Dotted Interval Graphs and High Throughput Genotyping

Yonatan Aumann * Moshe Lewenstein* Oren Melamud* Ron Pinter† Zohar Yakhini‡

Abstract

We introduce a generalization of interval graphs, which we call *dotted interval graphs (DIG)*. A dotted interval graph is an intersection graph of arithmetic progressions (=*dotted intervals*). Coloring of dotted intervals graphs naturally arises in the context of high throughput genotyping. We study the properties of dotted interval graphs, with a focus on coloring. We show that any graph is a *DIG* but that DIG_d graphs, i.e. DIGs in which the arithmetic progressions have a jump of at most d, form a strict hierarchy. We show that coloring DIG_d graphs is NP-complete even for $d = 2$. For any fixed d, we provide a $\frac{7}{8}d$ approximation for the coloring of DIG_d graphs.

1 Introduction

Overview. Interval graphs have been extensively studied and have many applications [5]. A graph is an interval graph if the nodes correspond to intervals on the real axis, and there is an edge between two nodes iff their corresponding intervals overlap. We introduce a generalization of interval graphs, which we call *Dotted Interval Graphs (DIG)*, in which instead of solid intervals we consider "Dotted Intervals", i.e. segments of a "dotted line". Formally, a *Dotted Interval (DI)* is an arithmetic progressions of integer values. Thus, the nodes of a dotted interval graph correspond to arithmetic progressions and there is an edge between two nodes iff their respective arithmetic progressions share a point. Dotted interval graphs naturally arise in the context of high throughput genotyping, as explained in Section 1.1.
In this paper we study the properties of DIGs, with a focus on coloring.

Summary of Result. First, we show that unlike interval graphs, any (countable) graph is a dotted interval graph:

Theorem 1 *Every graph with a countable number of nodes is a DIG.*

Thus, we consider restricted DIGs where in each dotted interval the *jump* (i.e. the distance between consecutive points) is bounded by some constant d. We denote the class of such graphs by DIG_d. We show that these graphs form a strict hierarchy:

Theorem 2 *For all* $d \geq 1$, $DIG_d \subsetneq DIG_{d+1}$. *In addition,* $\bigcup_{d=1}^{\infty} DIG_d \subsetneq DIG$.

*Dept. of Computer Science, Bar-Ilan University, Ramat Gan, Israel; {aumann,moshe,orenmel}@cs.biu.ac.il
†Dept. of Computer Science, Technion, Israel; pinter@cs.technion.ac.il
‡Agilent Laboratotries and Dept of Computer Science, Technion, Haifa, Israel;zohar_yakhini@agilent.com

Next we consider the coloring of DIGs (which is the problem arising in the genotyping application). Coloring general DIGs is as hard as coloring general graphs. Thus, we consider the coloring of DIG_d graphs, and seek algorithms that are polynomial for a fixed d (i.e. Fixed Parameter Tractable in d). We show that coloring remains hard even for a fixed d:

Theorem 3 *Coloring DIG_d graphs is NP-complete for any $d \geq 2$.*

Note that DIG_1 graphs are interval graphs, which are polynomially colorable. Thus, there is a sharp distinction between DIG_1 and DIG_2.
Approximate coloring is also hard for general DIGs, but for any fixed d a constant approximation is possible:

Theorem 4 *For $d = 2$, there is a polynomial algorithm that guarantees a $\frac{3}{2}$ approximation for the coloring of DIG_2 graphs (given their DIG_2 representation). For any fixed $d > 2$, there is a polynomial algorithm that guarantees a $\frac{7}{8}d$ approximation for the coloring of DIG_d graphs (given their DIG_d representation).*

Finding the maximum clique in DIGs, on the other hand, is "easy".

Theorem 5 *Finding the maximum clique in DIG_d graphs is fixed parameter tractable in d (given their DIG_d representation).*

1.1 DIG Coloring and High Throughput Genotyping

Genotyping is the process by which elements of the genetic composition of individuals are determined. Specifically, genotyping usually refers to the process of determining the specific *allele* (genetic variant) present in a particular DNA sample, out of a set of known polymorphisms (multiple possible variants). Genotyping is routinely performed in thousands of laboratories around the world every day, with applications ranging from prenatal disease detection to forensics, and from crop enhancements to evolutionary biology.
One common type of genetic variation is characterized by multiple repeats of a fixed sequence of bases, within an otherwise conserved genomic background. These types of variations are called *microsatellite polymorphisms.* Different alleles differ in the number of times the *repeat sequence* is repeated. High allelic variability makes microsatellites powerful genetic markers, and therefore an extremely valuable tool for genome mapping in a variety of organisms, including the human [9, 14].
There are many different methods for microsatellite genotyping. It is out of the scope of this paper to describe these methods in detail (see [19] for a review on microsatellite genotyping methods). However, the general structure of many of the methods consists of two main steps. In the first step, the microsatellite together with some fixed flanking area is isolated (see Figure 1). In the second step, the length of the resulting DNA fragment is measured. Let F be the length of the fixed flanking area, Δ be the length of the repeat sequence, and n the number of the repeats of the repeat sequence. Then, the total length of the isolated segment is $F + n\Delta$. Thus, if F and Δ are known, then n can be reconstructed from the total length.
Since microsatellite genotyping is a costly procedure, we seek to reduce the number of times the procedure is applied. This can be done by applying the procedure to several sites simultaneously, a process called *multiplexing.* In this case, however, we will get several measurements for the total length, one for each of the sites in the genotyping assay. It is thus necessary to be able to uniquely determine what measured length corresponds to which site. For this, we must guarantee that all possible outcomes of one site are distinct from those of any other jointly measured sites. We show

that the problem of finding the minimal number of assays to genotype a given set of microsatellites can be modelled as a coloring problem in DIGs.
For any polymorphic microsatellite site p let F be the length of the flanking area for p and Δ the length of the repeat sequence. Let ℓ and h be the minimum and maximum number of repeats possible in p, respectively. Then, the possible lengths of the DNA fragment corresponding to p are: $S_p = \{F + n\Delta : \ell \leq n \leq h\}$, which is an arithmetic progression, i.e. a dotted interval. Two sites can be genotyped together iff their respective dotted intervals do not share a point. Consider the DIG G formed by the entire set of dotted intervals corresponding to the set of sites to be genotyped. Any coloring of G corresponds to a partition of the sites into sets that can be jointly genotyped, and vice versa.

1.2 Related Work

Interval Graphs and Generalizations. Interval graphs have been studied extensively (see [5]). There a several known generalization for interval graphs. *Circular arc graphs* are a generalization in which the nodes are arcs on a circle, instead of intervals on the line. Circular arc graphs are a proper generalization of interval graphs, but do not contain all graphs. It is known that coloring of circular arc graphs is NP-Complete [4], and there is a $\frac{5}{3}$ approximation algorithm for their coloring [15]. See [5, 3, 11, 4] for additional properties of circular arc graphs.
Another generalization of interval graphs are *t-interval graphs.* In a t-interval graph each node corresponds to t intervals, and there is an edge between two nodes iff any of their corresponding intervals overlap. Clearly, every finite graph is a t-interval graph for some t. For a fixed t, [2] show how to color t-interval graph G in $2t(\omega(G') - 1)$ colors, where $\omega(G')$ is the clique number of the underlying (simple) interval graph. See [6, 16, 17] for additional results on t-interval graphs.
Another generalization of interval graphs are *t-track interval graphs* (also called *union graphs* and *separated t-interval graphs*). In a t-track interval graph we consider t separate lines - *tracks*, and for each node there is a corresponding interval in each of the tracks. There is an edge between two nodes iff there exists a track in which their corresponding intervals overlap. For any $t > 1$, t-track interval graphs are a proper subset of t-interval graphs. See [10, 7] for studies on t-track intervals graphs.

