1er Cours : Introduction
MPRI 2011–2012

Michel Habib
habib@liafa.jussieu.fr
http://www.liafa.jussieu.fr/~habib

Chevaleret, septembre 2011

Schedule

First question

Why such a course ?

Fagin's theorems in structural complexity

Characterizations of P and NP using graphs and logics fragments.

NP

The class of all graph-theoretic properties expressible in existential second-order logic is precisely NP.

P

The class of all graph-theoretic properties expressible in Horn existential second-order logic with successor is precisely P.

Practical issues

Graphs made up with vertices and edges (or arcs) provide a very powerful tool to model real-life problems. Weights on vertices or edges can be added.
Many applications involve graph algorithms, in particular many facets of computer science !
Also many new applications in social sciences (Web, Facebook, Google+ . . .).

The great importance of the right model

It is crucial to use all the characteristics of the problem you want to solve and find a good model. Two examples from Peter Winkler's nice book on Mathematical Puzzles :
All the discrete structures considered her are supposed to be finite.

1. An odd number of soldiers in a field in such a way that all pairwise distances are different. Each soldier is told to keep an eye on the nearest other soldier.
   Show that at least one soldier is not being watched.
2. A problem on rectangles :
   A large rectangle of the plane is partitioned into smaller rectangles, each of witch has either integer height or integer width (or both). Prove that the large rectangle also has this property.

Graph invariants

A graph invariant is a function $G \to f(G) \in N$, which is invariant under isomorphism (i.e. if G and H are isomorphic then $f(G) = f(H)$).

Which algorithms ?

- Optimal ?
- Linear
- Efficient ?
- Simple ?
- Easy to program

In our work on optimization problems on graphs, we will met several complexity barriers :

- Polynomial versus NP hard
  In this case an algorithm running in $O(n^{75}.m^{252})$ could be a nice result !
- Linear versus Boolean matrix multiplication
  Algorithms running in $O(n.m)$, hard to find a lower bound !
- Practical issues need linear algorithms in order to be applied
  - on huge graphs such as Web graph ....
  - many times such as inheritance in object oriented programming.
  - Need for heuristics !

Our Claims or thesis

An efficient algorithm running on a discrete structure is always based :

- on a theorem describing a combinatorial structure
- a combinatorial decomposition of this discrete structure.
- or in some other cases a geometric representation of the structure provides the algorithm.

Examples

- Chordal graph recognition and maximal clique trees (particular case of **treewidth**).
- Transitive orientation and modular decompostion.
- Max Flow and decomposing a flow in a sum of positive circuits.
- Greedy algorithms for minimum spanning trees and matroids

Uses of geometric representations

- Computing a maximum clique (of maximum size) or an efficient representation of an interval graph.
- Numerous algorithms on planar graphs use the existence of a dual graph. (Ex : flows transform into paths)

Academic year : 2011-2012

- Research team : Distributed algorithms and graphs, LIAFA :
- Michel Habib (Pr Univ. Paris Diderot)
  Jean-Sébastien Séréni (CR1, CNRS)
  Pierre Charbit (MdC, Univ. Paris Diderot)
- Nicolas Trotignon (CR1 CNRS, LIP at Lyon)
- contact : Prenom.nom@liafa.jussieu.fr

Schedule of the course

1. Introduction (this course)
2. Chordal and interval graphs and their nice structure
3. Algorithms for modular decomposition and extensions
4. Treewidth and other width parameters (parametrized complexity)
5. Graph Searches, a new approach
6. Diameter Computations

Examples of algorithmic problems

- shortest paths, reachability (existence of a path)
- Center and diameter computations
- Graph encoding, distance labeling
- Routing
- Sandwich (or completion) problems
- Colorations

Decompositions used here

- modular decomposition
- split decomposition
- Cliquewidth)
- Treewidth, pathwidth
- Branchwidth and rankwidth

Graph classes used here

- Tournaments (Tournois), posets, bipartite graphs
- Cographs (P_4-free)
- Interval graphs
- Chordal
- Comparability
- Asteroïdal triple free
- Distance-hereditary
- Permutation
- Perfect Graphs
- Convex, bi-convex, . . .

Typical questions

- Recognition algorithms, if possible linear-time and **robust**.
- Efficient decomposition algorithms
- Computations of compact encodings and representation
- Random generation and enumeration
- Routing protocols and diameter estimation
- Computation of some invariant (for example mini coloring or max clique).

