Discrete Optimization
Lecture-14

Ngày 22 tháng 11 năm 2011

Graph COLORING

How many provinces are in Vietnam?

Graph COLORING

How many provinces are in Vietnam?
In how many colors can you color them so that two provinces sharing borders will have distinct colors?

A brief history

In 1854 Francis Guithrie noticed that he could color the map of England using only four colors so that two counties sharing a border were assigned different colors. He wondered whether every map had this property. He asked his brother who relayed the question to his professor. And so was born the famous four color problem which was not resolved until 1976.

A brief history

In 1854 Francis Guithrie noticed that he could color the map of England using only four colors so that two counties sharing a border were assigned different colors. He wondered whether every map had this property. He asked his brother who relayed the question to his professor. And so was born the famous four color problem which was not resolved until 1976.

Graph coloring has many applications. We shall see some applications, prove some basic properties, visit some algorithms with performance guarantees and get an understanding why graph coloring is a hard computational problem.

A brief review

Recall:

- The chromatic number of a graph G, $\gamma(G)$ is:

A brief review

Recall:

- The chromatic number of a graph G, $\gamma(G)$ is:
- The smallest number of colors needed to color the vertices of G so that two vertices connected by an edge get different colors.

A brief review

Recall:

- The chromatic number of a graph G, $\gamma(G)$ is:
- The smallest number of colors needed to color the vertices of G so that two vertices connected by an edge get different colors.
- The independence number of a graph G, $\alpha(G)$ is:

A brief review

Recall:

- The chromatic number of a graph G, $\gamma(G)$ is:
- The smallest number of colors needed to color the vertices of G so that two vertices connected by an edge get different colors.
- The independence number of a graph G, $\alpha(G)$ is:
- The size of the largest set of vertices no two are connected by an edge.

A brief review

Recall:

- The chromatic number of a graph G, $\gamma(G)$ is:
- The smallest number of colors needed to color the vertices of G so that two vertices connected by an edge get different colors.
- The independence number of a graph G, $\alpha(G)$ is:
- The size of the largest set of vertices no two are connected by an edge.
- The clique number of a graph G, $\omega(G)$ is:

A brief review

Recall:

- The chromatic number of a graph G, $\gamma(G)$ is:
- The smallest number of colors needed to color the vertices of G so that two vertices connected by an edge get different colors.
- The independence number of a graph G, $\alpha(G)$ is:
- The size of the largest set of vertices no two are connected by an edge.
- The clique number of a graph G, $\omega(G)$ is:
- The size of the largest set of vertices every two are connected by an edge.

A brief review

Recall:

- The chromatic number of a graph G, $\gamma(G)$ is:
- The smallest number of colors needed to color the vertices of G so that two vertices connected by an edge get different colors.
- The independence number of a graph G, $\alpha(G)$ is:
- The size of the largest set of vertices no two are connected by an edge.
- The clique number of a graph G, $\omega(G)$ is:
- The size of the largest set of vertices every two are connected by an edge.
- The chormatic index of a graph G, $\gamma_1(G)$ is:

A brief review

Recall:

- The chromatic number of a graph G, $\gamma(G)$ is:
- The smallest number of colors needed to color the vertices of G so that two vertices connected by an edge get different colors.
- The independence number of a graph G, $\alpha(G)$ is:
- The size of the largest set of vertices no two are connected by an edge.
- The clique number of a graph G, $\omega(G)$ is:
- The size of the largest set of vertices every two are connected by an edge.
- The chormatic index of a graph G, $\gamma_1(G)$ is:
- The smallest number of colors needed to color the edges so that two edges sharing a vertex will have distinct colors.

Applications examples

1. The following generic situation is common in many applications. Assume that you have k jobs and n processors that can process the jobs (printers connected to a network, cranes in construction sites, lecture rooms at a university etc.). Some jobs cannot use the same processor. We can construct a graph whose vertices are connected by an edge if they cannot be processed by the same processor.

