



SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

P_3 -Hull Numbers and Infection Times of Unit Interval Graphs

av

Alrik Sandberg

2024 - No K8

P_3 -Hull Numbers and Infection Times of Unit Interval Graphs

Alrik Sandberg

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Per Alexandersson

2024

Abstract

It is well known that the P_3 -hull number and the infection time are difficult to calculate for large graphs. We find that it is possible to calculate the P_3 -hull number, and find at least one contagious set for certain, specific, types of unit interval graphs. Furthermore we show that the number of unit interval graphs with a possible maximal infection time of $n - 2$ is constant for $n \geq 4$ vertices. These graphs with a maximal possible infection time also share a specific structure. This means that it is possible to easily find all unit interval graphs of this type for a given number of vertices. We also show that the number of unit interval graphs of size n with a P_3 -hull number of n is given by the $(n + 1)$ th Fibonacci number. Lastly, we show how the number of unit interval graphs consisting of n vertices that has P_3 -hull number equal to 2 is equal to 2 times the $(n - 2)$ th Catalan number.

Sammanfattning

Det är välkänt att P_3 -hull tal och infektionstiden är svårberäknade för stora grafer. Vi visar att det är möjligt att beräkna P_3 -hull tal och att hitta minst en smittsam mängd för vissa specifika typer av enhetsintervallgrafer. Vi visar också att antalet enhetsintervallgrafer med en längsta möjliga infektionstid $n - 2$ är konstant för grafer bestående av $n \geq 4$ noder. Dessa grafer med en längsta möjliga infektionstid har alla en specifik struktur. Detta gör det möjligt att lätt hitta alla enhetsintervallgrafer av denna typ för ett givet antal noder. Vi visar även att antalet enhetsintervallgrafer av storlek n med P_3 -hull tal n fås från Fibonacci talen. Till sist visar vi att antalet enhetsintervallgrafer bestående av n noder med ett P_3 -hull tal lika med 2 är två gånger det $(n - 2)$:a Catalantalet.

CONTENTS

1	Introduction	5
2	Background information	7
2.1	Infections in graphs	7
2.2	Unit interval graphs	10
3	Data	15
3.1	Infection times	15
3.2	P_3 -hull number	18
4	Conclusion	23
	References	25
	Appendix	27

1

INTRODUCTION

Infection spread in different types of graphs is something that has been extensively studied (see for example [FPR18, GPT⁺21, GGS21]). Two properties of particular interest are the P_3 -hull number of a graph as well as the associated *infection time*.

Unit interval graphs are a special type of undirected graphs. There does not exist a general formula for computing the P_3 -hull number nor the longest or shortest infection time for the P_3 -hull number for graphs within this family of graphs.

By using a program created in Python it is possible to find all possible *contagious sets* for unit interval graphs consisting of a given number of vertices. From these contagious sets it is then possible to calculate the P_3 -hull number as well as the infection times for all of the unit interval graphs. By collecting all this data for all unit interval graphs consisting of up to 10 vertices we want to investigate whether it is possible to more accurately calculate and predict both the exact P_3 -hull number and infection time. By analysing the data we also find some patterns that we generalise and prove.

2

BACKGROUND INFORMATION

2.1 INFECTIONS IN GRAPHS

We begin by defining some basic concepts and terms to be used when discussing infections in undirected graphs.

Definition 2.1 (Seed [FPR18]). A *seed* is an initially infected vertex in a graph G . A set of seeds, A_0 , consists of some initially infected vertices of the graph.

Seeds are not the only concept required to describe how an infection spreads in a graph. It is also important to describe how an infection spreads throughout a graph from the initial infected vertices. To do this we first need to define the *infection number*.

Definition 2.2 (Infection number [GPT⁺21, FPR18]). The *infection number*, r , is the number of infected neighbouring vertices required for the infection to spread to a new vertex.

Throughout this text we assume that we are working with an infection number of $r = 2$ unless otherwise stated.

It is now possible to describe an infection in a graph. An infection always begins with a set of seeds within a graph, as well as a stated infection number. The seeds spread the infection to any non-infected vertex with at least r number of infected neighbours. This process is then repeated until the infection has either spread to every vertex in the graph, or there are no more vertices that can be infected.

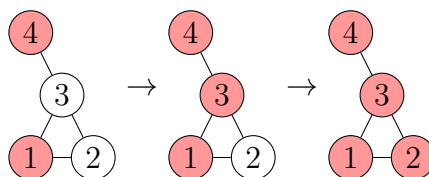


Figure 1: This shows an infection spreading through a graph with 4 vertices and $r = 2$, with $A_0 = \{1, 4\}$.

If a set of seeds manages to spread an infection throughout the whole graph, as in Figure 1, they form a *contagious set*.

Definition 2.3 (Contagious set [FPR18]). A **contagious set**, A_0 , of a graph G is a set consisting of seeds at time 0 that eventually fully infects G .

It is easy to see that all graphs must have at least one contagious set, since the set of all vertices of the graph always is a contagious set. However, as we can see in Figure 1 the set of all vertices is not necessarily the only contagious set of a graph. It is important to note that a graph can have contagious sets of different sizes. Most importantly, for any contagious set C and set of seeds S it holds that if $C \subseteq S$ then S is also a contagious set.

A contagious set has a two interesting properties. These are the size of the set as well as its *infection time*. We begin by defining the infection time.

