(k, l)-Colourings and Ferrers Diagram Representations of Cographs

Dennis A. Epple and Jing Huang*

Abstract

For a pair of natural numbers k, l, a (k, l)-colouring of a graph G is a partition of the vertex set of G into (possibly empty) sets S_1, S_2, ..., S_k, C_1, C_2, ..., C_l such that each set S_i is an independent set and each set C_j induces a clique in G.

The (k, l)-colouring problem, which is NP-complete in general, has been studied for special graph classes such as chordal graphs, cographs and line graphs. Let ˆκ(G) = (κ_0(G), κ_1(G), ..., κ_{θ(G)-1}(G)) and ˆλ(G) = (λ_0(G), λ_1(G), ..., λ_{χ(G)-1}(G)) where κ_l(G) (respectively, λ_k(G)) is the minimum k (respectively, l) such that G has a (k, l)-colouring. We prove that ˆκ(G) and ˆλ(G) are a pair of conjugate sequences for every graph G and when G is a cograph, the number of vertices in G is equal to the sum of the entries in ˆκ(G) or in ˆλ(G). Using the decomposition property of cographs we show that every cograph can be represented by Ferrers diagram. We devise algorithms which compute ˆκ(G) for cographs G and find an induced subgraph in G that can be used to certify the non-(k, l)-colourability of G.

Key words: (k, l)-colouring, bichromatic number, Ferrers diagram, cograph, box co-cograph, cotree, algorithm, complexity

1 Introduction

Let G be a graph and k, l ≥ 0 be natural numbers. A (k, l)-colouring of G is a partition of the vertex set of G into (possibly empty) sets S_1, S_2, ..., S_k, C_1, C_2, ..., C_l such that each S_i is an independent set and each C_j induces a clique in G. The concept of (k, l)-colourings encompasses the classical colouring and clique covering of graphs; indeed, a (k, 0)-colouring is just a k-colouring and a (0, l)-colouring is a partition of G into at most l cliques. A graph is (k, l)-colourable if it has a (k, l)-colouring. Thus bipartite graphs are exactly the (2, 0)-colourable graphs and split graphs are precisely the (1, 1)-colourable graphs [14].

*Department of Mathematics and Statistics, University of Victoria, Victoria, B.C., Canada V8W 2Y2; huangj@uvic.ca
The bichromatic number $\chi^b(G)$ of $G$ is the least integer $r$ such that, for all $k, l$ with $k + l = r$, $G$ is $(k, l)$-colourable. The notion of bichromatic number arose in the study of extremal graphs, motivated by classical results of Turán [19] and of Erdős, Stone and Simonovits (cf. [12]). This parameter has been studied by Prömel and Steger [18] under the name of $\tau$-parameter, by Bollobás and Thomason [3] under the name of colouring number, and by Axenovich, Kézdy and Martin [2] under the name of binary chromatic number. The parameter is tied to the speed of hereditary properties and edit distance, cf. [1, 2, 3]. A counterpart of the bichromatic number $\chi^b(G)$ is the notion of the cochromatic number $\chi^c(G)$, which is the least integer $r$ such that $G$ is $(k, l)$-colourable for some $k, l$ with $k + l = r$, cf. [17].

It is not surprising that computing the bichromatic number of a graph is an NP-hard problem, cf. [11]. Brandstädt [4, 5] proved that the problem of deciding whether a graph is $(k, l)$-colourable is NP-complete for fixed $k, l$ with $k \geq 3$ or $l \geq 3$ and polynomial time solvable otherwise. A graph is chordal if it does not contain an induced $C_k$ for each $k \geq 4$ and is a cograph if it does not contain an induced $P_4$. It is proved in [16] that a chordal graph is $(k, l)$-colourable if and only if it does not contain $(l + 1)K_{k+1}$, the disjoint union of $l + 1$ copies of $K_{k+1}$ (see definition below). The $(k, l)$-colourability of cographs and line-graphs have been studied in [8, 9, 10, 11, 13].

Every graph $G$ satisfies $\chi^b(G) \leq \chi(G) + \theta(G) - 1$ where $\chi(G)$ and $\theta(G)$ are the chromatic number and the clique covering number of $G$ respectively, cf. [1, 18]. Graphs which satisfy this inequality with equality have been characterized in [11]. It turns out that all these graphs are cographs. To describe the characterization, we recall the recursive definition of cographs. The disjoint union of graphs $G$ and $H$, denoted by $G + H$, is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$.

Let $\mathcal{C}$ be the set of graphs defined recursively as follows:

- $K_1 \in \mathcal{C}$;
- if $G \in \mathcal{C}$, then $\overline{G} \in \mathcal{C}$;
- if $G, H \in \mathcal{C}$, then $G + H \in \mathcal{C}$.

**Theorem 1.1.** [6] The following statements are equivalent for a graph $G$:

1. $G \in \mathcal{C}$;
2. $G$ is a cograph (i.e., $G$ does not contain an induced $P_4$);
3. for every induced subgraph $H \neq K_1$ of $G$, either $H$ or $\overline{H}$ is disconnected. \(\square\)

Let $\mathcal{B}$ be a set of graphs constructed recursively as follows:

- $K_1 \in \mathcal{B}$;
- if $G \in \mathcal{B}$, then $\overline{G} \in \mathcal{B}$;
• if $G, H \in \mathcal{B}$ with $\chi(G) = \chi(H)$, then $G + H \in \mathcal{B}$.

