained from an optimal colouring of G_1 or G_2. It was proved by Walsh [21] that the total number of complete graphs generated by using the contraction–connection method is at most $B(n)$, the nth Bell number which equals the number of ways to partition a set of n distinct elements into disjoint nonempty subsets. Asymptotically, $B(n)$ grows faster than c^n for any constant c but much slower than n factorial.

To end this section, we remark that if $\mathscr{F}$ has bounded treewidth, then $\mathscr{F} - k\mathrm{e}$, $\mathscr{F} + k\mathrm{e}$, and $\mathscr{F} + k\mathrm{v}$ all have bounded treewidth, and thus Vertex Colouring is fixed-parameter tractable for all of them [3]. Finally, for algorithms in the paper, we use the adjacency list representation for input graphs.

3. General graphs

In general, it appears that we need to know a modulator of the input graph in order to solve the vertex colouring problem on parameterized $\mathscr{F}$ graphs. Therefore, we will separate the issue of finding a modulator, and assume that a modulator of the input graph is also given as input when we discuss colouring problems in this section. Note that whenever $\mathscr{F}$ graphs are recognizable in polynomial time, a modulator of a parameterized $\mathscr{F}$ graph can be found in polynomial time for each fixed k by exhaustive search.

Given an efficient algorithm for solving Vertex Colouring on $\mathscr{F}$ graphs, how can it help us in solving the problem on parameterized $\mathscr{F}$ graphs? It is not clear in general; however, for some $\mathscr{F}$, it is possible to utilize the algorithm for $\mathscr{F}$ graphs to obtain an efficient colouring algorithm for parameterized $\mathscr{F}$ graphs. In this section, we present such an algorithm to optimally colour $\mathscr{F} - k\mathrm{e}$ graphs for $\mathscr{F}$ closed under edge contraction. We also solve Vertex Colouring for certain $\mathscr{F} + k\mathrm{e}$ graphs by using algorithms for computing chromatic polynomials of $\mathscr{F}$ graphs, and for certain $\mathscr{F} + k\mathrm{v}$ graphs by using algorithms for solving List Colouring on $\mathscr{F}$ graphs.

We consider $\mathscr{F} - k\mathrm{e}$ graphs first. It appears that, amongst parameterized $\mathscr{F}$ graphs, $\mathscr{F} - k\mathrm{e}$ graphs are the easiest in terms of the hardness of solving Vertex Colouring. As we will see, if $\mathscr{F}$ is closed under edge contraction, then Vertex Colouring is solvable in polynomial time for $\mathscr{F} - k\mathrm{e}$ graphs whenever it is solvable in polynomial time for $\mathscr{F}$ graphs. This is quite useful as many families of graphs are closed under edge contraction, for instance, planar graphs, chordal graphs, split graphs, interval graphs, cographs, as well as graphs that are closed under taking minors. Note that the following theorem can be also stated in terms of Vertex Colouring instead of finding an optimal colouring.

Theorem 3.1. *Let $\mathscr{F}$ be a family of graphs closed under edge contraction, and $T(m,n)$ denote the time to compute an optimal colouring of an $\mathscr{F}$ graph. Then an optimal colouring of an $\mathscr{F} - k\mathrm{e}$ graph G, given a modulator of G, can be found in time* $O(2^k \max\{T(m+k,n), m+n+k\})$.

Proof. We use the connection–contraction method. Let E_k be a modulator of G. Then $G + E_k \in \mathscr{F}$. Pick an arbitrary edge uv in E_k (note that uv is not an edge in G) and construct from G two graphs $G + uv$ and $G(u = v)$. Clearly $G + uv$ is an $\mathscr{F} - (k-1)\mathrm{e}$ graph with modulator $E_k - uv$. Furthermore, since $G(u = v)$ equals $(G + uv) \cdot uv$, $G(u = v)$ is also an $\mathscr{F} - (k-1)\mathrm{e}$ graph by Lemma 2.2 and the corresponding edges of $E_k - uv$ in $G(u = v)$ is a modulator of $G(u = v)$. Therefore, the problem of finding an optimal colouring of G is reduced to the problem of finding optimal colourings of two $\mathscr{F} - (k-1)\mathrm{e}$ graphs, and we recursively apply the connection–contraction method to these two $\mathscr{F} - (k-1)\mathrm{e}$ graphs and their modulators inherited from E_k. The recursion terminates when an input graph is an $\mathscr{F}$ graph, and an optimal colouring of the graph is computed directly.