Applications examples

1. The following generic situation is common in many applications. Assume that you have *k* jobs and *n* processors that can process the jobs (printers connected to a network, cranes in construction sites, lecture rooms at a university etc.). Some jobs cannot use the same processor. We can construct a graph whose vertices are connected by an edge if they cannot be processed by the same processor.

The chromatic number of this graph is the smallest number of processors needed to do all jobs.

Applications examples

1. The following generic situation is common in many applications. Assume that you have *k* jobs and *n* processors that can process the jobs (printers connected to a network, cranes in construction sites, lecture rooms at a university etc.). Some jobs cannot use the same processor. We can construct a graph whose vertices are connected by an edge if they cannot be processed by the same processor.

The chromatic number of this graph is the smallest number of processors needed to do all jobs.

2. Allocations of wavelengths for wireless communication devices (mobile phones, radio stations, TV, police communications, Firefighters, etc.). Some, depending for instance of distances cannot be assigned overlapping frequencies. So construct a graph whose vertices are the devices and connect two of them by an edge if they must be assigned different wavelength.

Applications examples

1. The following generic situation is common in many applications. Assume that you have *k* jobs and *n* processors that can process the jobs (printers connected to a network, cranes in construction sites, lecture rooms at a university etc.). Some jobs cannot use the same processor. We can construct a graph whose vertices are connected by an edge if they cannot be processed by the same processor.

The chromatic number of this graph is the smallest number of processors needed to do all jobs.

2. Allocations of wavelengths for wireless communication devices (mobile phones, radio stations, TV, police communications, Firefighters, etc.). Some, depending for instance of distances cannot be assigned overlapping frequencies. So construct a graph whose vertices are the devices and connect two of them by an edge if they must be assigned different wavelength.

The chromatic number of this graph is the smallest number of

Some easy observations

a. An even cycle is 2-chromatic, an odd cycle is 3-chromatic.

Some easy observations

a. An even cycle is 2-chromatic, an odd cycle is 3-chromatic.
b. If G_1 is a subgraph of G then $\gamma(G_1) \leq \gamma(G)$.

Some easy observations

a. An even cycle is 2-chromatic, an odd cycle is 3-chromatic.
b. If G_1 is a subgraph of G then $\gamma(G_1) \leq \gamma(G)$.
c. $\gamma(G) \geq \omega(G)$.

Some easy observations

a. An even cycle is 2-chromatic, an odd cycle is 3-chromatic.
b. If G_1 is a subgraph of G then $\gamma(G_1) \leq \gamma(G)$.
c. $\gamma(G) \geq \omega(G)$.
d. $\gamma_1(G) \geq \Delta(G)$.

Some easy observations

a. An even cycle is 2-chromatic, an odd cycle is 3-chromatic.
b. If G_1 is a subgraph of G then $\gamma(G_1) \leq \gamma(G)$.
c. $\gamma(G) \geq \omega(G)$.
d. $\gamma_1(G) \geq \Delta(G)$.

(2-colorable graphs)

Recall: we can efficiently decide whether a graph G is bipartite, or it can be efficiently colored by two colors and if not we can easily produce a certificate that proves that it cannot.

Some easy observations

a. An even cycle is 2-chromatic, an odd cycle is 3-chromatic.
b. If G_1 is a subgraph of G then $\gamma(G_1) \leq \gamma(G)$.
c. $\gamma(G) \geq \omega(G)$.
d. $\gamma_1(G) \geq \Delta(G)$.

(2-colorable graphs)

Recall: we can efficiently decide whether a graph G is bipartite, or it can be efficiently colored by two colors and if not we can easily produce a certificate that proves that it cannot.

Question

What about 3 colors?

Some easy observations

a. An even cycle is 2-chromatic, an odd cycle is 3-chromatic.
b. If G_1 is a subgraph of G then $\gamma(G_1) \leq \gamma(G)$.
c. $\gamma(G) \geq \omega(G)$.
d. $\gamma_1(G) \geq \Delta(G)$.

(2-colorable graphs)

Recall: we can efficiently decide whether a graph G is bipartite, or it can be efficiently colored by two colors and if not we can easily produce a certificate that proves that it cannot.

