UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences
Département d'Informatique

Characterization and complexity of Thin Strip Graphs

Abdeselam El-Haman Abdeselam

Promotor : Prof. Jean Cardinal

Master Thesis in Computer Sciences

Academic year 2018 - 2019

You may want
to write a dedication here

“*You may also include one or more general quotes related to your topic.*”

Name of the author, date

“*Another quote.*”

Name of the author, date

Acknowledgment

I want to thank ...

Contents

Introduction

Talk about what this is etc...

Chapter 1

Background

1.1 Graphs and intersections

Classes hereditaires

1.1.1 Graphs

A graph G is defined as $G = (V, E)$, where V is the set of vertices and E the set of edges, where $E \subseteq \binom{V}{2}$. The vertices $v, w \in V$ such that $e = vw \in E$ links are called the *endpoints* of e.

Definition 1 *An embedding of a graph G into a surface Σ is a mapping of G in Σ where the vertices correspond to distinct points and the edges correspond to simple arcs connecting the images of their endpoints. [**?**].*

A graph G is planar if there is an embedding of this graph that does not have any crossing between the edges.

Definition 2 *Let $G = (V, E)$ and $S \subset V$, an induced subgraph is a graph H of G whose vertex set is S and its edge set $F = \{vw : v, w \in S, vw \in E\}$.*

Definition 3 *Let $G = (V, E)$ its complement graph $\overline{G}$ is the graph such that its edge set is defined as: $\{vw : v, w \in V, vw \notin E\}$.*

Definition 4 *H is called a minor of G if H can be constructed by deleting edges and vertices, or contracting edges.*

Theorem 5 (Kuratowski) *A graph G is planar if and only if it does not contain K_5 or $K_{3,3}$ as a minor or a induced subgraph.*

Definition 6 *A path P_n in a graph G is a sequence of vertices $v_1 v_2 v_3 \ldots v_n$ such that $v_i v_{i+1} \in E$.*

Definition 7 *A cycle C_n in a graph $G = (V, E)$ is a path $v_1 \dots v_n$ such that $v_1 = v_n$.*

Definition 8 *A chord of a cycle C_n with $n \geq 4$ is an edge that connects two non consecutive vertices of C_n.*

Definition 9 *A triangular chord of a cycle is a chord that will create a new triangle (C_3).*

Definition 10 *A graph $G = (V, E)$ is complete if every pair of distinct $v_1, v_2 \in V$ are adjacent. This is denoted K_n with n the size of the graph. If G is an induced graph of H then G is a clique of H.*

Definition 11 *A graph G is bipartite if there exist two disjoint subsets $A, B \subset V$ such that $A \cup B = V$ and each edge $e \in E$ has an endpoint on A and the another on B.*

Definition 12 *A bipartite graph G with bipartitions A and B is complete bipartite if every pair of vertices $v \in A, w \in B$ are adjacent. It is denoted as $K_{n,m}$, being n and m the size of each bipartition.*

Definition 13 *An induced forbidden subgraph of a graph class X is a graph such that if it is the induced subgraph of a graph G, we know that $G \notin X$.*

The coloration of a graph is a color assignment to each vertex such that the color of the two endpoints of every edge of the graph is different.

Definition 14 *The chromatic number of a graph $\chi(G)$ is the smallest number of colors needed to have an acceptable coloration of G.*

Definition 15 *The clique number of a graph $\omega(G)$ is the size of the biggest clique of G. We can observe that for every graph: $\chi(G) \geq \omega(G)$.*

Definition 16 *A perfect graph is a graph that respects this condition for every induced subgraph:*

$$\omega(G) = \chi(G)$$

Theorem 17 (Lovasz) *G is perfect if and only if $\overline{G}$ is perfect.*

1.1.2 Intersection graphs

Definition 18 *The intersection graph of a collection ζ of objects is the graph (ζ, E) such that $c_1c_2 \in E \Leftrightarrow c_1 \cap c_2 \neq \varnothing$.*

Definition 19 *A partial order is a binary relation $\leq$ over a set A satisfying these axioms:*

- *$a \leq a$ (reflexivity).*

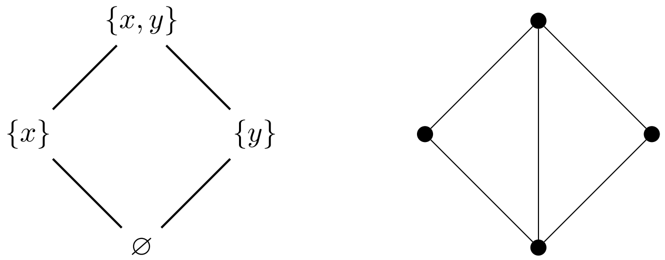


Figure 1.1: On the left, Hasse diagram of a poset of the power set of 2 elements ordered by inclusion. On the right, the comparability graph of this poset.

