Sur quelques généralisations polynomiales de la décomposition modulaire.

Vincent Limouzy

LIAFA – Université Paris Diderot

December 3, 2008
Outline of the Thesis

Part I. Generalizations of Modular Decomposition

• Homogeneous relations and modular decomposition.
• Umodular decomposition: a new point of view.

Part II. Efficient Algorithms

• Overlap Components.
• NLC-2 graphs recognition algorithm.
1 A brief Introduction to Homogeneous Relations
   First encounter
   Modular decomposition
   Results

2 Umodules
   Arbitrary relations
   Local congruence 2
   Self complemented families
   Undirected graphs
   Tournaments

3 Overlap components

4 Perspectives
   Homogeneous relations
   Overlap components
   NLC-width
Basic definitions

Modules and Modular decomposition
Basic definitions

Modules and Modular decomposition

Module
Basic definitions

Modules and Modular decomposition

Substitution / Contraction
Generalizing
Why and How?

Modular decomposition
• Social sciences,
• Bioinformatics,
• Computer science
• ...

Known generalizations
Role coloring: Everett & Borgatti’91
proven NP-complete by Fiala & Paulusma’05 that this problem

Desired properties of the generalizations
• Polynomial computation
• Good structural properties
• Decomposition tree

• Compact encoding of the family
• ...

**Summary**

**Module**
A *module* is a set of vertices which have the same neighborhood outside.
Module
A module is a set of vertices which have the same neighborhood outside.

Role
A “role” in a graph is a set of vertices which plays the same role.
**Module**
A *module* is a set of vertices which have the same neighborhood outside.

**Homogeneous Relations**
*Homogeneous relation* is something in between...

**Role**
A “*role*” in a graph is a set of vertices which plays the same role.
Homogeneous Relations
Definition

Let $X$ be a finite set. A Homogeneous Relation is a collection of triples on $X$, noted $H(a|b, c)$ fulfilling the following properties:

1. **Reflexivity**: $H(a|x, x)$,
2. **Symmetry**: $H(a|x, y) \equiv H(a|y, x)$ and
3. **Transitivity**: $H(a|x, y)$ and $H(a|y, z) \Rightarrow H(a|x, z)$
**Definition**

Let $X$ be a finite set. A **Homogeneous Relation** is a collection of triples on $X$, noted $H(a|b, c)$ fullfiling the following properties:

1. **Reflexivity**: $H(a|x, x)$,
2. **Symmetry**: $H(a|x, y) \equiv H(a|y, x)$ and
3. **Transitivity**: $H(a|x, y)$ and $H(a|y, z) \Rightarrow H(a|x, z)$

$H(a|b, c)$

$a$ is said to be **homogeneous** with respect to $b$ and $c$, or

$a$ does not distinguish $b$ from $c$. 
An example

\[ X = \{a, b, c, d\} \]

Let \( H \) be defined as follows:

\[
\begin{align*}
H(a|c, d), & \quad H(a|b, b), \\
H(b|a, c), & \quad H(b|c, d), \quad H(b|a, d), \\
H(c|a, a), & \quad H(c|b, b), \quad H(c|d, d), \\
H(d|b, c), & \quad H(d|a, a).
\end{align*}
\]

**Homogeneous relation ~ Equivalence relations**

To each element \( x \) of \( X \), thanks to the transitivity property we can associate an equivalence relation \( H_x \) defined on \( X \setminus \{x\} \)
### Equivalence relation

| \( H_a \) | \{b\}, \{c, d\} |
| \( H_b \) | \{a, c, d\} |
| \( H_c \) | \{a\}, \{b\}, \{d\} |
| \( H_d \) | \{a\}, \{b, c\} |

### Matrix representation

\[
\begin{pmatrix}
0 & 1 & 2 & 2 \\
1 & 0 & 1 & 1 \\
1 & 2 & 0 & 3 \\
1 & 2 & 2 & 0
\end{pmatrix}
\]
Graphic Homogeneous Relations

Graphic
A homogeneous relation $H$ is graphic if there exists a graph $G$ s.t.

$$\forall v \text{ of } V(G), \ H_v = N(v), N(v)$$

Theorem
A homogeneous relation $H$ is graphic iff $\forall x, y, z \in X$, $H$ does not contain:

1. $H(x|y, z) \land H(y|x, z) \land H(z|x, y)$
2. $H(x|y, z) \land \overline{H(y|x, z)} \land \overline{H(z|x, y)}$
Graphic Homogeneous Relations

**Graphic**
A homogeneous relations $H$ is **graphic** if there exists a graph $G$ s.t.

$$\forall v \text{ of } V(G), \ H_v = N(v), \overline{N(v)}$$

**Theorem**
A homogeneous relation $H$ is graphic iff $\forall x, y, z \in X, \ H$ does not contain:

1. $H(x | y, z) \land H(y | x, z) \land \overline{H(z | x, y)}$
2. $\overline{H(x | y, z)} \land \overline{H(y | x, z)} \land \overline{H(z | x, y)}$

$$
\begin{align*}
H_a &= \{b, c\}, \{d\} \\
H_b &= \{a, c\}, \{d\} \\
H_c &= \{a, b, d\} \\
H_d &= \{a, b\}, \{c\}
\end{align*}
$$
Homogeneous Relations Properties

*Local Congruence*

Maximum number of classes associated to an element.

*Example*

\[
\begin{align*}
H_a & = \{b\}, \{c,d\} \\
H_b & = \{a,c,d\} \\
H_c & = \{a\}, \{b\}, \{c\} \\
H_d & = \{a\}, \{b,c\}
\end{align*}
\]
**Definition**

A **Module** in a Homogeneous relation $H$ is a set $M$ such that:

$$\forall m, m' \in M \text{ and } \forall x \in X \setminus M \text{ we have:}$$

$$H(x|m m')$$

**Family of modules**

$M_H$: family of modules.

**Example**

$H_a = \{b\}, \{c, d\}$; $H_b = \{a, c, d\}$; $H_c = \{a\}, \{b\}, \{d\}$; $H_d = \{a\}, \{b, c\}$.

The modules are $\{a\}, \{b\}, \{c\}, \{d\}, \{a, b, c, d\}$ and $\{c, d\}$. 
Basic Properties

Definition (Overlap)
Let \( A \) and \( B \) be subsets of \( X \). \( A \) overlaps \( B \) if:

\[
A \varsubsetneq B \equiv A \setminus B \neq \emptyset \text{ and } B \setminus A \neq \emptyset \text{ and } A \cap B \neq \emptyset
\]

Proposition (Intersecting family)
Let \( H \) be a homogeneous relation on \( X \), and let \( M \) and \( M' \) modules of \( H \) s.t. \( M \varsubsetneq M' \) then:

\[
M \cap M' \in \mathcal{M}_H \text{ and } M \cup M' \in \mathcal{M}_H
\]

Theorem (Gabow’95)
\( \mathcal{M}_H \) can be stored in space \( O(n^2) \)
Results on Homogeneous Relations

**Modular Decomposition**

On Arbitrary Homogeneous relations:

- Primality: \(O(n^2)\)
- Decomposition algorithm: \(O(n^3)\)

On *good* Homogeneous relations

- Primality: \(O(n^2)\)
- Decomposition algorithm: \(O(n^2)\)

Where \(n\) is the cardinality of the ground set \(X\).

**Good Homogeneous Relations**

The modules family on *good* homogeneous relations forms a weakly partitive family.
Umodules
**Umodules**

**Definition**
Let $H$ be a homogeneous relation defined on $X$, a **Umodule** $U$ is a set such that:

\[ \forall u, u' \in U \text{ and } \forall x, x' \in X \setminus U : \]

\[ H(u|xx') \iff H(u'|xx') \]
**Definition**

Let $H$ be a homogeneous relation defined on $X$, a **Umodule** $U$ is a set such that:

$$\forall u, u' \in U \text{ and } \forall x, x' \in X \setminus U :$$

$$H(u|xx') \iff H(u'|xx')$$
$\mathcal{U}_H$ is the family of umodules.