Multiplexed Genotyping. Multiplexing is routinely performed in many genotyping studies. However, choosing what sites to jointly process is usually determined based on ad-hoc heuristics. There is little theoretical work on the algorithmic aspects of multiplexing.
Kivioja et al. [8] consider the problem of optimizing multiplexed transcription profiling. In this

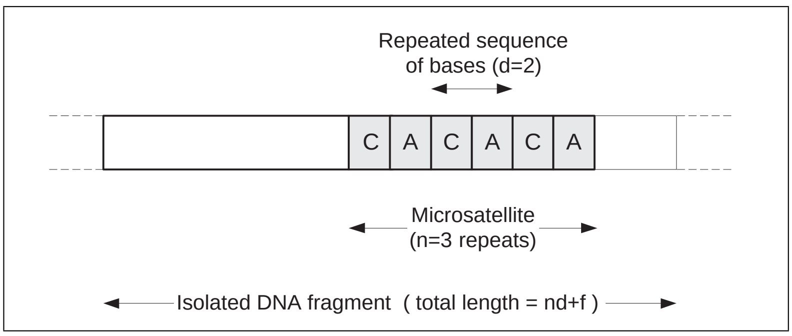

Figure 1: **Schematic Illustration of a Microsatellite**

case the aim is to measure transcriptional expression level of multiple genes, using hybridization probes. Kivioja et al. provide a 2-approximation algorithm for this optimization problem.
The relationship between microsatellite multiplexing and graph coloring was established in the patent [18]. The patent applies general graph coloring methods and does not provide specific coloring algorithms for DIGs. The patent also provides a technique to establish a per-instance lower bound on the coloring, based on the maximal number of dotted intervals that are over any point. We note that by using the maximal clique, as described in Section 5 a tighter lower bound can be established.
In [1] the authors consider the problem of multiplexing in SNP genotyping (SNPs are another type of genetic polymorphism). [1] provides theoretical hardness results for this problem, as well as heuristic algorithms that perform well in practice.

2 Definitions

In this Section we provide the formal definition for DIG and DIG_d graphs.

Definition 1 *Let* $F = \{S_1,...,S_n\}$ *be a family of sets.* $G = (V, E)$ *is the* **intersection graph** *of* F*, if* $|V| = |F|$ *and* $\forall\ i, j\ (v_i, v_j) \in E$ *if and only if* $i \neq j$ *and* $S_i \cap S_j \neq \emptyset$.

A ***dotted interval*** is a sequence of integers (dots) on the real line, where the distance between any two consecutive dots is some fixed distance k, called the dotted interval's *jump*. We denote by $DI(x, y, k, o)$ ($x \leq y$; $x, y \bmod k = o$; $k \in N\backslash\{0\}$; $0 \leq o < k$), the dotted interval such that $m \in DI(x, y, k, o)$ iff $x \leq m \leq y$ and $m \bmod k = o$. The parameter o is called the dotted interval's *offset*. x or y (or both) can be assigned the value $-\infty$ or ∞ respectively, in either case the DI is an infinite countable set of dots.

Definition 2 *A graph* $G = (V, E)$ *is a* **dotted interval graph (DIG)** *if it is an intersection graph of a set* D *of Dotted Intervals.*
A graph $G = (V, E)$*, is a* **dotted interval graph with maximal jump d***, denoted as* DIG_d*, if it is an intersection graph for a set of dotted intervals* D*, such that for each* $DI(x, y, k, o) \in D$*,* $k \leq d$ *(i.e. all the dotted intervals' jumps are at most* d*).*

By a slight abuse of notation we also denote by DIG the set of all DIG graphs, and by DIG_d, the set of all DIG_d graphs. It is easy to see that for finite and countable graphs, DIG_1 is exactly the class of interval graphs. However, dotted interval graphs with higher jumps reveal quite different properties. For instance, DIG_2 graphs can contain circles of arbitrary lengths (see Figure 2), while interval graphs are known to be chordal (maximal circle length is 3).

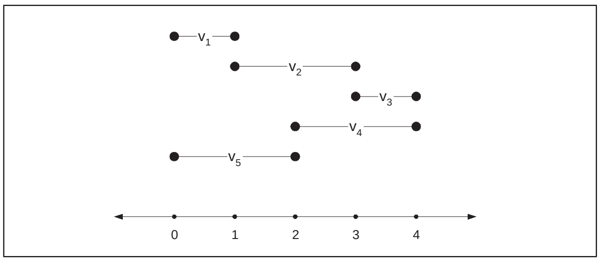

Figure 2: **A** DIG_2 **representation of** C_5**.**

Consider a dotted interval $I = DI(x, y, k, o)$. The beginning of I (denoted *begin*(I)) is x, the ending of I (denoted *end*(I)) is y. The *span* of I is the interval $[x, y]$.
The *overlap* of a number of DIs is the intersection of their spans. If this intersection is not empty, then they are considered *overlapping*. A dotted interval is *above* point p if its span includes p.
A dotted interval $DI(x, y, k, o)$ where $x = -\infty$ and $y = \infty$ is called an ***infinite dotted interval***. Such a dotted interval will be denoted as $DI_\infty(k, o)$. It follows directly from the DI definition that $DI_\infty(k, o) = \{kn + o\}_{n \in Z}$.

3 Properties of DIGs

3.1 Every Graph is a DIG

The following propositions demonstrate useful properties of infinite dotted intervals.

Proposition 3.1 *Two infinite dotted intervals $DI_\infty(k_1, o_1)$ and $DI_\infty(k_2, o_2)$ meet iff*

$$(o_1 - o_2) \bmod \gcd(k_1, k_2) = 0$$

Proof: The dots of $DI_\infty(k_1, o_1)$ and $DI_\infty(k_2, o_2)$ are the sets $\{k_1 n + o_1\}_{n \in Z}$ and $\{k_2 m + o_2\}_{m \in Z}$, respectively. In order for the two DIs to meet they need to have at least one common dot.
This means that the two DIs meet iff $\exists n', m' \in Z$ such that the following *meeting* condition is satisfied:

$$k_1 n' + o_1 = k_2 m' + o_2$$

or

$$o_1 - o_2 = k_2 m' - k_1 n'$$

or

$$o_1 - o_2 = \gcd(k_1, k_2) \cdot (p_2 m' - p_1 n')$$

where $p_1, p_2 \in Z$.

1. ($\Rightarrow$) If the two DIs meet then the meeting condition above is satisfied and obviously $(o_1 - o_2) \bmod \gcd(k_1, k_2) = 0$ must hold.
2. ($\Leftarrow$) If $(o_1 - o_2) \bmod \gcd(k_1, k_2) = 0$ then $(o_1 - o_2) = b \cdot \gcd(k_1, k_2)$ where $b \in Z$. The *Extended Euclidean algorithm* shows that $\exists n'', m'' \in Z$ such that $\gcd(k_1, k_2) = (k_2 m'' - k_1 n'')$ by describing a deterministic algorithm that calculates n'', m''. Using this we get $(o_1 - o_2) = (k_2 \cdot bm'' - k_1 \cdot bn'')$. So, choosing $n' = bn''$ and $m' = bm''$ satisfies the meeting condition, shown above, and the two DIs meet.

∎

Proposition 3.2 *If* $(o_1 - o_2) \bmod \gcd(k_1, k_2) = 0$ *then* $DI_\infty(k_1, o_1) \cap DI_\infty(k_2, o_2) = DI_\infty(\frac{k_1 \cdot k_2}{\gcd(k_1, k_2)}, o_3)$ *(where $\cap$ means intersection of sets).*

Proof: According to proposition 3.1, $(o_1 - o_2) \bmod \gcd(k_1, k_2) = 0$ means that $DI_\infty(k_1, o_1)$ and $DI_\infty(k_2, o_2)$ must have some meeting point $c \in DI_\infty(k_1, o_1) \cap DI_\infty(k_2, o_2)$. So $x \in DI_\infty(k_1, o_1) \cap DI_\infty(k_2, o_2)$ iff $(x - c) \bmod k_1 = 0$ and $(x - c) \bmod k_2 = 0$ and then $DI_\infty(k_1, o_1) \cap DI_\infty(k_2, o_2)$ is the set $\{\frac{k_1 \cdot k_2}{\gcd(k_1, k_2)} n + c\}_{n \in Z} = DI_\infty(\frac{k_1 \cdot k_2}{\gcd(k_1, k_2)}, (c \bmod \frac{k_1 \cdot k_2}{\gcd(k_1, k_2)}))$.