To analyze the complexity of this algorithm, we consider its recursion tree. Since the tree is a binary tree of height at most k, it has at most 2^k leaves and $2^k - 1$ internal vertices. Each leaf takes at most $T(m+k,n)$ time since it is an $\mathscr{F}$ graph with at most $m+k$ edges and n vertices, and each internal node takes at most $O(m+n+k)$ time. Therefore the total time is $O(2^k(T(m+k,n)+m+n+k))$, which is $O(2^k \max\{T(m+k,n), m+n+k\})$. □

In fact, Theorem 3.1 implies that, for $\mathscr{F}$ closed under edge contraction, the problem of finding an optimal colouring of $\mathscr{F} - k$e graphs is not only polynomial-time solvable but also fixed-parameter tractable whenever the problem for $\mathscr{F}$ graphs is polynomial-time solvable and the problem of finding a modulator of an $\mathscr{F} - k$e graph is fixed-parameter tractable. For instance, from the facts that a modulator of a chordal$-k$e graph can be found in time $O(f(k)(m+n))$ [6,20] and an optimal colouring of a chordal graph can be found in time $O(m+n)$ [13], we deduce from Theorem 3.1 that an optimal colouring of a chordal$-k$e graph can be found in time $O(f(k)(m+n))$, which is linear time for each fixed k.

We now turn our attention to $\mathscr{F} + k$e graphs. It is unclear how a colouring algorithm for $\mathscr{F}$ graphs can help in obtaining colouring algorithms for $\mathscr{F} + k$e graphs. However, for some $\mathscr{F}$, when an algorithm for computing chromatic polynomials of $\mathscr{F}$ graphs is available, we can use it to compute chromatic polynomials of $\mathscr{F} + k$e graphs, and hence solve Vertex Colouring for $\mathscr{F} + k$e graphs.

Theorem 3.2. *Let $\mathscr{F}$ be a family of graphs closed under identification of nonadjacent vertices, and $T(m,n)$ denote the time to compute the chromatic polynomial of an $\mathscr{F}$ graph. Then the chromatic polynomial of an $\mathscr{F} + k$e graph G, given a modulator of G, can be computed in time* $O(2^k \max\{T(m,n), m+n\})$.

Proof. Let E_k be a modulator of G. Then $G - E_k \in \mathscr{F}$. For each $e \in E_k$, $G - e \in \mathscr{F} + (k-1)$e, and it follows from Lemma 2.3 that $G \cdot e \in \mathscr{F} + (k-1)$e. Furthermore, $E_k - e$ is a modulator of $G - e$, and the corresponding edges of $E_k - e$ in $G \cdot e$ is a modulator of $G \cdot e$. Since for each edge e of G, the chromatic polynomial of G satisfies $\chi(G,t) = \chi(G-e,t) - \chi(G \cdot e,t)$, we can use this recurrence relation with e being an edge in a modulator to compute the chromatic polynomial of G recursively. Using an analysis similar to that in the proof of Theorem 3.1, we obtain the claimed complexity. □

We note that split graphs and nonbipartite graphs are closed under identification of nonadjacent vertices. Since the chromatic polynomial of a split graph can be easily computed in linear time and a modulator of a split$+k$e graph can be found in uniformly polynomial time [6], it follows from Theorem 3.2 that Vertex Colouring on split$+k$e graphs is fixed-parameter tractable. It should be pointed out, however, that Theorem 3.2 is not as useful as Theorem 3.1 because very few families of graphs are closed under identification of nonadjacent vertices and, in the meantime, admit polynomial algorithms for computing their chromatic polynomials.

Finally, we consider $\mathscr{F} + k$v graphs. Again, it is unclear how a colouring algorithm for $\mathscr{F}$ graphs can help in obtaining colouring algorithms for $\mathscr{F} + k$v graphs. Nevertheless, we can use algorithms for List Colouring on $\mathscr{F}$ graphs to obtain algorithms for Vertex Colouring on $\mathscr{F} + k$v graphs.