- *if $a \leq b$ and $b \leq a$ then $a = b$ (antisymmetry).*
- *if $a \leq b$ and $b \leq c$ then $a \leq c$ (transitivity).*

Definition 20 *A partially ordered set or poset $(S, \leq)$ where S a set and $\leq$ a partial order on S.*

Definition 21 *A graph $G = (V, E)$ is a comparibility graph if there exists a partial order $\leq$ such that $vw \in E \Leftrightarrow v \leq w$ or $w \leq v$. Equivalently, G is a comparability graph if it is the comparability graph of a poset. For example, the Hasse diagram (figure **??**) is a comparability graph where the relation is inclusion.*

Definition 22 *A graph $G = (V, E)$ is a co-comparability graph if its complement is a comparability graph.*

Interval graphs

An interval graph is a graph G that is the intersection graph of a collection of closed intervals in $\mathbb{R}$. If the length of each interval is unitary, then G is a unit interval graph (UIG).

Theorem 23 *G is an interval graph if and only if every simple cycle of four or more points has a chord. [**?**]*

Theorem 24 *An interval graph is a unit interval graph if and only if it has no induced subgraph $K_{1,3}$ [**?**].*

Another interesting class of interval graphs are mixed unit interval graphs, where each interval can be closed, open, open-closed or closed-open. In this paper we will denote those four classes like this:

$$\mathcal{I}^{++} = \{[x, y] : x, y \in \mathbb{R}, x \leq y\}$$
$$\mathcal{I}^{--} = \{(x, y) : x, y \in \mathbb{R}, x \leq y\}$$

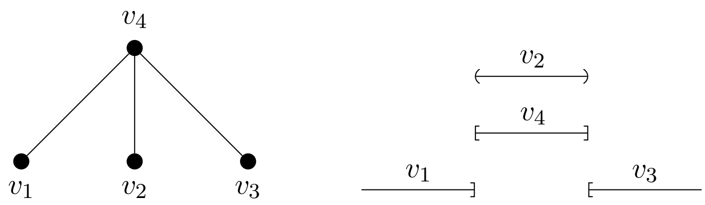


Figure 1.2: Representation of $K_{1,3}$ as a MUIG.

$$\mathcal{I}^{+-} = \{[x, y) : x, y \in \mathbb{R}, x \leq y\}$$
$$\mathcal{I}^{-+} = \{(x, y] : x, y \in \mathbb{R}, x \leq y\}$$

$\mathcal{I}$ will be replaced by $\mathcal{U}$ when we are talking about unit mixed interval graphs and their class is denoted MUIG.

Theorem 25 *The classes of the graphs $\mathcal{U}^{--}$, $\mathcal{U}^{++}$, $\mathcal{U}^{-+}$, $\mathcal{U}^{+-}$, and $\mathcal{U}^{-+} \cup \mathcal{U}^{+-}$ are the same (equivalent for $\mathcal{I}$). [?]*

Unlike for UIG class, $K_{1,3}$ is a MUIG as seen in figure **??**. Some characterizations have been already found for these classes of graphs [?] [?].

Exploit these characterizations!! $\rightarrow$ explain them and use them to characterize UUIG.

Disks

A disk graph G is a graph that is an intersection graph of disks on the plane, when the size of the disk is unitary, we talk about unit disk graphs. This class of graphs is important for this thesis, as thin strip graphs are a sub-class of unit disk graphs.

We will refer to the unit disk graph class as UDG and an example of a realization can be found in the figure **??**.

Induced forbidden subgraphs The characterization of this class with respect to its induced forbidden subgraphs has been studied [?].

Theorem 26 (Atminas-Zamaraev) *For every integer $k > 1$, $\overline{K_2 + C_{2k+1}}$ is a minimal induced subgraph.*

Theorem 27 (Atminas-Zamaraev) *For every integer $k > 4$, $\overline{C_{2k}}$ is a minimal induced subgraph.*

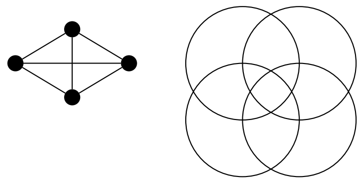


Figure 1.3: Realization of a UDG (Unit Disk Graph).

