AARHUS UNIVERSITY

Coversheet

This is the accepted manuscript (post-print version) of the article.

Contentwise, the post-print version is identical to the final published version, but there may be differences in typography and layout.

How to cite this publication

Please cite the final published version:

Demange M., Olsen M. (2018) A Note on Online Colouring Problems in Overlap Graphs and Their Complements. In: Rahman M., Sung WK., Uehara R. (eds) WALCOM: Algorithms and Computation. WALCOM 2018. Lecture Notes in Computer Science, vol 10755. Springer, Cham

Publication metadata

Title:	<i>A Note on Online Colouring Problems in Overlap Graphs and Their Complements</i>
Author(s):	<i>Demange M., Olsen M.</i>
Journal:	<i>WALCOM: Algorithms and Computation. WALCOM 2018. Lecture Notes in Computer Science</i>
DOI/Link:	<i>https://doi.org/10.1007/978-3-319-75172-6_13</i>
Document version:	<i>Accepted manuscript (post-print)</i>

General Rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Note on Online Colouring Problems in Overlap Graphs and their Complements

Marc Demange¹ and Martin Olsen^{2*}

¹ RMIT University, School of Science, Melbourne, Australia
`marc.demange@rmit.edu.au`

² BTECH, Aarhus University, Denmark `martino@btech.au.dk`

Abstract. We consider online versions of different colouring problems in interval overlap graphs, motivated by stacking problems. An instance is a system of time intervals presented in non-decreasing order of the left endpoints. We consider the usual colouring problem as well as b -bounded colouring and the same problems in the complement graph. We also consider the case where at most b intervals of the same colour can include the same element. For these versions, we obtain a logarithmic competitive ratio with respect to the maximum ratio of interval lengths. The best known ratio for the usual colouring was linear, and to our knowledge other variants have not been considered. Moreover, pre-processing allows us to deduce approximation results in the offline case. Our method is based on a partition of the overlap graph into permutation graphs, leading to a competitive-preserving reduction of the problem in overlap graphs to the same problem in permutation graphs. This new partition problem by itself is of interest for future work.

1 Introduction

The problem we consider is originally motivated by stacking problems (see, e.g., [1, 11]). After unloading a ship, containers are stored in the port before being uploaded again on another ship or truck. Due to space and logistics constraints, stored containers are stacked such that the container needing to leave first is preferably on the top of a stack, and in this work, we will consider the *perfect case* where this constraint is always satisfied. The problem is then to arrange the containers so as to minimise the space used on the floor. Additional constraints, like a maximum height of the stacks, are also natural. In a real context, the arrival time of ships and the final destination of their containers are not known or partially known in advance and one needs to stack the newly arrived containers without knowing details about the future ones. This constraint motivates the *online* version [14], where containers are supposed to arrive in any order and are allocated to stacks without taking into account future containers. A similar online model motivated by track assignment problems in railway optimisation is studied in [3, 4] with a constraint on the number of trains per track. In this

* corresponding author.

context, another natural restriction is the *midnight condition* that states that all trains are at the station at some time.

The stacking problem (in the perfect case) and the track assignment problem can be modelled as a colouring problem³ in a specific class of graphs, called *overlap graphs*, representing incompatibilities between containers in a single stack (resp., trains on a track). Vertices correspond to the set R of time intervals during which containers are stored in the port and two vertices are linked if the related intervals *overlap*, i.e., they intersect but none is contained in the other. In this case, the related containers need to be in different stacks. R is referred to as an *interval system* and all intervals are assumed to have non negative left endpoints. If all intervals in a system intersect, the related graph is a *permutation graph* [6, 7], formally defined later. So, our work deals with *online colouring of overlap graphs and permutation graphs*. The cases where stacks have a maximum height and tracks have a maximum capacity motivate variants of graph colouring, precisely the *b-bounded load colouring* defined later and the *b-bounded colouring*, where colour classes cannot exceed the size b . In a theoretical perspective we address a generalisation, called *H-colouring*, for a hereditary property H ; it includes in particular *(b-bounded) clique covering* corresponding to *(b-bounded) colouring* in the complementary graph. Main definitions are given in the next section and for all graph notions not defined here, the reader is referred to [7].

Related work

For the offline version, 4-colourability has been shown NP-complete in overlap graphs [16]. Colouring permutation graphs can be done in polynomial time, even in an online set-up using a First-Fit strategy if the permutation is presented from left to right, as explained in the next section. However, the *b-bounded colouring* problem is NP-hard in permutation graphs for $b \geq 6$ [9]. Clique covering is also known to be hard in overlap graphs [10] and other NP-hard versions of stacking problems are considered in [1, 11].