Theorem 3.3. *If* List Colouring *is solvable in time* $O(T(m,n))$ *for $\mathscr{F}$ graphs, then* Vertex Colouring *for an $\mathscr{F} + k$v graph G, given a modulator of G, can be solved in time* $O(B(k)\max\{T(m,n), m+n+k\})$, *where $B(k)$ is the kth Bell number.*

Proof. Let V_k be a modulator of G. Then $G' = G - V_k$ is an $\mathscr{F}$ graph. Let $Z_t = \{1,\ldots,t\}$ be a set of t colours. We use the following method to determine whether G is t-colourable. If $G[V_k]$ is a clique, then we colour vertices in V_k by colours from 1 to $|V_k|$ respectively, and then assign to each vertex $v \in V(G) - V_k$ a list $L(v)$ of admissible colours with respect to Z_t, i.e., $L(v)$ contains colours in Z_t that have not been used by neighbours of v in V_k. Clearly, G is t-colourable iff G' is L-colourable. Otherwise, $G[V_k]$ contains two nonadjacent vertices u and v, and we construct from G two graphs $G + uv$ and $G(u = v)$. Then G is t-colourable iff $G + uv$ or $G(u = v)$ is t-colourable, and we recursively determine whether at least one of these two graphs is t-colourable. Note that $G + uv$ is an $\mathscr{F} + k$v graph with modulator V_k, and $G(u = v)$ is an $\mathscr{F} + (k-1)$v graph with the corresponding vertices of V_k in $G(u = v)$ as a modulator of $G(u = v)$.

For the time complexity of the above algorithm, we note that, as shown by Walsh [21], the number of leaves in the recursion tree is at most $B(k)$, the kth Bell number. Therefore, the time for determining whether an $\mathscr{F} + k$v graph is t-colourable is $O(B(k)(T(m,n) + m + n + k))$, which equals the claimed complexity. □

We note that List Colouring can be solved in polynomial time for complete graphs by transforming it into a matching problem. Let $\mathscr{K}$ be the family of complete graphs. Then a modulator of a $\mathscr{K} + k$v graph G is a k-vertex cover in the complement of G and thus can be found in uniformly polynomial time [9]. It follows from Theorem 3.3 that Vertex Colouring on $\mathscr{K} + k$v graphs is fixed-parameter tractable. Unfortunately, Theorem 3.3 is of limited use because List Colouring is NP-complete for almost all interesting families of graphs, and even remains NP-complete for complete bipartite graphs [18].

4. Split graphs

In this section, we consider Vertex Colouring on parameterized split graphs. Note that an optimal colouring of a split graph can be found in linear time [14], and that Vertex Colouring is NP-complete on parameterized split graphs when k is not fixed. We will prove that Vertex Colouring on both split $+\,k$e and split $-\,k$e graphs is linear-time solvable for each fixed k. On the other hand, we will prove that the problem on split $+\,k$v graphs is fixed-parameter intractable ($W[1]$-hard) but polynomial-time solvable for each fixed k.

First we consider split $+\,k$e graphs and split $-\,k$e graphs. By Theorems 3.1 and 3.2, given a modulator, an optimal colouring of a split $-\,k$e graph can be found and the t-colourability of a split $+\,k$e graph can be determined both in uniformly linear time. Unfortunately, although we can find a modulator of a split $+\,k$e or split $-\,k$e graph in uniformly polynomial time [6], we do not know how to find it in uniformly linear-time. Furthermore, no result in the previous section enables us to find an optimal colouring of a split $+\,k$e graph in uniformly polynomial time. Here we will derive a uniformly linear-time algorithm for finding an optimal colouring of a split $+\,k$e or split $-\,k$e graph by considering a larger family of graphs.

The *splittance* of a graph G is the minimum number of edges that need to be added to or deleted from G in order to make G a split graph. Therefore both split+ke graphs and split $-$ ke graphs are graphs of splittance at most k. Hammer and Simeone [15] showed that the splittance of a graph can be determined from the degree sequence of the graph, and gave a linear algorithm for finding a minimum set of edges that need to be added to or deleted from the graph to make the graph a split graph. We now show that an optimal colouring of a graph of splittance k can be found in uniformly linear time.

Theorem 4.1. *For each fixed k, an optimal colouring of a graph of splittance k can be found in linear time.*

Proof. First we show that an optimal colouring of a split + ke graph G, given a modulator E_k of G, can be found in time $O(m+n+f(k))$ for some function $f(k)$ independent of m and n. Since $G-E_k$ is a split graph, we can partition its vertices into an independent set I and a clique K in linear time. Let E' be the set of edges in E_k whose ends are all inside I. Then $G-E'$ is also a split graph and $\{I,K\}$ partitions its vertices into an independent set I and a clique K. Denote by H the subgraph of G induced by vertices incident with edges in E'. Then $V(H) \subseteq I$, $E(H)=E'$ and $G-V(H)$ is a split graph. Furthermore, H is only connected to vertices in the clique K.