**Proposition (Union closed)**

Let $\mathcal{U}$ and $\mathcal{U}'$ be two umodules of $H$ such that $\mathcal{U} \circ \mathcal{U}'$ then:

$$\mathcal{U} \cup \mathcal{U}' \in \mathcal{U}_H$$
Crossing families

**Definition (Cross)**
Let $A$ and $B$ be two subsets of $X$. $A$ crosses $B$ if:

\[
A \circ B \equiv A \cap B \text{ and } A \cup B \neq X
\]

**Definition (Crossing family)**
Let $X$ be a finite set and $\mathcal{F}$ be a family of subset. $\mathcal{F}$ is said to be crossing if:

\[
\forall A, B \in \mathcal{F} \text{ such that } A \circ B \text{ and } A \cup B \text{ and } A \cap B \text{ belong to } \mathcal{F}.
\]
Proposition

Let $H$ be a homogeneous relation of Local Congruence 2 (LC2) and $\mathcal{U}_H$ is a crossing family.
**Proposition**

Let $H$ be a homogeneous relation of Local Congruence 2 (LC2) and: $\mathcal{U}_H$ is a crossing family.

**Sketch of Proof**

$\cup$: from the previous proposition.

$\cap$: Let $A$ and $B$ be two umodules. By hypothesis we have:

$$H(a|x, b) \iff H(y|x, b) \iff H(z|x, b)$$
$$H(b|x, a) \iff H(y|x, a) \iff H(z|x, a)$$

we obtain:

$$H(y|a, b) \iff H(z|a, b)$$
**Proposition**

Let $H$ be a homogeneous relation of Local Congruence 2 (LC2) and: $\mathcal{U}_H$ is a crossing family.

**Sketch of Proof**

∪: from the previous proposition.

∩: Let $A$ and $B$ be two umodules. By hypothesis we have:

$H(a|x, b) \iff H(y|x, b) \iff H(z|x, b)$

$H(b|x, a) \iff H(y|x, a) \iff H(z|x, a)$

we obtain:

$H(y|a, b) \iff H(z|a, b)$

**Theorem (Gabow’95 & Bernath’04)**

Crossing families defined on a ground set $X$ can be stored in $O(n^2)$ space.
**Bipartitive families**

Let $X$ be a finite set, and let $\mathcal{B} = \{\{B^1_1, B^2_1\}, \ldots, \{B^1_l, B^2_l\}\}$ be a set of bipartitions of $X$.

*Definition (Bipartitive families – Cunningham & Edmonds’80)*

$\mathcal{B}$ is a **bipartitive family** if for all overlapping bipartitions $\{B^1_k, B^2_k\}$ and $\{B^1_j, B^2_j\}$ we have:

$$\begin{align*}
\{B^1_k \cup B^1_j, B^2_k \cap B^2_j\}, & \quad \{B^1_k \cup B^2_j, B^2_k \cap B^1_j\} \\
\{B^2_k \cup B^1_j, B^1_k \cap B^2_j\}, & \quad \{B^2_k \cup B^2_j, B^1_k \cap B^1_j\}
\end{align*} \in \mathcal{B}$$

\[ B_j^1 \quad X \quad B_j^2 \]

\[ B_k^1 \quad X \quad B_k^2 \]

\[ B_j^1 \cup B_k^1 \quad B_j^1 \cap B_k^2 \quad B_j^2 \cup B_k^1 \quad B_j^2 \cap B_k^2 \]

\[ B_j^1 \cap B_k^1 \quad B_j^1 \cup B_k^1 \quad B_j^2 \cap B_k^2 \quad B_j^2 \cup B_k^2 \]
Bipartitive families

**Theorem (Cunningham & Edmonds’80)**

Let $\mathcal{B}$ be a bipartitive family defined on $X$. There exists a unique unrooted tree encoding $\mathcal{B}$. Its size is $O(n)$. 
**Definition**

Let $H$ be a Homogeneous Relation defined on $X$. $H$ is said to be self-complemented iff:

$$\forall U \in \mathcal{U}_H, \ X \setminus U \text{ belongs to } \mathcal{U}_H$$

**Theorem**

Let $\mathcal{U}_H$ be self-complemented then $\mathcal{U}_H$ form a bipartitive family.
4 Points condition

Let \( H \) be a homogeneous relation on \( X \). For all \( x, x', m, m' \) of \( X \) we have:

- \( H(m|x|x') \land H(m'|x|x') \land H(x|m|m') \Rightarrow H(x'|m|m') \)
- \( \overline{H(m|x|x')} \land \overline{H(m'|x|x')} \land \overline{H(x|m|m')} \Rightarrow \overline{H(x'|m|m')} \)

