Graph Coloring Problems

Guillermo Durán

Departamento de Ingeniería Industrial
Facultad de Ciencias Físicas y Matemáticas
Universidad de Chile, Chile

XII ELAVIO, February 2007, Itaipava, Brasil

The Four-Color problem

The Four-Color problem

The Four-Color Conjecture was settled in the XIX century:

Every map can be colored using at most four colors in such a way that adjacent regions (i.e. those sharing a common boundary segment, not just a point) receive different colors.

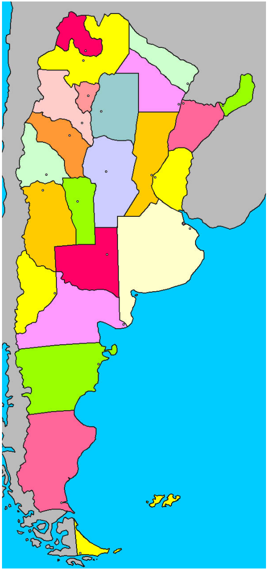
In terms of graphs...

Clearly a graph can be constructed from any map, the regions being represented by the vertices of the graph and two vertices being joined by an edge if the regions corresponding to the vertices are adjacent.

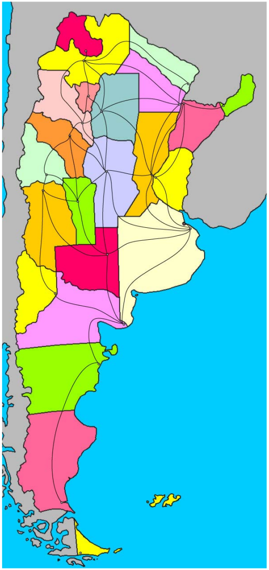
The resulting graph is planar, that is, it can be drawn in the plane without any edges crossing.

So, the Four-Color Conjecture asks if the vertices of a planar graph can be colored with at most 4 colors so that no two adjacent vertices use the same color.

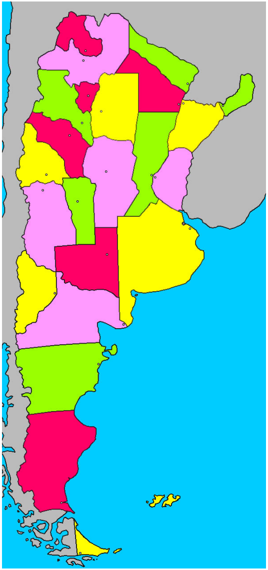
Example...



Example...



Example...



History

The Four-Color Conjecture first seems to have been formulated by Francis Guthrie. He was a student at University College London where he studied under Augusts De Morgan.

After graduating from London he studied law but some years later his brother Frederick Guthrie had become a student of De Morgan. Francis Guthrie showed his brother some results he had been trying to prove about the coloring of maps and asked Frederick to ask De Morgan about them.

Guthrie

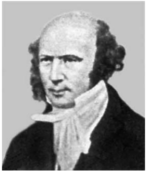
De Morgan

De Morgan was unable to give an answer but, on 23 October 1852, the same day he was asked the question, he wrote a letter to Sir William Hamilton in Dublin:

A student of mine asked me today to give him a reason for a fact which I did not know was a fact - and do not yet. He says that if a figure be anyhow divided and the compartments differently colored so that figures with any portion of common boundary line are differently colored - *four colors may be wanted, but not more* - the following is the case in which four colors are wanted. Query cannot a necessity for five or more be invented. ... If you retort with some very simple case which makes me out a stupid animal, I think I must do as the Sphynx did...

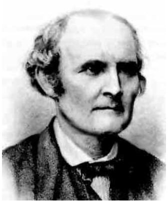
Hamilton replied on 26 October 1852 (showing the efficiency of both himself and the postal service):

I am not likely to attempt your quaternion of colors very soon.



Hamilton

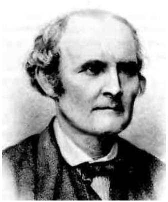
The first published reference is found in Arthur Cayley's, *On the colorings of maps*, Proc. Royal Geographical Society 1, 259–261, 1879.



Cayley

The first published reference is found in Arthur Cayley's, *On the colorings of maps*, Proc. Royal Geographical Society 1, 259–261, 1879.

On 17 July 1879 Alfred Bray Kempe announced in Nature that he had a proof of the Four-Color Conjecture.



Cayley

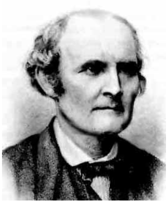
Kempes

The first published reference is found in Arthur Cayley's, *On the colorings of maps*, Proc. Royal Geographical Society 1, 259–261, 1879.

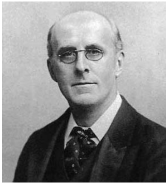
On 17 July 1879 Alfred Bray Kempe announced in Nature that he had a proof of the Four-Color Conjecture.

Kempe was a London barrister who had studied mathematics under Cayley at Cambridge and devoted some of his time to mathematics throughout his life.

At Cayley's suggestion Kempe submitted the Theorem to the American Journal of Mathematics where it was published in the ends of 1879.



Cayley



Kempe

Idea of Kempe's proof

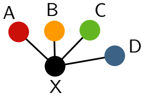
Kempe used an argument known as the method of *Kempe chains*.

If we have a map in which every region is colored red, green, blue or yellow except one, say X. If this final region X is not surrounded by regions of all four colors there is a color left for X. Hence suppose that regions of all four colors surround X.

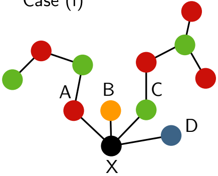
If X is surrounded by regions A, B, C, D in order, colored red, yellow, green and blue then there are two cases to consider.

(i) There is no chain of adjacent regions from A to C alternately colored red and green.

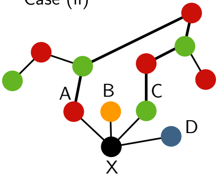
(ii) There is a chain of adjacent regions from A to C alternately colored red and green.



Case (i)



Case (ii)

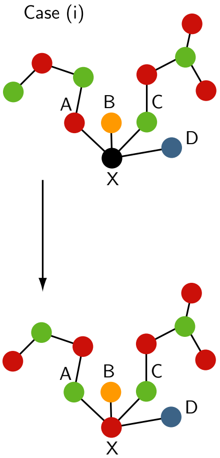


Cases:

(i) There is no chain of adjacent regions from A to C alternately colored red and green.

(ii) There is a chain of adjacent regions from A to C alternately colored red and green.

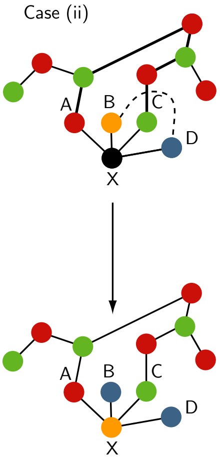
If (i) holds there is no problem. Change A to green, and then interchange the color of the red/green regions in the chain joining A. Since C is not in the chain it remains green and there is now no red region adjacent to X. Color X red.



Cases:

(i) There is no chain of adjacent regions from A to C alternately colored red and green.

(ii) There is a chain of adjacent regions from A to C alternately colored red and green.



If (i) holds there is no problem. Change A to green, and then interchange the color of the red/green regions in the chain joining A. Since C is not in the chain it remains green and there is now no red region adjacent to X. Color X red.