To construct an optimal colouring of G, we first compute an optimal colouring g of $G-V(H)$ in linear time, and then try to extend the colouring to H. Let $\{1,\ldots,j\}$ be the set of colours used in the colouring g. Then $j \leqslant \chi(G) \leqslant j+k+1$, since $j \leqslant \chi(G-E') \leqslant j+1$ and the addition of an edge in E' to $G-E'$ requires at most one new colour to recolour one end of the edge.

For an integer t between j and $j+k+1$, we construct a t-colouring of G, if it exists, by solving the following list colouring problem of H. For each vertex v in H, its admissible colours are the colours in $\{1,\ldots,t\}$ not used by neighbours of v in $G-V(H)$. Note that H contains at most k edges. Thus if v has more than k admissible colours, it can always be coloured by one of its admissible colours since v has at most k neighbours in H. Therefore, we compute, for each v, up to $k+1$ admissible colours and assign them to v as the list $L_t(v)$ of admissible colours. This takes $O(m+n+k^2)$ time since it takes $O(d(v)+k)$ time to obtain $L_t(v)$ for each v, where $d(v)$ is the degree of v. Now since H is only connected to the clique K of $G-V(H)$, an L_t-colouring of H extends the j-colouring g of $G-V(H)$ to a t-colouring of G. By exhaustive search, we can construct an L_t-colouring of H, if it exists, in time $O((2k)^{k+1})$ as H has at most $2k$ vertices and $|L_t(v)| \leqslant k+1$. Therefore we can construct an optimal colouring of G by using a binary search on t for $j \leqslant t \leqslant j+k+1$ in $O(m+n+(2k)^{k+1}\log k)$ time.[1]

[1] In fact, we can reduce $(2k)^{k+1}\log k$ to $B(k+1)k^3$ by applying the connection–contraction method on each connected component of H independently, and then solving list colouring problems on complete graphs by matching.

Now for a graph G of splittance k, we first use the linear algorithm of Hammer and Simeone [15] to find disjoint sets E_1 and E_2 of edges so that $|E_1| + |E_2| = k$ and $G + E_1 - E_2$ is a split graph. Set $k_1 = |E_1|$ and $k_2 = |E_2|$ and let $\mathscr{F}$ be the family of split $+\, k_2$e graphs. Then $G + E_1$ is an $\mathscr{F}$ graph with modulator E_2 and G is an $\mathscr{F} - k_1$e graph with modulator E_1. Since split $+\, k_2$e graphs are closed under edge contraction and it takes $O(m + n + f(k_2))$ time to find an optimal colouring of a split $+\, k_2$e graph, it follows from Theorem 3.1 that an optimal colouring of G can be found in time $O(2^{k_1}(m + n + k_1 + f(k_2)))$, which is linear time for fixed k as $k = k_1 + k_2$. □

Corollary 4.2. *For each fixed k, an optimal colouring of a split $+\, k$e or split $-\, k$e graph can be found in linear time.*

The vertex colouring problem appears to be difficult for split $+\, k$v graphs, and Theorem 3.3 is not applicable since List Colouring is NP-complete on split graphs. However, we can use polynomial algorithms for solving List Colouring on partial k-trees to find an optimal colouring of a split $+\, k$v graph in polynomial time for each fixed k.

Theorem 4.3. *An optimal colouring of a split $+\, k$v graph G can be obtained in time* $O(n^{k+2} k \log k)$.

Proof. First find a modulator V_k of G, which can be done easily in time $O(n^{k+2})$. Partition the vertex set of $G - V_k$ into a clique K and an independent set I, which can be done in linear time. Clearly $|K| \leqslant \chi(G) \leqslant |K| + k + 1$. Now, given an integer t between $|K|$ and $|K| + k + 1$, we can construct a t-colouring of G, if it exists, by solving a list colouring problem as follows. Let $H = G[I \cup V_k]$. Then H is a partial k-tree. Let $Z_t = \{1, \dots, t\}$ be a set of t colours. First, colour each vertex in the clique K by a distinct colour in $\{1, \dots, |K|\}$. Then assign to each vertex $v \in I \cup V_k$ a list $L_t(v)$ of admissible colours with respect to Z_t. Clearly, G is t-colourable iff H is L_t-colourable, and an L_t-colouring of H yields a t-colouring of G. Since H is a partial k-tree, we use an algorithm of Jansen and Scheffler [18] to construct an L_t-colouring of H in time $O((|I| + |V_k|)t^{k+1}k)$. By using a binary search on t for $|K| \leqslant t \leqslant |K| + k + 1$, we can construct an optimal colouring of G in time $O(n^{k+2} k \log k)$. □

