Graphs and hypergraphs : algorithmic and algebraic complexities

Laurent Lyaudet

Laboratoire de l'Informatique du Parallélisme
École Normale Supérieure de Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07, France
e–mail: laurent.lyaudet@ens-lyon.fr; fax: +33 4 72 72 80 80
Laboratoire d'Informatique Fondamentale d'Orléans
Université d'Orléans, Rue Léonard de Vinci, BP 6759 45067 Orléans cedex 2, France
e–mail: laurent.lyaudet@univ-orleans.fr; fax: +33 2 38 41 71 37

1 Introduction

The ubiquity of graphs or hypergraphs in algorithmic[1] and in complexity theory is notoriously known since the origin of these problematics. One can cite the numerous data structures using graphs (trees most of the time), flow algorithms, the many optimization problems on graphs that are NP-complete (proper coloring, maximum clique, ...), the proof of polynomiality of $2-\mathsf{SAT}$, the matroids seen as hereditary hypergraphs with an exchange axiom, etc.

This ubiquity is the consequence of apparently two opposite properties: A graph is a simple object, hence it appears almost everywhere. It can bear very complex structures which yield both very complex problems and very simple solutions. It is now well-known that structures appearing in random graphs are rich enough to provide simple proofs of the existence of particular objects, which are much harder to exhibit without using the probabilistic method. Trying to cope with this richness which is frequently the root of the NP-completeness (or worse) of numerous graph problems, we often restrain ourselves to classes of graphs where the additional constraints would give a loam in which intuition could germinate. This has encouraged the study of topological classes such as planar graphs or classes of graphs with bounded genus, algebraic classes such as Cayley graphs, classes of intersection graphs such as intervals graphs, circle graphs, or permutation graphs.

The subject of my study is the nature of the links between classes of graphs and classes of complexity.

Hierarchical decompositions of graphs Another approach to restrain the complexity of graphs is to decompose one graph into simple elements while preserving its connectivity. These decompositions usually have a tree-structure. Historically, the first type of graphs decompositions is the modular decomposition. Other well-known hierarchical decompositions are the path and tree decompositions (re)introduced by Robertson and Seymour[23, 24] (Halin introduced tree decompositions in 1976 with a different name [15]). Some width parameters are associated with these decompositions that measure the "sizes" of the simple elements used to decompose the graph.

These decompositions have a lot of applications. Indeed, Courcelle [7] showed that all decision problems expressible in monadic second order logic (on the vocabulary τ_1 or τ_2) are decidable in

[1] science of algorithms (standard neologism)

linear time on classes of graphs with bounded treewidth. The vocabulary τ_1 has a unique binary relation representing the adjacency between two vertices. The vocabulary τ_2 contains a binary relation representing the incidence between a vertex and an edge, and two unary predicates allowing to distinguish vertices and edges.

These works have been further expanded upon in two directions: Firstly these results have been extended to optimization or counting problems by extending the monadic second order logic with "counting operators". Secondly a new kind of hierarchical graph decompositions, the clique-decompositions has been introduced by Courcelle, Engelfriet and Rozenberg [8].

Graphs and complexities Since Fagin's Theorem [13], demonstrated in 1974, we know the importance of graphs for descriptive complexity theory. Indeed, it shows that a family of structures (its language) is in NP if and only if it is axiomatizable in existential second order logic (on vocabularies with an order); one family of such structures can always be interpreted as a family of graphs.

Let me say two words of algebraic complexity and more precisely Valiant's model before pursuing the links between graphs and complexities. While in boolean complexity we classify languages that are sets of words on an alphabet, in Valiant's model the alphabet is replaced by a field $\mathbb{K}$, words are replaced by polynomials over $\mathbb{K}$, and languages become sequences of polynomials. The complexity of these families of polynomials is not measured with Turing machines but is measured with arithmetic circuits (directed acyclic graphs with addition or multiplication gates). Thus the class VP (an algebraic analog of P) corresponds to the sequences of polynomials of polynomial degrees computed by a sequence of arithmetic circuits of polynomial sizes. The class VNP is an algebraic analog of NP. Another important class is the one of arithmetic formulas (also called expressions, or terms), denoted $\mathrm{VP_e}$ (e for expression), which corresponds to sequences of polynomials computed by a sequence of arithmetic circuits of polynomial sizes where the underlying graphs are trees.

This importance appears also in the numerous NP-complete graph problems as well as the combinatorial characterizations of the most important #P-complete problems (or VNP-complete in Valiant's model) such as the permanent and the hamiltonian [28, 29]. These combinatorial characterizations are frequently the bridge leading to the definition of "easy" instances for these problems. A famous result of this kind is the result by Fisher, Kasteleyn, and Temperley in 1961 [14, 17, 26] showing that the number of perfect matchings of a planar graph can be calculated in polynomial time using Pfaffian. Another result of this kind is by Courcelle, Makowsky, and Rotics [10] proving that the permanent and hamiltonian[2] of bounded treewidth graphs can be efficiently evaluated.

