



HAL
open science

A new tractable combinatorial decomposition

Binh-Minh Bui-Xuan, Michel Habib, Vincent Limouzy, Fabien de Montgolfier

► **To cite this version:**

Binh-Minh Bui-Xuan, Michel Habib, Vincent Limouzy, Fabien de Montgolfier. A new tractable combinatorial decomposition. RR-07016, 2007. lirmm-00157502v1

HAL Id: lirmm-00157502

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00157502v1>

Submitted on 26 Jun 2007 (v1), last revised 17 Dec 2007 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A new tractable combinatorial decomposition

Binh-Minh Bui-Xuan¹ Michel Habib² Vincent Limouzy²
Fabien de Montgolfier²

June 26, 2007

Abstract

This paper introduces the *umodules*, a generalisation of the notion of module in graph theory. The structure to be decomposed, so-called homogeneous relation, captures among other undirected graphs, tournaments, digraphs, and 2-structures. Our resulting decomposition scheme when restricted to undirected graphs generalises the well-studied modular graph decomposition, and meets the recently introduced bi-join decomposition. All other cases up to our knowledge lead to new notions.

First some properties of the umodule family are presented. Polynomial-time algorithms for non-trivial umodule existence test and for maximal umodule computation are then provided. When the input structure fulfills some natural axioms, the umodule family is shown to own a unique decomposition tree. We provide various algorithms to compute this tree in polynomial time: their exact performance depends on some size assumption.

Among other our theory applies to two particular cases: undirected graphs and tournaments. First, the latter tree-decomposition time in these two cases is linear in the size of the input structure. Besides, our work here can also be seen as a unification of the bi-join undirected graph decomposition and of a new tournament decomposition. From this viewpoint, we address the total decomposability of those structures, and obtain strong structural relationship between the so-called cographs and round tournaments. We then show how our theory provides a very natural manner to obtain several results on the so-called round tournaments, including characterisation by forbidding induced subgraphs, recognition, isomorphism testing, and feedback vertex set computation.

1 Introduction

In graph theory modular decomposition is now a well-studied notion [25, 11, 34, 18, 17], as well as some of its generalisations [16, 31, 36, 37]. As having been rediscovered in other fields, the notion also appears under various names, including intervals, externally related sets, autonomous sets, partitive sets, and clans. Direct applications of modular decomposition include tractable constraint satisfaction problems [13], computational biology [24], graph clustering for network analysis, and graph drawing.

Besides, in the area of social networks, several vertex partitioning have been introduced in order to catch the idea of putting in the same part all vertices acknowledging similar behaviour, in other words finding regularities [45]. Modular decomposition provides such a partitioning, yet seemingly too restrictive for real life applications. The concept of a role [19, 20] on the other hand seems promising, however its computation unfortunately is *NP*-hard [21, 22]. As a natural consequence, there is need for the search of *relaxed*, but *tractable*, variations of the modular decomposition scheme. A step following this direction has generalised graph modules to those of larger combinatorial structures, so-called homogeneous relations [8, 9, 10]. This paper follows the same research stream, and weakens the definition of module in order to further decompose. Fortunately we obtain a new tractable variation of modular decomposition, that we now introduce.

Modular decomposition is based on modules, more precisely vertex subsets without splitters: a vertex exterior to a given vertex subset is called a splitter of the subset if it distinguishes some elements of the subset from others. For instance an undirected graph splitter is linked with some vertices, while not linked to some

¹LIRMM, Univ. Montpellier buixuan@lirmmfr

²LIAFA, Univ. Paris Diderot. {habib,limouzy,fm}@liafa.jussieu.fr.

Research supported by the French ANR project “Graph Decompositions and Algorithms (GRAAL)”

other ones. Therefore, the “outside” of a module constitutes for all vertices of the module the same ordered partition. For instance, all vertices of an undirected graph module have the same neighbourhood. We here address unordered-modules, so-called *umodules* for short: the outside of a umodule constitutes for all vertices of the umodule the same unordered partition. For undirected graphs, if N is the “outsider” neighbourhood of a vertex of a umodule, then N either is the outsider neighbourhood, or the outsider non-neighbourhood, of all vertices of the umodules. Then same holds if N is the outsider non-neighbourhood of the first vertex (see Fig. 1). A umodule thus can be seen as a vertex subset without *interior splitters*: there are no two vertices inside the umodule that do not have the same unordered exterior partition.

This paper first displays umodule tractability by giving an $O(|X|^4 \log |X|)$ time computation of the maximal umodules of an arbitrary homogeneous relation over a finite set X . If the input structure satisfies some local configuration axiom, so-called *the four elements condition*, a decomposition scheme is provided, which generalises the “decomposition frame with the intersection and transitivity properties” of Cunningham [16]. Notice that Cunningham’s theory has also been reconsidered under the formalism of “bipartitive families” [35] and “unrooted set families” [31]. This has important structural consequences, both theoretical and algorithmic. For instance in this case a compact tree representation of the umodule family with $|X|$ leaves is available. This tree can be computed in $O(|X|^4 \log |X|)$ time using the previous algorithm. Moreover, when the input structure also satisfies a second axiom, so-called *the local congruence 2 condition*, this computation can be quickened to be in $O(|X|^2)$ time.

In order to show the importance of our decomposition scheme, the two particular cases of undirected graphs and of tournaments, which satisfy at the same time the *four elements* and *local congruence 2* conditions, will be detailed. For undirected graphs our decomposition scheme meets the recently introduced bi-join decomposition [36, 37]. The decomposition scheme for tournaments up to our knowledge is novel. In both cases, a nice transformation can be made in order to reduce the umodular decomposition problem to a general modular decomposition problem (as studied in [8, 9, 10]). This transformation is inspired from the so-called *Seidel-switch*, which is an important non-trivial transformation on graphs [42]. Following the transformation, we obtain linear time umodular graph and tournament decomposition algorithms.

We deepen the study and address the total umodular decomposability of undirected graphs and of tournaments, namely when any “large enough” sub-structure is umodular decomposable. Surprisingly enough, this results to strong structural relationship between the important graph class of cographs (see e.g. [6]) and the tournament class of locally transitive tournaments, which also is known as round tournaments, a sub-class of locally semicomplete digraphs (refer to [5] for more details). We then show how our theory provides a very natural manner to obtain several results on round tournaments, including characterisation by forbidding induced subgraphs, recognition, isomorphism testing, and feedback vertex set computation.

2 Notations

A (loopless finite simple undirected) graph $G = (V, E)$ is such that V is a finite set, and $E \subseteq \{A \subseteq V \wedge |A| = 2\}$. The (open) neighbourhood of a vertex x of G is denoted by $N(x) = \{v \mid v \in V \wedge \{x, v\} \in E\}$. The closed neighbourhood of x is $N[x] = N(x) \cup \{x\}$. The neighbourhood of x , restricted on the vertex subset $X \subseteq V$, is $N_X(x) = N(x) \cap X$.

A (loopless finite simple) directed graph $G = (V, A)$ is such that V is a finite set, and $A \subseteq V^2 \setminus \{(v, v) \mid v \in V\}$. The in-neighbourhood of a vertex x of G is denoted by $N^-(x) = \{v \mid v \in V \wedge (v, x) \in A\}$. The out-neighbourhood of a vertex x of G is denoted by $N^+(x) = \{v \mid v \in V \wedge (x, v) \in A\}$.

For both undirected and directed graphs, we denoted by $G_{[X]}$ the induced subgraph of G on the vertex subset $X \subseteq V$.

An oriented graph is an undirected graph along with an orientation on the edges. A tournament is an oriented graph obtained from a clique, namely a complete undirected graph. Oriented graphs and tournaments also are considered as directed graphs.

3 Umodule, an enlarged notion of module

Let X be a finite set. The family of all subsets of X is denoted by $\mathcal{P}(X)$. A *reflectless* triple is $(x, y, z) \subseteq X^3$ with $x \neq y$ and $x \neq z$. Reflectless triples will be denoted by $(x|yz)$ instead of (x, y, z) since the first element plays a particular role. Let H be a boolean relation over the reflectless triples of X . Given $x \in X$, we define H_x as the binary relation on $X \setminus \{x\}$ such that $H_x(y, z) \Leftrightarrow H(x|yz)$.

3.1 Homogeneous Relation & Module [8, 9, 10]

Homogeneous relations and modules are defined as follows.

Definition 1 (Homogeneous Relation and Module) [8, 9, 10] *H is a homogeneous relation on X if, for all $x \in X$, H_x is an equivalence relation on $X \setminus \{x\}$ (i.e. it fulfills the symmetry, reflexivity and transitivity properties). A subset $M \subseteq X$ is a module of H if*

$$\forall m, m' \in M, \quad \forall x \in X \setminus M, \quad H(x|mm').$$

Equivalently, a homogeneous relation H can be seen as a mapping from each $x \in X$ to a partition of $X \setminus \{x\}$, namely the equivalence classes of H_x . If $\neg H(x|mm')$ we say that x *distinguishes* m from m' , or x is a *splitter* of $\{m, m'\}$. A module M is *trivial* if $|M| \leq 1$ or $M = X$. The family of modules of H is denoted by \mathcal{M}_H , and \mathcal{M} when no confusion occurs. H is *modular prime*, or *M -prime*, if \mathcal{M}_H is reduced to the trivial modules.

Homogeneous relations generalise graphs and *2-structures*, where modular decomposition still applies under the different but equivalent name of *clan decomposition* [17, 18]. Roughly, a 2-structure is a complete digraph $G = (X, X^2)$ along with an edge colouration $C : X^2 \rightarrow \mathbb{N}$ (see e.g. [17, 18]). Thus, a digraph is a 2-structure using two colours, denoting the existing and absent arcs. Notice that there is no need of the concept of *adjacency* in a homogeneous relation. Besides, a straightforward relation can be derived from graphs and 2-structures as follows.