Definition 2.4 (Infection time [FPR18, GGS21]). The **infection time**, $\tau(G)$, for a contagious set A_0 of a graph G is the time required for the contagious set A_0 to infect the whole of G . It is therefore the smallest t such that the set of infected vertices at time t is

$$A_t = V(G).$$

Note that $\tau(G)$ depends on A_0 and r , but these parameters are usually clear from the context.

The infection time of a graph can be understood as the number of time steps required for a contagious set to spread the infection through the whole graph. For the graph and contagious set shown in Figure 1 therefore, the infection time is $\tau(G) = 2$.

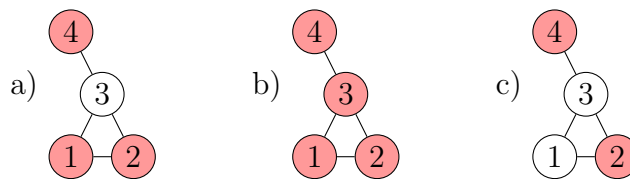


Figure 2: Three more contagious sets to the graph from Figure 1, all three of which have different sizes and infection times.

This brings us to the next important concept, namely the P_3 -hull number.

Definition 2.5 (P_3 -Hull Number [GPT⁺21, FPR18]). *The P_3 -hull number, $m(G, r)$, of a graph G is the cardinality of the smallest contagious set of G for the given infection number r .*

There exist some simple limitations to the P_3 -hull number and the infection time of a given graph, G . We begin with the P_3 -hull number. If the number of vertices of G is known then we have an upper limit to $m(G, r)$ that is easily calculated.

Corollary 2.6. *For a graph, G , its P_3 -hull number has to lie in the interval*

$$r \leq m(G, r) \leq |V(G)|$$

assuming $|V(G)| \geq r$.

For $r = 2$ this gives us an interval of

$$2 \leq m(G, 2) \leq |V(G)|$$

assuming that G has at least 2 vertices.

We can create a similar interval for the possible infection time of G .

Corollary 2.7. *Assuming that $|V(G)| \geq r$ then the graph, G , has*

$$\tau(G) \leq |V(G)| - r$$

and a contagious set, A_0 , of G has a maximum possible infection time, $\tau_{max}(G)$, of

$$\tau_{max}(G) \leq |V(G)| - |A_0|.$$

For $r = 2$ we therefore have

$$\tau(G) \leq |V(G)| - 2$$

and a minimum size for $|A_0| \geq 2$, again assuming that G consists of at least 2 vertices.

Unfortunately the intervals from Corollary 2.6 and Corollary 2.7 are not very useful as $|V(G)|$ becomes large. The intervals then become too wide to provide much useful information.

2.2 UNIT INTERVAL GRAPHS

The rest of this text is limited to looking at infections in a specific type of graphs called *unit interval graphs*. Unit interval graphs is a sub-family of regular *interval graphs* and we therefore begin by defining what these are.

Definition 2.8 (Interval graph [SZZ04]). An *interval graph* is an undirected graph $G(V, E)$ where each vertex, v , in G corresponds to an interval on the real number line, $i \subset \mathbb{R}$ and (u, v) is an edge of G if and only if the intersection $i_u \cap i_v$ is non-empty.

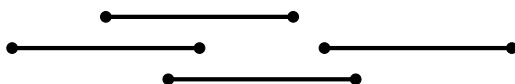


Figure 3: The interval realisation of the graph from Figures 1 and 2. Since there exists an interval realisation the graph is an interval graph.

The intervals in Figure 3 are all of equal length, 1. This means that the graph they represent is a special type of interval graph called a unit interval graph. Compared to regular interval graphs, unit interval graphs are more simple to work with since all distances are uniform within a graph.

Definition 2.9 (Unit interval graph [GGS21, SZZ04]). A *unit interval graph* is a type of interval graph where every interval has length 1. We always assume that vertices are labeled $1, \dots, n$ according to the order of the unit intervals, sorted from leftmost to rightmost.

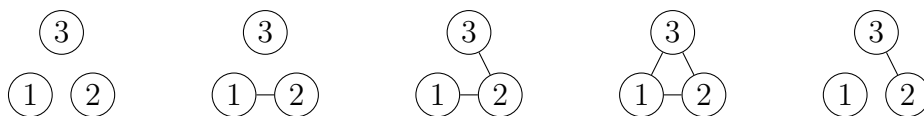


Figure 4: All possible unit interval graphs consisting of 3 vertices.

Definition 2.10. We denote the set composed of all unit interval graphs consisting of n vertices as $UIG(n)$.

This means that Figure 4 shows all graphs in the set $UIG(3)$.

There are several different ways to represent unit interval graphs and some of these are important to understand as they will be used in this text. *Dyck paths* are the representations that we use the most. They give us an easy to understand, visual way to show unit interval graphs.

Definition 2.11 (Dyck Path [Pon23]). A **Dyck path** is a walk in a $n \times n$ square grid, starting from $(0,0)$ and ending in (n,n) using steps $(1,0)$ (East, E) and $(0,1)$ (North, N), and weakly lying above the $y = x$ diagonal.