1.2 Complexity

Complexity theory has the objective to establish lower bounds on how efficient an algorithm can be for a given problem [?]. This approach let us have a reference point to establish the difficulty of a problem.

Definition 28 *Let Σ be a finite alphabet, Σ^* every word derived from Σ, $L \subseteq \Sigma^*$ is a decision problem.*

Definition 29 *A decider for a decision problem A is an deterministic algorithm V where*

$$A = \{w | V \textit{accepts } w\}$$

A is polynomially decidable if it has a polynomial time decider [?].

Definition 30 *A verifier for a decision problem A is an deterministic algorithm V where*

$$A = \{w | V \textit{accepts } \langle w, c\rangle \textit{ for some string } c\}$$

A is polynomially verifiable if it has a polynomial time verifier [?].

1.2.1 P vs NP

Definition 31 *A problem $L \in \mathcal{P}$ if L is polynomially decidable.*

Definition 32 *A problem $L \in \mathcal{NP}$ if L is polynomially verifiable. Thus, $\mathcal{P} \subseteq \mathcal{NP}$.*

To prove a bound of complexity on an unknown problem L we have to find another problem with already known complexity and find equivalences between those two. This can be achieved through *reductions*.

Definition 33 *A reduction of a problem L to a problem M is a mapping of an instance of L (I_L) to an isntance of M (I_M) such that I_L is true for the problem L if and only if I_M is true for the problem M. This is noted $L \leq M$ and $L \leq_P M$ if the reduction is done in polynomial time.*

With this concept we can define new complexity classes. $\mathcal{NP}$-hard is the set of problems so that we can reduce every $\mathcal{NP}$ problem to. The set of problems that are both $\mathcal{NP}$-hard and $\mathcal{NP}$ are called $\mathcal{NP}$-complete. This is generalized to every complexity class ($\mathcal{P}$, $\exists\mathbb{R}$, RP, etc...)

Satisfiability problem The satisfiability problem (SAT) is to decide the satisfiability of a CNF formula ϕ. A CNF formula is a boolean formula that is a conjunction of multiple clauses c_k. A clause is a disjunction of multiple literals. A literal may be a variable or a negation of a variable.

Theorem 34 (Cook-Levin) *SAT is $\mathcal{NP}$-complete.*

Clique problem The clique problem is to find a maximum clique of a graph G.

Theorem 35 *CLIQUE is $\mathcal{NP}$-complete. [?]*

Theorem 36 *CLIQUE is QPTAS when applied to disk graphs. [?]*

Theorem 37 (Clark-Colbourn) *CLIQUE is $\mathcal{P}$ when applied to unit disk graphs. [?]*

1.2.2 $\exists\mathbb{R}$ complexity class

$\exists\mathbb{R}$ is the class that describes the problems that can be reduced to *the existential theory of the reals*[?]. The existential theory of the reals is the problem of deciding if a sentence of this form is true:

$$(\exists X_1 \dots \exists X_n) : F(\exists X_1, \dots, \exists X_n)$$

where F is a quantifier-free formula in the reals. In other words, it is a conjuntion of clauses where each clause is a real polynomial inequality where each variable X_k is a real number. We can see that ETR is NP-hard because SAT can be reduced to it.

Proof. Let's take an instance of SAT ϕ_{SAT} with clauses c_k and variables x_k, we can construct an instance of ETR ϕ_{ETR} where we can construct variables in the domain $\{0, 1\}$ with this equality, so for each variable X_k:

$$X_k - X_k^2 = 0$$

Each literal of each clause will be positive or negative depending if the literal is cancelled in ϕ_{SAT}:

$$x_k \to l = X_k$$
$$\neg x_k \to l = (1 - X_k)$$

Then for each clause we can have a polynomial for which the sum of the values of every literal in the clause must be greater than one, so that at least one literal is true:

$$\sum_{l \in c_k} l \geq 1$$

With this proof, it is easy to see that ϕ_{ETR} is valid if and only if ϕ_{SAT} is also valid. $\square$

This result can show us that $P \subseteq NP \subseteq \exists\mathbb{R}$.