To our knowledge, the best known polynomial approximation for colouring overlap graphs is a $(\log n)$ -approximation [2], where n is the number of vertices. It applies to the larger class of *subtree filament* graphs [10]. A 2-approximation in overlap graphs is claimed in [16], but it is contradicted in [11]. For clique covering overlap graphs, a $(\log \beta(G))$ -approximation is proposed in [10], where $\beta(G)$ denotes the optimal value for the graph G . This is improved in [15]: for an instance G_R defined by an interval system R , a $2(1 + \log \tilde{\alpha}(R))$ -approximation is proposed, where $\tilde{\alpha}(R)$ is the maximum number of pairwise disjoint intervals. If $\alpha(G_R)$ is the *independence number* of G_R , we have $\tilde{\alpha}(R) \leq \alpha(G_R) \leq \beta(G_R)$.

Online colouring of overlap graphs has mainly been considered in [4], where a $O\left(\frac{L}{\ell}\right)$ -competitive algorithm is proposed if intervals are processed from *left to right*⁴, L and ℓ being the largest and smallest interval lengths, respectively. It is improved into $O\left(\frac{\log^2 L}{\log \log L}\right)$ if ℓ is known in advance. Online colouring of

³ Colour vertices using the minimum number of colours such that adjacent vertices have different colours.

⁴ i.e., in non-decreasing order of their left endpoint

permutation graphs, however, has been widely studied (see, e.g. [13]). In [4], online b -bounded colouring is considered in permutation graphs and a $(2 - 1/\sigma)$ -competitive algorithm is proposed, where σ is the minimum between b and the offline optimal value; moreover it is the best possible competitive ratio. For overlap graphs processed from left to right, it is shown ([4]) that a constant competitive ratio cannot be achieved. The proof can easily be transformed to show that a $K(\log \log n)$ -competitive ratio cannot be guaranteed for some constant K .

Our contribution

Here, we propose a $O(\log(\frac{L}{\ell}))$ -competitive online algorithm when intervals are processed from left to right; so an improvement from a linear to a logarithmic ratio and moreover, the result also holds for other H -colouring problems like clique covering, b -bounded colouring and b -bounded clique covering (Th. 2) as well as for b -bounded load colouring (Th. 3). These results are obtained using a partition of an overlap graph into permutation graphs, a new problem interesting by itself. It leads to a *competitive-preserving reduction* transforming an online algorithm in permutation graphs into an online algorithm in overlap graphs with a competitive ratio increased by a logarithmic factor (Th. 1). This narrows the gap between the best competitive ratio and the best lower bound.

Even though we mainly focus on the online version, we conclude by showing how to use our online algorithms for the offline cases. We derive a $O(\log \tilde{\alpha}(R))$ -approximation for several colouring problems (Th. 4). It slightly improves the best known approximation for colouring and clique covering in overlap graphs. Moreover, the same holds for the bounded versions, which constitutes, to our knowledge, the first approximations for these problems. It is worth noting that these approximation results are derived from online algorithms. Since a constant competitive ratio cannot be achieved, it highlights the question - left open - whether offline colouring is constant approximable in overlap graphs.

2 Preliminaries

$\mathbb{N}$, $\mathbb{N}^*$ and $\mathbb{Z}$ will denote the set of natural numbers, the set of positive natural numbers and the set of integers, respectively. In a graph G an *independent set* is a set of pairwise non-adjacent vertices while a *clique* is an independent set in the complement $\overline{G}$ of G . The *independence number* and the *clique number* of G denote the maximum size of an independent set and of a clique in G , respectively.

Overlap and permutation graphs

All real intervals are assumed to have non-negative left endpoints and $]a, b[$ and $]c, d[$ *overlap* if $a < c < b < d$. Given a system R of open intervals, the related *overlap graph*, $G_R = (R, E_R)$, is defined such that adjacent vertices correspond to overlapping intervals [6]. The *interval graph* associated with R has the same vertex set but two vertices are linked if the related intervals intersect. We refer to the clique number of this interval graph as the *load* of R , and we denote by $\tilde{\alpha}(R)$ its independence number.

Permutation graphs are usually defined from a permutation $\pi = (\pi_1, \dots, \pi_n)$. The vertex set of G_π is $\{\pi_1, \dots, \pi_n\}$ with an edge between π_i and π_j if $i < j$ and

Fig. 1: BBQ arrangement B of intervals (above) and the related permutation graph (below) associated with the permutation $\pi_B = (2, 3, 1, 5, 4, 6, 8, 10, 9, 7)$ or equivalently, for instance, the list $Q_B = (0.5, 1.5, 0, 2.5, 2.1, 3.5, 4.5, 5.5, 5.3, 4.1)$.