The join of $G$ and $H$, denoted by $G \vee H$, is the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H) \cup \{uv : u \in V(G), v \in V(H)\}$. It is easy to see that if $G, H \in \mathcal{C}$ then $G \vee H \in \mathcal{C}$ and if $G, H \in \mathcal{B}$ with $\theta(G) = \theta(H)$ then $G \vee H \in \mathcal{B}$.

Clearly, $\mathcal{B} \subseteq \mathcal{C}$. We call the graphs in $\mathcal{B}$ box cographs. It is proved in [11] that the box cographs are exactly the graphs $G$ which satisfy the inequality $\chi_b(G) \leq \chi(G) + \theta(G) - 1$ with equality.

**Theorem 1.2.** [11] A graph $G$ satisfies $\chi_b(G) = \chi(G) + \theta(G) - 1$ if and only if it is a box cograph.

A box cograph is of dimension $k$ times $l$ if it has chromatic number $k$ and clique covering number $l$. The graph $lK_k$ is a box cograph of dimension $k$ times $l$. No box cograph of dimension $k + 1$ times $l + 1$ is $(k, l)$-colourable. The following theorem asserts that they are exactly the forbidden (induced) subgraphs for $(k, l)$-colourable cographs. An equivalent statement of the theorem is proved in [13].

**Theorem 1.3.** [11] A cograph is $(k, l)$-colourable if and only if does not contain a box cograph of dimension $k + 1$ times $l + 1$ as an induced subgraph.

**Corollary 1.4.** For any cograph $G$,

$$\chi_b(G) = \max\{k + l : G \text{ contains a box cograph of dimension } k \text{ times } l\} - 1$$

Let $G$ be a graph and $k, l$ be natural numbers. We use $\kappa_l(G)$ to denote the minimum $k$ for which $G$ is $(k, l)$-colourable and use $\lambda_k(G)$ to denote the minimum $l$ for which $G$ is $(k, l)$-colourable. Write

$$\hat{\kappa}(G) = (\kappa_0(G), \kappa_1(G), \ldots, \kappa_{\theta(G) - 1}(G))$$

and

$$\hat{\lambda}(G) = (\lambda_0(G), \lambda_1(G), \ldots, \lambda_{\chi(G) - 1}(G)).$$

The knowledge of the values of $\hat{\kappa}(G)$ or in $\hat{\lambda}(G)$ can be directly used to determine whether $G$ is $(k, l)$-colourable and hence to compute the bichromatic number $\chi_b(G)$. Indeed, a graph $G$ is $(k, l)$-colourable if and only if $\kappa_l(G) \leq k$ or equivalently, if and only if $\lambda_k(G) \leq l$.

In this paper, we show that $\hat{\kappa}(G)$ and $\hat{\lambda}(G)$ are a pair of conjugate sequences for every graph $G$, that is, the Ferrers diagrams of $\hat{\kappa}(G)$ and $\hat{\lambda}(G)$ are conjugate to each other. We prove that, when $G$ is a cograph, the number of vertices in $G$ is equal to the sum of the entries in either of the sequences. This is not true in general. Using the decomposition property of cographs, we show that every cograph can be drawn in a shape similar to a Ferrers diagram for a sequence of numbers. We devise efficient bottom-up and top-down
algorithms on cotrees of cographs. The bottom-up algorithm calculates the sequence $\hat{\kappa}(G)$ and the top-down algorithm finds a box cograph of specified dimension which certifies the non-$(k,l)$-colourability of the input graph $G$.

Algorithms for the $(k, l)$-colourability of cographs have been studied by Gimbel, Kratsch, and Stewart [15] and by Demange, Ekim and de Werra [8]. In [15], an algorithm for the computation of the cochromatic number of a cograph using its cotree was presented. This algorithm, which runs in time $O(n^2)$, implicitly uses $(k, l)$-colourings of cographs. Demange, Ekim and de Werra [8] gave two different algorithms concerning the $(k, l)$-colouring of cographs. The first, which also uses cotrees, calculates a maximum $(k, l)$-colourable induced subgraph of a cograph (thereby also checking the $(k, l)$-colourability of the cograph itself) in time $O((k^3l + kl^3)n)$. For the purpose of the second algorithm, it is shown that if $G$ is a $(k, l)$-colourable cograph (with $l \geq 1$) and $C$ a maximum clique of $G$, then $G - C$ is $(k, l - 1)$-colourable. Using this, the algorithm finds a $(k, l)$-colouring of a cograph (if one exists) by successively removing $l$ maximum cliques and finding a $k$-colouring of the remaining graph. The algorithm runs in time $O(n(m + n))$ where $m$ is the number of edges of the graph. An adaptation of this idea is presented which calculates the cochromatic number in time $O(n^{3/2})$. Our algorithms for computing $\hat{\kappa}(G)$ and for finding certificates for the non-$(k, l)$-colourability of cographs $G$ matches the same complexity $O(n^2)$ as the algorithms in [8, 15]. In fact, the algorithm for $\hat{\kappa}(G)$ can be implemented to run in time $O(n \log n)$.