Definition 2 (Standard Homogeneous Relation) [8, 9, 10] *The standard homogeneous relation $H(G)$ of a directed graph $G = (X, A)$ is defined such that, for all $x, u, v \in X$, $H(G)(x|uv)$ is true if and only if the two following conditions hold:*

1. *either both u and v or none of them are in-neighbours of x , and*
2. *either both u and v or none of them are out-neighbours of x .*

This also holds for undirected graphs, tournaments, oriented graphs, and can be extended to a 2-structure (G, C) : $H(G)(x|uv)$ is true if and only if $C(x, u) = C(x, v)$ and $C(u, x) = C(v, x)$.

Proposition 1 *Let G be a graph, resp. tournament, oriented graph, directed graph, 2-structure. Modules of its standard homogeneous relation $H(G)$ are modules of G , resp. clan if G is a 2-structure, in the usual sense [25, 34, 18, 17].*

3.2 Umodules

We now introduce the central notion of this paper which, thanks to Proposition 2 (below), can be seen as a proper generalisation of the classical modules/clans. Besides, from the subsequent Proposition 4, umodules can also be seen as a dual notion to the generalised modules (in the sense of [8, 9, 10]).

Definition 3 (Umodules) *A subset U of X is a umodule of H if*

$$\forall u, u' \in U \quad \forall x, x' \in X \setminus U \quad H(u|xx') \iff H(u'|xx')$$

Roughly, elements of a umodule come from the same “school of thinking”: if one element of a umodule differentiates, resp. mixes together, some exterior elements, so does every element of the umodule. A umodule U is *trivial* if $|U| \leq 1$ or if $|U| \geq |X| - 1$. The family of umodules of H is denoted by \mathcal{U}_H , and \mathcal{U} when no confusion occurs. H is *umodular prime*, or *U -prime*, if \mathcal{U}_H is reduced to the trivial umodules. It follows from definition that

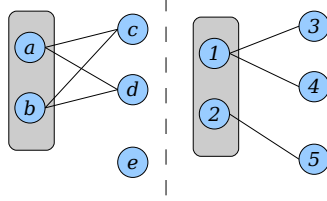


Figure 1: Modules and umodules in a graph: $\{a, b\}$ is a module and also a umodule, $\{1, 2\}$ is a umodule but is not a module.

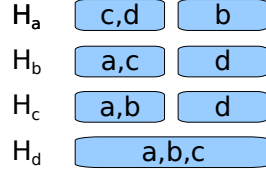


Figure 2: A homogeneous relation with a module which is not a umodule. $\{a, b\}$ is a module: they belong to the same equivalence class in both H_c and H_d . $\{a, b\}$ is not a umodule: c and d belong to the same class in H_a , and to different classes in H_b .

Proposition 2 *If H is the standard homogeneous relation of an undirected graph (resp. a digraph, or a 2-structure), then any module of H also is umodule of H .*

The two following basic propositions link umodules to the 1-intersecting families framework as defined in [28].

Proposition 3 *Let \mathcal{U} be the umodule family of a homogeneous relation. For any two umodules $U, U' \in \mathcal{U}$, if $U \cap U' \neq \emptyset$ then $U \cup U' \in \mathcal{U}$.*

Proposition 4 *If M is a module of a homogeneous relation H over a finite set X , then $X \setminus M$ is a umodule of H .*

In case of undirected graphs, a natural question arises [15]: for which graphs the notions of module and umodule coincide? The following result solves this problem. For convenience, as in the case of module, the umodules of a graph G refer to the umodules of the standard homogeneous relation of G .

Notice in this case that the complementary of a umodule also is a umodule. A *threshold graph* is a graph that can be constructed from the single vertex graph by repeated applications of the following two operations: addition of a single isolated vertex to the graph; and addition of a single dominating vertex to the graph. Equivalently, a threshold graph is a graph with no induced P_4 , C_4 , nor $\text{co-}C_4$, where P_4 denotes the four vertex path, C_4 the four vertex cycle, and $\text{co-}C_4$ the dual graph of C_4 .

Proposition 5 (Characterisation of threshold graphs) *G is a threshold graph if and only if in all induced subgraph of G , every umodule is either a module or the complementary of a module (or both).*

Proof: If G is not threshold, there are four vertices a, b, c , and d which induce either a P_4 , a C_4 , or a $\text{co-}C_4$. For all three cases, in the induced subgraph $G[\{a, b, c, d\}]$, the vertex subset $\{b, c\}$ is a umodule which is neither a module nor the complementary of a module.

Conversely, let $U \subseteq V'$ be a umodule of some induced subgraph $G' = (V', E')$ of G such that U is neither a module nor the complementary of a module of G' . Let $W = V' \setminus U$. That U is not the complementary of a module implies the existence of $a \in U$ and $b, c \in W$ such that a is a splitter of $\{b, c\}$, i.e. $\neg H(a|bc)$ with H being the standard homogeneous relation of G . W.l.o.g. we suppose that ab is an edge, and ac is a non-edge. Let A be the set containing all neighbours of b that belong to U , and $D = U \setminus A$. Let B be the set containing all neighbours of a that belong to W , and $C = W \setminus B$. Using the fact that U is a umodule,

and that $a \in A$, $b \in B$, and $c \in C$, one can deduce for all $x \in A, y \in B, z \in C, t \in D$ that xy and zt are edges while xz and yt are non-edges (this corresponds to bi-joins, which are detailed in see Section 7.1.1). Moreover, U is not a module, and we can deduce that there is a vertex d belonging to D . Finally, one can check that $G[\{a, b, c, d\}]$ is either a P_4 , a C_4 , or a $\text{co-}C_4$. \square

Threshold graphs are known to be one of the smallest graph classes (see e.g. [6]). Therefore for most graphs umodules and modules differ, and Section 7.1.1 is devoted to the umodular decomposition of undirected graphs. However, before deepening decomposition issues, let us first display umodule tractability.

4 Algorithmic Tractability for the general case

This section addresses the general case of an arbitrary homogeneous relation H over a finite set X . As far as we are aware, there is no evidence of a umodule decomposition scheme for arbitrary homogeneous relations. The only valuable objects to compute thus seem to be the maximal umodules with respect to some cut. Using this computation, we also provide a polynomial time umodule existence test (U-primality test).

4.1 Maximal Umodes Computation

Partitions will be ordered with respect to the usual partition lattice: $\mathcal{P} = \{P_1, \dots, P_p\}$ is *coarser* than $\mathcal{Q} = \{Q_1, \dots, Q_q\}$, and \mathcal{Q} is *thinner* than \mathcal{P} , if every part Q_i is contained in some P_j . It is noted $\mathcal{Q} \leq \mathcal{P}$ and $\mathcal{Q} < \mathcal{P}$ if the partitions are different. Let S be a subset of X . As the umodule family \mathcal{U} is closed under union of intersecting members (Proposition 3), the inclusionwise maximal umodules included in either S or $X \setminus S$ form a partition of X , denoted by $MU(S) = MU(X \setminus S)$. In other words, this is the coarsest partition of X into umodules of H , which is thinner than $\{S, X \setminus S\}$. Roughly, it gives an indication on how the umodules are structured w.r.t. S : a umodule either is included in a umodule of $MU(S)$, or properly intersects S , or properly intersects $X \setminus S$, or trivial.

Definition 4 Let H be a homogeneous relation over X . Let $C \subseteq X$. The relation R_C on C is defined as:

$$\forall x, y \in C, R_C(x, y) \text{ if } \forall a, b \in (X \setminus C) \quad H(x|ab) \iff H(y|ab).$$

This clearly is an equivalence relation on C . Furthermore, C is a umodule if and only if R_C only has one equivalence class. Let us define a *refinement* operation, the main algorithmic tool for constructing $MU(S)$.

Definition 5 Let \mathcal{P} be a partition of X and C a part of \mathcal{P} . Let C_1, \dots, C_k be the equivalence classes of R_C . $\text{Refine}(\mathcal{P}, C)$ is the partition obtained from \mathcal{P} , by replacing part C by the parts C_1, \dots, C_k . A partition \mathcal{P} is *refinable* by C if $\text{Refine}(\mathcal{P}, C) \neq \mathcal{P}$. \mathcal{P} is *unrefinable* if for every part C of \mathcal{P} , we have $\mathcal{P} = \text{Refine}(\mathcal{P}, C)$.

Let us say that $\text{Refine}(\mathcal{P}, C)$ is the *refinement* of \mathcal{P} using part C . $\text{Refine}(\mathcal{P}, C)$ is thinner than \mathcal{P} since every part of \mathcal{P} is included in a part of $\text{Refine}(\mathcal{P}, C)$. The *Refine* operation is the algorithmic tool we use for constructing $MU(S)$.

Lemma 1 Let H be a homogeneous relation over X , U a umodule of H , and \mathcal{P} a partition of X . If U is included in a part of \mathcal{P} , then for any part C of \mathcal{P} , U is included in a part of $\text{Refine}(\mathcal{P}, C)$. Moreover, a part C of \mathcal{P} is a umodule if and only if \mathcal{P} is not refinable by C .

Proof: Let U be an umodule and P be the part containing U . For the first statement let consider $\mathcal{Q} = \text{Refine}(\mathcal{P}, C)$ where C is a part of \mathcal{P} . If $P \neq C$, then P remains a part of \mathcal{Q} and still contains U . Else if $P = C$ then, as U is a umodule, the vertices of U cannot be separated using refinement. The proof for the second statement is immediate since C is a umodule if and only if $\mathcal{P} = \text{Refine}(\mathcal{P}, C)$. \square

Following the lemma, it is easy to write the simple partition refinement Algorithm 1. Subsequently we explain how to improve the algorithm for better performance.

Data: $S \subseteq X$
Result: $MU(S)$
 $\mathcal{P} \leftarrow \{S, X \setminus S\}$
while *there exists an unmarked part C in \mathcal{P}* **do**
 if $\mathcal{P} = \text{Refine}(\mathcal{P}, C)$ **then**
 \perp mark C
 else
 \perp $\mathcal{P} \leftarrow \text{Refine}(\mathcal{P}, C)$

Algorithm 1: Standard refinement algorithm

In this standard partition refinement algorithm, each part of the current partition may be *marked*, which means “processed”. When \mathcal{P} is replaced by $\text{Refine}(\mathcal{P}, C)$, the marked parts of $\text{Refine}(\mathcal{P}, C)$ are exactly the marked parts of \mathcal{P} .