$\pi_i > \pi_j$ (see Figure 1); we equivalently define G_Q from any list $Q = (q_1, \dots, q_n)$ of pairwise distinct numbers. It is a well known subclass of overlap graphs: in particular the overlap graph induced by a system of pairwise intersecting intervals (midnight condition) is a permutation graph (see, e.g. [4]). We call such a set of intersecting intervals a *brochette of intervals* and two brochettes are *independent* if every two intervals in different brochettes are disjoint; a set of pairwise independent brochettes is called a *BBQ arrangement*. Given a BBQ arrangement B with k brochettes $B_1, \dots, B_k$, we denote by $G_{B_i} = (V_i, E_i)$ the permutation graph associated with B_i . The overlap graph G_B associated with B is then the disjoint union of $G_1, \dots, G_k$, which is clearly a permutation graph:

Proposition 1. *BBQ arrangements always induce permutation graphs.*

Figure 1 gives an example of a BBQ arrangement B with the related permutation graph G_B , the corresponding permutation π and an example of a list Q_B satisfying $G_B = G_{Q_B}$ if we rename by q_i the vertex π_i , for $i = 1, \dots, n$. Q_B is chosen without positive integral values to avoid any confusion with values in π .

H -colouring problems and their online version

Graph colouring and its generalisations are extensively studied, in particular for their applications in scheduling. In the Minimum H -colouring problem, colour classes are constrained to satisfy a fixed hereditary property H . For overlap graphs, defined by an interval system R , suppose we are given a hereditary property H on interval systems, i.e., whenever a system satisfies H , so does any subsystem. Then, the Minimum H -colouring problem is to partition any interval system R into a minimum number of colour classes that all satisfy H . Table 1 lists the H -colouring problems we consider here and the related properties H .

Problem	Property H satisfied by each colour class $R' \subset R$
colouring	$G_{R'}$ is an independent set (no overlap) .
b -bounded colouring	$G_{R'}$ is an independent set and $ R' \leq b$.
clique covering	$G_{R'}$ is a clique (every two intervals overlap).
b -bounded clique covering	$G_{R'}$ is a clique and $ R' \leq b$.
b -bounded load colouring	$G_{R'}$ is an independent set and R' is of load at most b .

Table 1: Examples of H -colouring problems in an overlap graph defined by R .

Other examples of H -colouring include cocolouring [12] or split-colouring [5]. In the online version of colouring problems, vertices are presented one by one and each time a new vertex is presented, one needs to irrevocably decide its colour. If we number the vertices $v_1, \dots, v_n$ in the order of presentation, an online colouring algorithm decides the colour of vertex v_k using only the structure of the subgraph induced by vertices $v_1, \dots, v_k$ and the colours of vertices $v_1, \dots, v_{k-1}$.

It is sometimes relevant to impose a specific order of presentation for the vertices. In this work, the instance is an overlap graph defined by a system of intervals and we assume that intervals are presented in non-decreasing order of their left endpoints, called the *left to right* order. For instance, in Figure 1, the labelling of intervals, written from left to right, corresponds to the related permutation π_B . Note that π_B cannot be performed online but an equivalent list of numbers such as Q_B can.

Consequently, when considering a permutation graph defined by a BBQ arrangement presented from left to right, one can use any online algorithm using a list of numbers presented from left to right. In this case in particular, the greedy *First Fit algorithm*, that assigns the first possible colour to the new presented vertex, determines an optimal colouring (see, e.g., [4]).

The quality of an online algorithm is characterised by the *competitive ratio*. For a colouring problem, we denote by $\beta(G)$ the optimal value for G : it is the chromatic number $\chi(G)$ of G for the usual colouring problem, the bounded chromatic number $\chi_b(G)$ for the b -bounded colouring problem, or more generally $\chi_H(G)$ for the H -colouring problem. Given an online algorithm A for H -colouring, we denote by $\lambda(G, \mathcal{O})$ the value of the online solution computed for G presented in the order $\mathcal{O}$. A is said to guarantee the competitive ratio of ρ if we have $\lambda(G, \mathcal{O}) \leq \rho \times \beta(G)$ for any graph G and order $\mathcal{O}$ within the considered model. Since we only consider the order from left to right, we will omit the parameter $\mathcal{O}$.

Even if it is not systematically considered in the literature, it is relevant to analyse the complexity of online algorithms. An online algorithm is said to be *polynomial* if each step can be performed in polynomial time with respect to the number of already presented vertices. In the offline case, the approximation ratio of a polynomial algorithm is defined in a similar way as the competitive ratio.

3 Partitioning an overlap graph into permutation graphs

Our general strategy is to partition an overlap graph into permutation subgraphs: given an interval system R , we partition it into BBQ arrangements, called *clusters*, $\mathcal{P}_k$, $k \in B \subset \mathbb{N}$, each inducing a permutation graph. As we will show, this can be done online when intervals are presented from left to right.