Question

What about 3 colors?

Answer

Surprisingly, this is a very hard computational problem! It is indeed so hard that a safe security password scheme can be based on it.

3-SAT is reducible to 3-coloring of a graph

Imagine that you have a magic box. Inside it lives a very smart machine. It accepts graphs G with n vertices and within $p(n)$ nano-seconds it will decide whether the graph can be properly colored by 3 colors.

3-SAT is reducible to 3-coloring of a graph

Imagine that you have a magic box. Inside it lives a very smart machine. It accepts graphs G with n vertices and within $p(n)$ nano-seconds it will decide whether the graph can be properly colored by 3 colors.

Question

What makes this process so special ?

3-SAT is reducible to 3-coloring of a graph

Imagine that you have a magic box. Inside it lives a very smart machine. It accepts graphs G with n vertices and within $p(n)$ nano-seconds it will decide whether the graph can be properly colored by 3 colors.

Question

What makes this process so special ?

Answer

It turns out that if a magic box that accepts $3 - SAT$ *instances and can decide efficiently whether a given instance is satisfiable exists, then a large class of problems belonging to the class NP (to be discussed later) can be efficiently solved.*

3-SAT is reducible to 3-coloring of a graph

Imagine that you have a magic box. Inside it lives a very smart machine. It accepts graphs G with n vertices and within $p(n)$ nano-seconds it will decide whether the graph can be properly colored by 3 colors.

Question

What makes this process so special ?

Answer

It turns out that if a magic box that accepts $3 - SAT$ instances and can decide efficiently whether a given instance is satisfiable exists, then a large class of problems belonging to the class NP (to be discussed later) can be efficiently solved.

3-SAT is reducible to 3-coloring of a graph

Imagine that you have a magic box. Inside it lives a very smart machine. It accepts graphs G with n vertices and within $p(n)$ nano-seconds it will decide whether the graph can be properly colored by 3 colors.

Question

What makes this process so special ?

Answer

It turns out that if a magic box that accepts $3-SAT$ *instances and can decide efficiently whether a given instance is satisfiable exists, then a large class of problems belonging to the class NP (to be discussed later) can be efficiently solved.*

This means that we can also use the 3*-coloring magic box to do the same thing!*

Theorem

Given a 3-SAT instance with n variables and m clauses we can construct a graph G with $< 8m$ vertices such that G is 3-colorable if and only if the given $3-SAT$ instance is satisfiable.

Theorem

Given a 3-SAT instance with n variables and m clauses we can construct a graph G with $< 8m$ vertices such that G is 3-colorable if and only if the given $3-SAT$ instance is satisfiable.

(Proof)

The graph will be constructed in steps:
Step 1: We start with 3 special vertices labeled T, F, B that form a triangle. They will also be our 3 colors.This guarantees that they will requre 3 distinct colors.

Theorem

Given a 3-SAT instance with n variables and m clauses we can construct a graph G with $< 8m$ vertices such that G is 3-colorable if and only if the given $3-SAT$ instance is satisfiable.

(Proof)

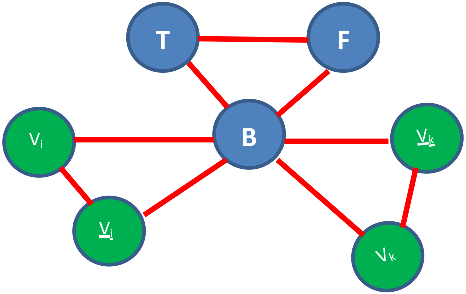
The graph will be constructed in steps:
Step 1: We start with 3 special vertices labeled T, F, B that form a triangle. They will also be our 3 colors.This guarantees that they will requre 3 distinct colors.

Theorem

Given a 3-SAT instance with n variables and m clauses we can construct a graph G with $< 8m$ vertices such that G is 3-colorable if and only if the given $3-SAT$ instance is satisfiable.