Theorem 1 *For every $S \subseteq X$, Algorithm 1 computes the coarsest umodule partition thinner than $\{S, X - S\}$ in $O(|X|^3)$ time.*

Proof: Let us examine the correctness of the algorithm. From first point of Lemma 1, if a part C is marked, then C is a umodule. Consequently, when the algorithm terminates, \mathcal{P} is unrefinable. In order to prove that \mathcal{P} , when the algorithm terminates, is indeed $MU(S)$, one can notice that lemma 1 implies the following invariant: *There is no umodule partition \mathcal{Q} such that $\mathcal{P} < \mathcal{Q} < \{S, X - S\}$, where $<$ is the natural ordering in the partition lattice of X .* Therefore, starting from $\{S, X - S\}$ the algorithm constructs a strictly decreasing chain of partitions of X ending at $MU(S)$.

For time complexity, we notice that: at each step of the “while” loop of the algorithm, either a part is marked (and will never be unmarked, or split), or a part is broken by *Refine* into at least two new parts. As the number of part of \mathcal{P} is bounded by $|X|$, each of the events above can happen at most $|X|$ time and the “while” loop runs $O(|X|)$ times. According to Lemma 2, *Refine* can be implemented in $O(|X|^2)$ time. Using a stack or any other data structure, it is not hard to find an unmarked part in constant time. As the “while” loop performs $O(1)$ -time computations plus one call to *Refine*, the overall algorithm complexity is $O(|X|^3)$. \square

Furthermore, the algorithm can be improved as follows.

Lemma 2 *It is possible to compute $\text{Refine}(\mathcal{P}, C)$ in $O(|X|^2)$ time.*

Proof: We first show how to test for $R_C(x, y)$. Compute, for every element x of C , a partition $\mathcal{H}(x, C) = \{P_x^1, \dots, P_x^{k(x)}\}$ of $X \setminus C$. It is the restriction of H_x to $X \setminus C$, i.e $P_x^i = H_x^i \setminus C$. It is easy to build in $O(|X|)$ time for each element of C . Then we have $R_C(x, y)$ if and only if $\mathcal{H}(x, C)$ is exactly the same partition than $\mathcal{H}(y, C)$. It can be tested in $O(|X|)$ time, but performing this for each couple of elements of C would lead to an $O(|X|^3)$ time implementation of *Refine*(\mathcal{P}, C). Let us instead consider $\mathcal{H}(x, C)$ as a b bit vectors (with $b = |X \setminus C| = O(|X|)$). Looking for duplicates among these vectors can be performed easily, by bucket sorting them on their first bit, then the second, and so on. A scan of all vectors (i.e. of all elements of C) compute the pairwise equal vectors, i.e the R_C equivalent elements of C . It is then easy to split C and to update \mathcal{P} , in $O(|X|^2)$ time. \square

Theorem 2 *For every $S \subseteq X$, the coarsest umodule partition thinner than $\{S, X - S\}$ can be computed in $O(|X|^2 \log(|X|))$ time.*

Proof: Using the well-known Hopcroft’s partition refinement rule it is possible to improve the above algorithm. The idea is to avoid at each step to consider the biggest part, see [40]. Thus, to compute $MU(S)$ assuming that $|S| \leq |X - S|$, we first partition $X - A$ using the “neighbourhoods lists” of all $a \in A$. If we assume a data structure which links each edge ay to its opposite edge ya . We can associate in the meantime to each element $a \in A$ a bitvector representing how $X - A$ sees a . These $|A|$ bitvectors of size $|X - A|$ can be sorted in $O(|X| \cdot |X - A|) \in O(|X|^2)$. Using Hopcroft’s rule, a vertex a can only be explored at most $O(\log(|X|))$ time, which yields the announced complexity. \square

4.2 Testing for U-primality

By definition, if $MU(S)$ contains only trivial umodules, then all nontrivial umodules of X intersect both S and $X \setminus S$. This allows to test the umodular primality of a homogeneous relation in $O(|X|^4 \log |X|)$ time using the following brute force algorithm: test if $MU(\{x, y\})$ is trivial for all pair of elements $\{x, y\}$; output a non trivial umodule if any. Notice that it is hard to have a better performance using the MU algorithm as toolbox, since there exist homogeneous relations having $X \setminus \{x, y\}$ as unique non trivial umodule.

Theorem 3 *It is possible to check in $O(|X|^4 \log(|X|))$ time if a homogeneous relation H is U-prime and, if not, to output a nontrivial umodule.*

5 Local Congruence and Crossing Families

Definition 6 (Local congruence) *Let H be a homogeneous relation on X . For $x \in X$, the congruence of x is the maximal number of elements that x pairwise distinguishes. In other words, it is the number of equivalence classes of H_x . The local congruence of H is the maximum congruence of the elements of X .*

Remark 1 *The standard homogeneous relation of an undirected graph or a tournament has local congruence 2. This value is 3 for an antisymmetric directed graph or a directed acyclic graph. The value is 4 for digraphs.*

When the local congruence of H is 2, so-call *LC2 condition* for short, we obtain the following structural property on its umodule family.

Definition 7 (Crossing family) $\mathcal{F} \subseteq 2^X$ is a crossing family if, for any $A, B \in \mathcal{F}$, that $A \cap B \neq \emptyset$ and $A \cup B \neq X$ implies $A \cap B \in \mathcal{F}$ and $A \cup B \in \mathcal{F}$ (see e.g. [41] for further details).

Proposition 6 *The umodules of a homogeneous relation with local congruence 2 form a crossing family.*

Proof: Without any assumption on the relation, the union of two overlapping umodules is also a umodule. Now let us consider two overlapping sets $A, B \in \mathcal{F}$, with $A \cap B \neq \emptyset$ and $A \cup B \neq X$, by hypothesis $A \setminus B$ and $B \setminus A$ are non-empty, $a \in A \setminus B$ and $b \in B \setminus A$. Moreover to be relevant $A \cap B$ must contain at least two elements otherwise the intersection is obviously a umodule. So $y, z \in A \cap B$. And finally $x \in X \setminus A \cup B$. By hypothesis we have $H(a|xb) \Leftrightarrow H(y|xb) \Leftrightarrow H(z|xb)$ and $H(b|xa) \Leftrightarrow H(y|xa) \Leftrightarrow H(z|xa)$ and as there are only two possible classes we have $H(y|ab) \Leftrightarrow H(z|ab)$. \square

Crossing families commonly arise as the minimizers of a submodular function. For instance, the minimum s, t -cuts of a network form a crossing family. Moreover such a family admits a compact representation in $O(|X|^2)$ space using a tree representation [23].

Considering undirected graphs or tournaments, it is easy to check that the umodule family necessarily is self-complemented. This will be developed in Section 7. But it should be notice that it is not the case for all relations with local congruence 2.

6 Self-complementarity and Bipartitive Families

As previsouly said, forcing the *LC2 condition* on a homogeneous relation H suffices to describe a polynomial-space structure coding the family of umodules of H . Examples of such a relation include standard homogeneous relations of graphs and tournaments. Moreover, those relations have stronger properties, which we will use to show a linear-space structure coding the umodule family.

Definition 8 (Four elements condition) *A homogeneous relation H fullfills the four elements condition if*

$$\forall m, m', x, x' \in X, \begin{cases} H(m|xx') \wedge H(m'|xx') \wedge H(x|mm') \Rightarrow H(x'|mm') \\ \neg H(m|xx') \wedge \neg H(m'|xx') \wedge \neg H(x|mm') \Rightarrow \neg H(x'|mm') \end{cases} .$$

Proposition 7 *Standard homogeneous relations of undirected graphs and tournaments satisfy the four elements condition.*

This is a light regularity condition, allowing to avoid examples similar to that of Fig. 2. Surprisingly enough, it suffices to make the umodule family behave in a very tractable manner (Proposition 8 and Corollary 1 below).

Definition 9 (Self-complementary condition) *A family \mathcal{F} of subsets of X is self-complemented if for every subset A , $A \in \mathcal{F}$ implies $X \setminus A \in \mathcal{F}$.*

Proposition 8 *If a homogeneous relation H fullfills the four elements condition then the family \mathcal{U} of umodules of H is self-complemented.*

Proof: Let us assume that U is a umodule and $X \setminus U$ is not, i.e. there are two elements x and x' of $X \setminus U$, and two elements m and m' of X such that $H(x|mm')$ but $\neg H(x'|mm')$. As U is a umodule, either both m and m' distinguish x from x' (i.e. $\neg H(m|xx')$ and $\neg H(m'|xx')$) or none of m and m' distinguish x from x' (i.e. $H(m|xx')$ and $H(m'|xx')$). The first case is forbidden by the first implication of the four elements condition, and the second case is forbidden by the second implication. \square

The *four elements condition* is quite convenient since it allows to shrink a umodule, hence apply the divide and conquer paradigm to solve optimisation problems. However, as far as umodules are concerned, the *self-complementary* relaxation is sufficient to describe a tree-decomposition theorem as can be seen in the following section. Finally, notice that the converse of Proposition 8 does not necessarily hold. The characterisation of relations having a self-complemented umodule family by a local axiom, such as the four elements condition, actually appears to be more difficult.

6.1 Tree Decomposition Theorem

The following results on bipartitions can be found in [16] under the name of “decomposition frame with the intersection and transitivity properties”, in [35] under the name of “bipartitive families” (the formalism used in this paper), and in [31] under the name of “unrooted set families”.

We call $\{X_i^1, X_i^2\}$ a *bipartition* of X if $X_i^1 \cup X_i^2 = X$ and $X_i^1 \cap X_i^2 = \emptyset$. Two bipartitions $\{X_i^1, X_i^2\}$ and $\{X_j^1, X_j^2\}$ *overlap* if for all $a, b = 1, 2$ the four intersections $X_i^a \cap X_j^b$ are not empty. A bipartition is *trivial* if one of the two parts is of size 1. Let $\mathcal{B} = \{\{X_i^1, X_i^2\}_{i \in 1, \dots, k}\}$ be a family of k bipartitions of X . The *strong* bipartitions of \mathcal{B} are those that do not overlap any other bipartition of \mathcal{B} . For instance, the trivial bipartitions of \mathcal{B} are strong bipartitions of \mathcal{B} .