Problems in $\exists\mathbb{R}$

In this section we will describe some problems that are $\exists\mathbb{R}$-complete and will give an overview of the proof.

The art gallery problem Given a simple polygon P (without crossings between every side), we introduce *guards*. A guard g is a point such that every point of the polygon is watched by a guard. A point p is watched by a point q if the segment pq is contained in P. The subset G, being G the set of guards and $G \subseteq P$, is optimum if it has the minimal cardinality covering the whole polygon.

The art gallery problem is to decide, given a polygon P and a number of guards k, whether there exists a configuration of k guards in G guarding the whole polygon. The art gallery problem is $\exists\mathbb{R}$-complete [**?**].

Proof idea First of all, we can see that the art gallery problem is in $\exists\mathbb{R}$ if we reduce this problem to ETR. If we have an instance (P, k) of the art gallery problem we can have a formula [**?**] like this:

$$\phi = \{\exists x_1 y_1, \dots x_k y_k \forall p_x p_y : \text{INSIDE-POLYGON}(p_x, p_y) \to \bigvee_{1 \leq i \leq k} \text{SEES}(x_i, y_i, p_x, p_y)\}$$

Where INSIDE-POLYGON returns 1 if $(p_x, p_y) \in P$ and SEES returns 1 if the segment $(x, y)(p_x, p_y) \in P$. ϕ is not a ETR formula, so we would like to construct a quantifier-free formula with the idea of ϕ. To achieve this, the main idea is to have a small set of points $Q \subseteq P$ such that if these points are watched, the whole polygon is watched. This subset Q is called the *witness set*. The only thing is now to create a polynomial for each point that ensures that the point is watched by a guard.

To finish the proof we have to prove that the art gallery problem is $\exists\mathbb{R}$-hard. For this part an $\exists\mathbb{R}$-complete problem has been deducted from ETR. For the problem ETR-INV we have a set of variables $\{x_1, \dots, x_n\}$ and a set of equations of this form:

$$x = 1, \quad x + y = z, \quad x \cdot y = 1$$

and the problem decides if it exists a solution to this set of equations such that the value of each variable is real in $[\frac{1}{2}, 2]$.

A reduction of ETR-INV is found to the art gallery problem by constructing a polygon P and finding a number g for that polygon such that the instance of ETR-INT is true if and only if P is covered by at most g guards.

Stretchability A pseudoline is a simple closed curve in the plane. The stretchability problem is to decide if given a pseudoline arrangement, it is equivalent to an arrangement of straight lines.

Proof idea ETR can be reduced to STRETCHABILITY due to Mnev's universality theorem. [**?**]

Unit disk graph recognition The unit disk graph recognition is the problem that decides if a graph G is a unit disk graph. Unit disk graph recognition is $\exists\mathbb{R}$-complete. [?]

Proof idea UDG recognition is a corollary of deciding whether a graph with a given length is realizable. This problem is $\exists\mathbb{R}$-complete.

The reduction is done from STRETCHABILITY [?]. The reduction is done by adding a vertex to V for each pseudoline intersection. For each three consecutive points u_1, u_2, u_3 along a pseudoline a widget will be added that will be only realizable if and only if the pseudoline can be stretched with the same arrangement.

1.3 Geometry

Definition 38 *$dist(a, b)$ denotes the distance between the points a and b and is calculated with:*

$$dist(a, b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

The intersection of convex objects is a matter well studied for multiple subjects. In our case, it is interesting to know some properties about the intersection of disks, those ones being convex objects.

A set S is convex if:

$$\forall p, q \in S\ \forall \lambda \in [0, 1] : (1 - \lambda)p + \lambda q \in S$$

1.3.1 Stabbing

A *stabbing* is a point that traverses a set of intersecting objects. A lot of research has been done [?] on the minimal amount of stabbings to cover every object in a set. Stabbings can also be done with more complex structures than points, in that case we are talking about *coverings*.

Theorem 39 (Helly) *Given a set Q of objects in $\mathbb{R}^d$, if for each subset of Q of size $d + 1$ their intersection is non empty, then $\bigcap_{q \in Q} \neq \varnothing$. [?]*

Theorem 40 *The problem that for a set of n disks whether there exists a regular n-gon whose vertices stab every disk of the set can be decided in $O(n^{10.5}/\sqrt{\log(n)})$ [?]*

1.3.2 Coin graphs

Penny graphs can be defined as disk graphs where the disks can just touch each other without overlapping. A famous theorem is derived from this class of graphs: the circle packing theorem.