Note then that each cluster is presented from left to right, which will allow us to apply known online algorithms for permutation graphs on each part (See Section 4). Even though the complement of an overlap graph is not always an overlap graph, the fact that the class of permutation graphs is stable by complementation will allow us to derive similar results for clique covering.

To help understanding the main idea, we first describe a (offline) decomposition that requires the lengths L and ℓ . Then, in Proposition 2, we show that this decomposition can be performed online, and we also show how to handle the case where L is not known in advance (ℓ will never be requested).

The offline decomposition:

Without loss of generality, we assume that the left endpoint of the first interval is 0. The decomposition works like a sieve based on nested discrete sets $\mathcal{S}_i$ defined as follows. Let λ be the first interval length. We define $k_L = \lceil \log_2(\frac{L}{\lambda}) \rceil + 1$ and $k_\ell = \lfloor \log_2(\frac{\lambda}{\ell}) \rfloor + 1$. For $-k_L \leq i \leq k_\ell$, we set $\mathcal{S}_i = \{k2^{-i}\lambda, k \in \mathbb{N}^*\}$ and $\mathcal{S}_i = \emptyset$ for $i < -k_L$ or $i > k_\ell$. Note that $\mathcal{S}_{-k_L} \subset \dots \subset \mathcal{S}_0 \subset \dots \subset \mathcal{S}_{k_\ell}$.

We see sets $\mathcal{S}_i$ as positions of vertical spikes of various lengths, as illustrated in Figure 2. The spikes in $\mathcal{S}_{-k_L}$ are the longest ones and $\mathcal{S}_{i+1}$ is obtained by adding a shorter spike between every two spikes in $\mathcal{S}_i$. Using the BBQ metaphor, we see the full instance as a steak sliced into pieces of meat (intervals) to be dropped and skewered on the first (longest) spike corresponding to an element of $\mathcal{S}_i$ for some i . The cluster $\mathcal{P}_i$ is defined as the intervals skewed at level i : it is the set of intervals that intersect $\mathcal{S}_i$ but do not intersect $\mathcal{S}_{i-1}$. Since $k_\ell > \log_2(\frac{\lambda}{\ell})$, we have for every $I \in R$, $I \cap \mathcal{S}_{k_\ell} \neq \emptyset$. So, no interval is left outside clusters.

We claim that each cluster $\mathcal{P}_i$ can be partitioned into independent brochettes, thus is a BBQ arrangement, and consequently, the graph associated with $\mathcal{P}_i$ is a permutation graph. Denoting $I =]a_I, b_I[$, we have indeed:

$$\forall I \in R : I \in \mathcal{P}_j \Leftrightarrow j = \min \left\{ i \in \{-k_L, \dots, k_\ell\} : \left\lceil \frac{2^i a_I}{\lambda} \right\rceil \leq \left\lfloor \frac{2^i b_I}{\lambda} \right\rfloor \right\} \quad (1)$$

If $j > -k_L$, Equation (1) means that the interval $\left[\frac{2^j a_I}{\lambda}; \frac{2^j b_I}{\lambda} \right]$ includes an integer while interval $\left[\frac{2^{j-1} a_I}{\lambda}; \frac{2^{j-1} b_I}{\lambda} \right]$ does not; if $j = -k_L$, then $\left[\frac{2^{-k_L} a_I}{\lambda}; \frac{2^{-k_L} b_I}{\lambda} \right]$ includes an integer. Note that, for $i = -k_L$, the distance $2^{k_L} \lambda$ between two distinct elements of $\mathcal{S}_{-k_L}$ is at least $2L$. Consequently,

$$\forall x_1, x_2 \in \mathcal{S}_i, I_1, I_2 \in \mathcal{P}_i : [x_1 \in I_1 \wedge x_2 \in I_2 \wedge x_1 \neq x_2] \Rightarrow I_1 \cap I_2 = \emptyset,$$

which proves the claim. A brochette in $\mathcal{P}_i$ corresponds to an element $s \in \mathcal{S}_i \setminus \mathcal{S}_{i-1}$; in this case, (s, i) (called a *skew*) is said to be *active*.

Since $\mathcal{S}_i = \emptyset$ for $i < -k_L$ or $i > k_\ell$, there are at most $(k_L + k_\ell + 1) \leq (\lfloor \log_2(\frac{L}{\ell}) \rfloor + 4)$ permutation graphs in the decomposition.

We know show that this decomposition can be performed online. Intervals are considered from left to right and Figure 2 corresponds to the time $t = t_0$.

Proposition 2. (*Partition into permutation graphs*)

Algorithm 1 described below is a polynomial online algorithm that partitions an overlap graph defined by an interval system presented from left to right into at most $(2 \lfloor \log_2(\frac{L}{\ell}) \rfloor + 7)$ permutation graphs defined by a BBQ arrangement. Moreover, if L is known in advance, a simplified version of the algorithm guarantees a number of at most $(\lfloor \log_2(\frac{L}{\ell}) \rfloor + 4)$ permutation graphs in the decomposition.