■

We are now ready to prove Theorem 1 (which we restate):

Theorem 1 *Every countable graph is a dotted interval graph.*

Proof: Given any countable graph $G(V, E)$ where $V = \{v_1, , \ldots, v_i, \ldots\}$ we construct a DIG representation for G as follows. Denote the $j-th$ prime number as p_j. For every i assign v_i the dotted interval $DI_\infty(k_i, o_i)$ which is constructed as a disjunction of primal jump DIs as follows: $DI_\infty(k_i, o_i) = DI_\infty(p_i, 0) \cap [DI_\infty(p_{m_1}, 1) \cap DI_\infty(p_{m_2}, 1) \cap \ldots \cap DI_\infty(p_{m_l}, 1)]$ where $m \in \{m_1, \ldots, m_l\}$ iff $(v_i, v_m) \notin E$ and $m < i$. Using proposition 3.2 it is easy to see that $DI_\infty(k_i, o_i) = DI_\infty((p_i \cdot p_{m_1} \cdot \ldots \cdot p_{m_l}), f_l)$, where f_l is the resulting offset. So every v_i is assigned $DI_\infty(k_i, o_i)$ where:

- The jump is:
$$k_i = p_i \cdot \prod_{j:(v_i, v_{m_j}) \notin E, m_j < i} p_{m_j}$$

- The offset $o_i = f_l$ has the following properties:
  1. $o_i \bmod p_i = 0$
  2. $o_i \bmod p_{m_j} = 1 \ \forall m_j$ such that $(v_i, v_{m_j}) \notin E$ and $m_j < i$

  These properties directly follow from the fact that o_i is actually a dot in $DI_\infty(k_i, o_i)$ and therefore is also a common dot to all of the original primal jump DIs.

Note that for every vertex v_i with a finite index i, the construction assigns a finite well-defined jump and offset to the corresponding DI.
The following will show that the intersection graph of the constructed DIs is indeed the graph G.

- $\forall v_i, v_j$ $(i < j)$ such that $(v_i, v_j) \notin E$:

  According to the constructed offsets' properties

  $(o_j - o_i) \bmod p_i = (o_j \bmod p_i) - (o_i \bmod p_i) = 1 - 0 \neq 0$

  and according to the jumps' construction $\gcd(k_i, k_j) = p_i \cdot k$, where $k \in N$.

  So it is clear that
$$(o_j - o_i) \bmod \gcd(k_i, k_j) \neq 0$$
  and the respective DIs indeed don't meet according to proposition 3.1

- $\forall v_i, v_j$ $(i < j)$ such that $(v_i, v_j) \in E$:

  According to jumps construction
$$\gcd(k_i, k_j) = \prod_{l:(v_i, v_{n_l}) \notin E, n_l < i} p_{n_l}$$
  According to constructed offsets' properties $\forall n_l$

  $(o_j - o_i) \bmod p_{n_l} = (o_j \bmod p_{n_l}) - (o_i \bmod p_{n_l}) = 1 - 1 = 0$

  So it is clear that $(o_j - o_i) \bmod \gcd(k_i, k_j) = 0$ and the respective DIs indeed meet according to proposition 3.1

■

3.2 The DIG_d Hierarchy

The following proposition demonstrates a useful property of overlapping dotted intervals.

Proposition 3.3 *Consider a graph $G \in DIG_d$. Let D be a DIG_d representation for G and S a set of dotted intervals $D \subseteq G$, where $|S| > d$, and every two dotted intervals in S overlap. There must exist at least two dotted intervals in S that meet.*

Proof: Let $\{DI_i\}_{i=1..d+1}$ denote $d+1$ dotted intervals in S. Consider $p = begin(DI_j)$ such that $\forall i$ $begin(DI_i) \leq begin(DI_j)$. This point is the rightmost beginning of all DIs in S. $\forall i$ $end(DI_i) \geq p$, must hold in order to satisfy the overlap assumption, and so p is in the span of all of these DIs. Since the maximal jump of all DIs is d, every one of the $d+1$ DIs must have at least one dot q such that $p \leq q < p + d$, but this means that at least two of these DIs must have the same dot, and so they meet.

∎

We are now ready to prove Theorem 2 (which we restate):

Theorem 2 *For all $d \geq 1$, $DIG_d \subsetneq DIG_{d+1}$. In addition, $\bigcup_{d=1}^{\infty} DIG_d \subsetneq DIG$.*

Proof: Consider the complete bipartite graph $K_{d+1,d+1} = (V, U, E)$ where $d \in N \setminus \{0\}$. Figure 3 illustrates a DIG_{d+1} representation for $K_{d+1,d+1}$ where $V = \{v_1, \ldots, v_{d+1}\}$ and $U = \{u_1, \ldots, u_{d+1}\}$. So $K_{d+1,d+1} \in DIG_{d+1}$.

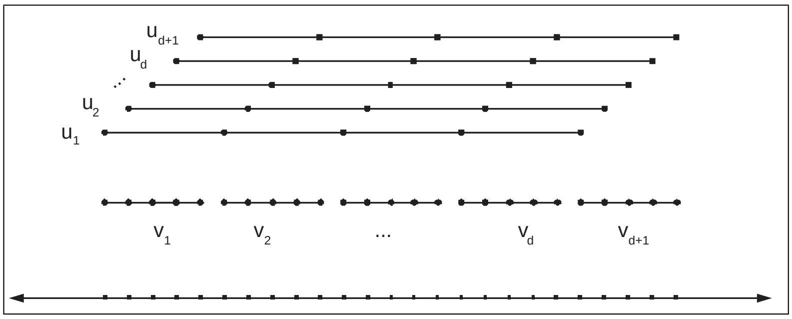

Figure 3: DIG_{d+1} **representation of** $K_{d+1,d+1}$

We show that $K_{d+1,d+1} \notin DIG_d$. Assume in contradiction that D is a DIG_d representation of $K_{d+1,d+1}$. Let $D_V \subseteq D$ be the representation of the vertices of V. Let $D_U \subseteq D$ be the representation of the vertices of U. Since V is an independent set and $|V| = d+1$, then according to Proposition 3.3, there must be at least two dotted intervals v_i, $v_j \in D_V$ that do not overlap. In the same way there must be at least two dotted intervals u_l, $u_m \in D_U$ that do not overlap. But this means it is not possible for both v_i and v_j to meet both u_l and u_m, which is in contradiction to the properties of the graph $K_{d+1,d+1}$ that D is representing. Thus, $DIG_d \subsetneq DIG_{d+1}$.
Consider the countable complete bipartite graph $K_{\infty,\infty}$. According to Theorem 1, $K_{\infty,\infty} \in DIG$. Assume, by contradiction, that $K_{\infty,\infty} \in \bigcup_{d=1}^{\infty} DIG_d$. This must mean that $\exists m \in N$ such that $K_{\infty,\infty} \in DIG_m$. But, as was demonstrated above $K_{m+1,m+1} \notin DIG_m$ and so obviously $K_{\infty,\infty} \notin DIG_m$. Thus, $\bigcup_{d=1}^{\infty} DIG_d \subsetneq DIG$.