Dyck paths can be visualised as a set of squares of equal size. To stay above the diagonal $y = x$ the first step of any Dyck path must be a North step, and the last step must be East. These are not included as squares in the figures of Dyck paths shown. Therefore, the squares go from $(0,1)$ to $(n-1,n)$ with each square in a figure lying above the diagonal. Instead, the vertices are numbered from 1 to n and shown lying on the $y = x$ diagonal (see Figures 5 and 6 for examples). It is also possible to represent a Dyck path as a sequence of numbers representing the number of squares on each line. These sequences are also shown to the left of the Dyck paths (see for example the sequence 0, 1, 2, 1 for Figure 5).

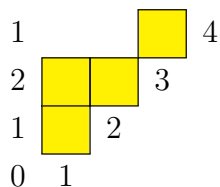


Figure 5: This shows the Dyck path of the unit interval graph from Figure 1. The Dyck path is NNNEENEE.

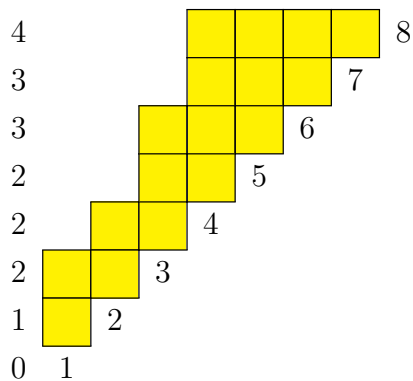


Figure 6: This shows the Dyck path of a unit interval graph with 8 vertices. The path is NNNENENNENNEEEEE.

The number of possible Dyck paths is given by the *Catalan number* (see A000108 in [Slo19]):

$$C_n = \frac{1}{n+1} \binom{2n}{n} \quad (1)$$

where n is the size of the grid of the Dyck path [Pon23]. We also note that the Catalan numbers for $0 \leq n \leq 8$ are

$$1, 1, 2, 5, 14, 42, 132, 429, 1430.$$

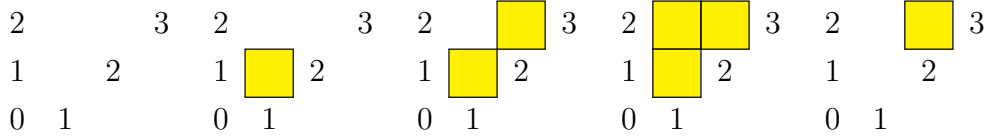


Figure 7: This shows all possible unit interval graphs consisting of 3 vertices (see Figure 4). The paths for the different graphs are NENENE, NNEENE, NNENEE, NNNEEE and NENNEE respectively.

In a Dyck path, every square belongs to the column of one vertex and the row of one vertex. When creating a graph from a Dyck path therefore, every square represents an edge between the column vertex and the row vertex (see Figure 4 and Figure 7). It is possible to prove (we omit the proof) that every such graph is a unit interval graph. We shall now prove the converse statement.

Theorem 2.12 (See e.g. [AP18]). *There exists a bijection between unit interval graphs and Dyck paths, meaning that every unit interval graph can be represented by a Dyck path. Therefore the number of unit interval graphs, $UIG(n)$, consisting of n vertices is the n th Catalan number C_n .*

From now on we use the Dyck path representation of a unit interval graph and the graph itself interchangeably.

Proof of Theorem 2.12. For any unit interval graph, G , consisting of n vertices it is possible to construct its corresponding Dyck path by starting at the first vertex. In order to stay above the diagonal $y = x$ we note that we have to take a North step to begin. Then there are two possibilities for vertex 1. It is either connected to vertex 2 or not. If they are connected we take one step North, and check if vertex 1 is connected to the next vertex.

We repeat this process until we reach a vertex, v , that is not connected to vertex 1. Since G is a unit interval graph, there must exist intersections between the intervals corresponding to all vertices $\{1, \dots, v - 1\}$. Therefore, all vertices in that set are connected to all other vertices in the set. Then, at vertex $v - 1$ we instead take a step East and check if vertex 2 is connected to vertex v . If it is we add another

North step and repeat the process from earlier, and if not we move one more step East. This can be repeated until we either check if v is connected to itself, in which case the answer always is yes, and we move North one step. We could also find that vertex 2 is connected to all vertices up to and including vertex n . We then move East until we have confirmed that no vertices are connected to vertex $n + 1$. It is impossible for any connection to exist since vertex $n + 1$ does not exist.

This process produces a walk from $(0, 0)$ consisting only of steps of type $(1, 0)$ or $(0, 1)$ that ends at (n, n) . It also never crosses below the diagonal $y = x$. This is the definition of a Dyck path (see Definition 2.11), which means that all unit interval graphs can be represented as Dyck paths. This means that any unit interval graph can be obtained from some Dyck path. \square

This means that the number of unit interval graphs we have to consider increases rapidly as the number of vertices increases.

Finally we introduce a theorem for calculating the P_3 -hull number for a very specific type of unit interval graph that we will call group connected graphs, using the Dyck path of the graph (see Figure 8 for an example).

