

3th Lecture : Partition refinement MPRI 2013–2014

Michel Habib
habib@liafa.univ-Paris-Diderot.fr
<http://www.liafa.univ-Paris-Diderot.fr/~habib>

Sophie Germain, octobre 2013

Schedule

Applications of partition refinement techniques

Modular decomposition

Uniqueness decomposition theorem

Partitive Families

Structural Aspects of Prime graphs

Factoring permutation

Three more cograph recognition algorithms

No course next week.

└ Applications of partition refinement techniques

Refining a partition

Definition

Let $S \subseteq V$, and $P = \{X_1, \dots, X_n\}$ be a partition of V .
 $Q = \text{Refine}(S, P) = \{X_1 \cap S, X_1 - S, \dots, X_n \cap S, X_n - S\}$
 S is called a pivot.

NB Some sets can be empty and then ignored.

▶ $\text{Refine}(S, P) \leq P$

▶ $\text{Refine}(S, P) = P$ iff S is a union of parts of P

Duality

$\text{Refine}(S, P) = \text{Refine}(\bar{S}, P)$

└ Applications of partition refinement techniques

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 of the first lecture

Propose a good algorithm to compute these classes

└ Applications of partition refinement techniques

Algorithm Folklore

Data: $G = (V, E)$ a graph with n vertices and m edges

Result: The classes of false twin vertices

$Q \leftarrow \{V\}$

for Every $x \in V$ **do**

$Q \leftarrow \text{Refine}(Q, N(x))$

end

└ Applications of partition refinement techniques

Proof

At the end, parts of Q have no splitter outside and therefore are modules.

Furthermore they have no splitter inside the part.

They are made up with false twins (non connected).

Complexity

$\sum_{x \in V} |N(x)| \in O(n + m)$

└ Applications of partition refinement techniques

Other applications

Detection of multi-occurency in a list of subsets

Just construct the incidence bipartite elements–subsets and compute the twins.

Recognition of a laminar family

Laminar Family

A family \mathcal{F} of subsets of a ground set X is laminar if :
 $\forall F', F'' \in \mathcal{F}$, either F', F'' are disjoint or included.
Such a family is ordered by inclusion with a forest structure.

Computing the tree structure

Sort the elements of \mathcal{F} by decreasing size.
Compute using partition refinement the sets contained in $F_0 \dots$
whole complexity in $O(\sum |F|_{F \in \mathcal{F}})$.

Degrees parts

Classification of the vertices in parts having the same degree.
A variation of the folklore algorithm for twins.

Generalized degree partition

Classification of the vertices in parts having the same degree with respects to the other parts. To compute this partition we can use a variation of the partition refinement.

DegreeRefine(P, S) :

computes the partition of S in parts having same degree with P
The computation of this partition is the first step of the main isomorphism algorithms.

Tree isomorphism using Partition refinement

Compute the generalized degree partitions of the two graphs G and H

Folklore Property

if G and H are isomorphic then their partitions are identical.

Particular case of trees

For trees the converse is also true.

Graph search

Most of the classical graph searches can be implemented using partition refinement and sometimes this gives a good way to obtain an optimal implementation.

Another exercise

Data: A family \mathcal{F} of subsets of V

Result: Compute the overlap components of \mathcal{F}

Partition refinement a kind of technique dual to Union-Find.

Complementary uses :

- ▶ x et y belong to the same part \rightarrow Union-Find
- ▶ x et y do not belong to the same part \rightarrow Partition refinement.

Generic Refinement Algorithm

Input : P a partition and S a set of pivots

While $S \neq \emptyset$

Choose $S \in \mathcal{S}$

$P = \text{Refine}(S, P)$

add all new generated parts to S

Hopcroft's rule

In many applications when a part C is cut into 2 parts : C', C'' : it is enough to consider as a pivot in the following only C' or C''

Hopcroft's rule

Choose the smallest half

This assures an $O(n \log n)$ algorithm.

proof

The number of time an element can be used in a pivot set, is bounded by $\log n$.

Variant

Avoid the biggest one

This also assures an $O(\log n)$ factor in the complexity of the algorithm

Historical Notes

This technique is very powerful not only for graph algorithms. First used by Corneil for Isomorphism Algorithms 1970
Hopcroft Automata minimisation 1971
Cardon and Crochemore string sorting 1981
J. Spinrad Graph Partitioning (generic tool vertex splitting) 1986
Paigue, Tarjan 1987 (generic tool presented on three problems)
...