∎

3.3 DIGs and Circular Arc Graphs

The following, which we state without proof, provides the relationship between DIG graphs and circular arc graphs:

Theorem 6 *There is a DIG_2 graph that is not a circular arc graph. For every $d \geq 1$, there is a circular arc graph that is not a DIG_d.*

4 NP-Hardness of the Optimal Coloring of DIG_d Graphs

The coloring of circular-arc graphs is known to be NP-Complete [4]. We prove the NP-Hardness of coloring DIG_2 graphs, using a polynomial reduction from the coloring of circular-arc graphs.
Let G be a circular arc graph, and let k be a natural number. The following polynomial reduction constructs a graph $H \in DIG_2$ such that G is k-colorable iff H is k-colorable.
If $k \geq |G|$ then G is clearly k-colorable. In this case simply output the empty graph which is in DIG_2 and obviously k-colorable, and we're done.
The *beginning* and *ending* of an arc a (also denoted as $begin(a)$ and $end(a)$) are the first point and the last point of arc a counter-clockwise, respectively. An arc a is regarded as *above* the point p if $begin(a) \leq p \leq end(a)$. The *frequency* of a point p (or $frequency(p)$) in a graph's circular-arc representation, is the number of arcs that are *above* this point.

If there is any point p on the circle of G's CAG representation such that $frequency(p) > k$, then G contains a clique whose size is larger than k, and it is clear that G is not k-colorable. In this case, simply output the graph $K_{k+1} \in DIG_2$. Since this $k+1$ clique is also obviously not k-colorable, we are done with this case.

For every arc a in G's CAG representation which is a complete circle let H's DIG_2 representation include an infinite dotted interval $DI_\infty(1,0)$. Since f such arcs and f such DIs always add exactly f distinct colors to the coloring of G and H, respectively, then w.l.o.g. we shall assume that there are no complete circle arcs in G's CAG representation.

Choose one of the points on G's circle whose frequency is maximal, and call it p_{split}. Then, for some $m \geq 0$ $frequency(p_{split}) = k - m$. Add to G's circle, m new arcs that are above p_{split} and intersect only with the $k - m$ arcs above p_{split}. We call the resulting graph G_1, and the k arcs above p_{split} *split arcs.*

Proposition 4.1 *G is k-colorable iff G_1 is k-colorable.*

Proof:

- ($\Leftarrow$) If G_1 is k-colorable then it is obvious that G is also k-colorable since $G \subseteq G_1$.
- ($\Rightarrow$) Assume that G is k-colorable. Use the same k-coloring to color G_1's arcs, except for the m additional arcs above p_{split}. Since this coloring uses only $k - m$ colors for the arcs above p_{split}, and the additional m arcs intersect only with those arcs, then it is possible to color the additional arcs with the m non-used colors at p_{split}. So G_1 is also k-colorable.

∎

Next we modify G_1's CAG representation in order that no split arc (i.e. arc above p_{split}) begins or ends at the same point as another split arc, without changing the represented graph. This will be achieved by iteratively breaking ties between any two arcs with the same beginning or end.

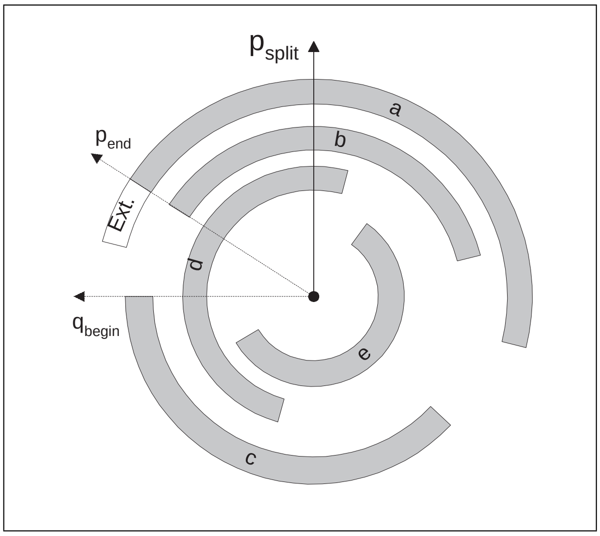

Figure 4: **Breaking ties on G_1**

A tie will be broken as follows (see Figure 4. Suppose that the two arcs a and b end at the same endpoint p_{end} (the method will be the same for two arcs with the same beginning). Scan the circle counter-clockwise until meeting the first beginning point q_{begin} of any arc such that $q_{begin} = p_{end} + \epsilon$ where $\epsilon > 0$ (points' coordinates are regarded as angular). Extend arc a's ending to $p_{end} + \frac{\epsilon}{2}$.

It is easy to observe that the described extension did not change the represented graph G_1, since the extended arc $a_{extended}$ meets no more and certainly no less arcs than did a.

Next we split the circular-arc representation of graph G_1 open at p_{split} turning it into the interval graph representation of a new graph G_2, where every split arc is represented by a corresponding pair of split intervals - a left split interval and a right split interval, and every other arc is represented by a single interval. See an illustration for this transformation in Figure 4.

Proposition 4.2 *G_1 is k-colorable iff G_2 has a k-coloring that assigns the same color to every corresponding pair of split intervals.*

Let $I(x, y)$ denote the interval $[x, y]$ on the real line. $I(x, y)$ is called an *even* interval if $x, y \in \{2n\}_{n \in Z}$, i.e. the interval's endpoints have even integral values. It is easy to see that every interval graph representation with a countable number of intervals can be modified in such a way that all intervals are even, without changing the represented graph. Therefore, assume w.l.o.g. that G_2's interval graph representation includes only such even intervals.

From G_2 we construct the DIG_2 representation of graph H as follows. For every left and right split intervals, $I_{left}(x, y)$ and $I_{right}(x, y)$, in G_2, include the left and right split dotted interval, $DI_{left}(x, y, 1, 0)$ and $DI_{right}(x, y, 1, 0)$. For any other interval, $I(x, y)$, in G_2, include the dotted interval $DI(x, y, 2, 0)$. For every corresponding pair of left and right split dotted intervals, $DI_{left}(x, y, 1, 0)$ and $DI_{right}(s, t, 1, 0)$, where $s > y$, include an additional *binding* dotted interval $DI_{binding}(y + 1, s - 1, 2, 1)$. These dotted intervals will have a *binding* effect on the coloring of corresponding pairs of split dotted intervals as will be shown.
Figure 5 illustrates the above construction.
We denote with $\{l_1, ..., l_k\}$, $\{r_1, ..., r_k\}$, $\{b_1, ..., b_k\}$, the left split, right split and binding dotted interval, respectively, where l_i, r_i are a corresponding pair of split dotted intervals, and b_i is their

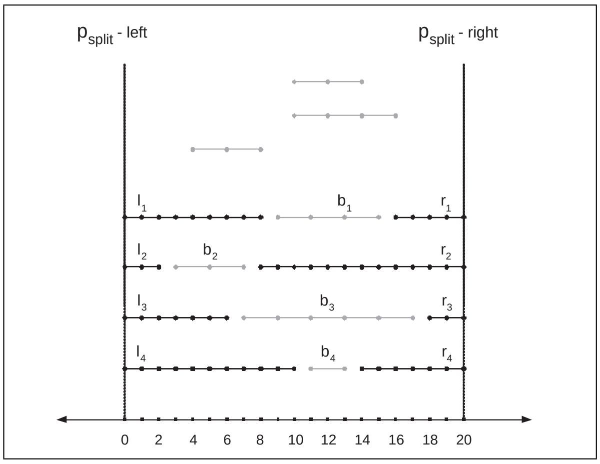

Figure 5: **An example for a constructed H graph** ($k = 4$)

binding dotted interval. Note that without the binding dotted intervals, graph H would represent the exact same graph as G_2 ($G_2 \subseteq H$). Also, the binding dotted intervals never meet the other (non-split) dotted intervals because the first consist only of odd points, while the latter only of even points.