If (ii) holds then there can be no chain of yellow/blue adjacent regions from B to D. [It could not cross the chain of red/green regions.] Hence property (i) holds for B and D and we change colors as above.

The Four-Color Theorem returned to being the Four-Color Conjecture in 1890.

Percy John Heawood, a lecturer at Durham England, published a paper called *Map coloring theorem*. In it he states that his aim is "*...rather destructive than constructive, for it will be shown that there is a defect in the now apparently recognised proof...*".

Although Heawood showed that Kempe's proof was wrong he did prove that every map can be 5-colored in this paper.

Heawood

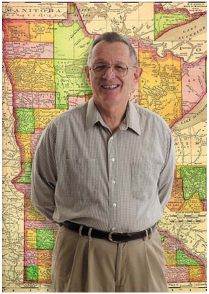
Exercise

Using Kempe's ideas, prove that every map can be 5-colored.

Hint: Every planar graph has at least one vertex of degree at most 5.

It was not until 1976 that the four-color conjecture was finally proven by Kenneth Appel and Wolfgang Haken at the University of Illinois. They were assisted in some algorithmic work by John Koch.

- K. Appel and W. Haken, *Every planar map is four colorable. Part I. Discharging*, Illinois J. Math. 21 (1977), 429–490.
- K. Appel, W. Haken and J. Koch, *Every planar map is four colorable. Part II. Reducibility*, Illinois J. Math. 21 (1977), 491–567.



Appel

Idea of the proof

If the four-color conjecture were false, there would be at least one map with the smallest possible number of regions that requires five colors. The proof showed that such a minimal counterexample cannot exist through the use of two technical concepts:

- An unavoidable set contains regions such that every map must have at least one region from this collection.
- A reducible configuration is an arrangement of countries that cannot occur in a minimal counterexample. If a map contains a reducible configuration, and the rest of the map can be colored with four colors, then the entire map can be colored with four colors and so this map is not minimal.

Idea of the proof

Using different mathematical rules and procedures, Appel and Haken found an unavoidable set of reducible configurations, thus proving that a minimal counterexample to the four-color conjecture could not exist.

Their proof reduced the infinitude of possible maps to 1,936 reducible configurations (later reduced to 1,476) which had to be checked one by one by computer.

However, the unavoidability part of the proof was over 500 pages of hand written counter-counter-examples (these graph colorings were verified by Haken's son!). The computer program ran for hundreds of hours.

But most of the researchers thought that there were two reasons why the Appel-Haken proof was not completely satisfactory.

- Part of the Appel-Haken proof uses a computer, and cannot be verified by hand, and
- Even the part that is supposedly hand-checkable is extraordinarily complicated and tedious, and no one has verified it in its entirety.

Ten years ago, another proof:

- N. Robertson, D. P. Sanders, P. D. Seymour and R. Thomas, *The four color theorem*, J. Combin. Theory Ser. B. 70 (1997), 2–44.
- N. Robertson, D. P. Sanders, P. D. Seymour and R. Thomas, *A new proof of the four color theorem*, Electron. Res. Announc. Amer. Math. Soc. 2 (1996), 17–25 (electronic).


Robertson


Sanders


Seymour


Thomas

Outline of the proof

The basic idea of the proof is the same as Appel and Haken's. The authors exhibit a set of 633 "configurations", and prove each of them is "reducible". Recall, that this is a technical concept that implies that no configuration with this property can appear in a minimal counterexample to the Four-Color Theorem. It has been known since 1913 that every minimal counterexample to the Four-Color Theorem should be a special structure, called "internally 6-connected triangulation".

In the second part of the proof they prove that at least one of the 633 configurations appears in every internally 6-connected planar triangulation. This is called proving unavoidability, and here the method used differs from that of Appel and Haken. The first part of proof needs a computer. The second part can be checked by hand in a few months, or, using a computer, it can be verified in about 20 minutes.

Why is this proof "better"?

The unavoidable set has size 633 as opposed to the 1476 member set of Appel and Haken, and the second part of the proof uses only about 30 rules, instead of the 300+ of Appel and Haken (and by computer can be verified in about 20 minutes against hundred of hours of the other proof).

At December 2004 in a scientific meeting in France, a joint group between people by Microsoft Research in England and INRIA in France announced the verification of the Robertson et al. proof by formulating the problem in the equational logic program Coq and confirming the validity of each of its steps (Devlin 2005, Knight 2005).

But in both cases (Appel and Haken, and Robertson et al.), the ‘proofs’ are not proofs in the traditional sense, because they contain steps that can never be verified by humans. Up today, a traditional mathematical proof is not known for the Four-Color Theorem.

Some basic concepts about computational complexity

Some basic concepts about computational complexity

- A *problem* is a general question to be answered, usually possessing several *parameters*, whose values are left unspecified.
- A problem is described by giving:
  1. A general description of all its parameters.
  2. A statement of what properties the answer (or *solution*) is required to satisfy.
- The difficulty of a problem is related to its structure and the length of the instance to be considered. This length is given by one or two parameters, for example, in the graph coloring problem, the number of vertices of the graph.

Some basic concepts about computational complexity

- In order to know the complexity of an algorithm we need to calculate the number of elementary arithmetic operations that the algorithm does to solve a given problem. This number is a function of the length of the instance.
- We say that a problem is in **P** if there exists an algorithm of polynomial complexity to solve it (the number of those operations is always upper bounded by a polynomial function in n, the input length).

NP-completeness theory

- It is applied to *decision problems*, problems whose answer is "YES" or "NOT" (but it is easy to see that this theory has several consequences on optimization problems).
- For example, the decision problem related to the graph coloring problem is the following: "Given a graph G and an integer number k, is there a valid coloring with at most k colors?"
- A decision problem π consists of a set D_π of instances and a subset $Y_\pi \subseteq D_\pi$ whose answer is "YES".

NP-completeness theory

- A problem $\pi \in$ **NP** if there exists a polynomial certificate to verify an instance of "YES" (this is, if I can verify in polynomial time that an instance of "YES" is right).
- So, it is not difficult to see that P $\subseteq$ NP.
- **Open Conjecture: P $\neq$ NP.**

Polynomial reduction

- Let π and π' be two decision problems. We say that $f : D_{\pi'} \rightarrow D_{\pi}$ is a polynomial reduction of π' in π if f can be computed in polynomial time and for every $d \in D_{\pi'}$, $d \in Y_{\pi'} \Leftrightarrow f(d) \in Y_{\pi}$. Notation: $\pi' \preccurlyeq \pi$.
- Note that if $\pi'' \preccurlyeq \pi'$ and $\pi' \preccurlyeq \pi$ then $\pi'' \preccurlyeq \pi$, because the composition of two polynomial reductions is a polynomial reduction.

NP-complete problems

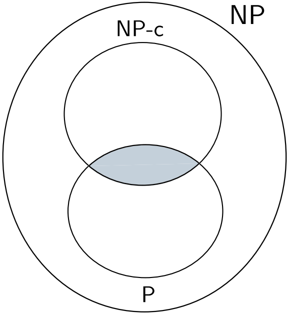
- A problem π is NP-complete if:
  1. $\pi \in$ NP.
  2. For every $\pi' \in$ NP, $\pi' \preccurlyeq \pi$.

- If a problem π verifies condition 2., we say that π is NP-hard (it is so "difficult" as all the problems in NP).