Can Vertex Colouring on split $+\, k$v graphs be solved in uniformly polynomial time? Unfortunately, it seems very unlikely as we will show that the problem is fixed-parameter intractable by a reduction from the following Independent k-Set problem, which is $W[1]$-complete [8].

Instance: Graph $G = (V, E)$.

Question: Does G contain an independent set of size k?

Theorem 4.4. Vertex Colouring *is $W[1]$-hard for split $+\, k$v graphs.*

Proof. We give a reduction from the Independent k-Set problem. Let $G = (V, E)$ be an arbitrary instance of Independent k-Set. For convenience, we assume $V = \{v_1, v_2, \dots, v_n\}$.

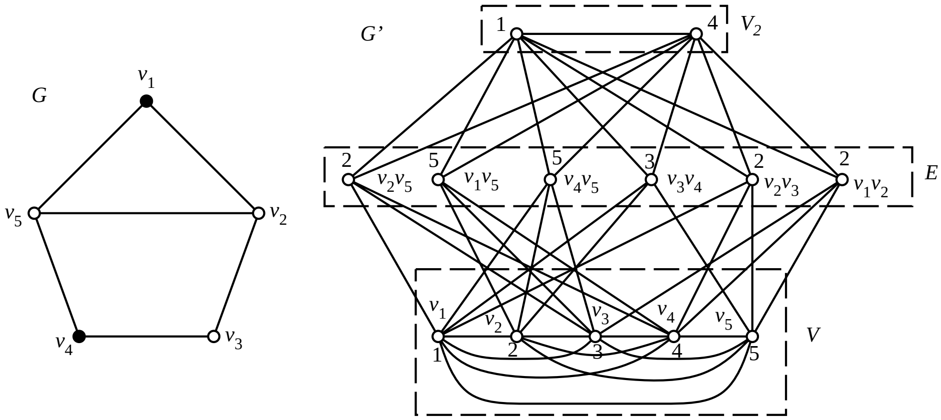

Fig. 1. The construction of G', where $k=2$ and $I=\{v_1,v_4\}$ is an independent set in G.

Construct from G a split $+\,k$v graph $G'=(V',E')$ as follows (see Fig. 1 for an example):

1. Set $V'=V\cup E\cup V_k$, where V_k is a set of k new vertices disjoint from $V\cup E$.
2. Connect every pair of vertices in V to form a complete graph on V, and connect every pair of vertices in V_k to form a complete graph on V_k.
3. For each vertex $v_iv_j\in E$, connect it with every vertex but v_i and v_j in V.
4. Connect every vertex in V_k with every vertex in E.

The construction clearly takes uniformly polynomial time. We now show that G contains an independent set of size k iff G' is n-colourable.

Suppose that G contains an independent set I of size k. Then $V-I$ is a vertex cover in G of size $n-k$, and we can construct a vertex n-colouring of G' as follows:

1. Colour vertex $v_i\in V$ by colour i.
2. Arbitrarily colour the k vertices in V_k by the k colours used for vertices in the independent set I of V.
3. For each vertex $v_iv_j\in E$, since $V-I$ is a vertex cover of G, at least one of v_i and v_j is in $V-I$. This implies that at least one of the colours used for vertices v_i and v_j (i.e., colours i or j) is not used for vertices in V_k. Therefore colour vertex v_iv_j by colour i if $v_i\in V-I$ and by colour j otherwise.