Proposition 4.3 *G_2 has a k-coloring that assigns the same color to every pair of split intervals iff H is k-colorable.*

Proof:

- ($\Rightarrow$) If G_2 has a k-coloring that assigns the same color to every corresponding pair of split intervals, then use this coloring for H's non-binding dotted intervals. Note that $\{l_1, ..., l_k\}$ and $\{r_1, ..., r_k\}$ are both cliques so every corresponding split pair (l_i, r_i) was assigned a different color $\chi(l_i) = \chi(r_i) = i$. Assigning the binding dotted intervals the same color as their corresponding split pair $\chi(b_i) = i$, gives a valid k-coloring, since the binding dotted intervals neither intersect their corresponding split pair intervals nor the other intervals, by construction. So H is k-colorable.

- ($\Leftarrow$) If H is k-colorable the following will show that G_2 has a k-coloring that assigns the same color to every pair of split intervals.

  Lemma 4.4 *For every $i \in \{1, .., k\}$ there exists a clique of size $k-1$, $K^i_{k-1} \subseteq H$, such that b_i and l_i intersect all of the vertices in K^i_{k-1}.*

  Proof:

  Consider any binding dotted interval b_i in graph H. We will show that for every j ($j \neq i$) there exists a dotted interval $a_j \in \{l_j, b_j, r_j\}$ such that both dotted intervals b_i and l_i intersect a_j, and the beginning of b_i (denoted as $begin(b_i)$) is always a dot of a_j.

  Since $begin(b_i)$ is an odd dot, and every odd dot on graph H's line is covered by exactly one of the dotted intervals $\{l_j, b_j, r_j\}$, then the following are all of the possible cases for $begin(b_i)$:

1. $begin(b_i) \in l_j$: this naturally means that b_i intersects l_j. It is also easy to see that l_i intersects l_j, since l_j covers all the dots to the left of $begin(b_i)$. See example in Figure 5, where $i = 1$ and $j = 4$.
2. $begin(b_i) \in b_j$: this naturally means that b_i intersects b_j. Since according to construction $begin(b_i) \neq begin(b_j)$, this must mean that $begin(b_j) < begin(b_i)$, and so l_i also intersects b_j. See example in Figure 5, where $i = 1$ and $j = 3$.
3. $begin(b_i) \in r_j$: this naturally means that b_i intersects r_j. Since $begin(r_j)$ is even while $begin(b_i)$ is odd, this must mean that $begin(r_j) < begin(b_i)$, and so l_i also intersects r_j. See example in Figure 5, where $i = 1$ and $j = 2$.

The above claim shows that $\forall i$ b_i and l_i intersect with the same $k-1$ dotted interval $\{a_j\}_{j \neq i}$. Since $\forall j$ $begin(b_i) \in a_j$, $\{a_j\}_{j \neq i}$ is the $k-1$ clique K^i_{k-1} (e.g. in Figure 5, l_1 and b_1 both meet all 3 vertices of the clique $\{r_2, b_3, l_4\}$). ∎

Since any coloring must use $k-1$ colors in order to color a $k-1$ clique, then according to Lemma 4.4 any k-coloring of graph H will have to use the same $k-th$ color for both l_i and b_i for every i.

The conclusion is that $\forall i$ $\chi(b_i) = \chi(l_i)$ must hold for any k-coloring of H. A symmetric argument to the above shows that $\forall i$ $\chi(b_i) = \chi(r_i)$ must also hold. So if graph H is k-colorable, all of its k-colorings fulfil the condition $\forall i$ $\chi(r_i) = \chi(l_i)$, i.e. the colors of every corresponding pair of split dotted interval are bound to be the same. Since $G_2 \subseteq H$ these k-colorings are obviously all valid k-colorings for G_2 too.

∎

Propositions 4.1, 4.2, 4.3, show that a circular-arc graph G is k-colorable iff a polynomial time constructed DIG_2 graph H is k-colorable. This means that the decision version of the coloring of DIG_2 graphs is NP-Complete, and the optimal coloring of DIG_2 graphs is NP-Hard. Since $\forall d$ s.t. $d > 2$, $DIG_2 \subseteq DIG_d$, the result is true for all DIG_d graphs where $d \geq 2$. This completes the proof of Theorem 3.

5 Coloring Approximations

Following are approximation algorithms for the coloring of DIG_2 and DIG_d graphs that establish Theorem 4. The algorithms require the DIG representation of the input graph. Let $\chi(G)$ and $maxclique(G)$ denote the chromatic number and the size of the maximum clique in graph G, respectively.

5.1 Coloring DIG_2 Graphs

Consider a graph $G \in DIG_2$, $|G| = n$ where $D_G = \{DI_i\}_{i=1..n}$ is a DIG_2 representation of G. Let $b_i = begin(DI_i)$ and assume w.l.o.g that if $b_i < b_j$ then $i < j$, i.e. the DIs are sorted by their beginnings. Let V_i denote the vertices that are represented by the DIs above the point b_i. G_i is a subgraph of G induced by the vertices of V_i. There are n such subgraphs in G. Note that these subgraphs are not necessarily distinct.

```
Color_2(G)
Input: Graph G ∈ DIG_2
Output: A coloring of G

for i ← 1 to n do
    foreach v ∈ G_i do
        if v has no color
            Try to assign a valid color from the colors of G_i's vertices
        if v still has no color
            Try to assign a valid color from the colors of G's vertices
        if v still has no color
            Assign a new color to v
```

Figure 6: Approximated coloring algorithm for DIG_2 graphs

The approximation coloring algorithm, $Color_2$, for G is as follows (see also Figure 6). Scan G's subgraphs from G_1 to G_n. For every G_i encountered, go over every vertex $v \in G_i$, that has not already been assigned a color in previous steps of the algorithm, in an arbitrary order, and assign a valid color to v. A valid color for a vertex v is a color that has not been previously assigned to any other vertex u which is adjacent to v in G. The algorithm tries to minimize the usage of colors by trying to reuse colors that have already been assigned. The strategy is to first try to reuse one of the colors that have been assigned to any of the vertices in G_i, and only if no such valid color is found, try colors from all G. Clearly, this algorithm performs a valid coloring of G in polynomial time. Let $Color_2(G)$ denote the number of colors used by the $Color_2$ algorithm to color graph G.

Theorem 7 *For every $G \in DIG_2$ (given G's DIG_2 representation), $Color_2(G) \leq \frac{3}{2} maxclique(G) \leq \frac{3}{2}\chi(G)$.*

Proof: Consider a graph $G \in DIG_2$, a DIG_2 representation of G, denoted D_G, and G's subgraphs $\{G_i\}_{i=1 \ldots n}$ as described above.
Before algorithm $Color_2$ colors G_i, the uncolored vertices of V_i are exactly the vertices whose DIs begin at b_i. Any vertex $v \in (\bigcup_{j=1}^{i-1} V_j) \setminus V_i$, is represented by a DI that ends before b_i and so never meets the vertices of G_i. This means that any color previously assigned to the vertices of $(\bigcup_{j=1}^{i-1} V_j) \setminus V_i$ and not yet assigned to any of V_i's vertices, is a valid color for any vertex of V_i. So the coloring of G_i will always use all of the colors of G before using a new color.
Consider a full coloring of graph G by algorithm $Color_2$, denoted C_G. Let $Color_2(G_i)$ denote the number of different colors assigned in C_G to the vertices of subgraph G_i. The above leads to the following corollary:

Corollary 5.1 *$Color_2(G) = max_{i=1 \ldots n}\{Color_2(G_i)\}$.*

Consider dotted intervals in D_G that contain only one dot. It is easy to see that all such dotted interval are in $\bigcup_{x \in Z}\{DI(2x, 2x, 2, 0), DI(2x+1, 2x+1, 2, 1)\}$. This means that we can assume w.l.o.g that all the dotted intervals in D_G that contain a single dot are represented as DIs with a jump of 2.
Let $V_{k,o}^i$ denote the vertices of G_i whose DIs' jump is k and offset is o. $G_{k,o}^i$ is the subgraph induced by the vertices of $V_{k,o}^i$. The distinct vertices sets $V_{1,0}^i$, $V_{2,0}^i$, $V_{2,1}^i$, contain all the vertices of G_i. It

is easy to observe the following characteristics of G_i (see illustration in Figure 7). $G^i_{1,0}$, $G^i_{2,0}$, and $G^i_{2,1}$ are cliques. The subgraph induced by $V^i_{1,0} \cup V^i_{2,0}$ and the subgraph induced by $V^i_{1,0} \cup V^i_{2,1}$ are both cliques. $G^i_{2,0}$, and $G^i_{2,1}$ are distinct. It follows from the above that $maxclique(G_i) = max\{\left|V^i_{1,0} \cup V^i_{2,0}\right|, \left|V^i_{1,0} \cup V^i_{2,1}\right|\}$.