2 Ferrers diagrams

Since $\kappa_0(G) \geq \kappa_1(G) \geq \cdots \geq \kappa_{\chi(G) - 1}(G)$ and $\lambda_0(G) \geq \lambda_1(G) \geq \cdots \geq \lambda_{\chi(G) - 1}(G)$, $\hat{\kappa}$ and $\hat{\lambda}$ are both monotonically non-increasing sequences, we can represent each of them by a Ferrers diagram. For example, the graph in Figure 1 has $\hat{\kappa} = (3, 3, 1)$ and $\hat{\lambda} = (3, 2, 2)$, which are represented by Ferrers diagrams in Figure 2. Note that these two sequences are conjugate to each other, that is, by reflecting any of the diagrams along the main diagonal we obtain the other diagram. We show below this is the case for every graph.

Figure 1: A graph

Theorem 2.1. For any graph $G$, $\hat{\kappa}(G)$ and $\hat{\lambda}(G)$ are a pair of conjugate sequences.

Proof: For each $j$ with $0 \leq j \leq \chi(G) - 1$, let

$$i_j^* = \max \{i : \kappa_i(G) \geq j + 1\}.$$
To prove that $\hat{\lambda}(G)$ is the conjugate sequence of $\hat{k}(G)$, it suffices to show that $\lambda_j(G) = i^*_j + 1$. By the definition of $i^*_j$, we have $\kappa_{i^*_j}(G) \geq j + 1$ and $\kappa_{i^*_j+1}(G) \leq j$. Since $\kappa_{i^*_j} \geq j + 1$, $G$ is not $(j, i^*)$-colourable which means that $\lambda_j(G) \geq i^* + 1$. Since $\kappa_{i^*_j+1}(G) \leq j$, $G$ is $(j, i^* + 1)$-colourable which means that $\lambda_j(G) \leq i^* + 1$. Therefore we have $\lambda_j(G) = i^* + 1$.

Theorem 2.1 allows us to convert $\hat{k}(G)$ into $\hat{\lambda}(G)$ and vice versa. In particular, it implies that the two Ferrers diagrams of $\hat{k}(G)$ and $\hat{\lambda}(G)$ have the same number of dots, that is,

$$\kappa_0(G) + \kappa_1(G) + \cdots + \kappa_{\theta(G) - 1}(G) = \lambda_0(G) + \lambda_1(G) + \cdots + \lambda_{\chi(G) - 1}(G).$$

The number of vertices of the graph in Figure 1 coincides with the number of dots in either of the Ferrers diagrams in Figure 2 but this is not true in general. For instance, the 4-vertex graph $P_4$ has $\hat{k} = (2, 1)$ whose Ferrers diagram has only three dots, whereas the 5-vertex graph $C_5$ has $\hat{k} = (3, 2, 1)$ whose Ferrers diagram has six dots. However, we will show that for every cograph $G$, the number of vertex in $G$ is always equal to the number of dots in either of the Ferrers diagrams of $\hat{k}(G)$ and $\hat{\lambda}(G)$ (see Theorem 2.8).

**Proposition 2.2.** Let $G, H$ be graphs and $k$ be a natural number. Then

$$\lambda_k(G + H) = \lambda_k(G) + \lambda_k(H).$$

**Proof:** It suffices to show that $G + H$ admits a $(k, \lambda_k(G) + \lambda_k(H))$-colouring but not a $(k, \lambda_k(G) + \lambda_k(H) - 1)$-colouring. For the first condition, let

$$S_1, S_2, \ldots, S_k, C_1, C_2, \ldots, C_{\lambda_k(G)}$$

be a $(k, \lambda_k(G))$-colouring of $G$ and

$$S'_1, S'_2, \ldots, S'_k, C'_1, C'_2, \ldots, C'_{\lambda_k(H)}$$

be a $(k, \lambda_k(H))$-colouring of $H$. Since $G + H$ contains no edges between $G$ and $H$, the sets $S_i \cup S'_i$ are independent for all $i$. Therefore

$$S_1 \cup S'_1, \ldots, S_k \cup S'_k, C_1, C_2, \ldots, C_{\lambda_k(G)}, C'_1, C'_2, \ldots, C'_{\lambda_k(H)}$$

is a $(k, \lambda_k(G) + \lambda_k(H))$-colouring of $G + H$.

Now suppose, $G + H$ admits a $(k, \lambda_k(G) + \lambda_k(H) - 1)$-colouring. Again, as $G + H$ has no edges between $G$ and $H$, every clique is completely contained in either $G$ or $H$. By the definition of $\lambda_k$, at least $\lambda_k(G)$ of the cliques are contained in $G$, while at least $\lambda_k(H)$ of the cliques are contained in $H$, implying that there are at least $\lambda_k(G) + \lambda_k(H)$ cliques in total, a contradiction. Therefore $G + H$ does not admit a $(k, \lambda_k(G) + \lambda_k(H) - 1)$-colouring. 

\[\square\]
**Corollary 2.3.** For any graphs $G, H$,

$$\hat{\lambda}(G + H) = \hat{\lambda}(G) + \hat{\lambda}(H),$$

where the addition is performed entrywise.

The two sequences $\hat{\lambda}(G)$ and $\hat{\lambda}(H)$ in Corollary 2.3 may have different length and if so we append zeros to the shorter one to make them the same length. By applying Proposition 2.2 and the fact that $\kappa_l(G) = \lambda_l(G)$, we can obtain the following equivalent statements for the join of two graphs.