Proposition 9 *If \mathcal{B} contains all trivial bipartitions of X , then there exists a unique tree $T(\mathcal{B})$.*

- with $|X|$ leaves, each leaf being labelled by an element of X .
- such that each edge e of $T(\mathcal{B})$ correspond to a strong bipartition of \mathcal{B} : the leaf labels of the two connected components of $T - e$ are exactly the two parts of a strong bipartition, and the converse also holds.

Let N be a node of $T(\mathcal{B})$ of degree k . The labels of the leaves of the connected components of $T - N$ form a partition X_1, \dots, X_k of X . For $I \subset \{1, \dots, k\}$ with $1 < |I| < k$, the bipartition $B(I)$ is $\{\bigcup_{i \in I} X_i, X \setminus \bigcup_{i \in I} X_i\}$.

Definition 10 (Bipartitive Family) *A family of bipartitions is a bipartitive family if it contains all the trivial bipartitions and if, for two overlapping bipartitions $\{X_i^1, X_i^2\}$ and $\{X_j^1, X_j^2\}$, the four bipartitions $\{X_i^a \cup X_j^b, X \setminus (X_i^a \cup X_j^b)\}$ (for all $a, b = 1, 2$) belong to \mathcal{B} .*

Theorem 4 *If \mathcal{B} is a bipartitive family, the nodes of $T(\mathcal{B})$ can be labelled complete, circular or prime, and the children of the circular nodes can be ordered in such a way that:*

- If N is a complete node, for any $I \subset \{1, \dots, k\}$ such that $1 < |I| < k$, $B(I) \in \mathcal{B}$.

- If N is a circular node, for any interval $I = [a, \dots, b]$ of $\{1, \dots, k\}$ such that $1 < |b - a| < k$, $B(I) \in \mathcal{B}$.
- If N is a prime node, for any element $I = \{a\}$ of $\{1, \dots, k\}$ $B(I) \in \mathcal{B}$.
- There are no more bipartitions in \mathcal{B} than the ones described above.

For a bipartitive family \mathcal{B} , the labelled tree $T(\mathcal{B})$ is thus an $O(|X|)$ -sized representation of \mathcal{B} , while the family \mathcal{B} can have up to $2^{|X|-1} - 1$ bipartitions of $|X|$ elements each. Furthermore, it allows to perform some algorithmic operations efficiently on \mathcal{B} . The self-complemented umodule families fit in this formalism. Indeed if we define the *bipartition closure* of the umodule family of a homogeneous relation H as $\mathcal{U}'(H) = \{\{U, X \setminus U\} \mid U \text{ is a umodule of } H\}$, then

Proposition 10 *If the umodule family of a homogeneous relation H is self-complemented, then its bipartition closure $\mathcal{U}'(H)$ is a bipartitive family.*

Proof: As $\mathcal{U}'(H)$ is self-complemented each part of a bipartition belonging to $\mathcal{U}'(H)$ is a umodule. Furthermore if the bipartitions $\{U, X \setminus U\}$ and $\{V, X \setminus V\}$ overlap (in the bipartition sense) then U and V overlap (in the set sense). According to Proposition 3 if U and V overlap $U \cup V$ is a umodule, and therefore $\{U \cup V, X \setminus (U \cup V)\} \in \mathcal{U}'(H)$. The self-complementary condition gives the results needed for the three other bipartitions. \square

Corollary 1 *There exists a unique $O(|X|)$ -sized decomposition tree that gives a description of all possible umodules of a homogeneous relation H fulfilling the self-complementary condition.*

This tree is henceforth called *umodular decomposition tree*. Notice that it is an unrooted tree, unlike the modular decomposition tree. The computation of this tree can be done in polynomial time using results of Section 4 as we are to describe next. However, it would be nice to find more efficient algorithms.

6.2 Tree Decomposition Algorithm

Let H be a self-complemented homogeneous relation, $T(H)$ be its umodular decomposition tree and U be a nontrivial strong umodule (if any). Let us examine some consequences of Theorem 4. Notice that two umodules overlap if and only if they are incident to the same node of $T(H)$. As H is self-complemented the union of two overlapping umodules is a umodule (Proposition 3) but also their intersection. The strong umodule U is an edge in $T(H)$ incident with two nodes A and B .

- If one of them, say A , is labelled prime then for any $x, y \notin U$ such that the least common ancestor of them in $T(H)$ is A , then $U \in MU(\{x, y\})$.
- If one of them, say A , is labelled circular then for any x belonging to the subtree rooted in the successor of U in the ordered circular list of A , and for any y belonging to the subtree rooted in the predecessor of U , then $U \in MU(\{x, y\})$.
- If one of them, say A , is labelled complete then the intersection, for all $x, y \notin U$ whose least common ancestor is A , the intersection of all parts of $MU(\{x, y\})$ containing U is exactly U .

As a consequence:

Theorem 5 *There exists an $O(|X|^4 \log(|X|))$ algorithm to compute the unique decomposition tree for a self-complemented umodule family.*

Proof: For every pair $\{x, y\}$ compute $MU(\{x, y\})$ in $O(|X|^2 \log(|X|))$ time (Theorem 1). That gives a family of at most $|X|^3$ umodules. Greedily compute the intersection of overlapping umodules of the family. It is possible in $O(|X|^4 \log(|X|))$ time: for each triple (a, b, c) look for the umodules containing exactly two of them, they overlap. Then we have all strong umodules. We just have to order them into a tree and to label its nodes, an easy task. \square

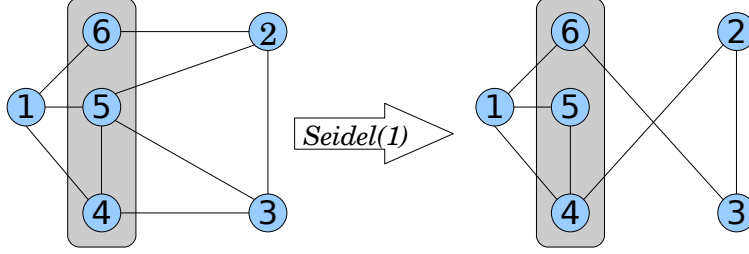


Figure 3: An example of a Seidel switch on an undirected graph

7 Seidel-switching Theorem, a potent Tractability

Standard homogeneous relations of graphs and tournaments are of local congruence 2, and their umodule families are self-complemented. Firstly this means we can either decompose those families using the crossing decomposition or using the bipartitive decomposition. Moreover, relations that satisfy both the self-complementary and *LC2* properties seem to own stronger potential. In particular, let us show a nice local transformation from the umodules of such a relation to the modules of another relation. This operation was first introduced in J. Seidel in [42] on undirected graphs. It was later studied by several authors interested in some computational aspects [14, 32] and structural properties [29, 30] and recently in [36, 37]. The operation is referred to as *Seidel switch* in [30], and we will adopt this terminology.

Roughly speaking, for an undirected graph $G = (V, E)$, and $W \subseteq V$, the Seidel switch on G, W is obtained by taking the complement of the edges between $G_{[W]}$ and $G_{[V \setminus W]}$ (but it should be noticed that $G_{[W]}$ and $G_{[V \setminus W]}$ remain unchanged). Actually we only use a particular case of this powerful operation, when applied to homogeneous relation, we identify the set W as one of the part of an element $x \in X$. For instance, on graphs the set W will be chosen as the neighbourhood of vertex. Moreover there is also a slight difference with the original operation, that is we remove from the transformation the element w such that $H_w^i = W$. For convenience, if H is a homogeneous relation on X and $s \in X$, we also refer to the equivalence classes of H_s as H_s^1, \dots, H_s^k .

Definition 11 (Seidel switch) *Let H be a homogeneous relation of local congruence 2 on X , and s an element of X . The Seidel switch at s transforms H into the homogeneous relation $H(s)$ on $X \setminus \{s\}$ defined as follows.*

$$\forall x \in X \setminus \{s\}, H(s)_x^1 = (H_x^1 \Delta H_s^j) \setminus \{s\} \text{ and } H(s)_x^2 = (H_x^2 \Delta H_s^j) \setminus \{s\}$$

with j such that $x \notin H_s^j$. where $A \Delta B$ denotes the symmetric difference of A and B .

An illustration on undirected graph is given in Figure 3.

Theorem 6 *Let H be a homogeneous relation of local congruence 2 on X such that \mathcal{U}_H is self-complemented. Let s be a member of X , and $U \subseteq X$ a subset containing s . Then, U is a umodule of H if and only if $M = X \setminus U$ is a module of the Seidel switch $H(s)$.*

Proof: Let $C = H_s^1 \cap M$ and $D = H_s^2 \cap M$. Since H is of local congruence 2, $\{C, D\}$ is a partition of M . Let $a \in U \setminus \{s\}$. Suppose that U is a umodule of H . Then, for all $y, z \notin U$, $H(a|yz)$ if and only if $H(s|yz)$. In other words, C is included in one class among H_a^1 and H_a^2 while D is included in the other class. As $C \subseteq H_s^1$ and $D \cap H_s^1 = \emptyset$, $C \cup D$ is included in one among the two classes $H(s)_a^i = H_a^i \Delta H_s^j$ ($i \in \{1, 2\}$ and j as in Definition 11). Hence, $M = C \cup D$ is a module of $H(s)$.