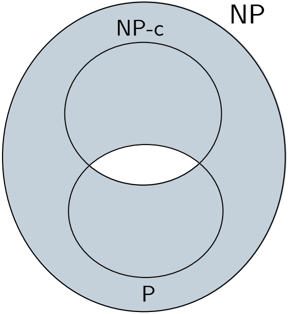
P $\neq$ NP? or P = NP?

- **If there is a problem $\pi \in$ NP-c $\cap$ P, then P=NP.**
    - If $\pi \in$ NP-c $\cap$ P, there is a polynomial time algorithm to solve π, because π is in P. On the other hand, as $\pi \in$ NP-c, for every $\pi' \in$ NP, $\pi' \preccurlyeq \pi$.
    - Let π' be in NP. We have to use the polynomial reduction which transforms instances of π' in instances of π, and then the polynomial time algorithm which solves π. It is easy to see that we obtain a polynomial time algorithm to solve π'.
- It is known any problem neither in NP-c $\cap$ P, nor in NP $\setminus$ P (in this last case, it would be proved that P $\neq$ NP).

Inclusions between the classes



if P=NP...



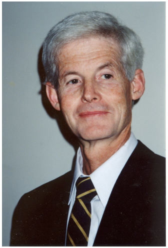
if P≠NP...

How do we have to do to prove that a problem is NP-complete?

Cook's Theorem (1971)

SAT is NP-complete.

The proof is direct: it is easy to see that SAT is in NP. Then, it is considered a general problem $\pi \in$ NP and a general instance $d \in D_\pi$. Using a polynomial non-deterministic Turing machine to solve π, it is generated in polynomial time a logic formula $\varphi_{\pi,d}$ such that $d \in Y_\pi$ if and only if $\varphi_{\pi,d}$ is satisfiable.



Cook

How do we have to do to prove that a problem is NP-complete?

Using Cook's Theorem, the standard technique to prove that a problem π is NP-complete uses the transitivity of $\preccurlyeq$, and consists in the following:

1. Prove that π is in NP.
2. Choose an appropriated problem π' belonging to NP-c.
3. Build a polynomial reduction f of π' in π.

The second condition of the definition holds using the transitivity: let π'' be a problem in NP. As π' is NP-c, $\pi'' \preccurlyeq \pi'$. But it was proved that $\pi' \preccurlyeq \pi$, so $\pi'' \preccurlyeq \pi$.

How do we have to do to prove that a problem is NP-complete?

Using Cook's Theorem, the standard technique to prove that a problem π is NP-complete uses the transitivity of $\preccurlyeq$, and consists in the following:

1. Prove that π is in NP.
2. Choose an appropriated problem π' belonging to NP-c.
3. Build a polynomial reduction f of π' in π.

The second condition of the definition holds using the transitivity: let π'' be a problem in NP. As π' is NP-c, $\pi'' \preccurlyeq \pi'$. But it was proved that $\pi' \preccurlyeq \pi$, so $\pi'' \preccurlyeq \pi$.

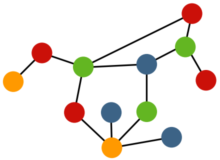
Some famous problems in NP-c

- Traveling Salesman Problem (TSP)
- Graph coloring
- Integer Programming

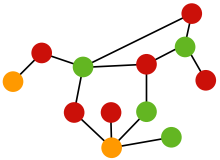
Graph coloring

Graph coloring

- A k-coloring of a graph G is an assignment of one color to each vertex of G such that no more than k colors are used and no two adjacent vertices receive the same color.
- A graph is called k-colorable iff it has a k-coloring.



4-coloring

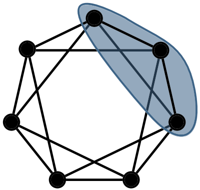


3-coloring

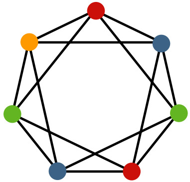
Chromatic number

- A clique in a graph G is a complete subgraph maximal under inclusion. The cardinality of a maximum clique is denoted by $\omega(G)$.
- The chromatic number of a graph G is the smallest number k such that G is k-colorable, and it is denoted by $\chi(G)$. An obvious lower bound for $\chi(G)$ is $\omega(G)$:

$$\omega(G) \leq \chi(G) \quad \forall G.$$



$\omega = 3$



$\chi = 4$

Applications

The problem of coloring a graph has several applications such as scheduling, register allocation in compilers, frequency assignment in Mobile radios, etc.

Example: Examination schedule

Each student must take an examination in each of his/her courses. Let X be the set of different courses and let Y be the set of students. Since the examination is written, it is convenient that all students in a course take the examination at the same time. What is the minimum number of examination periods needed?

Exercise

Model this problem as a coloring problem.

Computational complexity

The graph k-colorability problem is the following:

- INSTANCE: A graph $G = (V, E)$ and a positive integer $k \leq V$.
- QUESTION: Is G k-colorable?

This problem is NP-complete (Karp, 1972), and remains NP-c for $k = 3$.

Exercise

What happens for $k = 2$?

Karp

Planar graphs coloring

For planar graphs the paper by Robertson et al. gives a quadratic algorithm to four-color planar graphs, an improvement over the quadric algorithm by Appel and Haken.

Exercise

Does it mean that the k-colorability problem is polynomial for planar graphs?

Some easy properties about $\chi(G)$

- Let G be a graph with n vertices and $\overline{G}$ its complement. Then:
  - $\chi(G) \leq \Delta(G) + 1$, where $\Delta(G)$ is the maximum degree of G.
  - $\chi(G)\,\omega(\overline{G}) \geq n$
  - $\chi(G) + \omega(\overline{G}) \leq n + 1$
  - $\chi(G) + \chi(\overline{G}) \leq n + 1$

Brooks' Theorem

Brooks' Theorem (1941)

Let G be a connected graph. Then G is $\Delta(G)$-colorable, unless:

1. $\Delta(G) \neq 2$, and G is a $\Delta(G) + 1$-clique, or
2. $\Delta(G) = 2$, and G is an odd cycle.

Graph coloring algorithms

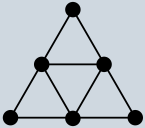
As it was said, it is not known a polynomial time algorithm to determine $\chi(G)$. Let us see the following no efficient algorithm (contraction-connection):

- Consider a graph G with two non-adjacent vertices a and b. The connection G_1 is obtained by joining the two non-adjacent vertices a and b with an edge. The contraction G_2 is obtained by shrinking $\{a, b\}$ into a single vertex $c(a, b)$ and by joining it to each neighbor in G of vertex a and of vertex b (and eliminating multiple edges).
- A coloring of G in which a and b have the same color yields a coloring of G_1. A coloring of G in which a and b have different colors yields a coloring of G_2.
- Repeat the operations of connection and contraction in each graph generated, until the resulting graphs are all cliques. If the smallest resulting clique is a k-clique, then $\chi(G) = k$.

Graph coloring algorithms

Exercise

Apply this method in the following graph



Chromatic polynomial

The chromatic polynomial of a graph G is defined to be a function $P_G(k)$ that expresses for each integer k the number of distinct possible k-colorings for a graph G.

- **Example 1:** If G is a tree with n vertices, then:

$$P_G(k) = k(k-1)^{n-1}$$

- **Example 2:** If G is a n-clique, then:

$$P_G(k) = k(k-1)(k-2)\dots(k-n+1)$$

Chromatic polynomial

The chromatic polynomial of a graph G is defined to be a function $P_G(k)$ that expresses for each integer k the number of distinct possible k-colorings for a graph G.