**Proposition 2.4.** Let $G, H$ be graphs and $l$ a natural number. Then

$$\kappa_l(G \lor H) = \kappa_l(G) + \kappa_l(H).$$

**Proof:** We have

$$\kappa_l(G \lor H) = \lambda_l(G \lor H) = \lambda_l(G + H) = \lambda_l(G) + \lambda_l(H) = \kappa_l(G) + \kappa_l(H).$$

**Corollary 2.5.** For any graphs $G, H$,

$$\hat{\kappa}(G \lor H) = \hat{\kappa}(G) + \hat{\kappa}(H),$$

where the addition is performed entrywise.

Using the Ferrers diagrams, we can also compute $\hat{\lambda}(G \lor H)$ and $\hat{\kappa}(G + H)$. Consider $G \lor H$. By Corollary 2.3, $\hat{\kappa}(G \lor H)$ is obtained by adding $\hat{\kappa}(G)$ and $\hat{\kappa}(H)$ entrywise. In terms of the Ferrers diagrams, we can picture this as putting the Ferrers diagrams of $\hat{\kappa}(G)$ and $\hat{\kappa}(H)$ beside each other with the rows lining up and moving all dots to the beginning of the row. This is equivalent to sorting the columns from largest to smallest. An example is given in Figure 3. We see that $\hat{\lambda}(G \lor H)$, being the conjugate of $\hat{\kappa}(G \lor H)$, is obtained by concatenating the sequence $\hat{\lambda}(G)$ with $\hat{\lambda}(H)$ and sorting the resulting sequence from largest to smallest. This way of concatenating two sequences prompts the following definition of $\ast$.

Let $\hat{a}$ and $\hat{b}$ be two non-increasing finite sequences of natural numbers. Then $\hat{a} \ast \hat{b}$ is the sequence obtained from concatenating $\hat{a}$ and $\hat{b}$ and sorting its entries from largest to smallest.

**Corollary 2.6.** For any graphs $G, H$,

$$\hat{\lambda}(G \lor H) = \hat{\lambda}(G) \ast \hat{\lambda}(H).$$
Figure 3: Ferrers diagrams of $\hat{\kappa}(G) = (3, 2, 2, 1)$, $\hat{\kappa}(H) = (3, 2, 1)$ and $\hat{\kappa}(G \vee H) = (6, 4, 3, 1)$.

**Corollary 2.7.** For any graphs $G, H$, 
\[
\hat{\kappa}(G + H) = \hat{\kappa}(G) \ast \hat{\kappa}(H).
\]

An interesting consequence of the above results is the following:

**Theorem 2.8.** For any cograph $G$, 
\[
\sum_{i \geq 0} \kappa_i(G) = \sum_{i \geq 0} \lambda_i(G) = |V(G)|.
\]

**Proof:** The first equality follows from Theorem 2.1. We will show 
\[
\sum_{i \geq 0} \kappa_i(G) = |V(G)|
\]
by induction on the number of vertices of $G$. If $G = K_1$, then $\hat{\kappa}(G) = (1)$, thus the statement holds. Assume that $G \neq K_1$ and the statement holds for all cographs with fewer vertices than $G$. Suppose $G$ is disconnected. Then there exist cographs $G_1, G_2$ such that $G = G_1 + G_2$ and we have $\hat{\kappa}(G) = \hat{\kappa}(G_1) \ast \hat{\kappa}(G_2)$. By the definition of the operation $\ast$ and the induction hypothesis,
\[
\sum_{i \geq 0} \kappa_i(G) = \sum_{i \geq 0} \kappa_i(G_1) + \sum_{i \geq 0} \kappa_i(G_2)
= |V(G_1)| + |V(G_2)|
= |V(G_1) \cup V(G_2)|
= |V(G)|.
\]
If on the other hand $G$ is connected, there exist cographs $G_1, G_2$ with $G = G_1 \vee G_2$ and
we obtain as above

\[
\sum_{i \geq 0} \kappa_i(G) = \sum_{i \geq 0} (\kappa_i(G_1) + \kappa_i(G_2)) \\
= \sum_{i \geq 0} \kappa_i(G_1) + \sum_{i \geq 0} \kappa_i(G_2) \\
= |V(G)|.
\]

It is possible to draw a cograph \(G\) on the Ferrers diagram of \(\hat{\lambda}(G)\) by adding edges in such a way that the dots in each row form an independent set and the dots in each column induce a clique in \(G\). Such a drawing of \(G\) is called a Ferrers diagram representation of \(G\).

\[G_1\] \hspace{1cm} \[G_2\] \hspace{1cm} \[G_1 + G_2\]
\[
\bullet \bullet \bullet \hspace{1cm} \bullet \bullet \bullet \hspace{1cm} \bullet \bullet \bullet \bullet \bullet \bullet \\
\bullet \bullet \bullet \hspace{1cm} \bullet \bullet \hspace{1cm} \bullet \bullet \bullet \bullet \bullet \\
\bullet \hspace{1cm} \bullet \hspace{1cm} \bullet \hspace{1cm} \bullet \\
\]

Figure 4: An illustration of the proof of Theorem 2.9.