Fig. 2: The steak is sliced into intervals to be dropped on several layers $\mathcal{P}_i$ of brochettes. Active skewes are thick, in particular $(s, 1)$, $(s, 0)$ and $(s, -1)$ are inactive if $s \in \mathcal{S}_{-2}$. If the distance between the brochettes in the top layer is at least $2L$ then each $\mathcal{P}_i$ is a BBQ arrangement.

Proof. As soon as a new interval is presented it will be assigned to a cluster corresponding to a BBQ arrangement. During this process a cluster will be called *open* if it is non-empty. If we decide to assign an interval to a not yet open cluster then we will open a new one for this interval. Note that, for a given L, ℓ, I , if the cluster $\mathcal{P}_j$ of I is determined by Relation (1), then j 's value does not change if ℓ is replaced by a smaller value $\tilde{\ell}$. If L is replaced by a larger value $\tilde{L}$ however, then j may change only if $j = -k_L$. Consequently, this decomposition can directly be performed online if L is known in advance with a maximum number of $(\lceil \log_2(\frac{L}{\ell}) \rceil + 4)$ BBQ-arrangements in the decomposition. In this case, the algorithm is simplified since the clusters $\mathcal{R}$ are not required (the simplified version of the algorithm can be easily derived).

If L is not known in advance, we need to make a specific treatment for any interval I_0 placed in $\mathcal{P}_{-k_{L_0}}$ where L_0 is the largest known interval length when I_0 is revealed. Algorithm 1 shows how to perform the partition dynamically. To handle the above mentioned problem, it creates two kinds of clusters, $\mathcal{Q}$ and $\mathcal{T}$. Roughly speaking a cluster $\mathcal{P}_{-k_{L_0}}$ in the previous decomposition might be doubled if L_0 is the largest known interval length when the cluster is open.

Line 6 ensures that, at any step of the online process, ℓ and L are respectively equal to the minimum and the maximum length of already revealed intervals. Once a set $\mathcal{S}_i$ is defined as non-empty, its definition will not change. New non-empty sets $\mathcal{S}_i$ are added when the values of L or ℓ are updated. Lines 10 and 12 determine in which cluster the interval should be put.

Consider first a cluster $\mathcal{T}_{j_0}$ for some j_0 . Intervals in $\mathcal{T}_{j_0}$ are assigned at lines 12, which means that $j_0 > -k_L$ each time an interval has been assigned to it. For any such interval $]a_I; b_I[$ the interval $\left[\frac{2^{j_0-1}a_I}{\lambda}; \frac{2^{j_0-1}b_I}{\lambda} \right]$ does not include any integer number, and consequently, neither do the intervals $\left[\frac{2^h a_I}{\lambda}; \frac{2^h b_I}{\lambda} \right]$ for $h < j_0$. Thus, Relation (1) will hold for any interval $I \in \mathcal{T}_{j_0}$ after the Algorithm stops and the previous analysis shows that $\mathcal{T}_{j_0}$ is a BBQ arrangement of intervals.

Consider now a cluster $\mathcal{Q}_{j_0}$; intervals have been assigned in this cluster at lines 10. Consider the last interval $I \in \mathcal{Q}_{j_0}$ revealed at the online step n_0 : denoting by L_0 the longest length of intervals revealed at steps $i \leq n_0$ we have $j_0 = -k_{L_0}$. All intervals in $\mathcal{Q}_{j_0}$ are of length at most L_0 and intersect $\mathcal{S}_{j_0}$ (see lines 8 and 10). The distance between any two elements in $\mathcal{S}_{j_0}$ is at least $2^{-j_0}\lambda \geq 2L_0$. Since all intervals are open we deduce that $\mathcal{Q}_{j_0}$ is a BBQ arrangement of intervals. All in all, the original overlap graph defined by an interval system is partitioned online into at most $(2k_L + k_\ell + 1) \leq (2 \lfloor \log_2(\frac{L}{\ell}) \rfloor + 7)$ BBQ arrangements of intervals, each revealed from left to right. Note finally that, at each step of the online process (new presented interval), the algorithm requires a linear complexity to decide in which cluster the interval should be added. It concludes the proof.

Algorithm 1 Partition into Permutation Graphs

Require: An overlap graph $G = (R, E_R)$ presented online from left to right (the maximum length L is not known in advance).