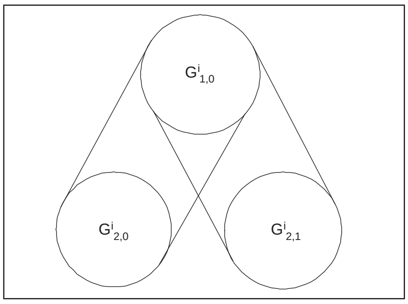

Figure 7: **The composition of a subgraph** G_i

Let $Color_2(G^i_{2,0} \cup G^i_{2,1})$ denote the number of different colors assigned in C_G to the vertices of subgraphs $G^i_{2,0}$ and $G^i_{2,1}$.

Lemma 5.2 $\forall i \; Color_2(G^i_{2,0} \cup G^i_{2,1}) \leq maxclique(G)$.

Proof: The proof is by induction on i.

$i = 1$

Let $b_1 \bmod 2 = \delta_1$. All vertices of G_1 are represented by DIs whose beginning is b_1, so $v \in G_1$ iff $v \in V^1_{1,0} \cup V^1_{2,\delta_1}$. Therefore, $Color_2(G^1_{2,0} \cup G^1_{2,1}) = Color_2(G^1_{2,\delta_1}) = \left|V^1_{2,\delta_1}\right| \leq maxclique(G_1) \leq maxclique(G)$.

$i = m$

Let $b_m \bmod 2 = \delta_m$. The induction assumption is $Color_2(G^{m-1}_{2,0} \cup G^{m-1}_{2,1}) \leq maxclique(G)$. So before $Color_2$ colors G_m, the vertices of $G^m_{2,0} \cup G^m_{2,1}$ were assigned less than $maxclique(G)$ different colors. Also, all the uncolored vertices in G_m are represented by DIs whose beginning is b_m, so in this step $Color_2$ colors only vertices in $V^m_{1,0} \cup V^m_{2,\delta_m}$. Assume, by contradiction, that $Color_2(G^m_{2,\delta_m} \cup G^m_{2,1-\delta_m}) > maxclique(G)$. This means there had to be a stage where the vertices of $G^m_{2,\delta_m} \cup G^m_{2,1-\delta_m}$ were previously assigned exactly $maxclique(G)$ different colors, and then $Color_2$ colored some vertex $v \in G^m_{2,\delta_m}$ with a color Ψ which was not used in $G^m_{2,\delta_m} \cup G^m_{2,1-\delta_m}$ before. Color Ψ cannot be a color of one of the vertices in $V^m_{1,0}$ because v meets all of them. Now, since $Color_2$ seeks to reuse the colors of G_m first, this necessarily means that there was no color from $G^m_{2,\delta_m} \cup G^m_{2,1-\delta_m}$ that was a valid color for v. $G^m_{2,1-\delta_m}$ is a clique that was fully colored at the beginning of this step and doesn't meet any of G^m_{2,δ_m}'s vertices, so it must be that the coloring of G^m_{2,δ_m} already contained all the colors of $G^m_{2,1-\delta_m}$, and therefore just before the coloring of v, the colors assigned to the vertices of G^m_{2,δ_m} are also the colors assigned to $G^m_{2,\delta_m} \cup G^m_{2,1-\delta_m}$, which are $maxclique(G)$ different colors. But this means that $\left|V^m_{2,\delta_m} \setminus \{v\}\right| \geq maxclique(G)$ or $\left|V^m_{2,\delta_m}\right| > maxclique(G)$, which is clearly impossible.

∎

Lemma 5.3 *$\forall i$ $Color_2(G_i) \leq \frac{3}{2}maxclique(G)$.*

Proof: Consider the two possible cases for the size of $V^i_{1,0}$, where $0 \leq m \leq \frac{1}{2}maxclique(G)$:

1. $\left|V^i_{1,0}\right| = \frac{1}{2}maxclique(G) + m$:
   $\left|V^i_{1,0}\right| + \left|V^i_{2,0}\right| \leq maxclique(G)$; $\left|V^i_{1,0}\right| + \left|V^i_{2,1}\right| \leq maxclique(G)$
   $\left|V^i_{2,0}\right| \leq \frac{1}{2}maxclique(G) - m$; $\left|V^i_{2,1}\right| \leq \frac{1}{2}maxclique(G) - m$
   Since the number of colors used for G_i is at most the number of its vertices:
   $Color_2(G_i) \leq \left|V^i_{1,0}\right| + \left|V^i_{2,0}\right| + \left|V^i_{2,1}\right| \leq \frac{3}{2}maxclique(G) - m \leq \frac{3}{2}maxclique(G)$

2. $\left|V^i_{1,0}\right| = \frac{1}{2}maxclique(G) - m$:
   $Color_2(G_i) \leq \left|V^i_{1,0}\right| + Color_2(G^i_{2,0} \cup G^i_{2,1}) \leq \left|V^i_{1,0}\right| + maxclique(G)$ (using Lemma 5.2) $\leq \frac{1}{2}maxclique(G) - m + maxclique(G) \leq \frac{3}{2}maxclique(G)$

∎

Lemma 5.3, Corollary 5.1 and the fact that $maxclique(G) \leq \chi(G)$ lead to $Color_2(G) \leq \frac{3}{2}maxclique(G) \leq \frac{3}{2}\chi(G)$, which completes the proof Theorem of 7. ∎

Figure 8 provides a tight example for the $\frac{3}{2}$ approximation factor guaranteed by Theorem 7. More specifically, the figure illustrates a DIG_2 representation of a graph with $4n$ vertices. It is easy to see that algorithm $Color_2$ uses $3n$ colors for this graph, where as the optimal coloring uses only $2n$ colors.