Theorem 2.13. *Consider a connected unit interval graph, G whose Dyck path touches the diagonal, $x = y + 1$, c times. Assume also that every subgraph $S \subset G$, consisting of the parts of G between two touches of the diagonal $x = y + 1$ are complete with all vertices of S connected to each other. Then the minimum number of seeds required to fully infect G is given by*

$$m(G, 2) = 2 + \left\lfloor \frac{c}{2} \right\rfloor.$$

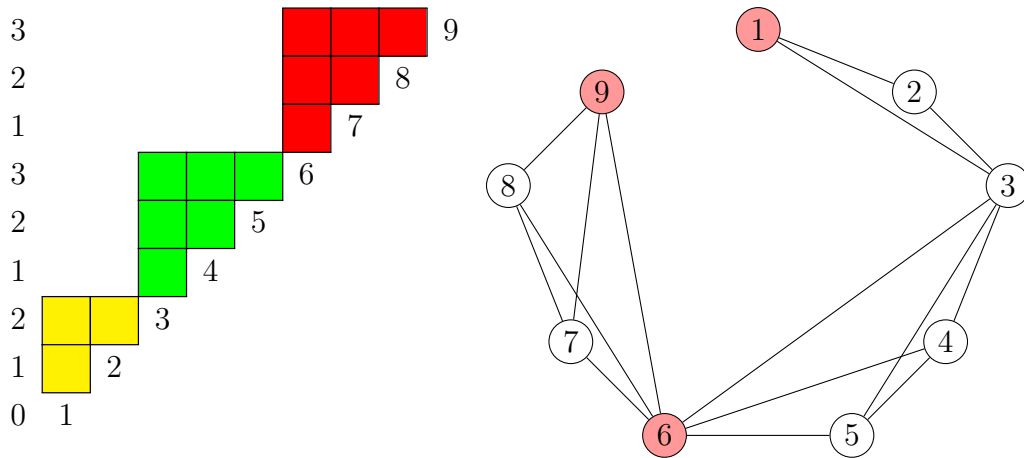


Figure 8: This shows a unit interval graph with $c = 2$ with all three components being complete sub-graphs. It has $m(G, 2) = 3$ and one of its contagious sets of that size is shown.

Proof of Theorem 2.13. We know that $m(G, 2) \geq 2$ since $r = 2$. Any time the Dyck path of G touches the line $x = y + 1$ it partitions the set of vertices into two components K_1, K_2 . Therefore, every time c is increased, another component is added and we have that $V(G) = K_1 \cup K_2 \cup \dots \cup K_{c+1}$ where every K_j is a complete subgraph of G . Any two components, i and $i + 1$, are only connected at a single vertex, $K_i \cap K_{i+1} = \{v_i\}$. We call v_i a connector between K_i and K_{i+1} . To fully infect one of these components, K_i , it is required that 2 of the vertices of K_i are infected.

The first and last components of G both only intersect one other component. Therefore both need to contain at least one seed each that are not v_1 or v_n to become infected. For the rest of the graph, if every component contains exactly one seed, s , the infection will be able to spread. This means that if every other v_i is a seed then it forms a contagious set.

Assume now that the number of components, K_i ($1 \leq i \leq n$) is odd, i.e. c is even. Then the number of connectors is even. Therefore the first or last component of G , K_1 or K_n , will have to contain a connector as a seed in order for every other connector to be a seed. This means that K_1 or K_n has to contain 2 seeds. It is then possible to add another component K_{n+1} without increasing the P_3 -hull number. Therefore, we have that

$$m(G, 2) = 2 + \left\lfloor \frac{c}{2} \right\rfloor.$$

See Figure 8 for an example. □

3

DATA

In order to investigate how the distribution of the infection time and P_3 -hull number of unit interval graphs of different number of vertices differ it is first required to perform a lot of calculations. To do this in a reasonable amount of time we decided to create and use a Python program (see Appendix 1).

This program allowed us to construct all possible unit interval graphs of a given number of vertices and then calculate their respective P_3 -hull number. It also calculated the infection time of each contagious set of the size of their P_3 -hull number which allowed us to find both the longest and shortest infection time as well as the P_3 -hull number of every unit interval graph up to size 10.

The data for $UIG(n)$ for $n \in \{3, \dots, 10\}$ vertices are presented in Tables 1, 2 and 3.

3.1 INFECTION TIMES

n	Time:	0	1	2	3	4	5	6	7	8
3		3	2	0	0	0	0	0	0	0
4		5	7	2	0	0	0	0	0	0
5		8	24	8	2	0	0	0	0	0
6		13	68	37	12	2	0	0	0	0
7		21	186	150	52	18	2	0	0	0
8		34	489	574	205	100	26	2	0	0
9		55	1271	2069	792	466	175	32	2	0
10		89	3274	7170	3054	1978	934	257	38	2

Table 1: Lists the shortest infection time possible for contagious sets the size of P_3 -hull number of all unit interval graphs, G in $UIG(n)$.

Looking at Table 1 there are exactly two unit interval graphs with a shortest infection time equal to the longest possible infection time for the given G is

$$\tau(G) = V(G) - 2 \quad (\text{Corollary 2.7}) \quad (2)$$

for $n \geq 3$. This also holds for $n = 2$, however this is a special case. When we have the infection number $r = 2$ then all graphs of size $n = 2$ must have an infection time, $\tau = 0$ and P_3 -hull number, $m = 2$. Since the number of possible unit interval graphs consisting of two vertices is equal to 2 this gives us the 2 graphs with a shortest (and longest) infection time of

$$\tau(G) = 2 - 2 = 0. \quad (3)$$

We can now formulate a theorem based on these observations.

Theorem 3.1. *For any $n \geq 2$ and infection number $r = 2$, any unit interval graph $G \in \text{UIG}(n)$ has a maximum possible infection time of $\tau_{\max}(G) = n - 2$. Moreover, there are exactly two graphs in $\text{UIG}(n)$ with a minimum infection time equal to this number.*

The proof of Theorem 3.1 is shown later.