Ensure: A partition of R , $(\mathcal{P}_1, \dots, \mathcal{P}_p)$ such that $G[\mathcal{P}_i]$ is a permutation graph.

```

1:  $\mathcal{T}_i \leftarrow \emptyset, i \in \mathbb{Z}$ 
2:  $\mathcal{Q}_i \leftarrow \emptyset, i \in \mathbb{Z}$ 
3:  $\ell \leftarrow \lambda, L \leftarrow \lambda$ 
4: When the first interval  $I$  is presented, set  $\lambda$  as its length and add  $I$  to  $\mathcal{T}_0$ 
5: for each new interval  $I = ]a_I, b_I[$  do
6:    $\ell \leftarrow \min\{b_I - a_I, \ell\}, L \leftarrow \max\{b_I - a_I, L\}$ 
7:    $k_L \leftarrow \lceil \log_2(\frac{L}{\lambda}) \rceil + 1, k_\ell \leftarrow \lfloor \log_2(\frac{\lambda}{\ell}) \rfloor + 1$ 
8:    $j \leftarrow \min\left\{i \in \{-k_L, \dots, k_\ell\} : \left\lceil \frac{2^i a_I}{\lambda} \right\rceil \leq \left\lfloor \frac{2^i b_I}{\lambda} \right\rfloor\right\}$ 
9:   if  $j = -k_L$  then
10:    Add  $I$  to  $\mathcal{Q}_j$ 
11:   else
12:    Add  $I$  to  $\mathcal{T}_j$ 
13:   end if
14: end for
15: The final partition is  $(\mathcal{T}_{-k_L+1}, \dots, \mathcal{T}_{k_\ell}) \cup (\mathcal{Q}_{-k_L}, \dots, \mathcal{Q}_0)$ 

```

4 Competitiveness through partitioning

Using Proposition 2, we reduce a large class of online H -colouring problems on overlap graphs to the same problem on permutation graphs:

Theorem 1. (*Online reduction*)

For any online algorithm for a H -Colouring problem guaranteeing a competitive ratio of ρ on permutation graphs defined by a BBQ arrangement presented from left to right, there is an online algorithm for the same problem on overlap graphs defined by an interval system presented from left to right guaranteeing the competitive ratio $(2 \lfloor \log_2(\frac{L}{\ell}) \rfloor + 7) \rho$. If L is known in advance the ratio is $(\lfloor \log_2(\frac{L}{\ell}) \rfloor + 4) \rho$.

Moreover, if the former online algorithm is polynomial, then the latter is polynomial as well.

Proof. We can partition online the overlap graph into $(2 \lfloor \log_2(\frac{L}{\ell}) \rfloor + 7)$ permutation graphs defined by a BBQ arrangement using Proposition 2. If L is known in advance, then the number of parts is reduced to $(\lfloor \log_2(\frac{L}{\ell}) \rfloor + 4)$.

Note that each BBQ arrangement in the decomposition is presented from left to right. This allows us to apply the online algorithm for permutation graphs on each BBQ arrangement using different colour sets. This defines a feasible H -colouring of the whole graph. Since the H -chromatic number is not increasing from a graph to a subgraph, the result immediately follows. Note finally that, since Algorithm 1 requires at each step a linear complexity, this reduction transforms a polynomial online algorithm for permutation graphs into a polynomial online algorithm in overlap graphs, which concludes the proof.

Combining this reduction and known competitive algorithms in permutation graphs presented from left to right leads to the following result:

Theorem 2. *The following online competitive ratios can be guaranteed by a polynomial online algorithm in overlap graphs defined by an interval system presented from left to right:*

1. $2 \lfloor \log_2(\frac{L}{\ell}) \rfloor + 7$ for (unbounded) colouring and clique covering;
2. $(2 \lfloor \log_2(\frac{L}{\ell}) \rfloor + 7) \left(2 - \frac{1}{\min\{b, \beta(G_R)\}}\right)$ for b -bounded colouring and b -bounded clique covering, where $\beta(G_R)$ represents the offline optimal value in G_R .

Proof. Given a BBQ arrangement B presented from left to right, one can perform online a list Q_B defining the permutation graph G_{Q_B} (see Figure 1). Moreover, the list Q_B^{-1} obtained by changing all signs in Q_B , also performed online, defines the complement graph $G_{Q_B^{-1}} = \overline{G_{Q_B}}$. Consequently, all results applying to online colouring permutation graphs also apply to online clique covering.

(1) For the usual colouring problem, the greedy algorithm First Fit is an online exact algorithm (1-competitive) for permutation graphs defined by a BBQ arrangement presented from left to right. Using the previous remark it gives as well an exact online algorithm for clique covering. We conclude using Theorem 1.

(2) For the online b -bounded colouring problem in a permutation graph G_Q associated with a list Q presented from left to right, an online algorithm guaranteeing the competitive ratio $2 - 1/\sigma$ is proposed in [4], where $\sigma = \min\{b, \chi_b(G_Q)\}$. We conclude the case of b -bounded colouring by noting that, if G_Q is a subgraph of an overlap graph G , we have: $2 - \frac{1}{\min\{b, \chi_b(G_Q)\}} \leq 2 - \frac{1}{\min\{b, \chi_b(G)\}}$. The same holds in the complementary graph $G_{Q_B^{-1}} = \overline{G_{Q_B}}$, which concludes the proof.

