The Tourist Problem: And Fun with Graph Modeling

Hon Wai Leong
Department of Computer Science
National University of Singapore
Email, FB: leonghw@comp.nus.edu.sg

Experience the fun of problem solving
The Tourist Problem

Organization

- The Tourist Problem
- Analysis and Simplifications
- Problem Modeling (with Graphs)
- Solving the Graph Model
- Mapping back the Solution
- Moral of the Story

Experience the fun of problem solving
The Tourist Problem…

Given: A list of tourists, each with his/her list of places to visit.
To do: Schedule bus rides for them so that each tourist visits all the places in his/her list.

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
<tr>
<td>Frances</td>
<td>BG, SZG, JB</td>
</tr>
<tr>
<td>Gary</td>
<td>CG, OR</td>
</tr>
<tr>
<td>Harry</td>
<td>JG, CG</td>
</tr>
</tbody>
</table>
Good to know the entities we are dealing with…

- **The Tourists:**
  \[ T = \{ A, B, C, D, E, F, G, H \} \]

- **The Attractions (Places):**
  \[ P = \{ BG, CG, JB, JG, OR, SI, VC, SZG \} \]

<table>
<thead>
<tr>
<th>Place</th>
<th>Common Name</th>
<th>Place</th>
<th>Common Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG</td>
<td>Botanical Gardens</td>
<td>CG</td>
<td>Chinese Gardens</td>
</tr>
<tr>
<td>JB</td>
<td>Jurong Birdpark</td>
<td>JG</td>
<td>Japanese Gardens</td>
</tr>
<tr>
<td>OR</td>
<td>Orchard Road</td>
<td>SI</td>
<td>Sentosa Island</td>
</tr>
<tr>
<td>SZG</td>
<td>Spore Zoological Gardens</td>
<td>VC</td>
<td>VivoCity</td>
</tr>
</tbody>
</table>
Some Simplifications: Consider

- Aaron \{ SZG, BG, JB \}
- Frances \{ SZG, BG, JB \}

Also consider

- David \{ JG, CG, OR \}
- Gary \{ CG, OR \}

An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
<tr>
<td>Frances</td>
<td>BC, SZG, JB</td>
</tr>
<tr>
<td>Gary</td>
<td>CC, OR</td>
</tr>
<tr>
<td>Harry</td>
<td>JG, CG</td>
</tr>
</tbody>
</table>

Simplification Rule:

If \( P(T_1) \subseteq P(T_2) \), then tourist \( T_1 \) can just “follows” tourist \( T_2 \). Thus, we can omit \( T_1 \) from consideration.

Oh, can also omit Harry

- Betty \{ CG, JG, BG \}
- Harry \{ CG, JC \}
The (Reduced) Tourist Problem...

**Given:** A list of tourist, each with his/her list of places to visit.

**To do:** Schedule bus rides for them so that each tourist visits all the places in his/her list.

### An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
</tbody>
</table>

\[ T = \{ A, B, C, D, E \} \]

\[ P = \{ BG, CG, JB, JG, OR, SI, VC, SZG \} \]
Given: A list of tourists, each with his/her list of places to visit.

To do: Schedule bus rides for them so that each tourist visits all the places in his/her list.

Solution: (Singapore 1-Day Tour)

Put all the tourists on one bus. Visit all eight places in 1 day.

An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
</tbody>
</table>

What’s Good: It works! One bus, one-day.

What’s Bad: Too rushed. NO time to see anything!

Not interesting!
Given: A list of tourists, each with his/her list of places to visit.
To do: Schedule bus rides for them so that each tourist visits all the places in his/her list, and C1: Each tourist visits at most one place a day.

Simple Solution: Schedule one trip to every place every day.

An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
</tbody>
</table>

What’s Good: It works! Finish in 3 days. (minimum!)
What’s Bad: Wasteful! 24 bus trips.

Also, not so interesting!
The Tourist Problem – v0.8

Given: A list of tourist, each with his/her list of places to visit.

To do: Schedule bus rides for them so that each tourist visits all the places in his/her list,

C1: Each tourist visits \textit{at most one place a day, and}

C2: There is \textit{at most one bus trip to each place}

Simple Solution:

\begin{itemize}
    \item Schedule \textit{one trip per day, each to a different place.}
\end{itemize}

An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
</tbody>
</table>

What’s Good: It works! 8 trips.
What’s Bad: It takes 8 days!

\textbf{But wait… Did you see something interesting?}
Given: A list of tourists, each with his/her list of places to visit.