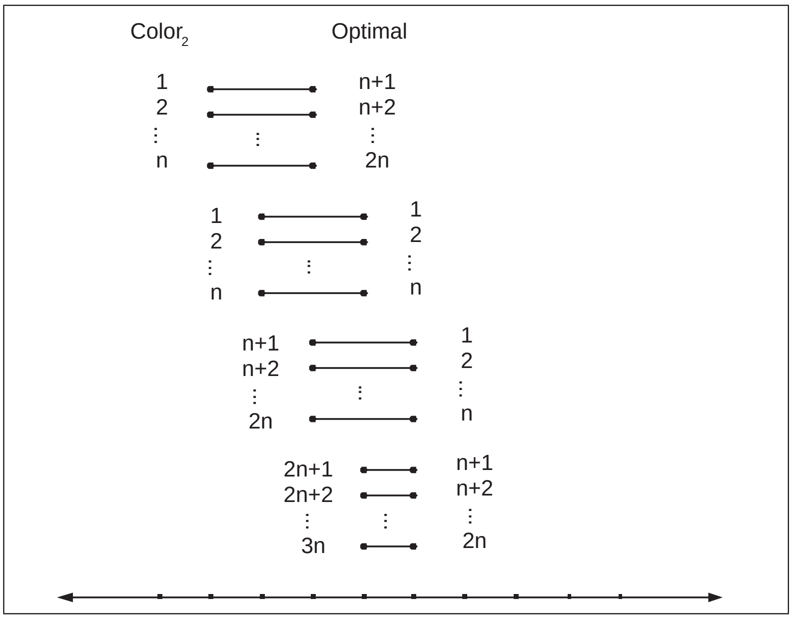

Figure 8: **A tight example for the approximation factor of** $Color_2$

Since $\chi(G) \leq Color_2(G)$ always holds, Theorem 7 also leads to an upper bound on the ratio between the chromatic number and maximum clique size of DIG_2 graphs as follows:

Corollary 5.4 *For every $G \in DIG_2$, $\chi(G) \leq \frac{3}{2}maxclique(G)$.*

$Color_d(G)$
Input: Graph $G \in DIG_d$
Output: A coloring of G

$C \leftarrow \emptyset$
for $i \leftarrow d$ **down to** 1 **do**
  if $i \notin C$ **then**
    if i mod $2 = 0$ **then**
      color $G_{i \cup \frac{i}{2}}$ with $Color_2$ (with a new set of colors)
      $C \leftarrow C \cup \{i, \frac{i}{2}\}$
    if i mod $2 = 1$ **then**
      color G_i with interval graph coloring (with a new set of colors)
      $C \leftarrow C \cup \{i\}$

Figure 9: Approximated coloring algorithm for DIG_d graphs

5.2 Coloring General DIG_d Graphs

Consider a graph $G(V, E) \in DIG_d$. Let D_G be a DIG_d representation of G. Let $V_{k,o}$, $k \leq d$, $o < k$, denote the set of vertices in G whose representation is a dotted interval with a jump k and an offset o, and $G_{k,o}$ the subgraph of G induced by the vertices of $V_{k,o}$. It is easy to see that $G_{k,o}$ is an interval graph.
Next, denote $V_k = \bigcup_{o=0}^{k-1} V_{k,o}$, and G_k the subgraph of G induced by the vertices of V_k. Using Proposition 3.1 it is easy to see that for every $o_1 \neq o_2$, G_{k,o_1} and G_{k,o_2} are distinct. G_k is a union of distinct interval graphs, which is also an interval graph. As an interval graph, G_k has a polynomial algorithm for optimal coloring, where $\chi(G_k = maxclique(G_k))$.
Denote $V^o_{k\cup 2k} = V_{k,o} \cup V_{2k,o} \cup V_{2k,k+o}$ where $2k \leq d$ and $o < k$, and let $G^o_{k\cup 2k}$ be the subgraph induced by the vertices of $V^o_{k\cup 2k}$. Let $D_{G^o_{k,2k}} \subseteq D_G$ be the set of DIs representing the subgraph $G^o_{k\cup 2k}$. A new DIG representation for $G^o_{k\cup 2k}$, denoted $D'_{G^o_{k,2k}}$, can be constructed by including in $D'_{G^o_{k,2k}}$ for every $DI(x, y, p, q) \in D_{G^o_{k,2k}}$ the dotted interval $DI(\frac{x-o}{k}, \frac{y-o}{k}, \frac{p-o}{k}, \frac{q-o}{k})$. It is easy to see that $D'_{G^o_{k,2k}}$ is a DIG_2 representation of $G^o_{k\cup 2k}$, and this means that $G^o_{k\cup 2k} \in DIG_2$.
Denote $V_{k\cup 2k} = \bigcup_{o=0}^{k-1} V^o_{k\cup 2k}$ where $2k \leq d$, and $G_{k\cup 2k}$ the subgraph induced by $V_{k\cup 2k}$. Note that $V_{k\cup 2k}$ contains all the vertices who are represented by DIs with jumps of either k or $2k$. Using Proposition 3.1 it is easy to see that for every $o_1 \neq o_2$, $G^{o_1}_{k\cup 2k}$ and $G^{o_2}_{k\cup 2k}$ are distinct. $G_{k\cup 2k}$ is a union of distinct DIG_2 graphs, which is also a DIG_2 graph. Using algorithm $Color_2$ to color $G_{k\cup 2k}$ gives $Color_2(G_{k\cup 2k}) \leq \frac{3}{2}\chi(G_{k\cup 2k})$ according to Theorem 7.
Algorithm $Color_d$ is as follows (see also Figure 9). Scan all possible jump sizes in the representation of G from d down to 1. For every jump i, if subgraph G_i was not yet colored, then color it in the following way. If i is even then *match* G_i to $G_{\frac{i}{2}}$ and use algorithm $Color_2$ to color $G_{i\cup\frac{i}{2}}$, else use an optimal interval graph coloring to color G_i. In every step the algorithm uses a different set of colors for the coloring, so the resulting coloring of G is clearly valid.
Denote m the number of matchings performed by $Color_d$ while coloring graph $G \in DIG_d$. This means: $Color_d(G) = \Sigma_{i=1}^{m} Color_2(G_{k_i\cup 2k_i}) + \Sigma_{j=1}^{d-2m}\chi(G_{k_j}) \leq \Sigma_{i=1}^{m}\frac{3}{2}maxclique(G_{k_i\cup 2k_i}) + \Sigma_{j=1}^{d-2m} maxclique(G_{k_j}) \leq \Sigma_{i=1}^{m}\frac{3}{2}maxclique(G) + \Sigma_{j=1}^{d-2m} maxclique(G) \leq (d - \frac{1}{2}m) \cdot maxclique(G)$.

Assume $d \bmod 4 = 0$. Then algorithm $Color_d$ performs at least $\frac{d}{4}$ matchings from $\{\frac{d}{2}+2n\}_{n=1\ldots\frac{d}{4}}$ to $\{\frac{d}{4}+n\}_{n=1\ldots\frac{d}{4}}$, and then $Color_d(G) \leq (\frac{7}{8}d) \cdot maxclique(G)$. For arbitrary d, $Color_d(G) \leq (\frac{7}{8}d+\frac{3}{8}) \cdot maxclique(G)$. Since, obviously, $maxclique(G) \leq \chi(G)$, it follows that:

Theorem 8 *For every $G \in DIG_d$ (given G's DIG_d representation), $Color_d(G) \leq (\frac{7}{8}d+\frac{3}{8})maxclique(G) \leq (\frac{7}{8}d+\frac{3}{8})\chi(G)$.*

Since $\chi(G) \leq Color_d(G)$ always holds, the following upper bound on the ratio between the chromatic number and maximum clique size of DIG_d graphs is established:

$$\chi(G) \leq (\frac{7}{8}d+\frac{3}{8}) \cdot maxclique(G)$$

6 Maximum Clique

The following section provides the proof for Theorem 5.

Proposition 6.1 *Consider the two dotted intervals $DI_1(x_1, y_1, k_1, o_1)$, $DI_2(x_2, y_2, k_2, o_2)$, and another two dotted intervals with the same jump and offset but modified beginning and end $DI'_1(x'_1, y'_1, k_1, o_1)$, $DI'_2(x'_2, y'_2, k_2, o_2)$. If the lengths of $overlap(DI_1, DI_2)$ and $overlap(DI'_1, DI'_2)$ are both greater than d^2 where $k_1, k_2 \leq d$ then DI_1 meets DI_2 iff DI'_1 meets DI'_2.*

Proof: Assume DI_1, DI_2, DI'_1, DI'_2 with overlaps longer than d^2 as described above.