To our knowledge this is the first non-obvious competitive analysis for online b -bounded colouring of overlap graphs and their complements. Note finally that Theorem 1 can be applied for a problem defined directly on an interval system R like, for example, b -bounded load colouring.

Consider a BBQ arrangement B processed from left to right, the brochettes $B_i, i = 1, \dots, k$, can be determined online. An online solution for b -bounded load colouring can be performed using the algorithm in [4] on each brochette, and

then, reusing the same colour set when passing to the next brochette. The overall number of colours is the maximum number of colours used in a single brochette and the same holds for a BBQ arrangement B : $\beta(B) = \max_{i=1, \dots, k} (\chi_b(G_{B_i}))$. Since $\min(b, \chi_b(G_{B_i})) \leq \min(b, \beta(B))$, for all $i = 1, \dots, k$, the algorithm in [4] gives a $\left(2 - \frac{1}{\min(b, \beta(B))}\right)$ -competitive algorithm for b -bounded load colouring in a BBQ arrangement. Using the same analysis as in Theorem 2-(2), we have:

Theorem 3. *There is a polynomial online algorithm for b -bounded load colouring in an interval system R presented from left to right with competitive ratio $(2 \lfloor \log_2(\frac{L}{\ell}) \rfloor + 7) \left(2 - \frac{1}{\min\{b, \beta(R)\}}\right)$, where $\beta(R)$ represents the related offline optimal value in R .*

5 Approximation of offline variants

We finally show how combining our online results with a preprocessing step gives new approximation results for various colouring problems in overlap graphs.

Proposition 3. *Given an interval system R of size n and $\varepsilon > 0$, we can modify R in $O(n \log n)$ -time, preserving the relative position of intervals (containment, overlapping and disjoint relation), so that in the new system R' , the maximum length $L(R')$ and the minimum length $\ell(R')$ satisfy $\frac{L(R')}{\ell(R')} \leq (2 + \varepsilon)\tilde{\alpha}(R)$.*

Proof. Given R , assume w.l.g. that the $2n$ endpoints are all distinct and sort them in increasing order as in [8]. It requires $O(n \log n)$ -time and since then the transformation is linear. The idea is to stretch and compress intervals without changing their relative position. Each time we move an endpoint of an interval, we may move accordingly other endpoints, keeping their relative order and a positive distance between two consecutive end-points.

Let $1 > \varepsilon > 0$ and define $\beta \leq \frac{\varepsilon}{\varepsilon+4}$. Now pick in polynomial time a set S with $\tilde{\alpha}(R)$ pairwise disjoint intervals; the complexity is $O(n)$ once the endpoints are sorted [8]. Stretch or compress intervals in S so that they all have length 1 and rearrange them such that the distance between the right endpoint of an interval and the left endpoint of the consecutive interval is β . We may move accordingly other endpoints of intervals in order to keep the same relative order is R .

Number intervals in S from left to right, $S = \{s_1, \dots, s_{|S|}\}$ with $|S| = \tilde{\alpha}(R)$ and denote by A the left endpoint of s_1 and by B the right endpoint of $s_{|S|}$. We now define three sub-intervals L_i , R_i and M_i for each $s_i \in S$, each of length β . L_i shares the left endpoint with s_i , R_i shares the right endpoint with s_i and M_i is on the middle of s_i . In other words, L_i and R_i are small zones in the left and right part of s_i , respectively, while M_i is a small zone in the middle. Since $\beta < \frac{1}{3}$, these three zones are disjoint. We first consider intervals with the right endpoint greater than B or the left endpoint less than A (note there is no left endpoint on the right of B neither right endpoint on the left of A by optimality of S). We compress these intervals so that all endpoints in the system are between $A - \frac{\beta}{2}$

and $B + \frac{\beta}{2}$; this can be done in linear time. Now we consider all intervals that are contained in one of the intervals s_i in S . We refer to this set of intervals as H_i . Two intervals in H_i intersect by optimality of S ; so their largest left endpoint is smaller than their smallest right endpoint. For every i we now move all the left endpoints of intervals in H_i to L_i without changing the relative order of endpoints in the whole system. We move the right endpoints to R_i in the same manner. Finally all the endpoints that are in some $s_i \setminus (L_i \cup R_i)$ (these endpoints have not been moved until now) are moved into M_i without changing their relative position. All in all, computing these modifications requires linear time.