Proposition

Let \( H \) be a Homogeneous relation fulfilling the 4 points condition then \( \mathcal{U}_H \) is self-complemented.
**Definition (Seidel switch)**

Let $G = (V, E)$ be a undirected loopless graph, and $S \subseteq V$, A Seidel switch on $G$ is the graph obtained by removing all the edges between $S$ and $\bar{S}$, and adding all the missing edges.
Definition (Seidel switch)
Let $G = (V, E)$ be a undirected loopless graph, and $S \subseteq V$, a Seidel switch on $G$ is the graph obtained by removing all the edges between $S$ and $\bar{S}$, and adding all the missing edges.
**Definition (Seidel switch)**
Let $G = (V, E)$ be a undirected loopless graph, and $S \subseteq V$, A Seidel switch on $G$ is the graph obtained by removing all the edges between $S$ and $\bar{S}$, and adding all the missing edges.

**Definition (Pointed Seidel switch)**
The pointed Seidel switch: $S = N(v)$
**Definition (Seidel switch)**

Let $G = (V, E)$ be a undirected loopless graph, and $S \subseteq V$, A Seidel switch on $G$ is the graph obtained by removing all the edges between $S$ and $\bar{S}$, and adding all the missing edges.

**Definition (Pointed Seidel switch)**

The **pointed Seidel switch**: $S = N(v)$
**Definition (Seidel switch on Homogeneous Relations)**

Let $H$ be a Homogeneous relation of local congruence 2 defined on $X$, the Seidel switch at an element $s$ is defined in the following way:

$$\forall x \in X \setminus \{s\}, \ H(s) = \begin{cases} 
H(s)_x^1 = (H^1_x \Delta H^i_s) \setminus \{s\} \\
H(s)_x^2 = (H^2_x \Delta H^i_s) \setminus \{s\}
\end{cases}$$
Definition (Seidel switch on Homogeneous Relations)

Let $H$ be a Homogeneous relation of local congruence 2 defined on $X$, the **Seidel switch** at an element $s$ is defined in the following way:

$$
\forall x \in X \setminus \{s\}, \quad H(s) = \begin{cases} 
H(s)_1 = (H_x^1 \Delta H_s^j) \setminus \{s\} \\
H(s)_2 = (H_x^2 \Delta H_s^j) \setminus \{s\}
\end{cases}
$$

Theorem

Let $H$ be a LC2 Homogeneous relation s.t. $U_H$ is self-complemented. Let $s$ an element of $X$, and let $U \subseteq X$ s.t. $s \in U$. Then

$U$ is a **umodule** of $H$ if and only if

$M = \bar{U}$ is a module of $H(s)$ (Homogeneous relation on $X - s$).
Algorithmic consequences

**Theorem**

Given a Self-complemented LC2 Homogeneous relation $H$ on $X$, its decomposition tree can be obtained in *linear time*. 

**Sketch of Proof**

• Pick an element $s$ of $X$.
• Seidel switch at $x$.
• Compute modular decomposition of $H(x)$.
• Add $x$ carefully.

□
**Theorem**
Given a Self-complemented LC2 Homogeneous relation $H$ on $X$, its decomposition tree can be obtained in **linear time**.

**Sketch of Proof**
- Pick an element $s$ of $X$.
- Seidel switch at $x$.
- Compute modular decomposition of $H(x)$.
- Add $x$ carefully.
**Definition (Bi-Joins de Montgolfier & Rao’05)**

Let $G = (V, E)$ a graph, a bi-join in $G$ is a bipartition $V_1, V_2$ of $V$, s.t. $V_1 = \{V_{1,1}, V_{1,2}\}$ and $V_2 = \{V_{2,1}, V_{2,2}\}$ and $V_{1,i}$ is completely connected to $V_{2,i}$ and $V_{1,i}$ is completely disconnected from $V_{2,j}$.

**Self complement**

The bi-joins of a graph are self-complemented.

**Bipartitivity**

Bi-joins of a graph form a bipartitive family.

There is a unique decomposition tree.
**Completely decomposable graphs**

*Theorem (de Montgolfier & Rao’05)*

The graphs completely decomposable w.r.t. Bi-join decomposition are the graphs without $C_5$, Bull, Gemma and co-Gemma as induced subgraphs.