To do: Schedule bus rides for them so that each tourist visits all the places in his/her list,

C1: Each tourist visits \textit{at most one place a day},
C2: There is \textit{at most one bus trip to each place, and}
C3: minimize the number of days to complete mission.

**Observation:**

On the same day, cannot schedule SZG and BG can schedule SZG and OR

**An Instance of Tourist Problem**

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
</tbody>
</table>

How to model all these constraints?
Activity Period #1:

Bus Scheduling DIY (Do It Yourself)

(5 minutes)
Activity Period 1: (5 minutes) [Tourist Problem ]

Bus Scheduling DIY: (Do It Yourself)

Tourist Problem Version 1.0
Given: A list of tourist, each with his/her list of places to visit.
To do: Schedule bus rides for them so that each tourist visits all the places in his/her list, and
C1: Each tourist visits at most one place a day,
C2: There is at most one bus trip to each place, and
C3: minimize the number of days to complete mission.

An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
</tbody>
</table>

Q1: Using the above information, try to schedule the bus trips and minimize the number of days needed to complete all the bus trips. (Make sure there are no conflicts.)

Day 1: __________________________
Day 2: __________________________
Day 3: __________________________
Day 4: __________________________
Review of Activity #1

- How many days did you use?
  - ____ days

- What was the main difficulty?
  - What if we are talking about 100 tourists?
  - … and 20 different attractions?

- Was there a lot of repetitive task?
  - How was the task?

- How can we do better?
The Graph Model

What is a graph?

- eg: \( y = \sin(bx) \)

No. Not this type of graph.
The Graph Model

Graph \( G = (V, E) \)

- \( V \) is a set of vertices, nodes (circles)
- \( E \) is a set of edges (connections)

An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
</tbody>
</table>

In our graph, nodes are places, and edges in the graph means conflicts.
Graph Model for the Tourist Problem

An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
</tbody>
</table>

The graph $G = (V, E)$ captures all the conflicts for our tourist problem instance.
Graph Model for the Tourist Problem

What’s good about the graph model?

- very simple!
- easy to spot conflicts

and the non-conflicts

\[ G = (V, E) \]

On Day 1, can schedule \( \text{SZG, OR} \)
[Any more? Why?]

On Day 2, can schedule \( \text{JB, CG, VC} \)

On Day 3, can schedule \( \text{BG, SI} \)

On Day 4, can schedule \( \text{JG} \)
Graph Coloring Problem

- Given a graph $G = (V, E)$, colour the vertices in $V$ so that any two vertices that are connected by an edge in $E$ will have different colors.

We want to minimize the number of colors.

$G = (V, E)$

Number of colours used to colour the graph $G$ = Number of days needed to complete the schedule
Activity Period #2:

Graph Colouring Exercises
(10 minutes)
Activity Period 2: (8 minutes) [The Tourist Problem]

The Tourist Problem

Your Name: _________________________________________

The tourist problem instance in the lecture can be modeled with the following graph. Two possible colorings of the graph are given in the lecture.

Q1: Give a different way to color the vertices of the graph on the left. How many colors?

Q1: # colors: ______

Q2: Try coloring the following graphs with the minimum number of colors.

Q2(a): # colors: ______

Q2(b): # colors: ______

Q3: What about this one (below, left)?

Q3: # colors: ______

Q4: Can you color the graph (above, right) with only three (3) colors.

Q4: With 3 colors? YES / NO

Hon Wai Le
Review of Activity #2

- Is Graph Colouring fun?
  - Did you *really* use different colours?

- How many colours was did you use (Q1)?

- What about the *cycles* (Q2):
  - Q2(a): $C_6$ (a cycle of length 6)?
  - Q2(b): $C_5$ (a cycle of length 5)?
  - What else can you say?

- What about the graph in Q3?

- What about Q4?
  - Why
1. What about the list of tourists on each bus? Can we get it from the graph model? 
NO. Why NOT.
### An Instance of Tourist Problem

<table>
<thead>
<tr>
<th>Tourist</th>
<th>Places of Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>SZG, BG, JB</td>
</tr>
<tr>
<td>Betty</td>
<td>CG, JG, BG</td>
</tr>
<tr>
<td>Cathy</td>
<td>VC, SI, OR</td>
</tr>
<tr>
<td>David</td>
<td>JG, CG, OR</td>
</tr>
<tr>
<td>Evans</td>
<td>CG, JG, SZG</td>
</tr>
<tr>
<td>Frances</td>
<td>BG, SZG, JB</td>
</tr>
<tr>
<td>Gary</td>
<td>CG, OR</td>
</tr>
<tr>
<td>Harry</td>
<td>JG, CG</td>
</tr>
</tbody>
</table>