Theorem 41 (Circle packing theorem) *The circle packing theorem states that every simple connected planar graph G is a penny graph. [?]*

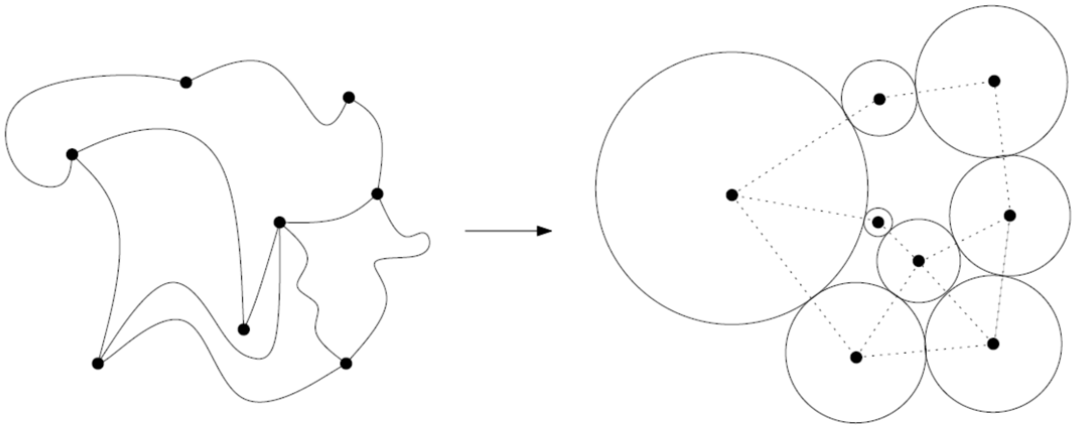


Figure 1.4: Circle packing of a planar graph. [**?**]

Corollary 42 *Planar graphs $\subseteq$ disk graphs [**?**].*

Chapter 2

Interval Graphs

In this chapter an overview of the MUIG and UUIG families will be given, with their characterization.

in the case of UUIG I want to find a more exhaustive characterization using MUUIG's families to use them in TSG. In this case it will be easy to proof complexity on TSG recognition

2.1 Mixed Unit Interval Graphs

Show and describe every family with demos from Joos' article

2.1.1 Families

Here we are going to define every family of forbidden induced subgraphs for MUUIG.

Theorem 43 *[Gilmore, Hoffman]A graph G is a comparability graph if and only if each odd cycle has at least one triangular chord. [**?**]*

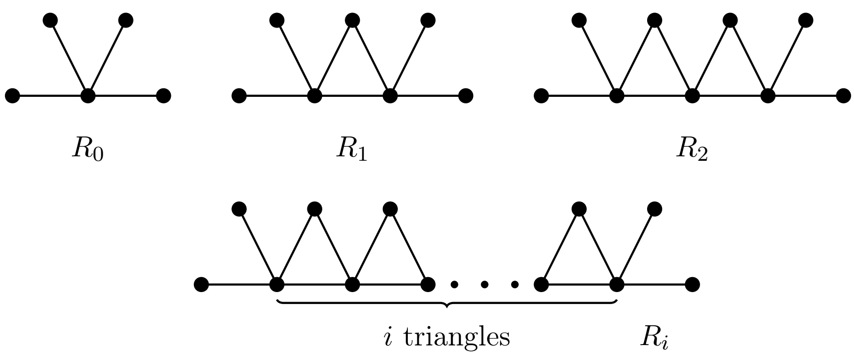
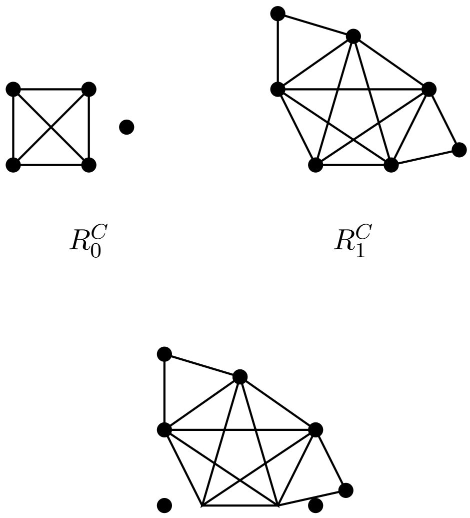


Figure 2.1: The class $\mathcal{R}$.