Conversely, suppose that f is an n-colouring of G'. Without loss of generality, we may assume that $f(v_i)=i$ for each vertex $v_i\in V$. Then for each vertex $v_iv_j\in E$ of G', $f(v_iv_j)$ equals either i or j since v_iv_j is adjacent to every vertex but v_i and v_j in V. Let Z_k be the set of k colours used for V_k. Then $S=\{1,2,\ldots,n\}-Z_k$ equals the set of colours used for the vertices in E since every vertex in V_k is adjacent to every vertex in E. From this, we deduce that $\{v_i\colon i\in S\}$ shares at least one vertex with each vertex in E. Therefore $\{v_i\colon i \in S\}$ is a vertex cover of size $n-k$ in G, and hence $V - \{v_i\colon i \in S\}$ is an independent set of size k in G. □

5. Bipartite graphs

The Vertex Colouring problem is trivially solvable in linear time for bipartite graphs, and one may expect that the addition of a few vertices or edges to a bipartite graph would not make the problem much harder on the new graph. Surprisingly, the addition of just a few vertices or edges to a bipartite graph drastically changes the complexity of Vertex Colouring from linear time to NP-complete!

Theorem 5.1. Vertex Colouring *is linear-time solvable for both bipartite* + 1v *graphs and bipartite* + 2e *graphs, but* NP-*complete on bipartite* + kv *graphs for every fixed* $k \geqslant 2$ *and bipartite* + ke *graphs for every fixed* $k \geqslant 3$.

Proof. Obviously, a bipartite + 1v graph is always 3-colourable, and it is not 2-colourable iff it is not a bipartite graph. Since bipartite graphs can be recognized in linear time, Vertex Colouring is linear-time solvable for bipartite + 1v graphs.

For a bipartite + 2e graph G, we first note that four colours always suffice. Since it is easy to tell in linear time whether G is 1- or 2-colourable, we need only consider the problem of determining whether G is 3-colourable. It is easy to see that G is 3-colourable if G contains no 4-clique, which implies that G is 3-colourable iff G contains no 4-clique. Therefore in order to solve Vertex Colouring in linear time, we need only devise a linear algorithm for determining whether G contains a 4-clique.

We make a few simple observations. First, if G is not bipartite then G contains an odd cycle C. Secondly, $G - E(C)$ is a bipartite + 1e graph and thus every 4-clique of G contains at least one edge in C. Thirdly, if an edge in C is contained in a 4-clique, then its two ends must have at least two common neighbours in G. Therefore we can determine whether G contains a 4-clique as follows: find an odd cycle C in G and for each edge in C check if it is contained in a 4-clique.

Algorithm 4-clique
Input: A bipartite + 2e graph G.
Output: "Yes" if G contains a 4-clique and "No" otherwise.
1. **if** G is bipartite
2. **then return** "No" and **stop**
3. **else** find an odd cycle C in G;
4. **for** each edge xy in C **do**
5. **if** $|N(x) \cap N(y)| \geqslant 2$
6. **then if** $G[N(x) \cap N(y)]$ contains an edge
7. **then return** "Yes" and **stop**;
8. **return** "No".

We now show that the above algorithm runs in linear time. We use sorted adjacency lists to represent G, which can be obtained from adjacency lists of G in linear time. It is clear that lines 1–3 take time $O(m+n)$. Line 5 takes $O(d(x)+d(y))$ time as adjacency lists are sorted, and line 6 takes $O(m+n)$ time. The loop of lines 4–7 may be executed $O(n)$ times; however, we will show that line 6 is executed at most 3 times, which implies that the total time for the loop of lines 4–7 is $O(m+n)$ instead of $O(n(m+n))$.

Let e and e' be two edges in G that makes $G' = G - \{e, e'\}$ bipartite. Then for every edge xy in G', x and y have no common neighbour in G' as G' is bipartite. In order for x and y to have at least two common neighbours in G, edge xy must be adjacent to both edges e and e'. If e and e' share a common vertex v (in this case, G contains no 4-clique), then edge xy must be incident with v or form a triangle with e and e'. Since the odd cycle C can have at most two edges incident with v, at most 3 edges in C meet the condition in line 5 and thus line 6 is executed at most 3 times in this case. Otherwise e and e' are vertex disjoint, and at most four edges in G can be adjacent to both e and e'. Together with edges e and e', at most 6 edges in G may meet the condition of line 5. But the odd cycle C can contain at most 3 of these 6 edges, and hence line 6 is also executed at most 3 times in this case. Therefore, we conclude that the whole algorithm runs in linear time.

For the NP-completeness part of the theorem, we give polynomial reductions from the following restricted version of the 1-Precolouring Extension problem, whose NP-completeness was established by Bodlaender et al. [5].

Instance: Bipartite graph $G = (X, Y; E)$ and three vertices $x_1, x_2, x_3 \in X$.

Question: Is there a 3-colouring of G that assigns vertices x_1, x_2, x_3 different colours?