Conversely, if M is a module of $H(s)$, then $C \cup D$ is included in either $H(s)_a^1$ or $H(s)_a^2$. Moreover, the definition of the Seidel switch can also be written as $H_a^i = H(s)_a^i \Delta H_s^j$ for $i \in \{1, 2\}$ and j as in Definition 11. Therefore, C is included in one class among H_a^1 and H_a^2 while D is included in the other class. In other words, for all $a \in U \setminus \{s\}$, and $y, z \notin U$, $H(a|yz)$ if and only if $H(s|yz)$. This implies for all $a, b \in U$, and $y, z \notin U$, $H(a|yz)$ if and only if $H(b|yz)$ and U is therefore a umodule. \square

Modular decomposition trees have well-known properties [11, 34]. They are rooted trees whose leaves are in one-to-one correspondence with elements of X . A node of the modular decomposition tree is exactly a strong module, a module that overlap (in the set sense) no other modules.

For a node N let F_1, \dots, F_k be the leaf-sets of its k children in the tree. When the family of umodules of H is bipartitive as it is the case in Theorem 6, the family of modules of any Seidel switch of H is a partitive set family [11], also known as *rooted set family* [31]. The following theorem from [11] describes the structure of partitive set families.

Theorem 7 *The nodes of a modular decomposition tree T can be labelled complete, linear or prime, and the children of the linear nodes can be ordered in such a way that:*

- *If N is a complete node, for any $I \subset \{1, \dots, k\}$ such that $1 < |I| < k$, $\bigcup_{i \in I} F_i$ is a module.*
- *If N is a linear node, for any interval $I = [a, \dots, b]$ of $\{1, \dots, k\}$ such that $1 < |b - a| < k$, $\bigcup_{i \in I} F_i$ is a module.*
- *If N is a prime node, for any element $I = \{a\}$ of $\{1, \dots, k\}$ $\bigcup_{i \in I} F_i$ is a module.*
- *There are no more modules than the ones described above.*

The relationships between the umodular decomposition tree of H and the modular decomposition tree of $H(v)$ are very tight:

Proposition 11 *Let H be a homogeneous relation of local congruence 2 on X such that \mathcal{U}_H is self-complemented. Let s be an element of X . The umodular decomposition tree \mathcal{T}_H of H and the modular decomposition tree $\mathcal{T}_H(s)$ of the Seidel switch $H(s)$ of H at s have the following properties:*

- *the two trees are exactly the same (same nodes and edges) excepted that the leaf with label s is missing in $\mathcal{T}_H(s)$ but present in \mathcal{T}_H .*
- *The node of \mathcal{T}_H that is adjacent to the leaf s corresponds to the root of $\mathcal{T}_H(s)$ (while \mathcal{T}_H is unrooted).*
- *A circular node of \mathcal{T}_U corresponds to a linear node of $\mathcal{T}_M(s)$. The orderings of the children are the same. The prime and complete nodes are the same in both trees.*

Proof: This is a consequence of Theorem 6. Every strong module of $H(s)$ gives a strong bipartition of \mathcal{T}_H , and the converse is true. Then for a node N of the modular decomposition tree, for any union $\bigcup_{i \in I} F_i$ of leaf-sets of children there is a bipartition $\{\bigcup_{i \in I} F_i, X \setminus (\bigcup_{i \in I} F_i) = B(I)\}$ using the notations defined above. For each bipartition of umodules of H , the part that contain s is dropped and the other part is put is the family of modules of $H(s)$. \square

A similar proof (and more detailed) with this result can be found in [31]. That article indeed describes the relationship between the consecutive-ones ordering and the circular-ones ordering of a boolean matrix, but the results (described in [31] as the transformation of a PQ-tree into a PC-tree) are the same. Notice that the modular decomposition tree of H can be trivial, while the one of its Seidel switch at s may be not.

Corollary 2 *The umodular decomposition tree of a self-complemented homogeneous relation of local congruence 2 on X can be computed in $O(|X|^2)$ time.*

Proof: Using a Seidel switch on any element will result in a homogeneous relation having the so-called *modular quotient* property [8]: every module of the relation also is a umodule. Then, the $O(|X|^2)$ -time modular decomposition algorithm for modular quotient relations depicted in [8] and Proposition 11 allow to conclude. \square

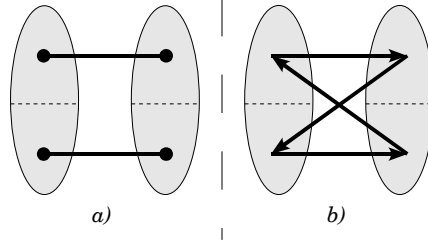


Figure 4: a. A bi-join (i.e. umodule) in an undirected graph, b. a umodule in a tournament

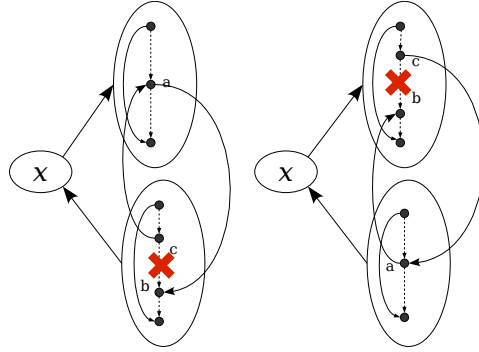


Figure 5: Forbidden arc configuration between neighbourhoods

7.1 Applications to undirected graphs and tournaments

7.1.1 Bi-join decomposition of undirected graphs

Let us now apply the umodular decomposition framework to graphs, or more exactly to the standard homogeneous relation of a graph. The resulting decomposition was already published in [36, 37]. We summarise here the main results of that paper and establish the link with umodules.

Definition 12 (bi-join) A bi-join of a graph $G = (X, E)$ is a bipartition $\{X^1, X^2\}$ of the vertex-set such that the edges between X^1 and X^2 form at most two disjoint complete bipartite graphs, and that for each $i, j = 1, 2$ every vertex of X^i is adjacent to a vertex of X^j .

Proposition 12 If $\{X^1, X^2\}$ is a bi-join of a graph then both X^1 and X^2 are umodules of G .

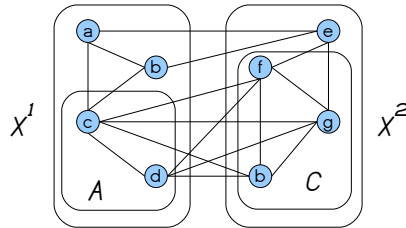


Figure 6: Example of a bi-join of a graph

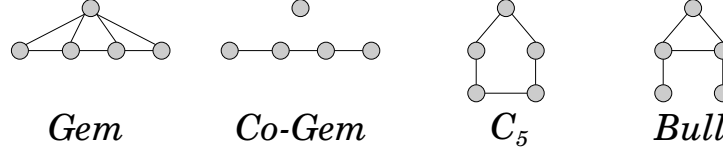


Figure 7: Forbidden induced subgraphs for Completely Bi-join Decomposable Graphs

Proof: Let A (resp. C) be the vertices of X^1 (resp. X^2) incident with the first complete bipartite graph, and B (resp. D) be the other vertices of X^1 (resp. X^2). Any vertex of X^1 distinguishes a vertex of C from a vertex of D , but can not distinguish two vertices from C , nor two vertices from D . X^1 is thus a umodule, and a similar proof holds for X^2 . \square

In [36, 37] the Seidel switch was used to derive most of the properties:

Proposition 13 *Let G be a graph. $\{X^1, X^2\}$ is a bi-join of G if and only if for every $v \in X^1$ (resp. X^2) X^2 (resp. X^1) is a module of the Seidel switch $G(v)$.*

It may be used to prove the converse of Proposition 12:

Proposition 14 *If U is a umodule of a graph $G = (X, E)$ then $\{M, X \setminus U\}$ is a bi-join of G .*

This is because the homogeneous relation of a tournament has local congruence 2 and is self-complemented (see Section 5).

Corollary 3 *The umodular decomposition of a graph equipped with its standard homogeneous relation is exactly its bi-join decomposition.*

Among the consequences exposed in [36, 37], bi-join (thus umodular) decomposition trees have no *circular* nodes.

Theorem 8 [37] *There is a unique unrooted decomposition tree T associated to an undirected graph G . All the nodes are labelled degenerate or prime. There is exactly two kind of degenerate nodes: The clique nodes K_n and the complete bipartite node $K_{n,m}$.*

7.1.2 Isomorphism of $(C_5, \text{bull}, \text{gem}, \text{co-gem})$ -free graphs

In this section, we prove that the isomorphism testing between two graphs totally decomposable w.r.t bi-join decomposition can be tested in linear time. This class of graph is studied by [37]. It is exactly the $(C_5, \text{bull}, \text{gem}, \text{co-gem})$ -free graphs (see Figure 7), and also exactly the graphs that can be obtained from a single vertex by a sequence of (twin, antitwin)-extensions.

It follows from definition that the decomposition tree has no *prime* nodes; furthermore, the decomposition tree alone is an $O(n)$ -sized encoding of the graph (like the cotree is an $O(n)$ -sized encoding of a cograph). We are then reduced to a tree isomorphism problem, as proven below.

Theorem 9 *Let G_1 and G_2 be graphs totally decomposable w.r.t. bi-join decomposition. Isomorphism between G_1 and G_2 can be tested in linear time.*

Proof: From Theorem 8 and [37], the decomposition tree of a graph is uniquely defined, and a decomposition tree with no prime nodes corresponds to exactly one graph. It is then sufficient to test for decomposition trees isomorphism.

It is possible to compute the decomposition trees of G_1 and G_2 in $O(n + m)$ (see [37]). Then the tree isomorphism is achieved in linear time [1]. Notice that decomposition trees are unrooted, and that the internal node labelling with a K or S is already known. \square

7.1.3 A New Tournament Decomposition

We have investigated in Section 7.1.1 the umodules of undirected graphs, and noticed that they lead to a nice decomposition. Similarly for tournaments our theory applies and we present a new tournament decomposition: the umodular decomposition. It is indeed the umodular decomposition of the standard homogeneous relation of the tournament. Actually this decomposition is more powerful than the general modular decomposition of [8, 9, 10], because every module of a tournament is a umodule, while umodular decomposition is able to decompose M-prime tournaments – those without nontrivial modules (Figure 8).