**Forbidden Subgraphs**

- $C_5$
- Bull
- Gemma
- co-Gemma
**Decomposition Algorithm**

1. Choose a vertex \( v \), proceed to a Seidel switch \( G \ast v \) \( O(n + m) \)
2. Compute modular decomposition of \((G \ast v) \setminus v\) \( O(n + m) \)
3. Turn the modular decomposition tree of \((G \ast v) \setminus v\) into the bi-join decomposition tree of \( G \) \( O(n + m) \)
Decomposition Algorithm

1. Choose a vertex $v$, proceed to a Seidel switch $G*v$
2. Compute modular decomposition of $(G*v) \backslash v$
3. Turn the modular decomposition tree of $(G*v) \backslash v$ into the bi-join decomposition tree of $G$

Complexity

$O(n + m)$

Completely Decomposable graph Recognition

1. Choose a vertex $v$, proceed to a Seidel switch $G*v$
2. Check if $(G*v) \backslash v$ is a cograph

Complexity

$O(n + m)$
**Tournaments**

*Umodes in tournaments*

![Diagram of umodes in tournaments]

**Locally transitive tournaments**

A tournament $T = (V, A)$ is locally transitive if for each vertex $v$, $T[N^+(v)]$ and $T[N^-(v)]$ are transitive tournaments.

**Completely decomposable tournaments**

Completely decomposable tournaments are exactly locally transitive tournaments.
**Forbidden characterization**

A tournament $T = (V, A)$ is completely decomposable w.r.t. umodular decomposition

$\iff$

We then check that only these graphs can produce a $\vec{C}_3$, after a Seidel switch

**Sketch of Proof**

A tournament is completely decomposable w.r.t. modular decomposition iff it is a transitive tournament. i.e. does not contain a $\vec{C}_3$
Simple Recognition Algorithm

Naive approach

- To check in $O(n^4)$ time if $T$ contains $\circlearrowright$ or $\circlearrowleft$ as induced sub-tournaments.
- To check for each vertex $v$ if $T[N^+(v)]$ and $T[N^-(v)]$ are transitive tournaments. We obtain a $O(n^3)$ time algorithm.
**Simple Recognition Algorithm**

**Naive approach**

- To check in $O(n^4)$ time if $T$ contains \( \bullet \longrightarrow \bullet \text{ or } \bullet \longrightarrow \bullet \) as induced sub-tournaments.
- To check for each vertex $v$ if $T[N^+(v)]$ and $T[N^-(v)]$ are transitive tournaments. We obtain a $O(n^3)$ time algorithm.

**Linear time algorithm**

1. Pick a vertex $v$ and check $T[N^+(v)]$ (A) and $T[N^-(v)]$ (B) are transitive tournaments
2. Check that the edges between A and B do not contain a forbidden configuration.
A Simple Recognition Algorithm

Proposition (Locally Transitive Tournament)
Let \( T = (V, A) \) a tournament, \( T \) is locally transitive iff:

(i) \( T[N^+(v)] \) and \( T[N^-(v)] \) are transitive tournaments,

(ii) If a vertex \( a \in T[N^+(v)] \) has an outgoing neighbor \( b \in T[N^-(v)] \) and an ingoing neighbor \( c \in T[N^-(v)] \) then \( (b, c) \in A \).

(iii) If a vertex \( a \in T[N^-(v)] \) has an outgoing neighbor \( b \in T[N^+(v)] \) and an ingoing neighbor \( c \in T[N^+(v)] \) then \( (b, c) \in A \).

The second step of the algorithm is equivalent to check the previous proposition.
Proposition (Locally Transitive Tournament)

Let \( T = (V, A) \) a tournament, \( T \) is locally transitive iff:

(i) \( T[N^+(v)] \) and \( T[N^-(v)] \) are transitive tournaments,

(ii) If a vertex \( a \in T[N^+(v)] \) has an outgoing neighbor \( b \in T[N^-(v)] \) and an ingoing neighbor \( c \in T[N^-(v)] \) then \((b,c) \in A\).

(iii) If a vertex \( a \in T[N^-(v)] \) has an outgoing neighbor \( b \in T[N^+(v)] \) and an ingoing neighbor \( c \in T[N^+(v)] \) then \((b,c) \in A\).