Let G' be the graph constructed from G by adding a triangle on three new vertices v_1, v_2, v_3, and 6 edges $\{x_1v_2, x_1v_3, x_2v_1, x_2v_3, x_3v_1, x_3v_2\}$. Then G' is a bipartite + kv graph for every $k \geqslant 2$ as $G' - \{v_1, v_2\}$ is bipartite. For every 3-colouring f of G', the connection between vertices x_1, x_2, x_3 and the triangle on $\{v_1, v_2, v_3\}$ ensures that vertices x_1, x_2, x_3 receive different colours. Therefore, the restriction of f to G produces a 3-colouring of G with x_1, x_2, x_3 receiving different colours. On the other hand, every 3-colouring f' of G with x_1, x_2, x_3 receiving different colours can be extended to a 3-colouring of G' by colouring vertices v_1, v_2 and v_3, respectively, with colours $f'(x_1)$, $f'(x_2)$ and $f'(x_3)$. This establishes the NP-completeness of Vertex Colouring on bipartite + kv graphs for every fixed $k \geqslant 2$.

Finally, let H be the graph constructed from G by adding three edges x_1x_2, x_2x_3, x_3x_1 to G. Then H is a bipartite + ke graph for every $k \geqslant 3$. Since vertices x_1, x_2, x_3 induce a triangle in H, it is clear that H is 3-colourable iff G is 3-colourable with x_1, x_2, x_3 receiving different colours, which proves the NP-completeness on bipartite + ke graphs for every fixed $k \geqslant 3$. □

6. Concluding remarks

We have studied the complexity of Vertex Colouring on parameterized graph families mainly from the fixed-parameter point of view. Our results on parameterized split graphs and parameterized bipartite graphs have revealed a colourful diversity of the complexity of Vertex Colouring on parameterized graph families: fixed-parameter tractable, fixed-parameter intractable, polynomial-time solvable, and NP-complete. This is a strong evidence that this line of research is worthwhile and interesting from both theoretical and practical points of view.

An encouraging sign for this line of research is that polynomial algorithms for several important graph problems on $\mathscr{F}$ can be used to obtain polynomial algorithms for parameterized $\mathscr{F}$ graphs whenever the parameter value is fixed. For instance, it can be shown that for Clique ([GT19] in [11]), Independent Set ([GT20] in [11]), and Vertex Cover ([GT1] in [11]), whenever a problem is polynomial-time solvable for a hereditary family $\mathscr{F}$ of graphs, the problem is fixed-parameter tractable for parameterized $\mathscr{F}$ graphs, given modulators of input graphs.[2] However, for many other graph problems, their complexity status on parameterized graphs is unknown and seems hard to determine. In this regard, Dominating Set ([GT2] in [11]) may be of interest in view of the tremendous amount of work in the literature concerning the problem [16,17].

Turning back to vertex colourings, we note that the complexity of Vertex Colouring on $\mathscr{F} - k$e graphs is unknown if $\mathscr{F}$ is not closed under edge contraction. How can we obtain polynomial algorithms in this case? Is there an $\mathscr{F}$ for which Vertex Colouring is NP-complete on $\mathscr{F} - k$e graphs? In particular, it would be interesting to know its complexity on permutation$-k$e graphs and comparability$-k$e graphs. For $\mathscr{F} + k$e and $\mathscr{F} + k$v graphs, we actually know very little about the complexity of their vertex colouring problem. Other than Theorems 3.2 and 3.3, we have very little idea for obtaining efficient algorithms. On the other hand, it seems also difficult to establish NP-completeness of the colouring problem on these parameterized graphs. It can be shown that if List Colouring with k colours (k is a fixed constant) is NP-complete for $\mathscr{F}$, then Vertex Colouring is NP-complete for $\mathscr{F} + k$v graphs and for $\mathscr{F} + k'$e graphs with $k' = \binom{k}{2}$. However this is not of much use in establishing NP-completeness of Vertex Colouring because for most $\mathscr{F}$, List Colouring with k colours is not or not known to be NP-complete. New ideas will be required to determine the complexity status of the colouring problem for various $\mathscr{F} + k$e graphs and $\mathscr{F} + k$v graphs, for instance, chordal $+ k$e graphs and chordal $+ k$v graphs.