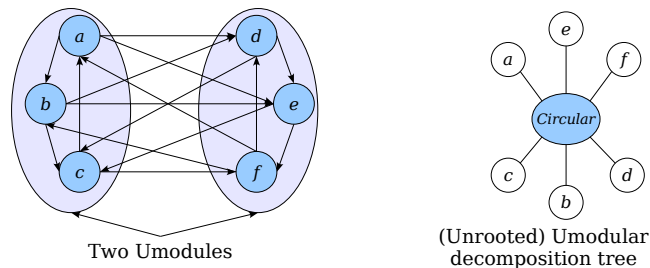


Figure 8: An example of a M-prime tournament which is not U-prime. The umodular decomposition tree is drawn on the right.

We can deduce from Proposition 11 some very interesting properties of the umodular decomposition of tournaments.

Corollary 4 *The umodular decomposition tree of a tournament has no complete node. And there exists a circular ordering of the vertices of the tournament such that every umodule of the tournament is a factor (interval) of this circular ordering.*

Proof: The first observation is obvious using Theorem 6. Furthermore any traversal of the umodular decomposition tree, respecting the order of the sons of a circular node, orders the leaf labels into the desired circular ordering. \square

This result was already known for modular decomposition [35]: there exists a (not circular) permutation of the vertices whose every module of the tournament is a factor. It is called *factorising permutation*.

Proposition 15 *The umodular decomposition tree of a tournament can be computed in $O(|X|^2)$ time.*

Proof: Again Theorem 6 says that one just has to perform a Seidel switch on a arbitrarily chosen vertex, then to compute the modular decomposition of the tournament. This can be done in linear (in fact $O(|X|^2)$) time using the algorithm from [33]. Proposition 11 tells how to cast the modular decomposition tree into the umodular one. \square

Given a graph decomposition scheme, is often worth to consider the totally decomposable graphs with respect to that scheme, namely the graphs in which every "large enough" subgraph admits a non trivial decomposition. In general this leads to the definition of very interesting class of graphs, such as cographs with modular decomposition or distance hereditary graphs with split decomposition. Totally umodular decomposable homogeneous relations may also be defined. Let us deepen the special case of standard homogeneous relations of tournaments.

7.2 Locally transitive tournaments

In this section, we focus on totally umodular decomposable tournaments. We first obtain strong structural relationship between the important graph class of cographs (see e.g. [6]) and the tournament class of locally

transitive tournaments, which also is known as round tournaments, a sub-class of locally semicomplete digraphs (refer to [5] for more details). We then show how our theory provides a very natural manner to obtain several results on round tournaments, including characterisation by forbidding induced subgraphs, recognition, isomorphism testing, and feedback vertex set computation. It is well-known that:

Proposition 16

- If T is an M -prime tournament then T contains an induced cycle with 3 vertices.
- T is totally decomposable w.r.t. modular decomposition if and only if it contains no induced cycle with 3 vertices (it is a transitive tournament).

7.2.1 Characterisation theorems

We have:

Theorem 10 *If T is an U -prime tournament then T contains a diamond (one of the induced subgraph described in Figure 9). T is totally decomposable w.r.t. umodular decomposition if and only if it is diamond-free.*

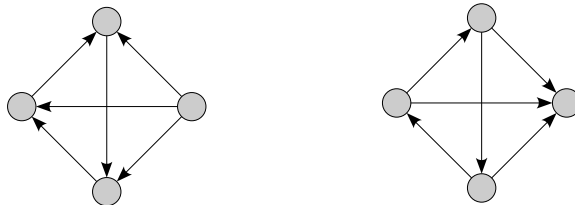


Figure 9: Minimal U-Prime Configurations in tournaments = forbidden subgraphs of a tournament totally decomposable w.r.t. umodular decomposition

Proof: Thanks to Theorem 6, T is U -prime if and only if for any vertex v a Seidel switch at v gives an M -prime tournament. Thanks to Proposition 16, one just has to check all the four-vertices tournaments where a Seidel switch on a vertex produces the cycle with 3 vertices. It is tedious but no hard. \square

Another characterisation is possible:

Definition 13 *A tournament T is locally transitive if for each vertex $x \in V(T)$, $T_{[N^+(x)]}$ and $T_{[N^-(x)]}$ are transitive tournaments.*

It is not hard to see the equivalence between the two classes, a classical result:

Proposition 17 *A tournament T is diamond-free if and only if it is locally transitive*

7.2.2 Recognition algorithm

Thanks to Theorem 10 the class membership can be checked in $O(|X|^4)$ time, and thanks to Proposition 17 in $O(|X|^3)$ time. The following condition provides however a faster test by checking only one vertex of the graph.

Another linear-time recognition algorithm was given by [12]. As far as we know, this french thesis was never published in english. We present here another linear-time recognition algorithm, based on the factorising permutation instead of so-called *circular ordering* (see below). Our algorithm is furthermore *certifying*: it outputs a diamond if the graph is not diamond-free, i.e. not locally transitive.

Proposition 18 *Let T be a tournament and x an arbitrary vertex. T is locally transitive if and only if*

1. $T_{[N^+(x)]}$ and $T_{[N^-(x)]}$ are transitive tournaments, and
2. if a vertex $a \in N^+(x)$ has an out-neighbour $b \in N^-(x)$ and an in-neighbour $c \in N^-(x)$ then $(b, c) \in T$.
3. if a vertex $a \in N^-(x)$ has an out-neighbour $b \in N^+(x)$ and an in-neighbour $c \in N^+(x)$ then $(b, c) \in T$.

Proof: Let us suppose T is totally decomposable. According to Proposition 17, (i) holds. If (ii) does not hold for some vertices a , b and c , i.e. if there is an arc (c, b) instead of (b, c) , then $\{a, b, c, x\}$ induce a forbidden configuration of Figure 9. Same if (iii) does not hold.

Conversely let us suppose that the three conditions hold. We shall prove that for every vertex, its in- and out-neighbourhoods are transitive. Then Proposition 17 tells T is totally decomposable w.r.t. umodular decomposition. For x , this is true thanks to (i). Let t be a vertex of $N^+(x)$. If $T_{[N^+(t)]}$ is not transitive then it contains a circuit (u, v, w) (with arcs, w.l.o.g., (u, v) and (v, w) and (w, u) .) As both $T_{[N^+(x)]}$ and $T_{[N^-(x)]}$ are transitive, the circuit overlaps them. Suppose w.l.o.g $u \in N^+(x)$ and $w \in N^-(x)$. Then (iii) is not true: take $a = w$ and $b = u$ and $c = t$.

If we suppose $T_{[N^+(t)]}$ is not transitive, it contains a circuit (u, v, w) with an arc (u, w) , $u \in N^+(x)$ and $w \in N^-(x)$. (iii) is also violated: take $a = w$ and $b = t$ and $c = u$.

Now let t be a vertex of $N^-(x)$. If $T_{[N^+(t)]}$ is not transitive then it contains a circuit (u, v, w) with an arc (u, w) , $u \in N^+(x)$ and $w \in N^-(x)$. (ii) is violated with $a = u$ and $b = w$ and $c = t$. And if $T_{[N^-(t)]}$ is not transitive then it contains a circuit (u, v, w) with an arc (u, w) , $u \in N^-(x)$ and $w \in N^+(x)$. (ii) is violated with $a = u$ and $b = t$ and $c = w$. \square

Theorem 11 *There exists an $O(|X|^2)$ -time certifying algorithm to recognize if a tournament is locally transitive.*

Proof: Condition (i) of the Proposition 18 can be tested in $O(|X|^2)$ time. Number $a_0 \dots a_k$ the vertices of $N^+(x)$ in increasing order along the transitive tournament $T_{[N^+(x)]}$, and $b_0 \dots b_l$ the vertices of $N^-(x)$ in increasing order. If a_i fulfills (ii), then its out-neighbourhood contains $b_0 \dots b_{f(i)}$ and its in-neighbourhood $b_{f(i)+1} \dots b_l$. This can be tested in $O(|X|)$ time. A similar test in $O(|X|)$ time is performed for each a_i and b_j , leading to an $O(|X|^2)$ -time algorithm. \square

Algorithm 2 presents a certifying implementation of this proof.

7.2.3 Umudular decomposition tree of a locally transitive tournament

Theorem 12 (Umudular decomposition tree of locally transitive tournaments) *The umodular decomposition tree of a locally transitive has only one single node. Moreover this node is a circular node.*

Proof: According to Theorem 6 for any x the Seidel switch at vertex x of a tournament T totally decomposable w.r.t. umodular decomposition gives a tournament $T(x)$ totally decomposable w.r.t. modular decomposition. According to Proposition 16 $T(x)$ is transitive: its modular decomposition tree has a single linear node. According to Proposition 11 the umodular decomposition tree of T only has a circular node. \square

It is well known that for encoding a cograph, it is enough to store its modular decomposition tree. Unfortunately, for a locally transitive tournament, the decomposition tree is not enough since it does not encode adjacencies between vertices.

The circular ordering of the vertices along this unique *circular* node is called a *circular factorising permutation*, since every umodule of G is an interval of this circular permutation, and the converse also holds, by definition of a circular node.

Further results on locally transitive tournaments are known:

Data: A Tournament $T = (V, A)$

Result:

Yes: A circular factorising permutation σ of T .