We claim that the new system R' satisfies $\frac{L(R')}{\ell(R')} \leq (2 + \varepsilon)\tilde{\alpha}(R)$. Indeed, note first that $L(R') \leq |B - A| + \beta = \tilde{\alpha}(R)(1 + \beta)$. Consider now an interval I in R' of length less than $(1 - 2\beta)$ (if any). I cannot be contained in an interval s_i due to modifications performed in H_i and it cannot have its two endpoints in $[A - \frac{\beta}{2}, B + \frac{\beta}{2}] \setminus \cup_i s_i$ (outside the zone covered by S) by optimality of S . So, it necessarily overlaps one interval $s_i \in S$ and by optimality of S it overlaps all intervals in H_i as well. We deduce that I has one of its endpoints in M_i and the other one outside s_i . Thus, I is of length at least $\frac{1}{2} - \frac{\beta}{2}$, which means that $\ell(R') \geq \frac{1}{2} - \frac{\beta}{2}$. Consequently, $\frac{L(R')}{\ell(R')} \leq 2\tilde{\alpha}(R) \left(\frac{1+\beta}{1-\beta} \right) \leq (2+\varepsilon)\tilde{\alpha}(R)$ since $\frac{4\beta}{1-\beta} \leq \varepsilon$. This concludes the proof.

We deduce approximation results using our online results with L a priori known. For clique covering overlap graphs, it improves by a factor 2 the ratio in [15].

Theorem 4. *For an interval system R , minimum colouring and clique covering G_R are $(\log \tilde{\alpha}(R) + c)$ -approximable, for a constant c . This ratio is doubled for b -bounded colouring and b -bounded clique covering G_R as well as b -bounded load colouring of R .*

6 Concluding remarks

Our contribution is twofold. First, we highlight the problem of partitioning an overlap graph into permutation graphs that seems interesting by itself. It is similar to the partition into layers in [2] but to our knowledge, it has not been considered in an online set-up until now. Our results motivate this approach that leads to improved competitive algorithms. Surprisingly, it leads to improved approximation algorithms in the offline case as well. The main advantage is to reduce colouring problems in overlap graphs to the same problems in permutation graphs. Since the partition can be done online, it leads to competitive-preserving reductions. Since the class of permutation graphs is stable by complement while the class of overlap graphs is not, this approach allows to solve the same problem in the complement of an overlap graph as well. This method could be used also for other combinatorial problems. Our results highlight two questions we leave open for the offline case: can we partition an overlap graph in less than a logarithmic factor of permutation graphs and are the considered problems constant approximable in overlap graphs?

Acknowledgments

The authors would like to thank anonymous referees for their helpful comments.

References

1. M. Avriel, M. Penn, and N. Shpirer. Container ship stowage problem: complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics*, 103(1-3):271–279, 2000.
2. J. Černý. Coloring circle graphs. *Electronic Notes in Discrete Mathematics*, 29:457–461, 2007.
3. S. Cornelsen and G. Di Stefano. Track assignment. *Journal of Discrete Algorithms*, 5(2):250–261, 2007.
4. M. Demange, G. Di Stefano, and B. Leroy-Beaulieu. On the online track assignment problem. *Discrete Applied Mathematics*, 160(7-8):1072–1093, 2012.
5. T. Ekin and D. de Werra. On split-coloring problems. *Journal of Combinatorial Optimization*, 10:211–225, 2005.
6. F. Gavril. Algorithms for a maximum clique and a maximum independent set of a circle graph. *Networks*, 3(3):261–273, 1973.
7. M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics 57)*. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 2004.
8. U. I. Gupta, D. T. Lee, and J. Y.-T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12(4):459–467, 1982.
9. K. Jansen. The mutual exclusion scheduling problem for permutation and comparability graphs. *Information and Computation*, 180:71–81, 2003.
10. J.M. Keil and L. Stewart. Approximating the minimum clique cover and other hard problems in subtree filament graphs. *Discrete Applied Mathematics*, 154(14):1983–1995, 2006.
11. F. G. König and M. E. Lübbecke. Sorting with complete networks of stacks. In *ISAAC 2008, Proceedings*, pages 895–906, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
12. L. Lesniak and H.J. Straight. The cochromatic number of a graph. *Ars Combinatoria*, 3:39–46, 1977.
13. S.D. Nikolopoulos and C. Papadopoulos. On the performance of the first-fit coloring algorithm on permutation graphs. *Information Processing Letters*, 75(6):265–273, 2000.
14. M. Olsen and A. Gross. Probabilistic analysis of online stacking algorithms. In *ICCL 2015, Proceedings*, volume 9335 of *Lecture Notes in Computer Science*, pages 358–369. Springer, 2015.
15. F. Shahrokhi. A new upper bound for the clique cover number with applications. *arXiv preprint arXiv:1502.06168*, 2015.
16. W. Unger. On the k-colouring of circle-graphs. In *STACS 88, Proceedings*, pages 61–72, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.