Finally, we address the problem of finding a modulator. In most cases it may be necessary to know a modulator in order to solve a problem efficiently on parameterized graphs. In fact, finding a modulator can be a bottleneck for efficiently solving some problems, such as Clique. There has been some work for finding modulators of parameterized graphs. In particular, it takes uniformly polynomial time to find modulators of the following families of graphs: chordal$-k$e graphs [6,20], planar $+ k$e graphs and planar $+ k$v graphs (since they are closed under taking minors), and parameterized $\mathscr{F}$ graphs for $\mathscr{F}$ admitting a finite forbidden induced subgraph characterization [6]. Nevertheless, the complexity status of finding modulators of parameterized $\mathscr{F}$ graphs is

[2] Hints for solving CLIQUE on parameterized graphs, where $\omega(G)$ denotes the clique number of G. For every vertex v in G, $\omega(G) = \max\{\omega(G - v), \omega(G[N(v)]) + 1\}$; for every edge uv in G, $\omega(G) = \max\{\omega(G - e), \omega(G[N(u) \cap N(v)]) + 2\}$; and for every pair of nonadjacent vertices u and v in G, $\omega(G) = \max\{\omega(G - u), \omega(G - v)\}$.

unknown for many $\mathscr{F}$. Is there a uniformly polynomial algorithm to find modulators of chordal $+\,k$e graphs or chordal $+\,k$v graphs? It would be also interesting to determine if the problems of finding modulators of bipartite $+\,k$e graphs or bipartite$+k$v graphs are fixed-parameter tractable.

Problems on parameterized graph families are abundant, and it is hoped that this paper will stimulate further research on parameterized graph families to enhance our knowledge about them from the perspectives of both traditional and parameterized complexity theories.

Acknowledgements

The author thanks the referees for constructive suggestions.

References

[1] S. Arnborg, D.G. Corneil, A. Proskurowski, Complexity of finding embeddings in a k-tree, SIAM J. Algebraic Discrete Methods 8 (1987) 277–284.
[2] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, J. Algorithms 12 (1991) 308–340.
[3] S. Arnborg, A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial k-trees, Discrete Appl. Math. 23 (1989) 11–24.
[4] H.L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (1996) 1305–1317.
[5] H.L. Bodlaender, K. Jensen, G.J. Woeginger, Scheduling with incompatible jobs, Discrete Appl. Math. 55 (1994) 219–232.
[6] L. Cai, Fixed-parameter tractability of graph modification problems for hereditary properties, Inform. Process. Lett. 58 (1996) 171–176.
[7] B. Courcelle, Recognizability and second-order definability for sets of finite graphs, Inform. and Comput. 85 (1990) 12–75.
[8] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness II: on completeness for $W[1]$, Theoret. Comput. Sci. 141 (1995) 109–131.
[9] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer, Berlin, 1999.
[10] S. Földes, P.L. Hammer, Split graphs, Congr. Numer. 19 (1977) 311–315.
[11] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman, San Francisco, CA, 1979.
[12] M.R. Garey, D.S. Johnson, L. Stockmeyer, Some simplified NP-complete graph problems, Theoret. Comput. Sci. 1 (1976) 237–267.
[13] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph, SIAM J. Comput. 1 (1972) 180–187.
[14] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[15] P.L. Hammer, B. Simeone, The splittance of a graph, Combinatorica 1 (3) (1981) 275–284.
[16] T.W. Haynes, S.T. Hedetniemi, P.J. Slater (Eds.), Fundamentals of Domination in Graphs, Marcel Dekker, New York, 1998.
[17] T.W. Haynes, S.T. Hedetniemi, P.J. Slater (Eds.), Domination in Graphs: Advanced Topics, Marcel Dekker, New York, 1998.
[18] K. Jansen, P. Scheffler, Generalized coloring for tree-like graphs, Discrete Appl. Math. 75 (1997) 135–155.
[19] D.S. Johnson, The NP-completeness column: an ongoing guide (16), J. Algorithms 6 (1985) 434–451.
[20] H. Kaplan, R. Shamir, R.E. Tarjan, Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs, SIAM J. Comput. 28 (1906–1922) 1999.

[21] T.R. Walsh, Worst-case analysis of Read's chromatic polynomial algorithm, Ars Combin 46 (1997) 145–151.
[22] D.B. West, Introduction to Graph Theory, Prentice-Hall, New York, 1996.

For further reading

L. Cai, B. Schieber, A linear-time algorithm for computing the intersection of all odd cycles in a graph, Discrete Appl. Math. 73 (1997) 27–34.