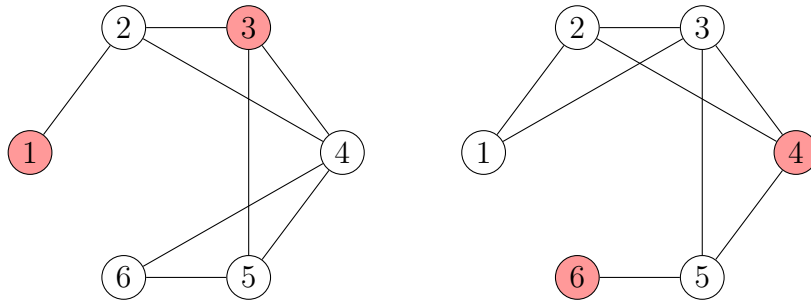


Figure 9: The only two unit interval graphs with $\tau_{\min}(G) = \tau_{\max}(G) = 4$ for $n = 6$ as mentioned in Theorem 3.1, along with one of their contagious sets.

There are similar patterns that occur when looking at the data in Table 2. There exists exactly 3 unit interval graphs with a maximal infection time for any n , when $n \geq 4$, that satisfies equation (2). It is thus possible to formulate a theorem similar to Theorem 3.1.

Theorem 3.2. *For any $n \geq 4$ and infection number $r = 2$, any unit interval graph $G \in \text{UIG}(n)$ has a maximum possible infection time of $\tau_{\max}(G) = n - 2$. Moreover, there are exactly 3 graphs with a maximum infection time equal to this number.*

n	Time:	0	1	2	3	4	5	6	7	8
3		3	2	0	0	0	0	0	0	0
4		5	6	3	0	0	0	0	0	0
5		8	16	15	3	0	0	0	0	0
6		13	40	50	26	3	0	0	0	0
7		21	96	152	121	36	3	0	0	0
8		34	224	445	454	224	46	3	0	0
9		55	514	1275	1555	1042	363	55	3	0
10		89	1166	3593	5107	4212	2043	519	64	3

Table 2: Lists the longest infection time possible for contagious sets the size of the P_3 -hull number of all unit interval graphs, G in $UIG(n)$.

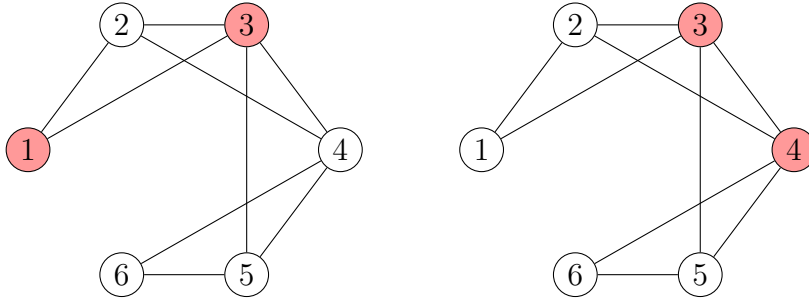


Figure 10: The only unit interval graph with $\tau_{max}(G) = 4$ for $n = 6$ as mentioned in Theorem 3.2 that does not have $\tau_{max}(G) = \tau_{min}(G)$, along with two of its contagious sets, one with $\tau(G) = 4$ and the other with $\tau(G) = 2$.

Graphs can have more than one contagious set of size $m(G, 2)$. Therefore it is not unexpected that some unit interval graphs with a contagious set with a maximal possible infection time also have another contagious set of the same size with a shorter infection time. This is something that is apparent from Theorem 3.1 and Theorem 3.2, as one of the 3 graphs with a maximal infection time also has another contagious set with a shorter infection time. This leads us to the proof of Theorem 3.1 and Theorem 3.2.

Proof of Theorem 3.1 and Theorem 3.2. In order to have a unit interval graph, G , with $\tau = n - 2$ the infection must spread to exactly one new vertex per time step. This means that G needs to be connected with $m(G, 2) = 2$.

Since we have $r = 2$ this means that all vertices except the first and last of G are required to have exactly 3 neighbours to spread the infection slowly enough. The first and last vertex are required to have either 3 or 4 neighbours combined. In order

for the graph to be connected this means that they either both have 2 neighbours or one has 1 neighbour and the other 2.

All of this leaves us with only three possible types of graphs. Firstly, we have the single graph that exists for any $n \geq 3$ where the first and last vertex both have 2 neighbours (see Figure 10 for an example). Secondly, we have graphs for $n \geq 4$ where either the first or last vertex only has 1 neighbour, while the other has 2. There exists exactly two of these graphs for any n . An example of these are shown in Figure 11. In total this leaves us with three graphs for each $n \geq 4$ that have a contagious set with a maximal infection time of $\tau(G) = n - 2$.