**Theorem 2.9.** Every cograph \(G\) has a Ferrers diagram representation.

**Proof:** The proof is by induction on the number of vertices of \(G\). If \(G = K_1\), then the Ferrers diagram consists of a single point. Assume that \(G\) has more than one vertex and every cograph on fewer vertices than \(G\) has a Ferrers diagram representation. Suppose, \(G\) is disconnected. Then there exist cographs \(G_1, G_2\) with \(G = G_1 + G_2\). By the induction hypothesis, \(G_1\) and \(G_2\) have a Ferrers diagram representation. Consider what happens if we write the two diagrams side by side (see the left side of Figure 4 for an example). Each column completely belongs to either \(G_1\) or \(G_2\), therefore forms a clique. Each row may have vertices from both \(G_1\) and \(G_2\), but the vertices in each of the two graphs form an independent set, and therefore the whole row must form an independent set in \(G_1 + G_2\). The diagram we have might not be a Ferrers diagram, though. However, this can be remedied by permuting the columns. As this does not change the sets of rows and columns, we obtain a Ferrers diagram representation of \(G\) (see the right side of Figure 4 for the example).

If \(G\) is connected, there exist cographs \(G_1, G_2\) with \(G = G_1 \lor G_2\) and we obtain a Ferrers diagram representation of \(G\) by writing the two Ferrers diagram representations of \(G_1\) and \(G_2\) on top of each other and permuting the rows instead. \(\Box\)
3 Cotrees

It follows from Theorem 1.1 that every cograph $G$ on at least two vertices can be either written as $G = G_1 + G_2$ or $G = G_1 \lor G_2$ for some cographs $G_1, G_2$. These characteristic properties of cographs allow us to represent a cograph $G$ as a tree, called the **cotree** of $G$, cf. [6]. The cotree of $G$, denoted by $T_G$, is a rooted tree where every internal node is labelled with either 0 or 1, which can be recursively constructed as follows.

- If $G = K_1$, then we define $T_G$ to be the rooted tree on a single vertex.
- If $G$ is disconnected, let $G_1, G_2, \ldots, G_t$ be the connected components of $G$. We take the cotrees of $G_1, G_2, \ldots, G_t$ and add an edge from each of the roots to a new vertex, which we label with a 0. The tree thus constructed, with the root at the new vertex, is the cotree $T_G$.
- If $G \neq K_1$ is connected, let $\overline{G_1}, \overline{G_2}, \ldots, \overline{G_t}$ be the connected components of $\overline{G}$. We take the cotrees of $G_1, \ldots, G_t$ and add an edge from each of the roots to a new vertex, which we label with a 1. The tree thus constructed, with the root at the new vertex, is the cotree $T_G$.

We remark that the construction implies that the cotree $T_G$ for a cograph $G$ is unique. Every leaf of the cotree $T_G$ represents a vertex of $G$ and every internal node represents the subgraph of $G$ induced by the vertices that are descendents of that node. Every 0-node represents a disconnected subgraph, every 1-node a connected subgraph. By the construction, all children of a 0-node represent connected cographs, thus are either 1-nodes or leaves. Similarly the children of a 1-node are either 0-nodes or leaves. Also, we note that two vertices of the cograph are adjacent if and only if the lowest common ancestor of the corresponding leaves is a 1-node. An example of a cotree is given in Figure 5. The corresponding cograph is shown in Figure 6 where the thick edges stand for complete adjacency.

![Figure 5: A cotree.](image_url)

It is known that cotrees can be constructed in linear time (cf. [7]). Algorithms which are implemented on cotrees can be classified into two types, depending on how they traverse on them. The bottom-up algorithm traverses the cotree from the leaves to the root,
while the top-down algorithm traverses the cotree from the root to the leaves. Examples of bottom-up algorithms include calculating the chromatic number, the cochromatic number, the number of cliques, and number of transitive orientations of a cograph (cf. [6, 15]). We will give in Section 4 two bottom-up algorithms, for calculating the sequence \( \hat{\kappa}(G) \) and for constructing the Ferrers diagram representation of a cograph \( G \) respectively.

Top-down algorithms are suited for example for finding an induced subgraph with certain properties, such as a maximum clique. Although not explicitly given in [6], a top-down algorithm for finding a maximum clique in a cograph can be derived from the bottom-up algorithm for the chromatic number of a cograph. We will also give in Section 4 a top-down algorithm for finding an induced box cograph of a given dimension in a cograph.

4 Calculating \( \hat{\kappa}(G) \) and certificates

By Theorem 1.3, a cograph is \((k, l)\)-colourable if and only if it does not contain a box cograph of dimension \( k+1 \) times \( l+1 \) as an induced subgraph. In this section, we devise a bottom-up algorithm which calculates for any cograph \( G \) the sequence \( \hat{\kappa}(G) \) from which it can be determined whether \( G \) is \((k, l)\)-colourable for any given \( k, l \). In the case when \( G \) is not \((k, l)\)-colourable a top-down algorithm will find an induced box cograph of dimension \( k+1 \) times \( l+1 \) in \( G \) which certifies that \( G \) is not \((k, l)\)-colourable.

Our bottom-up algorithm for the calculation of \( \hat{\kappa} \) of a cograph relies on the formulas for \( \hat{\kappa} \) given in Corollaries 2.5 and 2.7. The algorithm is similar in nature to the one presented in [15] for the cochromatic number. However, the presentation is much simpler due to the formulas for \( \hat{\kappa} \) established here.

Algorithm 4.1. (KAPPA)

- **INPUT**: Cotree \( T \) of a cograph \( G \).
- **INITIALIZATION**: Assign \( \hat{\kappa} = (1) \) to each leaf of \( T \).
Figure 7: KAPPA for the cotree from Figure 5.