R_0^C R_1^C

R_2^C (à completer)

Figure 2.2: The class $\mathcal{R}^C$.

An important family of forbidden induced subgraphs in paper TSG: $\mathcal{R}_k$

Lemma 44 *$\mathcal{R}$ is a family of co-comparability graphs.*

Proof. *To prove that $\mathcal{R}$ is a family of co-comparability graphs we have to prove that its complement is a comparability graph. You can see in Figure* **??** *the family of complements of $\mathcal{R}$ that we will call $\mathcal{R}^C$.*

For this proof we will analyze the topology of $R_k^C = (V, E)$. First of all we can define two disjoint subsets $A \cup B = V$ where $\#A = k + 4$ and $\#B = k + 1$. A is a clique and B is a set of vertices such that in R_k their degree is greater than 3. We can also observe that the induced subgraph A is actually a tree, so there is not a cycle in A.

We will try to find an odd cycle C such that we do not find any triangular chord in it:

- $\forall_{v \in C}\ v \in A$*: In this case, every cycle inside the cycle has a triangular chord.*
- $\exists_{v \in C}\ v \in B$*: In this case, we will have to find a cycle C with these conditions:*
  - *If $\#(C \cap B) \geq 3$, then a triangular chord is found.*
  - *If $\#(C \cap B) = 2$, we will have to have an odd number of vertices on A. It can be either one (so it creates a triangulation, because both of the vertices share the same vertex in A) or more than 3.*
  - *If $\#(C \cap B) = 1$, we will have to have an even number of vertices on A.*

    Proof formality to be improved

That is, because we cannot find an odd cycle without a triangular chord, the $\mathcal{R}^C$ is a family of comparability graphs, so $\mathcal{R}$ is a family of co-comparability graphs. $\square$

2.2 Unfettered Unit Interval Graphs

Try to characterize with known forbidden families of subgraphs (from MUIG? and TSG article)

Chapter 3

Thin Strip Graphs

Introduction of the chapter.

3.1 Definition

c-strip graphs are unit disk graphs such that the centers of the disks belong to $\{(x, y) : -\infty < x < \infty, 0 \leq y \leq c\}$. *The class is denoted by SG(c). We have SG(0) = UIG and SG(*∞*) = UDG. [?]*

Definition 45 *Thin strip graphs are defined as* $TSG = \bigcap_{c>0} SG(c)$.

Remark 46 *SG(0)* $\neq$ *TSG. We can construct a* $K_{1,3}$ *such that we have 3 vertices with the coordinates* $(1, 0)$, $(0, 0)$, $(1, 0)$ *and a last one* $(0, \varepsilon)$ *with* $\varepsilon > 0$ *and arbitrarily small as seen in Figure* **??**.

Theorem 47 *There is no constant t such that SG(t) = TSG.*

Since this class is newly defined we have to characterize it. For this purpose, some relations have been found between this class and interval graphs.

3.1.1 Interval graphs

Theorem 48 *MUIG* $\subsetneq$ *TSG.*

We can define a new class of graphs: unfettered unit interval graphs. These graphs are unit interval graphs where if two intersections touch, we can decide whether they intersect or not. We denote this class UUIG.

Complete description and properties of UUIG

Theorem 49 *TSG* $\subsetneq$ *UUIG.*

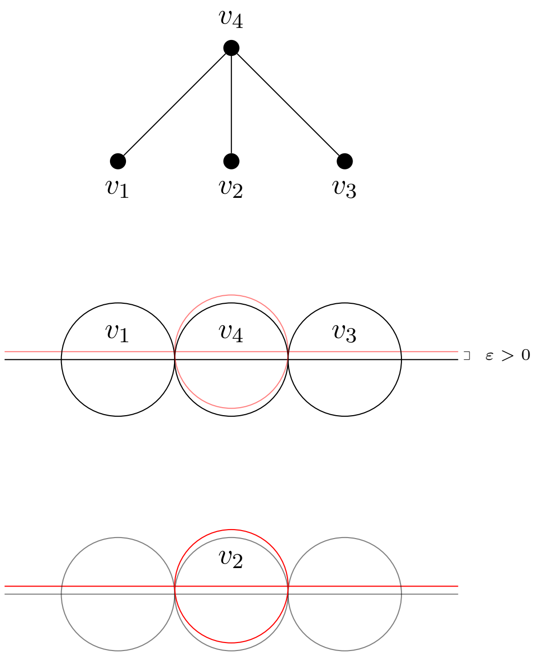


Figure 3.1: A construction of $K_{1,3}$ with a disk realization, being this graph a TSG.