Some applications

- ▶ Quicksort : Hoare, 1962
- ▶ Minimal deterministic automaton Hopcroft $O(n \log n)$ 1971.
- ▶ Relational coarsen partition Paige, Tarjan $O(n \log n)$ 1987
- ▶ Coarsest functional partition Paigue, Tarjan $O(n \log n)$ 1987 improved to $O(n)$ by Paigue, Tarjan, Bonic 1985 and Crochemore 1982.
- ▶ String sorting $O(n \log n)$
- ▶ Doubly Lexicographic ordering Paige and Tarjan 1987 $O(L \log L)$, using a 2-dimensional refinement technique. where $L = \#ones$ in the matrix using a 2-dimensional refinement technique.

Many other applications on graphs

Partition refinement has many applications in graph algorithm design, mainly for undirected graphs. Kind of **generic tool** to obtain efficient algorithms easy to understand.

Vertex splitting, (also called vertex partitioning) when the neighborhood $N(x)$ is used as a pivot set. Provides a linear algorithm if the neighbourhood of every vertex is used a constant number of times.

- ▶ Interval graph recognition $O(n + m)$ using partition refinement on maximal cliques, 1-consecutiveness property $O(n + m)$, Habib, McConnell, Paul and Viennot 2000.
- ▶ Modular decomposition,
- ▶ Cograph recognition $O(n + m)$, Habib, Paul 2000.
- ▶ Transitive orientation

In some application the order between parts matters and we play with ordered partitions.

Variations :

1. Parts are equipped with a counter representing its size.
2. Predicate "left-to" between parts in an ordered partition in $O(1)$.
3. Implement a backtrack

Exercise :

Implement the classical graph search : BFS and DFS using partition refinement.

Research aspects

1. Find an efficient way to implement a backtrack operation (Kind of UnRefine)
2. Generalize the applications of partition refinement to directed graphs

- 📄 J.E. Hopcroft, *A $n \log n$ algorithm for minimizing states in a finite automaton*, Theory of Machine and Computations, (1971) 189-196.
- 📄 A. Cardon and M. Crochemore, *Partitioning a Graph in $O(|A| \log |V|)$* , Theor. Comput. Sci., 19 (1982) 85-98.
- 📄 R. Paige and R. E. Tarjan, *Three Partition Refinement Algorithms*, SIAM J. Computing 16 : 973-989, 1987.
- 📄 M. Habib, R. M. McConnell, C. Paul and L. Viennot, *Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing*, Theor. Comput. Sci. 234 :59-84, 2000.

Modules

Modules

For a graph $G = (V, E)$, a **module** is a subset of vertices $A \subseteq V$ such that

$$\forall x, y \in A, N(x) \cap A = N(y) \cap A$$

The problem with this definition : must we check all subsets A ?

Trivial Modules

\emptyset , $\{x\}$ and V are modules.

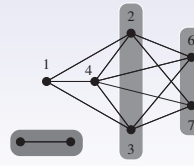
Prime Graphs

A graph is **prime** if it admits only trivial modules.

Examples

Characterization of Modules

A subset of vertices M of a graph $G = (V, E)$ is a **module** iff $\forall x \in V \setminus M$, either $M \subseteq N(x)$ or $M \cap N(x) = \emptyset$



Examples of modules

- ▶ connected components of G
- ▶ connected components of \overline{G}
- ▶ any vertex subset of the complete graph (or the stable)

- ▶ Modular decomposition (algorithmic aspects)
- ▶ But also an operation on graphs : Modular composition a graph grammar with a simple rule : replace a vertex by a graph
- ▶ Very natural notion, (re)discovered under many names in various combinatorial structures such as :clan, homogeneous set, ...
- ▶ An important tool in graph theory

Playing with the definition

Duality

A is a module of G implies A is a module of \overline{G} .

Easy observations

- ▶ No prime undirected graph with ≤ 3 vertices (false for directed graphs, as a directed triangle shows it)
- ▶ P_4 the path with 4 vertices is the only prime on 4 vertices.
- ▶ P_4 is isomorphic to its complement.

Twins and strong modules

Twins

$x, y \in V$ are false- (resp. true-) **twins** if $N(x) = N(y)$ (resp. $N(x) \cup \{x\} = N(y) \cup \{y\}$).

x, y are false twins in G iff x, y are true twins in \overline{G} .

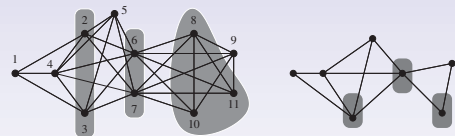
Classes of twins are particular modules (stable sets for false twins and complete for true twins).

Strong modules

A **strong module** is a module that does not strictly overlap any other module.

Modular partition

A partition \mathcal{P} of the vertex set of a graph $G = (V, E)$ is a **modular partition** of G if any part is a module of G .