- **0-NODE OPERATOR:** \( \hat{\kappa}(A) = \hat{\kappa}(A_1) \ast \hat{\kappa}(A_2) \ast \cdots \ast \hat{\kappa}(A_t) \).
- **1-NODE OPERATOR:** \( \hat{\kappa}(A) = \hat{\kappa}(A_1) + \hat{\kappa}(A_2) + \cdots + \hat{\kappa}(A_t) \).
- **OUTPUT:** \( \hat{\kappa}(G) \).

**Proof:** [Proof of Correctness] The correctness follows directly from the fact that \( \hat{\kappa}(K_1) = K_1 \) and from the formulas for \( \hat{\kappa} \) given in Corollaries 2.5 and 2.7 for the disjoint union and join of graphs.

An example of the output of KAPPA is shown in Figure 7 where the arguments of the leaves (all being (1)) have been omitted.

As we can easily calculate the cochromatic number and bichromatic number from \( \hat{\kappa} \), KAPPA can be seen as an algorithm for the \((k,l)\)-colourability, the cochromatic number and the bichromatic number of a cograph.

We will briefly discuss the complexity of KAPPA. Let \( n \) be the number of vertices of the cograph \( G \). Then the number of operations (\(*\) or \(+\)) performed by the algorithm is \( O(n) \). We will show that each operation only needs time \( O(n) \). Note that each sequence \( \hat{\kappa}(A) \) has length \( |A| \). To calculate \( \hat{\kappa}(A_1) \ast \hat{\kappa}(A_2) \), say, we need to sort the concatenated sequence of \( \hat{\kappa}(A_1) \) and \( \hat{\kappa}(A_2) \). Since both sequences are already sorted, we only need to scan each sequence once. As both sequences have length at most \( n \), this can be done in \( O(n) \). To calculate \( \hat{\kappa}(A_1) + \hat{\kappa}(A_2) \), we need to perform at most \( n \) additions, which is also \( O(n) \). Thus the algorithm can be implemented in time \( O(n^2) \), which matches the complexity of the algorithm from \cite{15}. However, it is possible to slightly modify KAPPA to an algorithm that runs in time \( O(n \log n) \). For the sake of explanation, we consider a more general invariant of cotrees, in which each internal node (again labeled either 0 or 1) has exactly two children. We call such a tree a pseudocotree. In general, a cograph can be represented by different pseudocotrees. But the number of nodes in any pseudocotree for a cograph \( G \) on \( n \) vertices is \( O(n) \). This means that the number of operations performed on any pseudocotree for \( G \) is \( O(n) \). Suppose that \( A \) is a node in the input pseudocotree for \( G \) and \( A_1 \) and \( A_2 \) are the two children of \( A \). We claim that \( \hat{\kappa}(A) \) can be calculated in time \( O(\min \{|A_1|, |A_2|\}) \). Indeed, assume without loss of generality that \( a = |A_1| \leq |A_2| \). If \( A \) is a 1-node, then \( \hat{\kappa}(A) \) can be obtained by adding the at most \( a \) entries of \( \hat{\kappa}(A_1) \) to the first entries of \( \hat{\kappa}(A_2) \), which can be done in time \( O(a) \). Suppose that \( A \) is a 0-node.
The operation on \( A \) requires to merge \( \kappa(A_1) \) into \( \kappa(A_2) \). To do it efficiently, we can store \( \kappa(A_2) \) in the form of \( (n_1^{\alpha_1}, n_2^{\alpha_2}, \ldots, n_q^{\alpha_q}) \) where \( n_1 > n_2 > \cdots > n_q \). Thus it takes at most \( O(a) \) scans of the entries \( n_i^{\alpha_i} \) for the merge of \( \kappa(A_1) \) into \( \kappa(A_2) \). Hence \( \kappa(A) \) can be calculated in time \( O(\min\{|A_1|, |A_2|\}) \) for each node \( A \) of the pseudocotree. This implies that the algorithm can be implemented to run in time \( O(n \log n) \).

A similar algorithm as KAPPA can be devised to calculate \( \lambda(G) \) for a cograph \( G \). This can be done simply by performing \( \hat{\lambda}(A) = \hat{\lambda}(A_1) + \hat{\lambda}(A_2) + \cdots + \hat{\lambda}(A_t) \) on each 0-node \( A \) and \( \hat{\lambda}(A) = \hat{\lambda}(A_1) \ast \hat{\lambda}(A_2) \ast \cdots \ast \hat{\lambda}(A_t) \) on each 1-node \( A \) of the cotree of \( G \). The correctness of this algorithm is justified by Corollaries 2.3 and 2.6.

We can use KAPPA to establish a top-down algorithm that finds a certain box cograph. As a reminder, the obstructions for \((k-1, l-1)\)-colourability of cographs are precisely the box cographs of dimension \( k \times l \), similar to the \( k \)-clique being the obstructions for \((k-1)\)-colourability. We use \([r]^s\) to denote the sequence consisting of \( s \) entries of \( r \).