- ($\Rightarrow$) If DI_1 meets DI_2 then according to proposition 3.1:

  $$(o_1 - o_2) \bmod \gcd(k_1, k_2) = 0$$

  Now according to proposition 3.2 we easily get that:

  $$DI'_1(x'_1, y'_1, k_1, o_1) \cap DI'_2(x'_2, y'_2, k_2, o_2) = DI'_3(\frac{k_1 \cdot k_2}{\gcd(k_1, k_2)}, o_3, -\infty, \infty) \cap overlap(DI'_1, DI'_2)$$

  But since the jump $\frac{k_1 \cdot k_2}{\gcd(k_1,k_2)} \leq d^2$ while $|overlap(DI'_1, DI'_2)| \geq d^2$ then clearly $DI'_1 \cap DI'_2 \neq \emptyset$ and so DI'_1 and DI'_2 also meet.

- ($\Leftarrow$) The second direction of the proof is exactly symmetric.

∎

We are now ready to prove Theorem 5 (which we restate):

Theorem 5 *Finding the maximal clique of DIG_d graphs is fixed parameter tractable in d.*

Proof: Consider graph $G \in DIG_d$, $|G| = n$. The following will show an algorithm for finding the maximal clique of G in time which is fixed parameter tractable in d.

Let x_i and y_j be the leftmost and rightmost endpoint of all DIs with finite endpoints in the DIG_d representation of G, respectively (if there are no finite endpoints, let x_i and y_i be equall to zero). Replace any DI coordinate $x = -\infty$ with $x = x_i - d^2$ and any DI coordinate $y = \infty$ with $y = y_i + d^2$. The outcome of this is a new representation which contains only finite DIs. It is easy to see according to proposition 6.1 that the new representation represents the same graph G, since this trimming of any dotted interval a_j into a'_j either does not change its overlap with other DIs, or the overlap grows smaller but stays at least d^2 long. In either case $\forall i$ a_i meets a_j iff a_i meets a'_j.

The following proposition shows that cliques in DIGs have a property that we call *locality*.

Proposition 6.2 *Consider a clique K. Let $\{DI_i\}$ be a DIG representation of K. There is at least one common point p such that:*

- $\forall i$ $p \in span(DI_i)$
- $\exists j$ *such that* $p = begin(DI_j)$

Proof: Consider $p = begin(DI_j)$ such that $\forall i$ $begin(DI_i) \leq begin(DI_j)$. This point is the rightmost beginning of all DIs in the clique K. All the DIs pertaining to the same clique meet each other. In order to meet it is obvious that it is necessary for their spans to overlap. So $\forall i$ $end(DI_i) \geq begin(DI_j) = p$, must hold, and point p satisfies the above conditions.

∎

Denote as H_p a *local* subgraph of G induced by the vertices that are represented by the DIs above the point p.
Applying proposition 6.2 to the maximal clique leads to the following corollary:

Corollary 6.3 *$maxclique(G) = \max_i\{maxclique(H_{begin(DI_i)})\}$ where $\{DI_i\}$ is a DIG_d representation of G.*

This means that in order to find the maximal clique of G it is enough to go over the n different DIs' beginnings, find each local graph's maximal clique, and return the maximal of these maximal cliques.
The following will show how to find the maximal clique of a local graph H_p:

- Trim all DIs in H_p so their beginnings and ends have a distance of at most d^2 from p. Again, according to proposition 6.1, the new representation represents the same graph, since this trimming of any dotted interval a_j into a'_j either does not change its overlap with other DIs, or the overlap grows smaller but stays at least d^2 long. In either case $\forall i$ a_i meets a_j iff a_i meets a'_j.
- Assign each $DI(x, y, k, o) \in H_p$ to a group $g(x, y, k, o)$. Since for every DI:
  1. $p - d^2 \leq x \leq p$
  2. $p \leq y \leq p + d^2$
  3. $k, o \leq d$

  the number of different groups required is at most d^6, while each group may contain up to n identical DIs.
- Since all DIs in the same group g have the exact same dots, then g is a clique. Also, if any DI in g_i meets any DI in g_j then all of the DIs in g_i meet all the DIs in g_j. Therefore, any maximal clique of H_p containing any single DI of g, contains all of the DIs of g.

- Construct a group graph M, $|M| \leq d^6$, where every group g_i is a vertex, and g_i is adjacent to g_j if the DIs of these groups meet. The maximal clique of H_p can be found by going over all the maximal complete subgraphs of M (meaning the complete subgraphs which are maximal in the sense that there is no other complete subgraph that properly contains them), and picking the one that contains the greatest sum of DIs in its groups.

The algorithm described above clearly finds the maximal clique of G in time which is fixed parameter tractable in d.

∎

References

[1] Yonatan Aumann, Efrat Manisterski, and Zohar Yakhini. Designing optimally multiplexed snp genotyping assays. In Gary Benson and Roderic D. M. Page, editors, *Proceedings of the Third International Workshop on Algorithms in Bioinformatics, WABI 2003*, volume 2812 of *Lecture Notes in Computer Science*, pages 255–283, 2003.

[2] Reuven Bar-Yehuda, Magnús M. Halldórsson, Joseph (Seffi) Naor, Hadas Shachnai, and Irina Shapira. Scheduling split intervals. In *Proceedings of the thirteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 732–741, 2002.

[3] A. Brandstadt, V.B. Le, J.P. Spinrad. *Graph classes - a survey.* SIAM, 1999.

[4] M.R. Garey, D.S. Johnson, G.L. Miller, C.H. Papadimitriou. The Complexity of coloring circular arcs and chords. In *SIAM Journal of Algebraic Discrete Methods*, 1(2):216-227, 1980.

[5] M.C. Golumbic. *Algorithmic graph theory and perfect graphs.* Elsevier, 2004.

[6] András Gyárfás. On the chromatic number of multiple interval graphs and overlap graphs. *Discrete Math.*, 55:161–166, 1985.

[7] András Gyárfás and Douglas West. Multitrack interval graphs. *Congr. Numer.*, 109:109–116, 1995.

[8] Teemu Kivioja, Mikko Arvas, Kari Kataja, Merja Penttila, Hans Soberlund, and Esko Ukkonen. Assigning probes into a small number of pools separable by electrophoresis. *Bioinformatics*, 18(1):199–206, January 2002.

[9] E.W. Knapik, A. Goodman, M. Ekker et al. A microsatellite genetic linkage map for zebrafish (Danio rerio). In *Nature Genetics*, 18:338-343, 1998.

[10] N. Kumar and N. Deo. Multidimensional interval graphs. *Congr. Numer.*, 102:45–56, 1994.

[11] T. A. McKee, F.R. McMorris. *Topics in intersection graph theory.* SIAM, 1999.

[12] W. S. Oetting, H. K. Lee, D. J. Flanders, G. L. Wiesner, T. A. Sellers, R. A. King. Linkage analysis with multiplexed short tandem repeat polymorphisms using infrared fluorescence and M13 tailed primers. In *Genomics*, 30:450-458, 1995.

[13] T. Pastinen, J. Partanen, A.C. Syvnen. Multiplex, fluorescent, solid-phase minisequencing for efficient screening of DNA sequence variation. In *Clin. Chem.*, 42:1391-1397, 1996.

[14] G.D. Schuler, M.S. Boguski, E.A. Stewart et al. A gene map of the human genome. In *Science*, 274:540-546, 1996.

[15] W. K. Shih and W. L. Hsu. An approximation algorithm for coloring circular-arc graphs. Technical report, Sinica, China, 1989.

[16] Jr. W. T. Trotter and F. Harary. On double and multiple interval graphs. *Journal of Graph Theory*, 3:205–211, 1979.

[17] D.B. West, D.B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. In *Discrete Applied Mathematics*, 8:295-305, 1984.

[18] Z. Yakhini, P. Webb, and R. Roth. *Partitioning of polymorphic DNAs.* US Patent 6,074,831, 2000.

[19] L. Zane, L. Bargelloni, and T. Patarnello. Strategies for microsatellite isolation: a review. *Molecular Ecology*, 11:1–16, 2002.