No: An obstruction.

begin

Pick a vertex $x \in V$

$\mathbf{A} \leftarrow N^+(x)$

$\mathbf{B} \leftarrow N^-(x)$

if $T_{[\mathbf{A}]}$ *is not a transitive tournament* **then**

└ **Failure: certificate is** *A dominated 3-circuit*

if $T_{[\mathbf{B}]}$ *is not a transitive tournament* **then**

└ **Failure: certificate is** *A anti-dominated 3-circuit*

for $i \leftarrow 1$ **to** k **do**

┌ **if** *there exists* $j \in \{0, \dots, l\}$ *such that* **then**

└ $\forall p \leq j$ $b_p \in N_{\mathbf{B}}^+(a_i)$

└ $\forall q > j$ $b_q \in N_{\mathbf{B}}^-(a_i)$

else

└ Find α and β such that $\alpha < \beta$

└ with $b_\alpha \in N_{\mathbf{B}}^-(a_i)$ and $b_\beta \in N_{\mathbf{B}}^+(a_i)$

└ **3-Circuit** $\leftarrow (a_i, b_\beta, x)$

└ **Dominating vertex** $\leftarrow b_\alpha$

└ **Failure: certificate is** *(Dominated vertex,3-Circuit)*

for $i \leftarrow 1$ **to** l **do**

┌ **if** *there exists* $j \in \{0, \dots, k\}$ *such that* **then**

└ $\forall p \leq j$ $a_p \in N_{\mathbf{A}}^+(b_i)$

└ $\forall q > j$ $a_q \in N_{\mathbf{A}}^-(b_i)$

else

└ Find α and β such that $\alpha < \beta$

└ with $a_\alpha \in N_{\mathbf{A}}^-(b_i)$ and $a_\beta \in N_{\mathbf{A}}^+(b_i)$

└ **3-Circuit** $\leftarrow (a_\alpha, b_i, x)$

└ **Anti-Dominating vertex** $\leftarrow a_\beta$

└ **Failure: certificate is** *(Anti-Dominated vertex,3-Circuit)*

Pick a vertex x

Compute a Seidel switch on x

$\sigma \leftarrow \text{Seidel}(x) \cup x$

return $\sigma(V)$ *a circular factorising permutation.*

end

Algorithm 2: Certifying recognition of a totally U -decomposable tournament. The certificate output on failure is a diamond; on success is a circular factorising permutation (the set of intervals of this permutation is exactly the set of umodules)

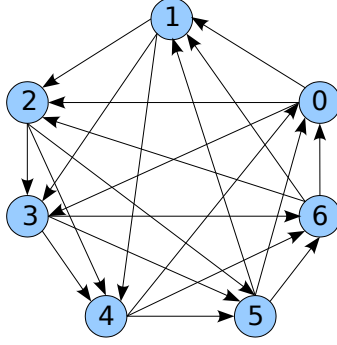


Figure 10: The complete circuit of 7 vertices ($k = 3$).

7.2.4 Circular structure of locally transitive tournaments

A circular structure result of Locally transitive tournament is known from Lopez and Rauzy, we recall it here.

Definition 14 A tournament $G = (V, E)$ is a complete circuit if the vertices can be numbered from 0 to $2k$ and if for every vertex numbered i , its out-neighbourhood is the vertices numbered from $i + 1$ to $i + k$ inclusively (modulo $2k + 1$).

Theorem 13 [27] Let $G = (V, E)$ be a locally transitive tournament. V can be partitioned into $V_0 \dots V_{2k}$, $k \geq 0$, and

- for each $0 \leq i \leq 2k$ $G[V_i]$ is a transitive tournament
- for $x \in V_i$ and $y \in V_j$, if there exists $a \leq k$ such that $i = a + j$ modulo $2k + 1$ then (x, y) is an arc of G , otherwise (y, x) is an arc of G

Notice than every V_i is a module of the graph, furthermore these modules are maximal wrt inclusion (the only module containing V_i is V).

Corollary 5 A nontrivial module M of a locally transitive tournament induces a transitive tournament.

The circular ordering of the $2k + 1$ strong modules, i.e. the circular partition $V_0 \dots V_{2k}$ as defined in Theorem 13. henceforth called *circular ordering*

We have seen another “circular structure” exists: the circular factorising permutation of the n vertices. These two circular orderings are not isomorphic however.

Let $G = ([0 \dots 2k], E)$ be the unique (up to isomorphism) complete circuits of $2k + 1$ vertices. Then let τ be the bijection

$$\tau(i) = ki \text{ modulo } 2k + 1$$

Let τ' be τ seen as a circular list

Proposition 19 The intervals of τ' are exactly the umodules of G

Proof: Thanks to the property of closure under union of overlapping umodules (Proposition 3) we just have to check that the umodules of tow vertices are exactly the pairs $\{i, i + k\}$ (additions are performed modulo $2k + 1$). This is easy to check. \square

This proposition can be generalised if G is locally transitive, but not a complete circuit.

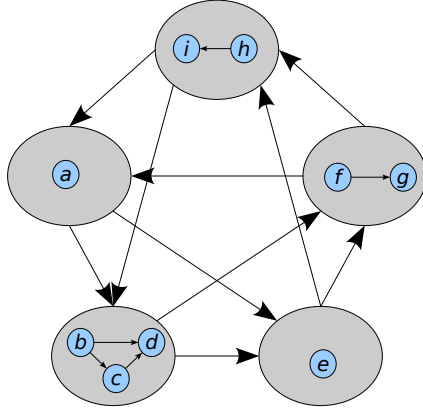


Figure 11: Example of locally transitive graph of 9 vertices: the complete circuit of 5 vertices where each vertex is substituted with a transitive tournament. A circular ordering is $(\{a\}, \{b, c, d\}, \{e\}, \{f, g\}, \{h, i\})$ while a factorizing permutation is $(a, e, i, h, d, c, b, g, f)$ (see Section 7.2.4)

Proposition 20 Let $G = (V, E)$ be a locally transitive tournament and $V_0 \dots V_{2k}$ be its circular ordering. Each V_i induce a transitive tournament, i.e its vertices form a chain $v_i^1 \dots v_i^{f(i)}$.

Let σ be the circular permutation such that

- Each V_i is a factor (interval) of σ
- The V_i follow consecutively following τ' , i.e V_0 then V_k then V_{2k} then $V_{3k} \dots$ (subscripts modulo $2k+1$)
- Within each V_i the ordering of vertices is the reverse of the ordering of the chain: $v_i^{f(i)} \dots v_i^1$.

The umodules of G are exactly the intervals of σ (i.e. σ is a circular factorising permutation of G). Furthermore, σ is the unique circular factorising permutation of G .

Figure 11 gives a example of the relationship between the circular ordering and the circular factorising permutation.

This proposition allows to construct the circular ordering, given the circular factorizing permutation computed by Algorithm 2.

A first step should identify the $2k+1$ induced tournaments. Two vertices u and v are *twins* if $N^+(u) \setminus \{v\} = N^+(v) \setminus \{u\}$. They are *consecutive twins* for a circular factorising permutation σ if they follow consecutively in σ . Let R be the transitive closure of the *consecutive twins* relation

Proposition 21 The equivalence classes of R are exactly the induced transitive tournaments $V_0 \dots V_{2k}$ of the circular ordering of a locally transitive tournament.

Proof: It is not hard to check that, in a tournament, two twins form a module of two vertices, and that the classes of the transitive closure R are thus modules. Then just apply Corollary 5. We just have to check that each class M of R is a *maximal* module: if not then there exists x such that $M \cup \{x\}$ is a module, but then either x and M sink, or x and M source, are twins, contradiction. \square

Then we can give another quadratic-time algorithm than the one of [12]

Theorem 14 The circular ordering of a tournament can be computed in $O(n^2)$

Proof: First Algorithm 2 computes the circular factorising permutation. Then the relation R of Proposition 21 can be computed by checking if the n pairs of consecutive vertices are twins or not. Then the sets $V_0 \dots V_{2k}$ are re-ordered using the inverse of τ as in Proposition 20. \square

7.2.5 Efficient storage of locally transitive tournaments

Definition 15 A composition of n is a list of k integer terms such that the sum of the terms is n . The composition is odd if k is odd.

An circular odd composition of n is a circular list of $2k + 1$ integer terms such that the sum of the terms is n . Notice that a circular list has reading direction: $\{1, 2, 3\}$ differs from $\{3, 2, 1\}$ but is same than $\{2, 3, 1\}$.

Brouwer [7] computed the number of locally transitive tournaments by establishing a bijection between them and “shift registers where the complement of the bit shifted out of the last position is shifted into the first position”. These results can be rephrased as:

Theorem 15 [7] There is a bijection between the totally decomposable tournaments of n vertices and the circular odd compositions of n elements.

Brouwer gave the first terms of the sequence, i.e. the number of locally transitive tournaments on n vertices, referred in Sloane encyclopedia [43] as A000016: 1, 1, 1, 2, 2, 4, 6, 10, 16, 30, 52, 94, 172, 316, 586, 1096, 2048, 3856, 7286, 13798, 26216, 49940, 95326, 182362, 349536, 671092, 1290556, 2485534, 4793492, 9256396, 17895736. He also gave the exact value:

$$\sum_{d|n} \frac{2^{d-1}}{d} \text{odd}\left(\frac{d}{n}\right) \sum_{e|\frac{n}{d}} \frac{\mu(e)}{e}$$

where μ is the Möbius function and $\text{odd}(x)$ is 1 if x odd, 0 otherwise.

Remark that the number of tournaments totally decomposable w.r.t. umodular decomposition is strictly larger than the number of tournaments totally decomposable w.r.t. to modular decomposition. For modular decomposition there exists indeed only *one* tournament totally decomposable with n vertices!

Theorem 16 A locally transitive unlabelled (resp. labelled) tournament of n vertices can be stored in $O(n)$ (resp. $O(n \log n)$) bits.

Proof: If the tournament is unlabelled, one just has to store the corresponding circular odd integer composition. This can be done using a standard encoding of compositions: a vector of $n - 1$ bits. If the k th term of the composition is x , it is stored by $x - 1$ ones followed by a zero. The last bit is always zero and thus can be omitted. For instance the composition $\{2, 3, 1, 1, 3\}$ of 10 is stored as $[1, 0, 1, 1, 0, 0, 0, 1, 1]$. This is a classical canonical encoding of compositions [2].

If the graph is labelled, the permutation of vertices is must also be stored, in $O(n \log n)$ bits. □

7.2.6 Minimum Feedback Vertex Set

The Minimum Feedback Vertex Set problem is NP-Hard on directed graphs [26, GT7], and remains NP-Hard on tournaments [44].

In this section show that the Minimum Feedback Vertex Set is polynomial on tournaments totally decomposable w.r.t. umodular decomposition.

Let us recall what a feedback vertex set is. A feedback vertex set of directed graph $G = (V, A)$ is a subset $V' \subseteq V$ such that each element of V' belongs to at least one circuit of G . The goal is to minimise the cardinality of V' .