**Algorithm 4.2. (BOX COGRAPH)**

- **INPUT:** Cotree \( T \) of a cograph \( G \) with \( \kappa_{i-1}(G) \geq k \) and \( \kappa \) for each node of \( T \).
- **INITIALIZATION:** Assign \( c(G) = [k]^l \) to the root of \( T \).
- **0-NODE OPERATOR:** For \( c(A) = [r]^s \), set \( c(A_i) \) such that
  \[
  c(A_i) = [r_i]^s_i, \\
  \kappa_{s_i-1}(G_i) \geq r_i, \\
  s_1 + s_2 + \cdots + s_t = s.
  \]
- **1-NODE OPERATOR:** For \( c(A) = [r]^s \), set \( c(A_i) \) such that
  \[
  c(A_i) = [r_i]^s_i, \\
  \kappa_{s_i-1}(G_i) \geq r_i, \\
  r_1 + r_2 + \cdots + r_t = r.
  \]
- **OUTPUT:** Leaves with \( c = (1) \) inducing a box cograph \( H \) of dimension \( k \times l \).

**Proof:** [Proof of Correctness] We start by showing that the two operators are well-defined. Let \( A \) be a node that got \( c(A) = [r]^s \) assigned. By the initialization and the definition of the operators, we know that

\[
\kappa_{s-1}(A) \geq r.
\]

Suppose \( A \) is a 0-node. Then

\[
A = A_1 + A_2 + \cdots + A_t
\]

and therefore

\[
\kappa(A) = \kappa(A_1) \ast \kappa(A_2) \ast \cdots \ast \kappa(A_t).
\]
As \( \kappa_{s-1}(A) \geq r \) implies that there are at least \( s \) entries greater than or equal to \( r \) in \( \hat{k}(A) \) and since \( \hat{k}(A) \) arises from the concatenation of \( \hat{k}(A_1), \hat{k}(A_2), \ldots, \hat{k}(A_t) \), we know that we can find values \( s_1, s_2, \ldots, s_t \) such that \( \hat{k}(A_i) \) contains at least \( s_i \) entries greater than or equal to \( r \) and \( s_1 + s_2 + \cdots + s_t = s \). Therefore the 0-operator is well-defined.

If \( A \) is a 1-node, then
\[
A = A_1 \lor A_2 \lor \cdots \lor A_t
\]
and we have
\[
\hat{k}(A) = \hat{k}(A_1) + \hat{k}(A_2) + \cdots + \hat{k}(A_t),
\]
thus
\[
\kappa_{s-1}(A) = \kappa_{s-1}(A_1) + \kappa_{s-1}(A_2) + \cdots + \kappa_{s-1}(A_t).
\]
Hence we can find values \( r_1, r_2, \ldots, r_t \) such that \( \kappa_{s-1}(A_i) \geq r_i \) and \( r_1 + r_2 + \cdots + r_t = r \). Therefore the 1-operator is well-defined.

To show that the vertices with \( c = (1) \) induce a box cograph of dimension \( k \) times \( l \), we first note that for any 0-node, the sum over the entries of \( c(A) \) is
\[
rs = (r_1 + r_2 + \cdots + r_t)s = r_1s + r_2s + \cdots + r_ts,
\]
which is the sum over all entries of all \( c(A_i) \). The same holds for 1-nodes. Hence the sum over all entries of all sequences assigned to the leaves equals the sum over the entries of the sequence assigned to the root, which is \( kl \). As the only possible assignments to the leaves are \( (1) \) and the empty sequence \( () \), we must have \( kl \) leaves with \( (1) \) assigned to them. It suffices to show that the graph \( H \) induced by these leaves satisfies \( \hat{k}(H) = [k]^l \). To do so, we apply KAPPA to \( T \), where we initialize the leaves by the arguments assigned to them by this algorithm. The operators of KAPPA are inverses to the ones of BOX COGRAPH. Therefore the output of KAPPA will be \( c(G) = [k]^l \). It follows that \( \hat{k}(H) = [k]^l \) and \( H \) is a box cograph of dimension \( k \) times \( l \).

An example of the output of BOX COGRAPH is shown in Figure 8, where \( \hat{k} \) is shown in round brackets to the left of each node and \( c \) in square brackets to the right of each node. For the leaves, \( \hat{k} \) and \( c \) have been omitted, except when \( c = [1] \).

![Figure 8: BOX COGRAPH with \( k = 4 \) and \( l = 2 \) for the cotree from Figure 5](image)

As a final algorithm, we present a bottom-up algorithm, calculating the Ferrers diagram representation of a cograph.
Figure 9: FERRERS DIAGRAM for the complement of the cotree from Figure 5.

Algorithm 4.3. (FERRERS DIAGRAM)

- **INPUT:** Cotree $T$ of a cograph $G$.
- **INITIALIZATION:** Assign $Y = \bullet$ to each leaf of $T$, labelled with the name of the leaf.
- **0-NODE OPERATOR:** $\mathcal{Y}(A)$ is the Ferrers diagram representation consisting of the columns of the Ferrers diagram representations $\mathcal{Y}(A_1), \mathcal{Y}(A_2), \ldots, \mathcal{Y}(A_t)$, sorted by size.
- **1-NODE OPERATOR:** $\mathcal{Y}(A)$ is the Ferrers diagram representation consisting of the rows of the Ferrers diagram representations $\mathcal{Y}(A_1), \mathcal{Y}(A_2), \ldots, \mathcal{Y}(A_t)$, sorted by size.
- **OUTPUT:** Ferrers diagram representation of $G$.

**Proof:** [Proof of Correctness] The correctness follows directly from the proof of Theorem 2.9.

An example of an output of FERRERS DIAGRAM is given in Figure 9. The cotree is the one from Figure 5 except that the labels of the internal nodes have been swapped for visual reasons (the graph represented by this cotree is the complement of the one from Figure 6).