### Alternative Representation

<table>
<thead>
<tr>
<th>Tourist</th>
<th>BG</th>
<th>CG</th>
<th>JB</th>
<th>JG</th>
<th>OR</th>
<th>SI</th>
<th>SZG</th>
<th>VC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Betty</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Cathy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>David</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evans</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frances</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gary</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Harry</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Hon Wai Leong, SoC, NUS

Copyright © 1987-- by Leong Hon Wai
Get Solutions to Tourist Problem (2)

Coloured graph ⇒ “Bus Schedule”

\[ G = (V, E) \]

1. What about the list of tourists on each bus?
2. What if you only have 2 buses?
   - can colour vertex VC green.
Coloured graph \( \Rightarrow \) “Bus Schedule”

\[ G = (V, E) \]

<table>
<thead>
<tr>
<th>Color</th>
<th>Day</th>
<th>Place</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>SZG, OR</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>JB, CG, VC</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>BG, SI</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>JG</td>
</tr>
</tbody>
</table>

1. What about the list of tourists on each bus?
2. What if you only have 2 buses?
3. What if BG is closed on Day 3?
   • Can we re-order the colours?
Coloured graph ⇒ “Bus Schedule”

\[ G = (V, E) \]

1. What about the list of tourists on each bus?
2. What if you only have 2 buses?
3. What if BG is closed on Day 3?
4. Can we use fewer colours (fewer days)?
Abstract Problem Modeling…

Real Life Problem

Abstract Model

Model Solution

Real Life Solution

Map to model

“Solve” the model

Map back from model
Nothing new. You do it *all the time.*

In a farm there are 15 chickens and goats. Together, there are 40 animal feet. How many chickens are there?

There are ?? chickens.

```
Map to model

Abstract Model

“Solve” the model

Model Solution

Map back from model
```
Modeling: An example

Using a Direct Method

In a farm there are 15 chickens and goats. Together, there are 40 animal feet. How many chickens are there?

There are ?? chickens.

Draw picture,
Count legs,
Adjust as needed.
Modeling: An example

Using modeling process

In a farm there are 15 chickens and goats. Together, there are 40 animal feet. How many chickens are there?

Use Algebra: define \( x \) & \( y \)

Let \# of chickens be \( x \), and \# of goats be \( y \),
Then, \( x + y = 15 \),
\( 2x + 4y = 40 \).

Solve equations

Solution: \( x = 5 \), \( y = 10 \).

Map back \( x \) & \( y \)

There are 5 chickens.**
Problem transformation...

the model transforms original problem to a new abstract problem that is easier to solve.
Bend a steel bar

Real Life Problem

Real Life Solution

Man bending steel rod

(Direct method)
Modeling: Another example (2)

- Bend a steel bar (using transformation)

Real Life

Problem

Real Life

Solution

Transformed

problem

Transformed

Solution

Solve

Problem

Bend it

easily

Heat

Bend it

easily

Cool down

Eureka!!

Heat becomes soft when red hot...
Abstract Problem Modeling…

- **Real Life Problem**
  - Map to model

- **Abstract Model**
  - Solve the model

- **Model Solution**

- **Real Life Solution**
  - Map back from model
Modeling: *Have you seen it b4?*

Nothing new. You do it *all the time.*

In a farm there are 15 chickens and goats. Together, there are 40 animal feet. How many chickens are there?

*Use Algebra:*
define $x$ & $y$

Let # of chickens be $x$, and # of goats be $y$,
Then, $x + y = 15$,
$2x + 4y = 40$.

*Solve equations*

Solution: $x = 5$,
$y = 10$.

There are 5 chickens.**

Map back from $x$ & $y$

There are ?? chickens.

Map to model

Use Algebra:
define $x$ & $y$
Problem \textit{transformation}... \\

\textbf{the model transforms} \textbf{original problem to a new abstract problem}
Modeling: Another example

- Bend a steel bar (3cm in diameter)

(Direct method)

Man bending steel rod
Bend a steel bar (using transformation)

Real Life Problem

Transformed problem

Solve problem

Bend it easily

Real Life Solution

Transformed Solution

Heat

Cool down

Eureka!!

becomes soft when red hot...
Modelling in Tourist Problem

Recap: Our Graph modelling...