- **Example 1:** If G is a tree with n vertices, then:

$$P_G(k) = k(k-1)^{n-1}$$

- **Example 2:** If G is a n-clique, then:

$$P_G(k) = k(k-1)(k-2)\dots(k-n+1)$$

Chromatic polynomial

Property: $P_G(k) = P_{G_1}(k) + P_{G_2}(k)$, where G_1 and G_2 are the graphs defined in the connection-contraction algorithm.

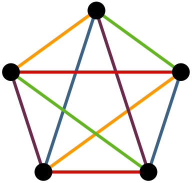
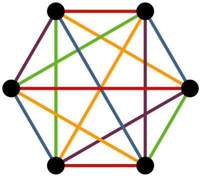
Exercise

Prove that the chromatic polynomial of a cycle C_n is:

$$P_{C_n}(k) = (k-1)^n + (-1)^n(k-1)$$

Chromatic index

- The chromatic index $\chi'(G)$ of a graph G is defined to be the smallest number of colors needed to color the edges of G so that no two adjacent edges have the same color.
- Clearly $\chi'(G) \geq \Delta(G)$, the maximum degree of G.
- A q-coloring of the edges of G is defined to be a partition of the edge set of G into q subsets that are matchings (a set of edges which do not share endpoints).

Chromatic index

- Property: If G is a complete graph with n vertices, then
  - $\chi'(G) = n - 1$, if n is even
  - $\chi'(G) = n$, if n is odd

- **Vizing's Theorem (1964):** Let G be a graph, then

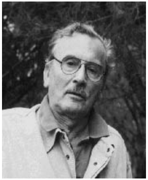
$$\Delta \leq \chi'(G) \leq \Delta(G) + 1.$$

- The problem of determining if there exists a $\Delta(G)$-coloring of a graph G is NP-complete (Holyer, 1981), even if the given graph is triangle-free with $\Delta(G) = 3$ (Koreas, 1997).

Perfect graphs

A famous class of graphs associated to graph coloring

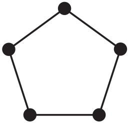
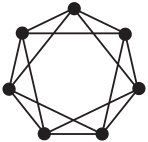
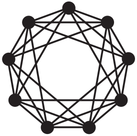
A graph G is perfect if $\omega(H) = \chi(H)$ for every induced subgraph H of G (Berge, 1961).



Berge

Berge conjectured two statements:

1. A graph is perfect if and only if its complement is perfect.
2. A graph is perfect if and only if it contains neither induced odd cycle of length at least five nor its complement.

The Perfect Graph Theorem (Lóvasz, 1972; Fulkerson, 1973)

A graph is perfect if and only if its complement is perfect.

Lóvasz

Fulkerson

Exercise

Prove that odd holes and their complements are not perfect.

The Strong Perfect Graph Theorem (Chudnovsky, Robertson, Seymour, Thomas, 2002)

A graph is perfect if and only if it contains neither induced odd cycle of length at least five nor its complement.

This work was published recently:

- Chudnovsky M., Robertson N., Seymour P. and Thomas R., *The Strong Perfect Graph Theorem*, Annals of Mathematics 164 (2006), 51–229.

Chudnovsky

Robertson

Seymour

Thomas

Polynomial (but no efficient!) recognition

The characterization by Chudnovsky et al. does not lead to a polynomial recognition of perfect graphs (the complexity of recognizing odd holes is open).

In 2002, two polynomial algorithms for recognizing perfect graphs were presented.

- *Recognizing Berge Graphs*, Chudnovsky and Seymour, 2002 (an $O(n^9)$ algorithm).
- *A Polynomial Algorithm for Recognizing Perfect Graphs*, Cornuéjols, Liu and Vušković, 2002 (an $O(n^{20})$ algorithm).

In 2005, it was published the following joint work:

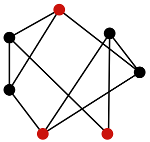
- Chudnovsky M., Cornuéjols G., Liu X., Seymour P. and Vušković K., *Recognizing Berge Graphs*, Combinatorica 25 (2005), 143–187.

Cornuéjols

Vušković

Another definition of perfect graphs

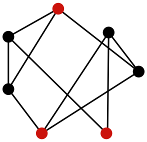
- An independent set (or *stable set*) in a graph G is a subset of pairwise non-adjacent vertices of G. The stability number $\alpha(G)$ is the cardinality of a maximum independent set of G.
- A clique cover of a graph G is a subset of cliques covering all the vertices of G. The clique-covering number $k(G)$ is the cardinality of a minimum clique cover of G.
- It is easy to see that $\alpha(G) = \omega(\overline{G})$ and $k(G) = \chi(\overline{G})$.
- So, by PGT: **A graph G is perfect when $\alpha(H) = k(H)$ for every induced subgraph H of G.**



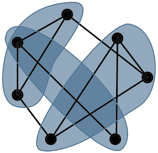
$\alpha = 3$

Another definition of perfect graphs

- An independent set (or *stable set*) in a graph G is a subset of pairwise non-adjacent vertices of G. The stability number $\alpha(G)$ is the cardinality of a maximum independent set of G.
- A clique cover of a graph G is a subset of cliques covering all the vertices of G. The clique-covering number $k(G)$ is the cardinality of a minimum clique cover of G.
- It is easy to see that $\alpha(G) = \omega(\overline{G})$ and $k(G) = \chi(\overline{G})$.
- So, by PGT: **A graph G is perfect when $\alpha(H) = k(H)$ for every induced subgraph H of G.**



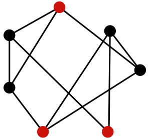
$\alpha = 3$



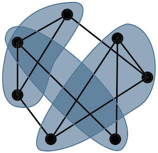
$k = 3$

Another definition of perfect graphs

- An independent set (or *stable set*) in a graph G is a subset of pairwise non-adjacent vertices of G. The stability number $\alpha(G)$ is the cardinality of a maximum independent set of G.
- A clique cover of a graph G is a subset of cliques covering all the vertices of G. The clique-covering number $k(G)$ is the cardinality of a minimum clique cover of G.
- It is easy to see that $\alpha(G) = \omega(\overline{G})$ and $k(G) = \chi(\overline{G})$.
- So, by PGT: **A graph G is perfect when $\alpha(H) = k(H)$ for every induced subgraph H of G.**



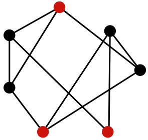
$\alpha = 3$



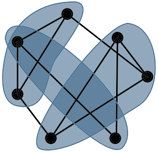
$k = 3$

Another definition of perfect graphs

- An independent set (or *stable set*) in a graph G is a subset of pairwise non-adjacent vertices of G. The stability number $\alpha(G)$ is the cardinality of a maximum independent set of G.
- A clique cover of a graph G is a subset of cliques covering all the vertices of G. The clique-covering number $k(G)$ is the cardinality of a minimum clique cover of G.
- It is easy to see that $\alpha(G) = \omega(\overline{G})$ and $k(G) = \chi(\overline{G})$.
- So, by PGT: **A graph G is perfect when $\alpha(H) = k(H)$ for every induced subgraph H of G.**



$\alpha = 3$

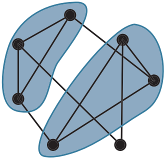


$k = 3$

Variations of perfect graphs

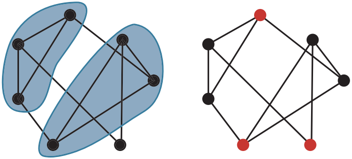
Clique-perfect graphs

- A clique-independent set is a collection of pairwise vertex-disjoint cliques. The clique-independence number $\alpha_c(G)$ is the size of a maximum clique-independent set of G.
- A clique-transversal of a graph G is a subset of vertices that meets all the cliques of G. The clique-transversal number $\tau_c(G)$ is the size of a minimum clique-transversal of G.
- Clearly, $\alpha_c(G) \leq \tau_c(G)$ for every graph G.
- A graph G is clique-perfect when $\alpha_c(H) = \tau_c(H)$ for every induced subgraph H of G.