Application domains

Themes

- Bioinformatics (mainly phylogeny and other graph problems)
- Networks and Distributed systems
- Analysis of huge graphs in social sciences (ranking, clustering)

GANG

An INRIA–LIAFA project on graphs an networks, dir. L. Viennot

SAE

A biological group at UPMC with E. Bapteste and P. Lopez

Social sciences

A collaboration with D. Cardon (Orange Lab) and C. Prieur (LIAFA)

Notation

For a finite loopless undirected graph $G = (V, E)$
V set of vertices
E edges set
$|V| = n$ and $|E| = m$

Some exercises

1. A linear algorithm for isomorphism on trees
2. Computation of the diameter of a tree
3. Propose an algorithm which determines if a clique of size $\Delta + 1$ exists in a graph with maximum degree Δ ?

Solution de l'exercice

- Propose an algorithm which determines if a clique of size $\Delta + 1$ exists in a graph with maximum degree Δ ?
- A solution
  Let S be the set of vertices with degree Δ of the graph.
  Until it exists $a \in S$ with a neighbour $\notin S$, delete a from S.
  The remaining connected components of size exactly $\Delta + 1$ are the solutions.
- The algorithm requires $O(n + m)$.

Bound on the number of edges

Triangle free graphs

Show that if G has no triangle then :
$|E| \leq \frac{|V|^2}{4}$

Planar graphs

Show that if G is a simple planar graph (i.e. without loop and parallel edge) then :
$|E| \leq 3|V| - 6$

Classes of twin vertices

Definition

x and y are called false twins, (resp. true twins) if $N(x) = N(y)$ (resp. $N(x) \cup \{x\} = N(y) \cup \{y\}$)

Exercise

Propose a good algorithm to compute these classes

Research Problems

- Find a linear-time algorithm that minimizes a deterministic automaton.
- Find a linear-time algorithm which computes a doubly lexicographic ordering of a 0-1 matrix, i.e. an ordering of the columns and the lines for which lines and columns appear to be lexicographically ordered.

Example of a doubly lexicographic ordering

	C1	C2	C3	C4	C5
L1	1	0	1	0	1
L2	0	1	0	1	1
L3	1	1	0	1	1
L4	0	0	0	0	1
L5	0	1	1	0	0

	C3	C1	C4	C5	C2
L4	0	0	0	1	0
L1	1	1	0	1	0
L5	1	0	0	0	1
L2	0	0	1	1	1
L3	0	1	1	1	1

- Such an ordering always exists
- Best algorithm to compute one :
  Paige and Tarjan [1987] proposed an $O(LlogL)$ where $L = n + m + e$ for a matrix with n lines, m columns and e non zero values, using partition refinement.
- For undirected graphs, such an ordering of the symmetric incidence matrix, yields an ordering of the vertices which has nice properties.

- Implicit hypothesis : the memory words have k bits with $k > \lceil log(|V|) \rceil$
- To be sure, consider the bit encoding level

- Adjacency lists
  $O(|V| + |E|)$ memory words
  Adjacency test : xy is an arc in $O(|N(x)|)$
- Adjacency Matrix
  $O(|V|^2)$ memory words (can be compressed)
  Adjacency test : xy is an arc in $O(1)$
- Customized representations, a pointer for each arc . . .

For some large graphs, the Adjacency matrix, is not easy to obtain and manipulate.
But the neighbourhood of a given vertex can be obtained. (WEB Graph or graphs is Game Theory)

Quicksands

- To compute this invariant or this property of a given graph G one needs to "see"(or visit) every edge at least once.
- False statement as for example the computation of twins resp. connected components on $\overline{G}$ knowing G.

Exercise

Can the advantages of the 2 previous representations can be mixed in a unique new one ?
Adjacency lists : construction in $O(n+m)$
Inicence matrix : cost of the query : $xy \in E$? in $O(1)$

In other words

Using $O(n^2)$ space, but with linear time algorithms on graphs ?

Auto-complemented representations

Initial Matrix

	1	2	3	4
1	1	1	1	0
2	0	0	1	0
3	1	0	1	1
4	1	0	0	0

Tagged Matrix

	$\overline{1}$	2	$\overline{3}$	4
1	0	1	0	0
2	1	0	0	0
3	0	0	0	1
4	0	0	1	0

- At most $2n$ tags (bits).
  $O(n + m')$ with $m' << m$.
  Dalhaus, Gustedt, McConnell 2000
- What can be computed using such representations ?
  Example : strong connected components of G, knowing $\overline{G}$?

- Find a minimum sized representation
- If G is undirected a minimum is unique.

What is an elementary operation for a graph ?

- Traversing an edge or Visiting the neighbourhood ?
- It explains the very few lower bounds known for graph algorithms on a RAM Machine.
- Our graph algorithms must accept any auto-complemented representation.
  Partition Refinement