<table>
<thead>
<tr>
<th>Graph Model</th>
<th>Tourist Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>places</td>
</tr>
<tr>
<td>Edges / Conflicts</td>
<td>tourist want to visit both places</td>
</tr>
<tr>
<td>Colors</td>
<td>bus trips to places</td>
</tr>
<tr>
<td>Others</td>
<td>The tourists</td>
</tr>
</tbody>
</table>
Moral of the Story

The Tourist Problem:
- Some problems are EASY. (*don’t* complicate them)
- Get a *simple* solution first.
  - then *analyze* it, *improve* it, *refine* it.
- Solution depend on the questions asked
- It is important to *ask* questions.
- Theoretical *modeling and analysis* are beneficial

Modeling
- Abstract modeling *simplifies* problem and solution!
- Abstract model is *transferable*.
- Models don’t *answer* everything.
Where else is Graph Colouring used?

- The Tourist Problem [done]
- Map Colouring
- Fish in a Tank
- Frequency assignment in wireless networks
- Time Table Scheduling
- And a whole lot more…

Experience the fun of problem solving
The Map Coloring Problem

We want to color countries, oceans, lakes, and islands on a map so that no two adjacent areas have the same color.

A legal colouring. But uses > 10 colours!

Can we do it with only 4 colours?
The Map Coloring Problem

We want to color countries, oceans, lakes, and islands on a map so that no two adjacent areas have the same color.

Two colors

Three colors

Four colors
Map and Graph Coloring

Map to be colored

Equivalent graph

Coloring the graph

(The Tourist Problem) Page 45
The Four Color Conjecture

Question: (from 1852…)

Can all maps be coloured using only four colours?

HISTORY:

- 1852 Conjecture (Guthrie → DeMorgan)
- 1878 Publication (Cayley)
- 1879 First proof (Kempe)
- 1880 Second proof (Tait)
- 1890 Rebuttal (Heawood)
- 1891 Second rebuttal (Petersen)
Does four colour suffices?

Martin Gardner published in Scientific American (April 1975) this map of 110 regions. He claimed that the map requires five colors and constitutes a counterexample to the four-color theorem.
Does four colour suffices?

Martin Gardner published in Scientific American (April 1975) this map of 110 regions. He claimed that the map requires five colors and constitutes a counterexample to the four-color theorem.

However, the coloring of Wagon clearly shows that this map is, in fact, four-colorable. (The coloring was obtained algorithmically using Mathematica)

Source: http://mathworld.wolfram.com/Four-ColorTheorem.html

April Fool Joke!
The Four Color Conjecture

HISTORY (continued):

- . . .
- 1891 Second rebuttal (*Petersen*)
- 1913 Reducibility, connexity (*Birkhoff*)
- 1922 Up to 25 regions (*Franklin*)
- 1969 Discharging (*Heesch*)
- 1976 Four Color Thm (*Appel & Haken*) @UIUC
- 1995 Streamlining (*Robertson & al.*)
- 2005 COQ proof (*Gonthier*)
In Fall 1979, I took a course CS313 Combinatorics by Ken Appel.
On 25-Oct-2015, his wife, Carole Appel visited the UIUC campus. At the Quad, I happened to meet her!
Activity Period #3:

Map Colouring &
Fish in a Tank
(10 minutes)
Review of Hands-on Activity #3

- How many colours did the map need?
  - You should never need more than 4 colours

- Did you know about the “Four-Colour Theorem”?

- How many fish tanks did you need?
Activity 4: Color These Maps
Use as few colors as possible

Real map: One color already used

Made-up map
## Summary of Problem Modelling

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Tourist Problem</th>
<th>Fish in a tank</th>
<th>Frequency Assignment</th>
<th>Map Coloring</th>
</tr>
</thead>
<tbody>
<tr>
<td>Places</td>
<td>tourist want to visit both places</td>
<td>cannot be placed in same tank</td>
<td>interference if placed too near</td>
<td>share a common border</td>
</tr>
<tr>
<td>Fishes</td>
<td>radio stations</td>
<td>Countries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Others</td>
<td>The tourists</td>
<td>--</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Other conflicts:**

- **Edges / Conflicts:** Bus trips to places, interference if placed too near, share a common border
- **Colors:** Fish tanks, signal frequencies, color
- **Others:** The tourists, --
Why CS dept teach abstract problems?

One abstract model, *many application*;
One graph-colouring program, *many applications.*
References…

On Graph Coloring and Applications:
2. http://www.colorado.edu/education/DMP/activities/graph/ddghnd03.html
3. Lots of other links available

On the Four Color Theorem:
End of Talk on Tourist Problem!

If you want to contact me,
go email, MSN, FB at leonghw@comp.nus.edu.sg