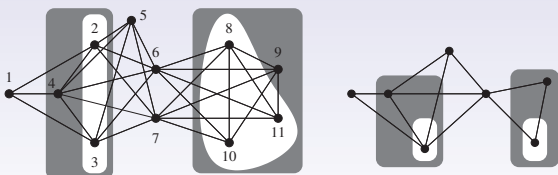


Let \mathcal{P} be a modular partition of a graph $G = (V, E)$. The **quotient graph** $G_{/\mathcal{P}}$ is the induced subgraph obtained by choosing one vertex per part of \mathcal{P} .

Lemma (Mohring Radermacher 1984)

Let \mathcal{P} be a modular partition of $G = (V, E)$.

$\mathcal{X} \subseteq \mathcal{P}$ is a module of $G_{/\mathcal{P}}$ iff $\cup_{M \in \mathcal{X}} M$ is a module of G .



Modular Decomposition Theorem

Theorem (Gallai 1967)

Let $G = (V, E)$ be a graph with $|V| \geq 4$, the three following cases are mutually exclusive :

1. G is not connected,
2. \overline{G} is not connected,
3. $G_{/\mathcal{M}(G)}$ is a prime graph, with $\mathcal{M}(G)$ the modular partition containing the maximal strong modules of G .

As a byproduct, we notice that a prime graph G satisfies :
 G and \bar{G} are connected

Modular decomposition tree

Tree

A recursive of this theorem yields a tree T in which :

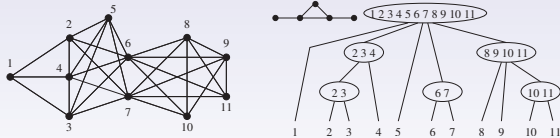
- ▶ The root corresponds to V
- ▶ Leaves are associated to vertices
- ▶ Each node corresponds to a strong module

There are 3 types of nodes :

Parallel, Series and Prime

Another explanation

The set of strong modules is nested into an inclusion tree (called the **modular decomposition tree** $MD(G)$ of G).



Partitive Families

Lemma

If M and M' are two overlapping modules then

- ▶ (i) $M \setminus M'$ is a module
- ▶ (ii) $M \cap M'$ is a module
- ▶ (iii) $M \cup M'$ is a module
- ▶ (iv) $M \Delta M'$ is a module

- ▶ A family satisfying (i) - (iv) is called a **partitive family**
- ▶ A family satisfying (i) - (iii) is called a **weak partitive family**

Chein Habib Maurer 1981

The set of all modules of an undirected graph (resp. a directed graph) constitutes a partitive family (resp. a weak partitive family).

Uniqueness decomposition theorem

Partitive (resp. weakly partitive) families admit a decomposition tree with two (resp. three) types of nodes :

- ▶ degenerate (also called fragile)
- ▶ prime
- ▶ (resp. linear)

This tree representation theorem for partitive (resp. weakly partitive) families $F \subseteq 2^{|X|}$, yields an encoding of these families in $O(|X|)$.

Modular Decomposition Theorem for Directed Graphs

Theorem (Chein, Habib, Maurer 1981)

Let $G = (V, E)$ be a directed graph with $|V| \geq 4$, the four following cases are mutually exclusive :

1. G is not connected, **Parallel node**
2. \bar{G} is not connected, **Series node**
3. G^* is not strongly connected, **Linear node**
4. $G_{/\mathcal{M}(G)}$ is a prime graph, with $\mathcal{M}(G)$ the modular partition containing the maximal strong modules of G , **Prime node**

Prime graphs are nested

Folklore Theorem

Let G be a prime graph ($|G| \geq 4$), then G contains a P_4 .

Theorem Schmerl, Trotter, Ille 1991 ...

Let G be a prime graph ($|G| = n \geq 4$), then G contains a prime graph on $n - 1$ vertices or a prime graph on $n - 2$ vertices.

A simple proof

A stronger statement, Cournier, Ille, 1991

For a prime graph there is at most one vertex not contained in a P_4 .

Proof

As a prime graph G is necessarily connected and if $\exists x \in V$ that does not belong to a P_4 . Every connected component of $\overline{N(x)}$ is a module, therefore $\overline{N(x)}$ must be a stable set.

If $\exists x \neq y \in V$ that does not belong to a P_4 ,

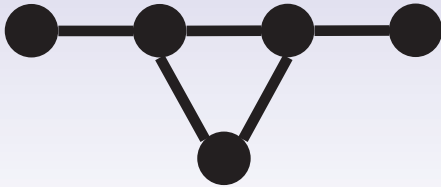
wlog assume $xy \notin E$

But then $y \in \overline{N(x)}$ and therefore : $N(y) \subseteq N(x)$.