We can now easily see that the graph where both the first and last vertices have 2 neighbours can have a shorter infection time than $\tau = n - 2$ if neither of the two seeds are the first or last vertex of G . However, the other two graphs only have contagious sets of size 2 with $\tau = n - 2$. This is because it is required that the vertex with only one neighbour is a seed in order to spread the infection throughout the whole graph. \square

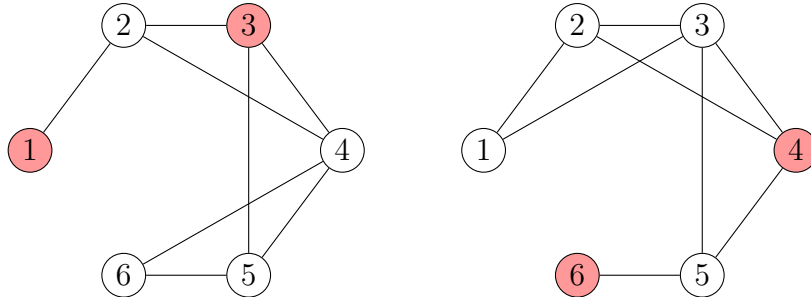


Figure 11: The two unit interval graph with $\tau_{max}(G) = 4$ for $n = 6$ as mentioned in Theorem 3.2 that do have $\tau_{max}(G) = \tau_{min}(G)$, along with a contagious set for each.

3.2 P_3 -HULL NUMBER

For the distribution of the P_3 -hull number in Table 3 there is an immediate, familiar number sequence that also appears in both Table 1 and Table 2, namely 3, 5, 8, 13, 21, 34, 55, 89. This is the Fibonacci sequence, F_n , for $4 \leq n \leq 11$ (see A000045 in [Slo19]). Therefore we formulate the next theorem.

Theorem 3.3. *Given the infection number, $r = 2$, the total number of graphs, $G \in UIG(n)$ with $V(G) = n$, that has a P_3 -hull number equal to n is equal to the*

n	P_3 -hull number:	2	3	4	5	6	7	8	9	10
3		2	3	0	0	0	0	0	0	0
4		4	5	5	0	0	0	0	0	0
5		10	12	12	8	0	0	0	0	0
6		28	33	33	25	13	0	0	0	0
7		84	98	98	78	50	21	0	0	0
8		264	306	306	250	174	96	34	0	0
9		858	989	991	822	598	369	180	55	0
10		2860	3279	3297	2763	2065	1355	757	331	89

Table 3: Lists the distribution of the P_3 -hull number of all unit interval graphs, G in $UIG(n)$.

$(n + 1)$ th Fibonacci Number. That is

$$|\{G \in UIG(n) : m(G, 2) = n\}| = F_{n+1}.$$

This sequence appears in all three tables since all unit interval graphs with a P_3 -hull number, $m(G, 2) = V(G)$ must have a maximal and minimal infection time $\tau(G) = 0$ since all vertices are infected from the start. This observation helps with the proof of Theorem 3.3.

Proof of Theorem 3.3. For any graph, G , to have a P_3 -hull number, $m(G, r)$, equal to its number of vertices, there cannot be any way for an infected vertex to spread the infection to another vertex. Therefore, since $r = 2$, no vertex within G can have more than one neighbour.

In order for G to have $m(G, 2) = V(G)$ all vertices in G have to be partitioned into blocks consisting of one or two vertices each where each vertex is connected only to other vertices within the same block.

This means that in order to describe all possible graphs with n vertices we need to look at all possible combinations of 1s and 2s that add up to n . The Fibonacci number F_n can be used to describe the number of sequences consisting of 1s and 2s that add up to $n - 1$ (see [A000045](#) in [\[Slo19\]](#)). Therefore the total number of graphs with $m(G, 2) = V(G)$ where $V(G) = n$ is equal to the $(n + 1)$ th Fibonacci number. \square

The number of unit interval graphs with P_3 -hull number, $m(G, 2) = 2$, for a given number of vertices shown in Table 3, column one, is twice the sequence of Catalan

numbers for $0 \leq n \leq 8$ (see [A068875](#) [Slo19]). This leads us to Theorem 3.4.

Theorem 3.4. *For any set of unit interval graphs $UIG(n)$ consisting of n number of vertices, the number of graphs, $G \in UIG(n)$, with $m(G, 2) = 2$ is equal to two times the $(n - 2)$ th Catalan number, that is*

$$|\{G \in UIG(n) : m(G, 2) = 2\}| = 2 \cdot C_{n-2}$$

for any $n \geq 2$.

Before we show the proof of Theorem 3.4 we first introduce a new lemma.

Lemma 3.5. *Any connected unit interval graph, G consisting of at least 2 vertices, whose Dyck path touches the diagonal $x = y + 1$ at most once, has $m(G, 2) = 2$.*

Lemma 3.5 differs from Theorem 2.13 for $c = 1$ by applying to graphs even where one or both components is not complete.

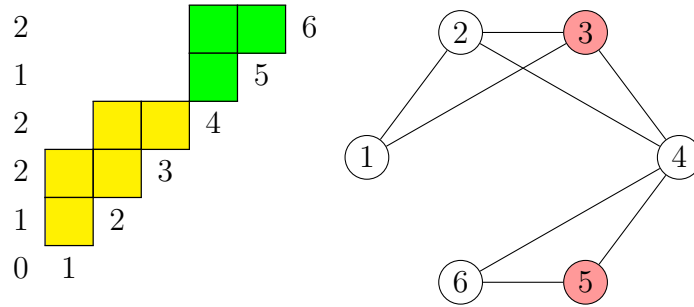


Figure 12: Shows one example of a unit interval graph with $c = 1$. From the Dyck path we can see that the yellow component is not complete. One example of a contagious set of size 2 is $\{3, 5\}$.

Figure 12 shows one unit interval graph where Lemma 3.5 is applicable while Theorem 2.13 is not. We now give the proof of the lemma.