Algorithm FERRERS DIAGRAM can be implemented in time $O(n \log n)$ similarly to Algorithm KAPPA. We again describe the implementation on a pseudocotree. To store the array of vertices in the Ferrers diagram representation, we keep track of the right neighbour and bottom neighbour (in the representation not the graph) for each point. We also calculate $\hat{\kappa}$ and $\hat{\lambda}$ and the order of the induced subgraph at each node of the
cotree. Suppose that $A$ is a node in the input pseudocotree and $A_1$ and $A_2$ are the two children of $A$. We claim that we can calculate the Ferrers diagram representation of $A$ from the Ferrers diagrams representation of $A_1$ and $A_2$ in $O(\min\{|A_1|, |A_2|\})$, implying that the algorithm can be implemented to run in time $O(n \log n)$. Assume without loss of generality $a = |A_1| \leq |A_2|$. If $A$ is a 0-node we insert the columns of the Ferrers diagram representation of $A_1$ into the Ferrers diagram representation of $A_2$. To find the locations for inserting the columns requires $O(a) \hat{}$ steps (using $\hat{k}$). For each column we insert we need to update the right neighbours of its own vertices and of the vertices of the new column to its left, again $O(a)$ \hat{} steps. Similarly, if $A$ is a 1-node we can insert the rows of the Ferrers diagram representation of $A_1$ into that of $A_2$ in $O(a)$ \hat{} steps, using $\hat{\lambda}$ to find the locations of insertion.

From the Ferrers diagram representation we can easily determine, whether a cograph is $(k, l)$-colourable and find an box cograph obstruction if not. For example, the graph in Figure 9 is not $(1, 3)$-colourable. A box cograph of dimension 2 times 4 is induced by the vertices $a, f, h, i, d, g, j, k$.

In this paper, we proved that the number of vertices in any cograph $G$ is equal to the sum of entries in either of $\hat{k}(G)$ or $\hat{\lambda}(G)$. We have seen examples (e.g., $P_4$ and $C_5$) for which this property does not hold. It would be interesting to characterize all (perfect) graphs for which this property holds.

**Acknowledgement** We would like to thank the anonymous referees for their helpful comments on the paper and suggestion of the above stated problem.

**References**

[1] N. Alon and U. Stav, *The maximum edit distance from hereditary graph properties*, J. Combin. Theory Ser. B 98 (2008), no. 4, 672–697.

[2] M. Axenovich, A. Kézdy, and R. Martin, *On the editing distance of graphs*, J. Graph Theory 58 (2008), no. 2, 123–138.

[3] B. Bollobás and A. Thomason, *Hereditary and monotone properties of graphs*, Algorithms Combin. 14 (1997), 70–78.

[4] A. Brandstädt, *Partitions of graphs into one or two independent sets and cliques*, Discrete Math. 152 (1996), no. 1-3, 47–54.

[5] A. Brandstädt, *Corrigendum: Partitions of graphs into one or two independent sets and cliques*, Discrete Math. 186 (1998), no. 1-3, 295.

[6] D.G. Corneil, H. Lerchs, and L.K. Stewart Burlingham, *Complement reducible graphs*, Discrete Appl. Math 3 (1981), 163–174.

[7] D.G. Corneil, Y. Perl, and L.K. Stewart, *A linear recognition algorithm for cographs*, SIAM J. Comput. 14 (1985), 926–934.
[8] M. Demange, T. Ekim, and D. de Werra, *Partitioning cographs into cliques and stable sets*, Discrete Optim. 2 (2005), 145–153.

[9] M. Demange, T. Ekim, and D. de Werra, *(p,k)-coloring problems in line graphs*, Theoret. Comput. Sci. 349 (2005), no. 3, 462–474.

[10] D. Epple, *Graph paritions and the bichromatic number*, PhD Thesis, University of Victoria, 2011.

[11] D.D.A. Epple and J. Huang, *A note on the bichromatic number of graphs*, J. Graph Theory 65 (2010), no. 2, 263–269.

[12] P. Erdős and A.H. Stone, *On the structure of linear graphs*, Bull. Amer. Math. Soc. 52 (1946), 1087–1091.

[13] T. Feder, P. Hell, and W. Hochstättler, *Generalized colourings (matrix partitions) of cographs*, Graph Theory in Paris, Trends Math., Birkhäuser, Basel, 2007, pp. 149–167.

[14] S. Földes and P.L. Hammer, *Split graphs having Dilworth number two*, Canad. J. Math. 29 (1977), no. 3, 666–672.

[15] J. Gimbel, D. Kratsch, and L. Stewart, *On cocolourings and cochromatic numbers of graphs*, Discrete Appl. Math. 48 (1994), 111–127.

[16] P. Hell, S. Klein, L.T. Nogueira, and F. Protti, *Partitioning chordal graphs into independent sets and cliques*, Discrete Appl. Math. 141 (2004), no. 1-3, 185–194.

[17] L. Lesniak-Foster and H.J. Straight, *The cochromatic number of a graph*, Ars Combin. 3 (1977), 39–46.

[18] H.J. Prömel and A. Steger, *Excluding induced subgraphs II: Extremal graphs*, Discrete Appl. Math. 44 (1993), 283–294.

[19] P. Turán, *On an extremal problem in graph theory*, Mat. Fiz. Lapok 48 (1941), 436–452 (in Hungarian).