By symmetry x, y must be false twins, a **contradiction**.

Nota Bene : this result also holds for infinite graphs!
 So does our previous proof.

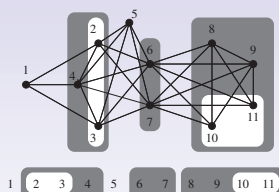
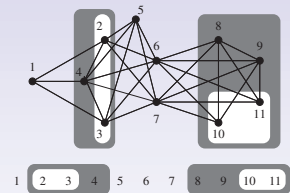
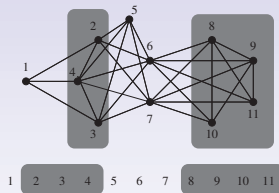
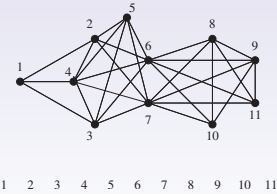
In fact if there is such a vertex $x \in V$,
 x is adjacent to the middle vertices of a P_4 .
 (Such a subgraph is called a **bull**).



A bull is isomorphic to its complement.

Factoring permutation

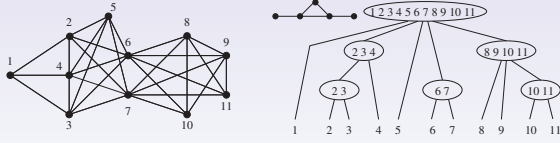
A **factoring permutation** of a graph $G = (V, E)$ is a permutation of V in which any strong module of G is a factor. [CH 97]



- ▶ From G to factoring permutation : $O(n + m \log n)$ [HPV99]
- ▶ From factoring permutation to $MD(G)$: $O(n + m)$ [CdMH01] [UY00] [BXHP05]

Another explanation

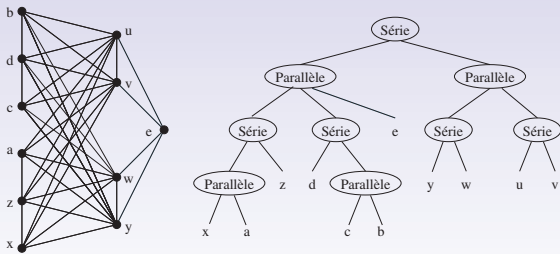
The set of strong modules is nested into an inclusion tree (called the **modular decomposition tree** $MD(G)$ of G).



A factoring permutation is simply a left-right ordering of the leaves of a plane drawing of $MD(G)$.

Consequence : it always exists factoring permutations.
 There are easier to compute than the modular decomposition tree.

Recall of the tree structure of cographs



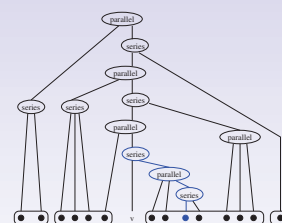
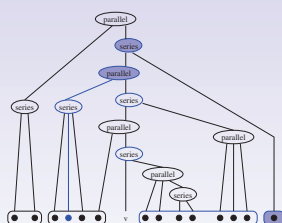
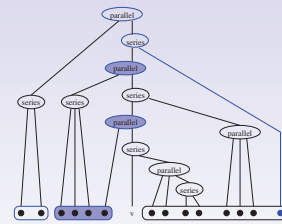
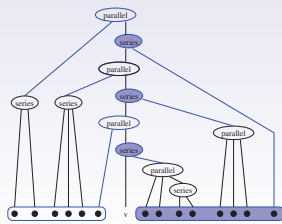
A very simple algorithm to build the tree from the factoring permutation.

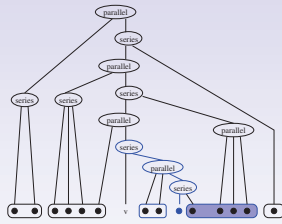
Factoring permutation

Factoring permutation (of cographs) via vertex partitioning

Starting with the partition $\{\overline{N(x)}, \{x\}, N(x)\}$, we maintain the following invariant :

It exists a factoring permutation smaller than the current partition.





Invariant

Any strong module is a factor of the partition.

1. Brute force, just partition refinement. Using recursivity. $O(nm)$
2. Using the rule : take the smallest half $O(m \log n)$
3. Using at most one vertex per part to refine the other parts.
Idea : only two parts (the extreme ones) refine nothing. Then we can restart the procedure to the closest one to the initial pivot.
In the whole a vertex is at most used twice as a pivot.
 $O(n + m)$

Modular decomposition algorithms via partition refinement are very similar than the cograph recognition algorithms just a little more complicated.