(Proof)

The graph will be constructed in steps:
Step 1: We start with 3 special vertices labeled T, F, B that form a triangle. They will also be our 3 colors.This guarantees that they will requre 3 distinct colors.
Step 2: With each boolean variable x_i we will associate two vertices labeled $v_i, \underline{v}_i$. For each i, $\{v_i, \underline{v}_i, B\}$ will also form a triangle.


T
F
B
Vi
Vj
Vk
Vk

Chứng minh.

Step 3: For each clause we add a subgraph consisting of six new vertices and the edges as described by the following figure: (see next slide).

Chứng minh.

Step 3: For each clause we add a subgraph consisting of six new vertices and the edges as described by the following figure: (see next slide).

Chứng minh.

Step 3: For each clause we add a subgraph consisting of six new vertices and the edges as described by the following figure: (see next slide).

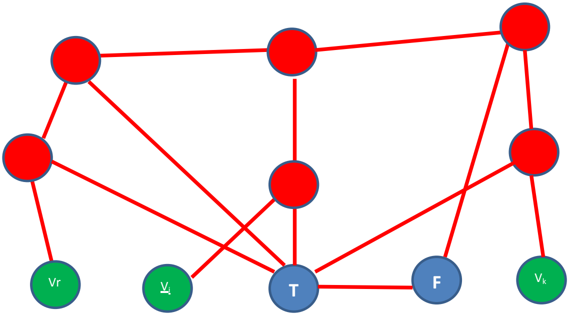
Step 4: Observe that if G is 3-colorable then either $v_i = T \ or \ \bar{v}_i = T$ but not both. This is a truth assignment for the 3-SAT instance.

Chứng minh.

Step 3: For each clause we add a subgraph consisting of six new vertices and the edges as described by the following figure: (see next slide).

Step 4: Observe that if G is 3-colorable then either $v_i = T$ *or* $\bar{v}_i = T$ but not both. This is a truth assignment for the 3-SAT instance.

Step 5: for instance, if in the clause $(x_i \vee \bar{x}_j \vee x_k)$ all boolean variables are F then the associated graph G will not be 3 colorable. □


Vr
Vi
T
F
Vk

Theorem

For any graph G, $\gamma(G) \leq \Delta(G) + 1$.

Theorem

For any graph G, $\gamma(G) \leq \Delta(G) + 1$.

Chứng minh.

Obvious. □

Theorem

For any graph G, $\gamma(G) \leq \Delta(G) + 1$.

Chứng minh.

Obvious. □

Remark (A coloring tool)

We can sequence the vertices in some order say $v_1, v_2, \ldots, v_n$ *and use a greedy coloring to color the vertices.*

Theorem

For any graph G, $\gamma(G) \leq \Delta(G) + 1$.

Chứng minh.

Obvious. □

Remark (A coloring tool)

We can sequence the vertices in some order say $v_1, v_2, \ldots, v_n$ and use a greedy coloring to color the vertices.

Theorem

For any graph G, $\gamma(G) \leq \Delta(G) + 1$.

Chứng minh.

Obvious. □

Remark (**A coloring tool)**

We can sequence the vertices in some order say $v_1, v_2, \ldots, v_n$ *and use a greedy coloring to color the vertices.*

This shows that determining the chromatic number is conceptually easy: just try all sequencings and choose the one that produces the smallest number of colors.

Theorem

For any graph G, $\gamma(G) \leq \Delta(G) + 1$.

Chứng minh.

Obvious. □

Remark (**A coloring tool)**

We can sequence the vertices in some order say $v_1, v_2, \ldots, v_n$ *and use a greedy coloring to color the vertices.*

This shows that determining the chromatic number is conceptually easy: just try all sequencings and choose the one that produces the smallest number of colors.

Yes, just try it on a graph with 1000 vertices...

Theorem (Brooks Theorem)

If a connected graph $G \neq K_{n+1}$ and $\Delta(G) = n > 2$ then $\gamma(G) \leq \Delta(G)$.

Theorem (Brooks Theorem)

If a connected graph $G \neq K_{n+1}$ and $\Delta(G) = n > 2$ then $\gamma(G) \leq \Delta(G)$.

Chứng minh.

Watch the blackboard. □