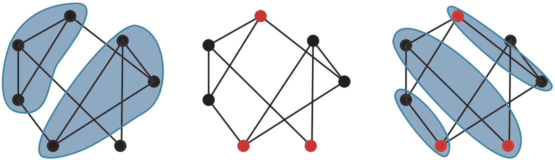
Clique-perfect graphs

- A clique-independent set is a collection of pairwise vertex-disjoint cliques. The clique-independence number $\alpha_c(G)$ is the size of a maximum clique-independent set of G.
- A clique-transversal of a graph G is a subset of vertices that meets all the cliques of G. The clique-transversal number $\tau_c(G)$ is the size of a minimum clique-transversal of G.
- Clearly, $\alpha_c(G) \leq \tau_c(G)$ for every graph G.
- A graph G is clique-perfect when $\alpha_c(H) = \tau_c(H)$ for every induced subgraph H of G.



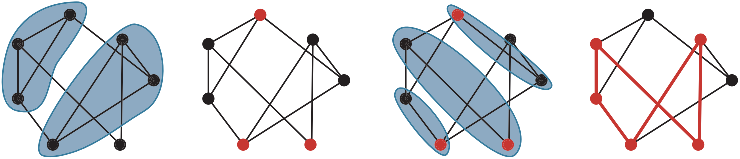
Clique-perfect graphs

- A clique-independent set is a collection of pairwise vertex-disjoint cliques. The clique-independence number $\alpha_c(G)$ is the size of a maximum clique-independent set of G.
- A clique-transversal of a graph G is a subset of vertices that meets all the cliques of G. The clique-transversal number $\tau_c(G)$ is the size of a minimum clique-transversal of G.
- Clearly, $\alpha_c(G) \leq \tau_c(G)$ for every graph G.
- A graph G is clique-perfect when $\alpha_c(H) = \tau_c(H)$ for every induced subgraph H of G.



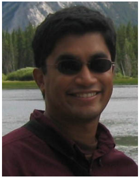
Clique-perfect graphs

- A clique-independent set is a collection of pairwise vertex-disjoint cliques. The clique-independence number $\alpha_c(G)$ is the size of a maximum clique-independent set of G.
- A clique-transversal of a graph G is a subset of vertices that meets all the cliques of G. The clique-transversal number $\tau_c(G)$ is the size of a minimum clique-transversal of G.
- Clearly, $\alpha_c(G) \leq \tau_c(G)$ for every graph G.
- A graph G is clique-perfect when $\alpha_c(H) = \tau_c(H)$ for every induced subgraph H of G.



Clique-perfect graphs

- The terminology "clique-perfect" has been introduced by Guruswami and Pandu Rangan in 2000, but the equality of the parameters α_c and τ_c was previously studied by Berge in the seventies.



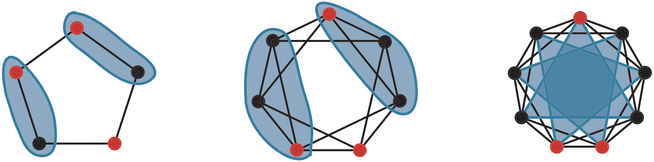
Guruswami

Pandu Rangan

- The complete list of minimal clique-imperfect graphs is still not known. Another open question concerning clique-perfect graphs is the complexity of the recognition problem.

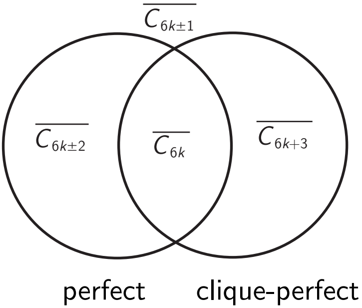
Question: is there some relation between clique-perfect graphs and perfect graphs?

- Odd holes $C_{2k+1}, k \geq 2$, are not clique-perfect: $\alpha_c(C_{2k+1}) = k$ and $\tau_c(C_{2k+1}) = k + 1$.
- Antiholes $\overline{C_n}, n \geq 5$, are clique-perfect if and only if $n \equiv 0(3)$ (Reed, 2000): $\tau_c(\overline{C_n}) = 3$ and $\alpha_c(\overline{C_n}) = 2$ or 3, being 3 only if n is divisible by three.



C_5 $\quad$ $\overline{C_7}$ $\quad$ $\overline{C_9}$

So we have the following scheme of relation between perfect graphs and clique-perfect graphs:



In several works, clique-perfect graphs have been characterized by a restricted list of forbidden induced subgraphs when the graph belongs to a certain class. Some of this characterizations lead to polynomial time recognition algorithms for clique-perfection within these classes.

- J. Lehel and Z. Tuza, *Neighborhood perfect graphs*, Discrete Mathematics 61 (1986), 93–101.
- F. Bonomo, M. Chudnovsky and G.D., *Partial characterizations of clique-perfect graphs*, Electronic Notes in Discrete Mathematics 19 (2005), 95–101.
- F. Bonomo and G.D., *Characterization and recognition of Helly circular-arc clique-perfect graphs*, Electronic Notes in Discrete Mathematics 22 (2005), 147–150.

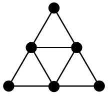
Example: The characterization for line graphs

Let H be a graph. Its line graph $L(H)$ is the intersection graph of the edges of H. A graph G is a line graph if there exists a graph H such that $G = L(H)$.

Theorem

Let G be a line graph. Then the following statements are equivalent:

1. No induced subgraph of G is and odd hole, or a pyramid.
2. G is clique-perfect.
3. G is perfect and it does not contain a pyramid.

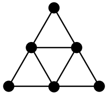


pyramid

Example: The characterization for line graphs

Line graphs have polynomial time recognition (Lehot, 1974).

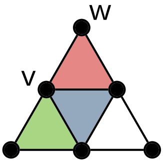
The recognition of clique-perfect line graphs can be reduced to the recognition of perfect graphs with no pyramid, which is solvable in polynomial time.



pyramid

Coordinated graphs

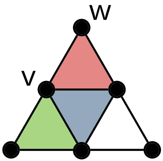
- Let v be a vertex of a graph G and $m(v)$ the number of cliques containing v.
- Let $M(G)$ be the maximum $m(v)$ for any v in G.
- Let $F(G)$ be the cardinality of a minimum partition of the cliques of G into clique-independent sets, that is, the smallest number of colors that can be assigned to the cliques of G so that intersecting cliques have different colors.
- Note that $F(G) \geq M(G)$ for any graph G.