The second step of the algorithm is equivalent to check the previous proposition.

Complexity

1. The first step is done in linear time.
2. The second step is done \( O(1) \) per edge between \( A \) and \( B \). Every edge is considered only once. Thus overall complexity is \( O(n^2) \).
**Isomorphism**

Thanks to the unicity of the structure obtained, we are able to decide in **linear time** if two completely decomposable tournaments are isomorph.

**Feedback Vertex Set**

The Feedback Vertex Set problem is polynomial on completely decomposable tournaments.
Algorithmic results

Primality testing : $O(n^3)$

Umodular decomposition : $O(n^5)$
Overlap Components
**Overlap components**

**The problem**
Let $X$ be a finite set, and let $\mathcal{F} = \{X_1, \ldots, X_t\}$ be a family of subsets of $X$.

**Input:** $\mathcal{F}$

**Output:** Overlap Components of $\mathcal{F}$.

Size of the data is $|X| + \sum_{i=1}^{t} |X_i|$, $n = |X|$ and $f = \sum_{i=1}^{t} |X_i|$.

**Overlap graph**
Let $OG = (\mathcal{F}, E)$ be the overlap graph of $\mathcal{F}$. $uv \in E$ iff $u \sqcap v$.

**Overlap component**
The overlap components of $\mathcal{F}$ are the connected components of $OG$. 
Examples

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

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

A pathologic example

\[ \mathcal{F} = \{ 1, 2, 3, n \} \]

\[ OG = \{ n, 1, 2, 3, 4 \} \]

\[ = K_n \]
A first idea

Naive approach
First compute OG and then output the connected components. But OG is not necessarily linear in the size of \( \mathcal{F} \).

Dahlhaus’s algorithm
Linear time and space algorithm to find overlap components of \( \mathcal{F} \) in \( O(n + f) \).

Our result
A drastic simplification of Dahlhaus’s algorithm.

Output a spanning subgraph of OG in time \( O(n + f) \).
A brief Introduction to Homogeneous Relations
First encounter
Modular decomposition
Results

Umodules
Arbitrary relations
Local congruence 2
Self complemented families
Undirected graphs
Tournaments

Overlap components

Perspectives
Homogeneous relations
Overlap components
NLC-width
Homogeneous Relations

Homogeneous relations

- Characterize “digraphic” and “oriented” homogeneous relations.
- Improve modular decomposition algorithm:
  1. Conjecture: a $O(n + m)$ algorithm
  2. a $O(n^2)$ algorithm for arbitrary Homogeneous relations.
Homogeneous Relations

Homogeneous relations

- Characterize “digraphic” and “oriented” homogeneous relations.
- Improve modular decomposition algorithm:
  1. Conjecture: a $O(n + m)$ algorithm
  2. a $O(n^2)$ algorithm for arbitrary Homogeneous relations.

Umodular decomposition

- Improve the $O(n^5)$ decomposition algorithm.
- Corresponding decomposition for directed and oriented graphs.
- Necessary and sufficient condition to characterize self-complemented families.
- Investigate Seidel minor properties.
Overlap Component and related problems

Overlap component

- Overlap-k component.
- Recognition specific properties of the overlap graph in linear time:
  1. Bipartite,
  2. Chain, tree
  3. ...

Partition refinement

To “implement” Least Common Ancestor (LCA) with partition refinement techniques.

Dynamic partition refinement.
Overlap Component and related problems

Overlap component

• Overlap-k component.
• recognition specific properties of the overlap graph in linear time:
  1 Bipartite,
  2 Chain, tree
  3 ...

Partition refinement

• To “implement” Least Common Ancestor (LCA) with partition refinement techniques.
• Dynamic partition refinement.
NLC-2 Graphs

NLC-width

- Improve recognition algorithm to $O(n.m)$.
- What about NLC-3 graphs?
- Is NLC-k a FPT problem?
NLC-2 Graphs

NLC-width

- Improve recognition algorithm to $O(n.m)$.
- What about NLC-3 graphs?
- Is NLC-k a FPT problem?

Clique-width

- Clique-width $\geq 4$?
- Is clique-width FPT?
Sagolun

Takk

תודה

Merci

Thank you