Proof of Lemma 3.5. From Theorem 2.13 we know that if the two components that G is divided into by the touching of the diagonal are both complete, then $m(G, 2) = 2$. Therefore we only need to show that this also holds if one or both components are not complete.

Since we know that G is connected it is enough that each component has 2 infected vertices that share at least one neighbour for the infection to spread to all

vertices of the component. We also know that both components share one vertex, k .

Therefore, if we place one seed in each component, where they both are connected with vertex k , then k will become infected. This means that both components will have 2 infected vertices that share at least one neighbour, a seed and k , which in turn will spread the infection to all other vertices of G . \square

This leads us to the proof of Theorem 3.4.

Proof of Theorem 3.4. We begin by looking at what properties unit interval graphs with $m(G, 2) = 2$ need to have. Since $r = m(G, 2)$ the graphs all need to be connected. Lemma 3.5, the number of times, c , the Dyck path of the graphs can touch the diagonal $x = y + 1$ is at most 1 since the removal of edges can only increase the P_3 -hull number.

First, we calculate the number of connected unit interval graphs of size n with $c = 0$. In order for a unit interval graph to not touch the diagonal at all, it is required that it is possible to remove 2 East steps from the top right and 2 North steps from the bottom left of the Dyck path. This is equivalent to taking all unit interval graphs of size $n - 2$ and adding these steps. Therefore there are C_{n-2} connected unit interval graphs of size n with $c = 0$.

We then need to calculate the number of connected unit interval graphs with $c = 1$. It is possible to construct all such unit interval graphs by taking all unit interval graphs of size $n - 2$ and splitting their Dyck path representation into two subgraphs. The first subgraph consists of the path of the unit interval graph until the first time it touches the $x = y + 1$ diagonal, and the second subgraph consists of the rest of the graph. We then raise the second subgraph as earlier, by adding 2 East steps to the top right and 2 North steps to the bottom left of the Dyck path of the graph. This will give us all possible unit interval graphs with $c = 1$ and there are exactly C_{n-2} of these graphs.

All in all this leaves us with $C_{n-2} + C_{n-2} = 2 \cdot C_{n-2}$ unit interval graphs of size n with P_3 -hull number equal to 2. \square

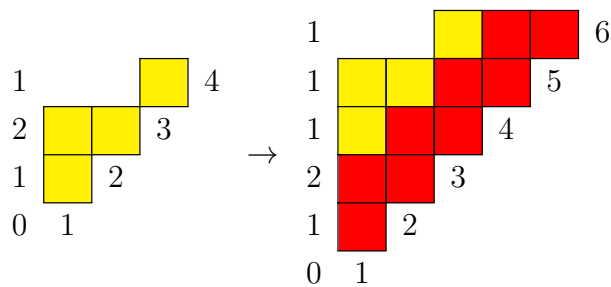


Figure 13: Shows an example of the first type of Dyck path transformation, for $c = 0$, mentioned in the Proof of Theorem 3.4 with the added parts coloured in red.

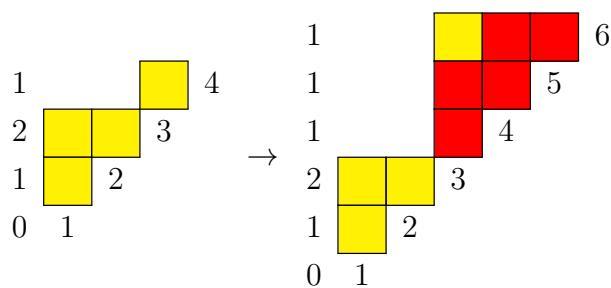


Figure 14: Shows an example of the second type of Dyck path transformation, for $c = 1$, mentioned in the Proof of Theorem 3.4 with the added parts coloured in red.

CONCLUSION

We have studied how to calculate the P_3 -hull number for different types of unit interval graphs. The distribution of the P_3 -hull number over the whole set of unit interval graphs of size n has also been investigated, and several patterns have been discussed. We have also examined the distribution of the infection times for contagious sets the size of the P_3 -hull number of unit interval graphs.

There are several avenues for continuing this work. One of the most interesting is to try to find a formula for calculating the P_3 -hull number of more general, connected, unit interval graphs than those consisting only of complete components.

REFERENCES

- [AP18] Per Alexandersson and Greta Panova. LLT polynomials, chromatic quasisymmetric functions and graphs with cycles. *Discrete Math.*, 341(12):3453–3482, December 2018.
- [FPR18] Daniel Freund, Matthias Poloczek, and Daniel Reichman. Contagious sets in dense graphs. *Eur. J. Comb.*, 68:66–78, February 2018. URL: <https://doi.org/10.1016/j.ejc.2017.07.011>.
- [GGS21] Lucia M González, Luciano N Grippo, and Martin D Safe. Formulas in connection with parameters related to convexity of paths on three vertices: caterpillars and unit interval graphs. *Australasian Journal of Combinatorics*, 79(3):401–423, 2021.
- [GPT⁺21] Luciano N. Grippo, Adrián Pastine, Pablo Torres, Mario Valencia-Pabon, and Juan C. Vera. On the P_3 -hull number of Kneser graphs. *The Electronic Journal of Combinatorics*, 28(3), July 2021. doi:10.37236/9903.
- [Pon23] Viviane Pons. A description of the Zeta map on Dyck paths area sequences. working paper or preprint, March 2023. URL: <https://hal.science/hal-04024002>.
- [Slo19] Neil J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. Online, 2019. URL: <https://oeis.org>.
- [SZZ04] Sayyed Bashir Sadjad and Hamid Zarrabi-Zadeh. Unit interval graphs, properties and algorithms. <http://www.cs.haifa.ac.il/~golumbic/courses/seminar-2013graph/sadjad-unit-interval04.pdf>, 2004. Accessed: 2023-12-10.