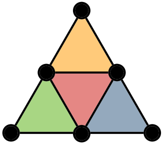
$m(v) = 3, m(w) = 1, M = 3$

Coordinated graphs

- Let v be a vertex of a graph G and $m(v)$ the number of cliques containing v.
- Let $M(G)$ be the maximum $m(v)$ for any v in G.
- Let $F(G)$ be the cardinality of a minimum partition of the cliques of G into clique-independent sets, that is, the smallest number of colors that can be assigned to the cliques of G so that intersecting cliques have different colors.
- Note that $F(G) \geq M(G)$ for any graph G.



$m(v) = 3, m(w) = 1, M = 3$



$F = 4$

Coordinated graphs

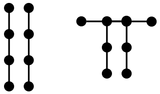
- We say that a graph G is coordinated if $F(H) = M(H)$, for every induced subgraph H of G (this class of graph was defined by Bonomo, D. and Groshaus in 2002).
- Property: Coordinated graphs are perfect.
- The complete list of minimal non-coordinated graphs is still not known. Again, they have been characterized by a restricted list of forbidden induced subgraphs when the graph belongs to a certain class (Bonomo, D., Soulignac and Sueiro, 2006).

Example: The characterization for complements of forests

A forest is a graph with no cycles.

Theorem

Let G be a complement of a forest. Then G is coordinated if and only if G contains neither $\overline{2P_4}$ nor $\overline{R}$ as induced subgraphs.



$2P_4$ and R

This theorem leads to a linear time recognition of coordinated graphs if the given graph is a complement of a forest.

The general recognition of coordinated graphs is NP-hard (Soulignac and Sueiro, 2006).

Some subclasses of perfect graphs

- A graph is an interval graph if it is the intersection graph of a set of intervals over the real line. A unit interval graph is the intersection graph of a set of intervals of length one.
- A split graph is a graph whose vertex set can be partitioned into a complete graph K and a stable set S. A split graph is said to be complete if its edge set includes all possible edges between K and S.
- A bipartite graph is a graph whose vertex set can be partitioned into two independent sets V_1 and V_2. A bipartite graph is said to be complete if its edge set includes all possible edges between V_1 and V_2.

Some subclasses of perfect graphs

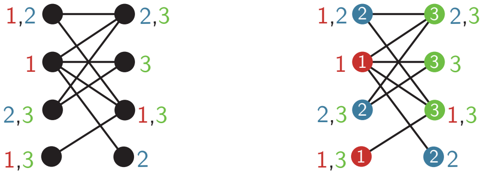
- A cograph is a graph with no induced P_4.
- The line graph of a graph is the intersection graph of its edges. Line graphs of bipartite graphs are perfect.
- A graph is distance-hereditary if the distance between any two vertices in a connected induced subgraph containing both is the same as in the original graph.
- A graph is a block graph if it is connected and every maximal 2-connected component is complete.

Extensions of the coloring problem

The list-coloring problem

In order to take into account particular constraints arising in practical settings, more elaborate models of vertex coloring have been defined in the literature. One of such generalized models is the list-coloring problem, which considers a prespecified set of available colors for each vertex.

- Given a graph G and a finite list $L(v) \subseteq \mathbb{N}$ for each vertex $v \in V$, the list-coloring problem asks for a list-coloring of G, i.e., a coloring f such that $f(v) \in L(v)$ for every $v \in V$.



The list-coloring problem

- Many classes of graphs where the vertex coloring problem is polynomially solvable are known, the most prominent being the class of perfect graphs [Grötschel-Lovász-Schrijver, 1981].
- Meanwhile, the list-coloring problem is NP-complete for perfect graphs, and is also NP-complete for many subclasses of perfect graphs, including split graphs, interval graphs, and bipartite graphs.
- Trees and complete graphs are two classes of graphs where the list-coloring problem can be solved in polynomial time. In the first case it can be solved using dynamic programming techniques [Jansen-Scheffler, 1997]. In the second case, the problem can be reduced to the maximum matching problem in bipartite graphs.

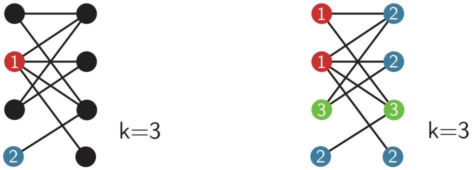
We are going to explore the complexity boundary between vertex coloring and list-coloring in different subclasses of perfect graphs. The goal is to analyze the computational complexity of coloring problems lying "between" (from a computational complexity viewpoint) these two problems.

The precoloring extension problem

Some particular cases of list-coloring have been studied.

- The precoloring extension (PrExt) problem takes as input a graph $G = (V, E)$, a subset $W \subseteq V$, a coloring f' of W, and a natural number k, and consists in deciding whether G admits a k-coloring f such that $f(v) = f'(v)$ for every $v \in W$ or not [Biro-Hujter-Tuza, 1992].

In other words, a prespecified vertex subset is colored beforehand, and our task is to extend this partial coloring to a valid k-coloring of the whole graph.

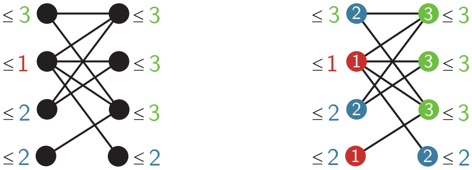


The μ-coloring problem

Another particular case of the list-coloring problem is the following.

- Given a graph G and a function $\mu : V \to \mathbb{N}$, G is μ-colorable if there exists a coloring f of G such that $f(v) \leq \mu(v)$ for every $v \in V$ [Bonomo-Cecowski, 2005].

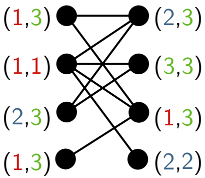
This model arises in the context of classroom allocation to courses, where each course must be assigned a classroom which is large enough so it fits the students taking the course.

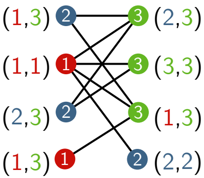


The (γ, μ)-coloring problem

A new variation of this problem is the following (Bonomo, D., Marenco, 2006).

- Given a graph G and functions $\gamma, \mu : V \to \mathbb{N}$ such that $\gamma(v) \leq \mu(v)$ for every $v \in V$, we say that G is (γ, μ)-colorable if there exists a coloring f of G such that $\gamma(v) \leq f(v) \leq \mu(v)$ for every $v \in V$.

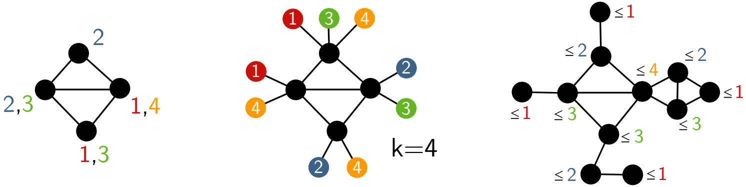




- The classical vertex coloring problem is clearly a special case of μ-coloring and precoloring extension, which in turn are special cases of (γ, μ)-coloring.
- Furthermore, (γ, μ)-coloring is a particular case of list-coloring.
- These observations imply that all the problems in this hierarchy are polynomially solvable in those graph classes where list-coloring is polynomial and, on the other hand, all the problems are NP-complete in those graph classes where vertex coloring is NP-complete.

General results

Since all the problems are NP-complete in the general case, there are also polynomial-time reductions from list-coloring to precoloring extension and μ-coloring. An example is shown in the figure, where we can see a list-coloring instance and its corresponding precoloring extension and μ-coloring instances.



These reductions involve changes in the graph, but are closed within some graph classes. This fact allows us to prove the following general results.