3.2 Characterization of UUIG

Chapter 4

σ-SG(c)

To better characterize SG(c) and TSG we can define a new class of graphs: σ-SG(c).

σ-SG(c) are unit disk graphs such that the center of the disks belong to $\{(x, y) : -\inf < x < \inf, y \in \{0, c\}\}$, so more intuitively we can say that the center of the disks are placed on two parallel horizontal lines.

4.1 Characterization of σ-SG(c)

Proposition 50 *An σ-SG(c) graph G (with $c < 1$) can be characterized by computing $\delta : A \times B \to E$ where $A, B \subseteq G$, A and B are UIG, and $A \cup B = \varnothing$:*

$$\delta(x, y) = \begin{cases} xy & \text{if } dist(x, y) \leq 1 \\ \varnothing, & \text{otherwise} \end{cases}$$

Proof. *(Idea) Let's take two subsets $A, B \subseteq G$ being G a SG(c)... Both of these subsets (A and B are UIG, because each element in each of these subsets is in the same line).*

Finish proof about characterization -> UIGs

Theorem 51 *σ-SG(ϵ) $\subsetneq$ TSG*

prove with 2 clique adjacency

Proof. *By definition, we know that σ-SG(ϵ) $\subset$ TSG because the area where the disks can be placed in σ-SG(ϵ) is included in the area in TSG.*

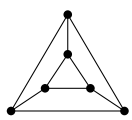


Figure 4.1: Forbidden graph in σ-SG(ϵ)

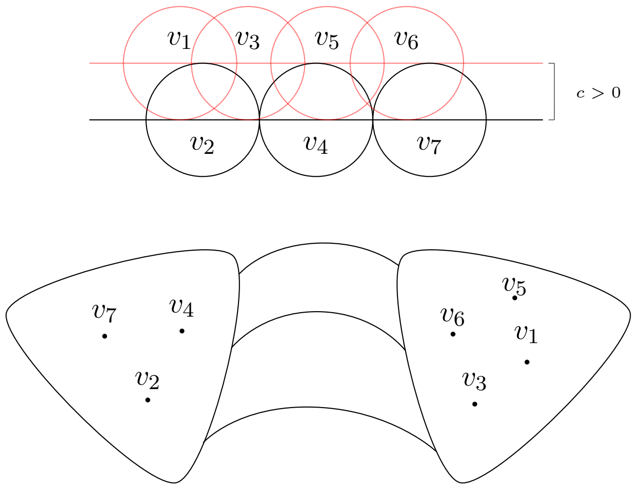


Figure 4.2: A representation of a σ-SG(c)

We can prove that σ*-SG*(ϵ) $\neq$ *TSG because we can construct a graph* G *such that* $G \in$ *TSG and* $G \notin \sigma$*-SG*(ϵ)*. This graph* F *is a net* graph as described in Figure* **??**.

Part 1 F *is a TSG because we can realize it as a TSG if we take as center of disks* $(0,0)$, $(0,z)$, $(0,\epsilon)$, $(1,0)$, $(1,z)$, $(1,\epsilon)$ *such that* $0 < z < \epsilon$.

Part 2 *Now we have to prove that* F *is a forbidden induced subgraph of* σ*-SG*(ϵ)*. We will try to construct it by taking a induced subgraph that is representable: we take* $F_{-1} = (V, E)$ *such that* $V(F_{-1}) = V(F) \setminus \{x\}$ *with* $x \in V(F)$*. We notice that* $V(F_{-1})$ *is* C_4 *(abcd) with a vertex e attached to two of its vertices (adjacent) creating a triangle abe. The only way to realize this is by taking* $a = (0,0)$, $b = (0,\epsilon)$, $c = (1,\epsilon)$, $d = (1,0)$ *and e*

How to prove for sigma?

Then in the hierarchy, is MUIG $\subsetneq$ σ*-SG*(ϵ) *or* $\subsetneq$ σ*-SG*(ϵ) *true?*

4.2 Induced forbidden subgraphs

Add forbidden subgraphs known for the moment with proofs.

Chapter 5

Complexity

5.1 Recognizing Thin Strip Graphs

Conclusions

The conclusions are to be written with care, because it will be sometimes the part that could convince a potential reader to read the whole document.

Todo list