The goal of this document is to defend the following thesis: The most significant classes of graphs, such as planar graphs or classes defined by the boundedness of some kind of width, are not only efficient tools to find efficient algorithms but more importantly they intrinsically capture part of the complexity of graphs and by extension part of both algorithmic and algebraic complexities. This should appear in the fact that the restrictions of graph problems (satisfying certain properties) to these classes yield complete problems for natural classes of complexity.

[2] but Hamiltonian polynomial according to the rules for scientific names.

2 Algebraic complexity

In the first part of this thesis, we consider expressiveness in Valiant's model of graph covers restricted to particular classes of graphs. By graph cover, we mean a set of edges of a graph such that each vertex is incident to at least one edge of the cover. The weight of a graph cover is the product of the weights of its edges. The graph covers we study are the directed cycle covers, the hamiltonian circuits and the perfect matchings. These three graphs covers are well-known combinatorial characterizations of the permanent and Hamiltonian polynomials when we take the sum of weights of all covers of a given type on a graph. Indeed, the permanent of a matrix M is defined by the following formula

$$\text{per}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}.$$

If $m_{i,j}$ is considered as the weight of the arc from vertex i to j, then the equivalence between the permanent of M and the sum of directed cycle covers of G_M is clear since any permutation can be written down as a product of disjoint cycles and this decomposition is unique.

The *Hamiltonian* polynomial ham(M) is defined similarly, except that we only sum over cycle covers consisting of a *single* cycle (hence the name).

The link between the permanent and bipartite perfect matchings can easily be seen if one considers the bijection between the directed cycle covers of a graph G and the perfect matchings of the graph G' constructed as follows. For each vertex v of G, we create two vertices v^+ and v^- in G'. For each arc from u to v with weight w, we put an edge of weight w between u^+ and v^-.

2.1 Expressiveness of graph covers on graphs of bounded treewidth

We first show in Chapter 3 that the three considered graph covers capture the complexity of arithmetic formulas when restricted to graphs of bounded treewidth. The results of this chapter and Chapter 6 have been obtained in collaboration with Uffe Flarup and Pascal Koiran.

To prove this result, we first show that every arithmetic formula can be expressed as the sum of directed cycle covers of a graph of treewidth 2 (Proposition 3.1.1), the sum of Hamiltonian circuits of a graph of treewidth 6 (Proposition 3.1.2), or the sum of perfect matchings of a graph of treewidth 2 (Proposition 3.1.4). The constructed graphs have size linear in the size of the formula.

Conversely, we show that the sum of directed cycle covers (Proposition 3.2.4), the sum of Hamiltonian circuits (Proposition 3.2.5), and the sum of perfect matchings (Proposition 3.2.6) of a graph of bounded treewidth can all be evaluated by a formula of size polynomial in the size of the graph. To prove these propositions, we exhibit in all three cases a dynamic programming algorithm that, given a tree-decomposition of width k of the graph, constructs a circuit computing the sum of graph covers. The constructed circuit has depth $O(k^2 \cdot d)$ where d is the depth of the tree-decomposition. The results follow from the equivalence between arithmetic formulas and arithmetic circuits of logarithmic depth and the following theorem of Bodlaender [3].

Theorem 1. *Let $G = \langle V, E\rangle$ be a graph of treewidth k with n vertices. Then there exists a tree-decomposition $\langle T, (X_t)_{t\in V_T}\rangle$ of G of width $3k + 2$ such that $T = \langle V_T, E_T\rangle$ is a binary tree of depth at most $2\lceil log_{\frac{5}{4}}(2n)\rceil$.*

Courcelle, Makowsky and Rotics show in [10] that the three graph covers on graphs of bounded treewidth are in VP. In fact, their construction combined with Bodlaender's Theorem is also a proof that these graph covers on graphs of bounded treewidth can be evaluated by arithmetic formulas. However their more general construction (which applies to all graph covers definable in monadic second order logic) gives circuits of size possibly super-exponential in the treewidth whereas our more direct construction is simply exponential in the treewidth.

Combining the proofs related to bounded treewidth graphs (Proposition 3.1.1, Bodlaender's Theorem, and Proposition 3.2.4, for example), we can in fact give a new demonstration of Brent's well-known result on the parallelization of arithmetic formulas (Corollary 3.2.7). This result is interpreted in the non-uniform Valiant's model by the standard inclusion $\mathrm{VP_e} \subseteq \mathrm{VNC}^1$.