General results

Theorem

Let $\mathcal{F}$ be a family of graphs with minimum degree at least two. Then list-coloring, (γ, μ)-coloring and precoloring extension are polynomially equivalent in the class of $\mathcal{F}$-free graphs.

Theorem

Let $\mathcal{F}$ be a family of graphs satisfying the following property: for every graph G in $\mathcal{F}$, no connected component of G is complete, and for every cutpoint v of G, no connected component of $G \setminus v$ is complete. Then list-coloring, (γ, μ)-coloring, μ-coloring and precoloring extension are polynomially equivalent in the class of $\mathcal{F}$-free graphs.

Since odd holes and antiholes satisfy the conditions of the theorems above, these theorems are applicable for many subclasses of perfect graphs.

Review: complexity table for coloring problems

Class	coloring	PrExt	μ-col.	(γ, μ)-col.	list-col.
Complete bipartite	P	P	P	P	NP-c
Bipartite	P	NP-c	NP-c	NP-c	NP-c
Cographs	P	P	P	?	NP-c
Distance-hereditary	P	NP-c	NP-c	NP-c	NP-c
Interval	P	NP-c	NP-c	NP-c	NP-c
Unit interval	P	NP-c	?	NP-c	NP-c
Complete split	P	P	P	P	NP-c
Split	P	P	NP-c	NP-c	NP-c
Line of $K_{n,n}$	P	NP-c	NP-c	NP-c	NP-c
Complement of bipartite	P	P	?	?	NP-c
Block	P	P	P	P	P

"NP-c": NP-complete problem, "P": polynomial problem, "?": open problem.

Review: complexity table for coloring problems

Class	coloring	PrExt	μ-col.	(γ, μ)-col.	list-col.
Complete bipartite	P	P	P	P	NP-c
Bipartite	P	NP-c	NP-c	NP-c	NP-c
Cographs	P	P	P	?	NP-c
Distance-hereditary	P	NP-c	NP-c	NP-c	NP-c
Interval	P	NP-c	NP-c	NP-c	NP-c
Unit interval	P	NP-c	?	NP-c	NP-c
Complete split	P	P	P	P	NP-c
Split	P	P	NP-c	NP-c	NP-c
Line of $K_{n,n}$	P	NP-c	NP-c	NP-c	NP-c
Complement of bipartite	P	P	?	?	NP-c
Block	P	P	P	P	P

"NP-c": NP-complete problem, "P": polynomial problem, "?": open problem.

As this table shows, unless P $=$ NP, μ-coloring and precoloring extension are strictly more difficult than vertex coloring, list-coloring is strictly more difficult than (γ, μ)-coloring and (γ, μ)-coloring is strictly more difficult than precoloring extension.

Review: complexity table for coloring problems

Class	coloring	PrExt	μ-col.	(γ, μ)-col.	list-col.
Complete bipartite	P	P	P	P	NP-c
Bipartite	P	NP-c	NP-c	NP-c	NP-c
Cographs	P	P	P	?	NP-c
Distance-hereditary	P	NP-c	NP-c	NP-c	NP-c
Interval	P	NP-c	NP-c	NP-c	NP-c
Unit interval	P	NP-c	?	NP-c	NP-c
Complete split	P	P	P	P	NP-c
Split	P	P	NP-c	NP-c	NP-c
Line of $K_{n,n}$	P	NP-c	NP-c	NP-c	NP-c
Complement of bipartite	P	P	?	?	NP-c
Block	P	P	P	P	P

"NP-c": NP-complete problem, "P": polynomial problem, "?": open problem.

It remains as an open problem to know if there exists any class of graphs such that (γ, μ)-coloring is NP-complete and μ-coloring can be solved in polynomial time. Among the classes considered here, the candidate classes are cographs, unit interval and complement of bipartite.

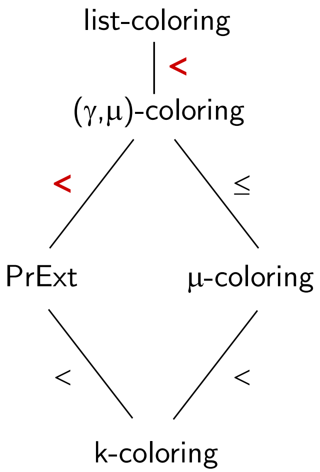
Review: complexity table for coloring problems

Class	coloring	PrExt	μ-col.	(γ, μ)-col.	list-col.
Complete bipartite	P	P	P	P	NP-c
Bipartite	P	NP-c	NP-c	NP-c	NP-c
Cographs	P	P	P	?	NP-c
Distance-hereditary	P	NP-c	NP-c	NP-c	NP-c
Interval	P	NP-c	NP-c	NP-c	NP-c
Unit interval	P	NP-c	?	NP-c	NP-c
Complete split	P	P	P	P	NP-c
Split	P	P	NP-c	NP-c	NP-c
Line of $K_{n,n}$	P	NP-c	NP-c	NP-c	NP-c
Complement of bipartite	P	P	?	?	NP-c
Block	P	P	P	P	P

"NP-c": NP-complete problem, "P": polynomial problem, "?": open problem.

For split graphs, precoloring extension can be solved in polynomial time, whereas μ-coloring is NP-complete. It remains as an open problem to find a class of graphs where the converse holds. Among the classes considered here, the candidate class is unit interval.

Review: hierarchy of coloring problems



(γ, μ)-coloring is polynomial for complete bipartite graphs

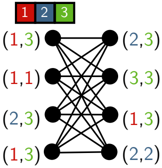
Proof: The following is a combinatorial algorithm that solves (γ, μ)-coloring in polynomial time for complete bipartite graphs.

Let $G = (V, E)$ be a complete bipartite graph, with bipartition $V_1 \cup V_2$, and let $\gamma, \mu : V \to \mathbb{N}$ such that $\gamma(v) \leq \mu(v)$ for every $v \in V$.

We have to consider two cases:

(i) There exists a vertex v such that $\gamma(v) = \mu(v)$.

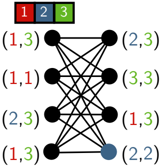
(ii) For every vertex v, $\gamma(v) < \mu(v)$.



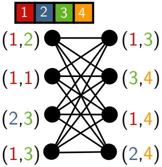
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \leq \mu(v) \leq \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:



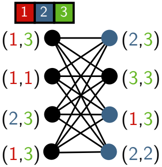
Example 2:



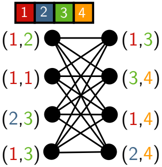
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \leq \mu(v) \leq \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:



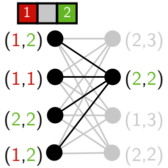
Example 2:



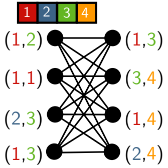
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \leq \mu(v) \leq \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:



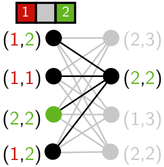
Example 2:



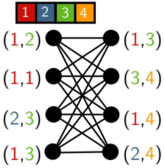
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \leq \mu(v) \leq \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:



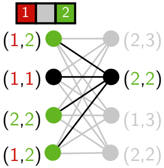
Example 2:



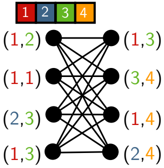
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \leq \mu(v) \leq \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:



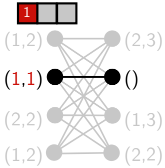
Example 2:



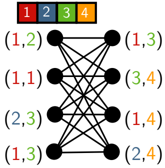
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \leq \mu(v) \leq \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:



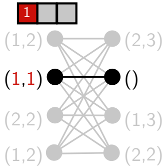
Example 2:



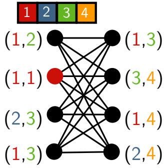
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \leq \mu(v) \leq \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:



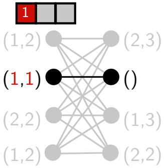
Example 2:



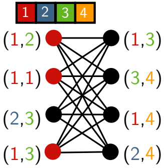
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \le \mu(v) \le \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:



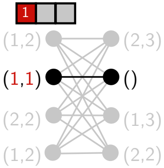
Example 2:



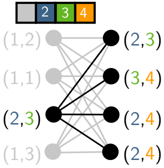
Case (i):

- If $\gamma(v) = \mu(v)$, the vertex v must be colored with color $\mu(v)$. Suppose $v \in V_2$. Since G is complete bipartite, no vertex of V_1 can use color $\mu(v)$.
- So, we can color with color $\mu(v)$ every vertex w of V_2 such that $\gamma(w) \leq \mu(v) \leq \mu(w)$ without affecting the feasibility of the problem.
- Then we remove those vertices and remove the color $\mu(v)$ from the universe of colors (we renumber the remaining colors so that they are still consecutive numbers).
- If some vertex of V_1 remains with no available color, the original graph was not (γ, μ)-colorable. Otherwise, we repeat this procedure until reaching either a coloring, or the non-colorability, or the case (ii).

Example 1:

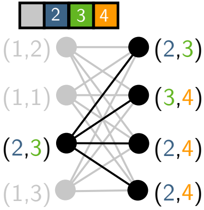


Example 2:



Case (ii):

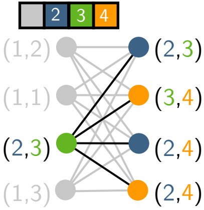
- If for every vertex v, $\gamma(v) < \mu(v)$, then every vertex has among its possible colors at least an odd color and an even color.
- So the graph is (γ, μ)-colorable, we can color the vertices of V_1 with odd colors and the vertices of V_2 with even colors.



□

Case (ii):

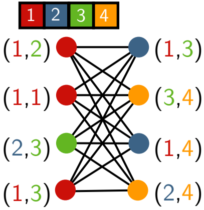
- If for every vertex v, $\gamma(v) < \mu(v)$, then every vertex has among its possible colors at least an odd color and an even color.
- So the graph is (γ, μ)-colorable, we can color the vertices of V_1 with odd colors and the vertices of V_2 with even colors.



□

Case (ii):

- If for every vertex v, $\gamma(v) < \mu(v)$, then every vertex has among its possible colors at least an odd color and an even color.
- So the graph is (γ, μ)-colorable, we can color the vertices of V_1 with odd colors and the vertices of V_2 with even colors.

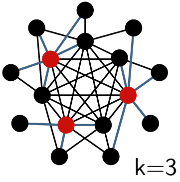


□

μ-coloring is NP-complete for split graphs

Proof: It is used a reduction from the dominating set problem on split graphs, which is NP-complete (A. Bertossi, 1984).

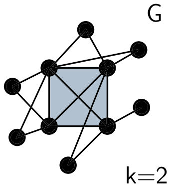
An instance of the dominating set problem on split graphs is given by a split graph G and an integer $k \geq 1$, and consists in deciding if there exists a subset D of $V(G)$, with $|D| \leq k$, and such that every vertex of $V(G)$ either belongs to D or has a neighbor in D. Such a set is called a dominating set.



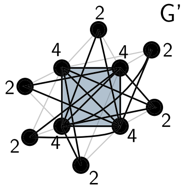
dominating set

Let G be a split graph and $k \geq 0$; $V(G) = K \cup I$, K is a complete and I is an independent set. We may assume G with no isolated vertices and $k \leq |K|$.

- We will construct a split graph G' and a function $\mu : V(G') \rightarrow \mathbb{N}$ such that G' is μ-colorable if and only if G admits a dominating set of cardinality at most k:
  - $V(G') = K \cup I$
  - K is a complete and I is an independent set in G'
  - for $v \in K$ and $w \in I$, $vw \in E(G')$ iff $vw \notin E(G)$
  - $\mu(v) = |K|$ for $v \in K$ and $\mu(w) = k$ for $w \in I$.

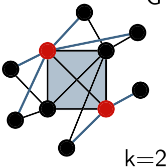


instance of split dominating set



instance of split μ-coloring

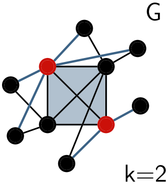
- Suppose first that G admits a dominating set D with $|D| \leq k$. Since G has no isolated vertices, G admits such a set $D \subseteq K$.



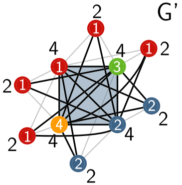
dominating set

- Let us define a μ-coloring of G' as follows:
  - color the vertices of D using different colors from 1 to $|D|$
  - color the remaining vertices of K using different colors from $|D|+1$ to $|K|$
  - for each vertex w in I, choose w' in D such that $ww' \in E(G)$ and color w with the color used by w'.

- Suppose first that G admits a dominating set D with $|D| \leq k$. Since G has no isolated vertices, G admits such a set $D \subseteq K$.



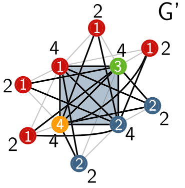
dominating set



μ-coloring

- Let us define a μ-coloring of G' as follows:
  - color the vertices of D using different colors from 1 to $|D|$
  - color the remaining vertices of K using different colors from $|D| + 1$ to $|K|$
  - for each vertex w in I, choose w' in D such that $ww' \in E(G)$ and color w with the color used by w'.

- Suppose now that G' is μ-colorable, and let $c : V(G') \to \mathbb{N}$ be a μ-coloring of G'. Since $\mu(v) = |K|$ for every $v \in K$ and K is complete in G', it follows that $c(K) = \{1, \ldots, |K|\}$.

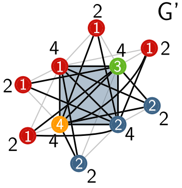


μ-coloring

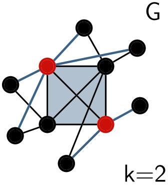
- Since $k \leq |K|$, for each vertex $w \in I$ there is a vertex $w' \in K$ such that $c(w) = c(w') \leq k$. Then $ww' \notin E(G')$, so $ww' \in E(G)$. Thus the set $\{v \in K : c(v) \leq k\}$ is a dominating set of G of size k.

□

- Suppose now that G' is μ-colorable, and let $c : V(G') \to \mathbb{N}$ be a μ-coloring of G'. Since $\mu(v) = |K|$ for every $v \in K$ and K is complete in G', it follows that $c(K) = \{1, \ldots, |K|\}$.



μ-coloring



dominating set

- Since $k \leq |K|$, for each vertex $w \in I$ there is a vertex $w' \in K$ such that $c(w) = c(w') \leq k$. Then $ww' \notin E(G')$, so $ww' \in E(G)$. Thus the set $\{v \in K : c(v) \leq k\}$ is a dominating set of G of size k.

□

Acknowledgments: To Flavia Bonomo for her valuable help in the preparation of this course.