APPENDIX

Listing 1: Python Code

```
import itertools

def vertices(n): # Creates an ordered list of vertices from
    1 to n.
    return list(range(1, n+1))

def ui_graph(connector_list, vertex_list): # Creates a
    graph from a sorted connection list and a sorted list of
    vertices for a unit interval graph. Returns a list of
    edges of the created graph.

    graph = [] # Creates a list of edges of the graph

    for i in vertex_list:
        edges = connector_list[i - 1]

        while edges > 0:
            new_edge = (i - edges, i)
            graph.append(new_edge)
            edges -= 1

    return graph

def infection_sim(graph, seeds, vertex_list): # Calculates
    the infection spread in the given graph with the given
    seeds. Returns the time it takes for equilibrium to be
    reached as well as the equilibrium graph.

    time = 0
    old_infected = seeds
```

```

while True:
    new_infected = []
    new_infected += old_infected

    for i in vertex_list:
        infected_neighbours = []

        if i not in old_infected:
            neighbours = [edge for edge in graph if i in
                           edge]

            for j in old_infected:
                infected = [edge for edge in neighbours
                              if j in edge]

                if infected:
                    infected_neighbours.append(
                        infected)

            if len(
                infected_neighbours) > 1:
                new_infected.append(i)

    new_infected.sort()

    if len(new_infected) == len(old_infected):

        if old_infected == vertex_list: # Checks if the
            infection has spread to the whole graph
            spread = True

        else:
            spread = False

        return old_infected, time, spread

    else:

```



```

        old_infected = new_infected
        time += 1

def new_layer(lists): # Creates the next layer of the
    connector list of the unit interval graph.
    next_layer = []

    for i in range(0, lists[-1] + 2):
        new_list = []
        new_list += lists
        new_list.append(i)
        next_layer.append(new_list)

    return next_layer

def connector_generator(n): # Generates all possible
    connector_list for a unit-interval graph with the given
    number of vertices.

    if n == 0:
        return []

    else:
        connectors = [[0]]
        connector_list = []

        for i in range(0, n):
            connector_list = connectors
            connectors = []

            for j in range(0, len(connector_list)):
                new_connectors = new_layer(connector_list[j]
                    ]) # Creates the next layer of the
                    connectors.
                connectors += new_connectors # Creates a

```

list of all possible connector_lists
including the next layer.

```
return connector_list
```

```
def seed_generator(vertex_list, no_seeds): # Generates all  
possible combinations of seeds for a given number of  
starting seeds and returns them as a list of lists.
```

```
tuple_seeds = list(itertools.combinations(vertex_list,  
no_seeds)) # Creates a list of tuples of all  
possible combinations of seeds of the given number.
```

```
seed_list = [list(seed) for seed in tuple_seeds] #  
Transforms the tuple list of seeds into a list of  
lists of seeds.
```

```
return seed_list
```

```
def run(no_vertices):
```

```
vertex_list = vertices(no_vertices)  
connector_list = connector_generator(no_vertices)  
results = []
```

```
for connector in connector_list:  
graph = ui_graph(connector, vertex_list)
```

```
for i in range(2, no_vertices + 1):  
contagious_sets = []  
seed_list = seed_generator(vertex_list, i)
```

```
for seeds in seed_list:  
spread, time, completion = infection_sim(  
graph, seeds, vertex_list)
```

```
if completion:  
contagious_sets += [seeds, time]
```

```

        if len(contagious_sets) > 0:
            data_point = (connector, contagious_sets)
            results += [data_point]
            break

    return results

def connector_to_dyck_path(connector_list): # Transforms a
connector_list into its corresponding binary dyck_path.
    length = len(connector_list)
    dyck_path = (2 * length) * [0]

    for i in range(length):
        dyck_path[(2 * i) - connector_list[i]] = 1

    return dyck_path

def data_compiler(raw_data): # Takes in raw data and
calculates the hull number as well as the longest and
shortest infection times for each data point.
    compiled_data_list = []

    for data_point in raw_data:
        hull_number = len(data_point[1][0])
        shortest_infection_time = len(data_point[0])
        longest_infection_time = 0

        for i in range(len(data_point[1])):

            if i % 2 == 1:
                time = data_point[1][i]

                if time > longest_infection_time:

```

```

        longest_infection_time = time

        if time < shortest_infection_time:
            shortest_infection_time = time

    compiled_data = (data_point[0], [hull_number,
        shortest_infection_time, longest_infection_time])
    compiled_data_list.append(compiled_data)

return compiled_data_list

def data_analysis(data): # Takes in a compiled data list
and returns the time distribution for the longest and
shortest infection time for the hull number.
    vertices = len(data[0][0])
    time_distribution_short = vertices * [0]
    time_distribution_long = vertices * [0]
    hull_number_distribution = (vertices + 1) * [0]

    for i in data:
        time_distribution_short[i[1][1]] += 1
        time_distribution_long[i[1][2]] += 1
        hull_number_distribution[i[1][0]] += 1

return time_distribution_short, time_distribution_long,
    hull_number_distribution

```