Another way of considering this problem is to find a minimum set whose removal will result in an acyclic graph.

Consequently in tournaments, the problem is equivalent to find the maximum sub-tournament induced, which is transitive.

Considering the structure of tournament totally decomposable, it not hard to be convinced that finding the maximum transitive sub-tournament induced can be done in polynomial time. Actually it suffices to find the vertex with the maximum out or in-degree and retrieve its neighbours and then output the complementary set.

Theses operations can be achieved in $O(n^2)$ -time.

7.2.7 Isomorphism

As far as we know, the status of the isomorphism problem is still unknown for tournaments. [4, 3]. [12] gave a linear-time algorithm for locally transitive tournaments isomorphism. It is not hard to see that, given the compact encoding given in Section 7.2.5, isomorphism can be tested in $O(n)$ time.

8 Extensions and further developments

We have presented the umodules and homogeneous relations focusing on graph theory field. But umodules may be found in many other objects. Let us briefly present an example.

8.1 Homogeneous relation based on a binary function

Let f be a binary function $X \times X \rightarrow Y$. The homogeneous relation *based on* f , written H_f , is defined as $H_f(s|ab)$ if and only if $f(s, a) = f(s, b)$ and $f(a, s) = f(b, s)$.

For instance on graphs f is the existence of an edge. On directed graph is the existence of an arc. And on a 2-structures $f(x, y)$ is the number of equivalence class of the couple (x, y) . It can also be seen as a colouring of the edge (x, y) .

Notice that weaker homogeneous relations can be defined from a binary function: the *left* homogeneous relation based on f , H_f^l , is defined as $H_f^l(s|ab)$ if and only if $f(s, a) = f(s, b)$. And the *right* homogeneous relation based on f , H_f^r , is defined as $H_f^r(s|ab)$ if and only if $f(a, s) = f(b, s)$. But these relation do not have the quotient properties, and have not the same umodules. We have:

Proposition 22 *If M is a umodule for H_f^r and for H_f^l then is a umodule for H_f .*

The proof is immediate from definition. Notice that the converse is not true. For instance for $X = \{a, b, c, d\}$ if $f(a, c) = f(a, d)$, $f(b, c) = f(b, d)$ and all other couples have pairwise different values, then $\{a, b\}$ is a umodule for H_f but neither for H_f^l nor for H_f^r . If f is a symmetric function, then the three homogeneous relations of course are the same. This is true for graphs and for symmetric 2-structures, for instance.

Proposition 23 *The principal ideals of a ring are umodules (w.r.t. its multiplication homogeneous relation).*

8.2 Further work

In this paper we study umodular decomposition applied to graphs, when the local congruence is 2, the next challenge is now to understand umodular decomposition of directed graphs or directed acyclic graphs, starting with the self-complemented case first.

Our U-primality test presented here is polynomial, but its asymptotic complexity can surely be reduced, especially when applied to particular combinatorial objects. Same remark holds for the umodular decomposition algorithm of self-complemented homogeneous relation of local congruence greater than two. One can expect better than $O(|X|^4 \cdot \log(|X|))$.

We have noticed here the great importance of the seidel switch operation, and following the notion of vertex minor as defined in [38, 39], let us called H a seidel minor of a graph G , if H can be obtained from G by the two following operations:

- delete a vertex,
- choose a vertex and do a seidel switch on this vertex

It could be of interest to study seidel minors.

References

- [1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Analysis of Computer algorithms*. Assison-Wesley, 1974.
- [2] G. E. Andrews. *The theory of partitions*. Addison-Wesley, 1976. Chapter 4: Compositions and Simon Newcomb’s problem.
- [3] V. Arvind, B. Das, and P. Mukhopadhyay. On isomorphism and canonization of tournaments and hypertournaments. In *17th International Symposium of Algorithms and Computation (ISAAC)*, volume 4288 of *LNCS*, pages 449–459, 2006.
- [4] L. Babai and E. M. Luks. Canonical labeling of graphs. In *15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 171–183, 1983.
- [5] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer-Verlag, 2001.
- [6] A. Brandstadt, V.B. Le, and J.P. Spinrad. *Graph Classes: A Survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 1999.
- [7] A. E. Brouwer. The enumeration of locally transitive tournaments, 1980. Technical report ZW138 of stichting mathematisch centrum, Amsterdam.
- [8] B.-M. Bui-Xuan, M. Habib, V. Limouzy, and F. de Montgolfier. Algorithmic aspects of a novel modular decomposition theory. Technical report, <http://hal.archives-ouvertes.fr/hal-00111235>, 2006. Submitted.
- [9] B.-M. Bui Xuan, M. Habib, V. Limouzy, and F. de Montgolfier. Homogeneity vs. adjacency: generalising some graph decomposition algorithms. In *32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 4271 of *LNCS*, June 2006.
- [10] B.-M. Bui-Xuan, M. Habib, V. Limouzy, and F. de Montgolfier. On modular decomposition concepts: the case for homogeneous relations. *Electronic Notes in Discrete Mathematics*, 27, 2006.
- [11] M. Chein, M. Habib, and M. C. Maurer. Partitive hypergraphs. *Discrete Mathematics*, 37(1):35–50, 1981.
- [12] E. Clarou. *Une hiérarchie de forçage pour les tournois indécomposables*. PhD thesis, Université Claude Bernard Lyon I, 1996.
- [13] D. A. Cohen, M. C. Cooper, and P. G. Jeavons. Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms. Technical Report CS-RR-06-06, Oxford University, 2006.
- [14] C. J. Colbourn and D. G. Corneil. On deciding switching equivalence of graphs. *Discrete Applied Mathematics*, 2(3):181–184, 1980.
- [15] D. G. Corneil. Private communication. Dagstuhl, 2007.
- [16] W. H. Cunningham. *A combinatorial decomposition theory*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 1973.
- [17] A. Ehrenfeucht, T. Harju, and G. Rozenberg. *The Theory of 2-Structures- A Framework for Decomposition and Transformation of Graphs*. World Scientific, 1999.
- [18] A. Ehrenfeucht and G. Rozenberg. Theory of 2-structures. *Theoretical Computer Science*, 3(70):277–342, 1990.
- [19] M. G. Everett and S. P. Borgatti. Role colouring a graph. *Mathematical Social Sciences*, 21:183–188, 1991.

- [20] M. G. Everett and S. P. Borgatti. Regular equivalence: General theory. *Journal of Mathematical Sociology*, 18:29–52, 1994.
- [21] J. Fiala and D. Paulusma. The computational complexity of the role assignment problem. In *30th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 817–828, 2003.
- [22] J. Fiala and D. Paulusma. A complete complexity classification of the role assignment problem. *Theoretical Computer Science*, 349(1):67–81, 2005.
- [23] H. N. Gabow. A representation for crossing set families with applications to submodular flow problems. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 202–211. ACM/SIAM, 1993.
- [24] J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari. Modular decomposition of protein-protein interaction networks. *Genome Biology*, 5(8), 2004.
- [25] T. Gallai. Transitiv orientierbare Graphen. *Acta Math. Acad. Sci. Hungar.*, 18:25–66, 1967.
- [26] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.
- [27] Lopez Grard and Rauzy Claire. Reconstruction of binary relations from their restrictions of cardinality 2, 3, 4 and (n-1). *Z. Math. Logik Grundlag. Math.*, 38(1):27–37 and 157–168, 1992. in two parts.
- [28] M. Habib and M. C. Maurer. 1-intersecting families. *Discrete Mathematics*, 53:91–101, 1985.
- [29] R. B. Hayward. Recognizing 3-structure: A switching approach. *Journal of Combinatorial Theory, Serie B*, 66(2):247–262, 1996.
- [30] A. Hertz. On perfect switching classes. *Discrete Applied Mathematics*, 94(1-3):3–7, 1999.
- [31] W.-L. Hsu and R. M. McConnell. PC-trees and circular-ones arrangements. *Theoretical Computer Science*, 296:99–116, 2003.
- [32] J. Kratochvíl, J. Nešetřil, and O. Zýka. On the computational complexity of Seidel’s switching. In *Fourth Czechoslovakian Symposium on Combinatorics, Graphs and Complexity (Prachatice, 1990)*, volume 51 of *Ann. Discrete Math.*, pages 161–166. North-Holland, Amsterdam, 1992.
- [33] R. M. McConnell and F. de Montgolfier. Linear-time modular decomposition of directed graphs. *Discrete Applied Mathematics*, 145(2):189–209, 2005.
- [34] R. H. Möhring and F. J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. *Annals of Discrete Mathematics*, 19:257–356, 1984.
- [35] F. de Montgolfier. *Décomposition modulaire des graphes. Théorie, extensions et algorithmes*. PhD thesis, Université Montpellier II, 2003.
- [36] F. de Montgolfier and M. Rao. The bi-join decomposition. In *ICGT '05, 7th International Colloquium on Graph Theory*, 2005.
- [37] F. de Montgolfier and M. Rao. Bipartitive families and the bi-join decomposition. Technical report, <http://hal.archives-ouvertes.fr/hal-00132862>, 2005. Submitted.
- [38] S.-I. Oum. *Graphs Of Bounded Rank Width*. PhD thesis, Princeton University, 2005.
- [39] S.-I. Oum. Rank-width and vertex-minors. *Journal of Combinatorial Theory, Series B*, 95(1):79–100, 2005.
- [40] R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

- [41] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer-Verlag, 2003.
- [42] J. J. Seidel. A survey of two-graphs. In *Colloquio Internazionale sulle Teorie Combinatorie (Rome, 1973), Tomo I*, pages 481–511. Atti dei Convegni Lincei, No. 17. Accad. Naz. Lincei, Rome, 1976.
- [43] N. J. A. Sloane. The on-line encyclopedia of integer sequences .
<http://www.research.att.com/~njas/sequences/>.
- [44] E. Speckenmeyer. On feedback problems in digraphs. In *15th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 411 of *LNCS*, pages 218–231, 1989.
- [45] D. R. White and K. P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5:193–234, 1983.