2.2 Expressiveness of graph covers on graphs of bounded pathwidth

We next show in Chapter 4 that we can restrict furthermore the class of graphs to bounded pathwidth graphs and still capture the complexity of arithmetic formulas. The results of this chapter and Chapter 5 have been obtained in collaboration with Uffe Flarup. While we have already the inclusion into arithmetic formulas due to the results for graphs of bounded treewidth, we concentrate in this chapter on proving results on the equivalence between certain types of bounded width circuits and arithmetic formulas.

Definition 1. *An arithmetic circuit φ has* bounded width $k \geq 1$ *if there exists a finite set of totally ordered layers such that:*

- *Each gate of φ is contained in exactly 1 layer.*
- *Each layer contains at most k gates.*
- *For every non-input gate of φ, if that gate is in some layer n, then its two inputs are in layer $n+1$.*

We first prove that the output of every weakly skew circuit of width k can be computed as the sum of weights of directed cycle covers of a graph of pathwidth $\lfloor \frac{7 \cdot k}{2} \rfloor - 1$ (Proposition 4.1.2), the sum of weights of Hamiltonian circuits of a graph of pathwidth $7 \cdot k + 2$ (Proposition 4.1.3), or the sum of weights of perfect matchings of a graph of pathwidth $7 \cdot k - 1$ (Proposition 4.1.4). These graphs have a size linear in the size of the weakly skew circuit.

Conversely, we show that the sum of directed cycle covers (Proposition 4.2.4), the sum of hamiltonian circuits (Proposition 4.2.2), and the sum of perfect matchings (Proposition 4.2.3) of a graph of bounded pathwidth can all be evaluated by a skew circuit of bounded width of size polynomial in the size of the graph.

A by-product of these proofs is the equivalence between weakly skew circuits of bounded width and skew circuits of bounded width (Corollary 4.3.1).

We prove in Theorem 4.3.3 that every arithmetic formula can be evaluated by a skew circuit of bounded width (hence by the sum of graph covers of a graph with bounded pathwidth), and reciprocally. Here we use the following Theorem of Ben-Or and Cleve [2].

Theorem 2. *Any arithmetic formula can be computed by a linear bijection straight-line program of polynomial size that uses three registers.*

Let $R_1, \ldots, R_m$ be a set of m registers, a linear bijection straight-line (LBS) program is a vector of m initial values given to the registers plus a sequence of instructions of the form

(i) $R_j \leftarrow R_j + (R_i \times c)$, or
(ii) $R_j \leftarrow R_j - (R_i \times c)$, or
(iii) $R_j \leftarrow R_j + (R_i \times x_u)$, or
(iv) $R_j \leftarrow R_j - (R_i \times x_u)$,

where $1 \leq i, j \leq m$, $i \neq j$, $1 \leq u \leq n$, c is a constant, and $x_1, \ldots, x_n$ are variables (n is the number of variables). We suppose without loss of generality that the value computed by the LBS program is the value in the first register after all instructions have been executed.

Summing our results on bounded pathwidth graphs and bounded treewidth graphs, our results on bounded width (weakly) skew circuits (Theorem 3.0.1 and Theorem 4.0.1), Brent's Theorem, Ben-Or and Cleve's Theorem, and folklore, we obtain the following theorem.

Theorem 3. *Let (f_n) be a family of polynomials with coefficients in a field $\mathbb{K}$. The following properties are equivalent:*

- *(f_n) can be evaluated by a family of arithmetic formulas with polynomial sizes.*
- *(f_n) can be evaluated by a family of arithmetic circuits with logarithmic depths.*
- *(f_n) can be evaluated by a family of linear bijection straight-line programs with polynomial sizes using a bounded number of registers.*
- *(f_n) can be evaluated by a family of straight-line programs with polynomial sizes using a bounded number of registers.*
- *(f_n) can be evaluated by a family of skew circuits with bounded width and polynomial sizes.*
- *(f_n) can be evaluated by a family of weakly skew circuits with bounded width and polynomial sizes.*
- *There exists a family (M_n) of matrices with polynomial sizes and bounded treewidth such that the entries of M_n are 0, 1, constants or variables of f_n and $f_n = \mathrm{per}(M_n)$.*
- *There exists a family (M_n) of matrices with polynomial sizes and bounded treewidth such that the entries of M_n are 0, 1, constants or variables of f_n and $f_n = \mathrm{ham}(M_n)$.*
- *There exists a family (M_n) of symmetric matrices with polynomial sizes and bounded treewidth such that the entries of M_n are 0, 1, constants or variables of f_n and $f_n = \sum_{P \in \mathcal{P}(M_n)} W(P)$, where $\mathcal{P}(M_n)$ is the set of perfect matchings of G_{M_n}.*
- *There exists a family (M_n) of matrices with polynomial sizes and bounded pathwidth such that the entries of M_n are 0, 1, constants or variables of f_n and $f_n = \mathrm{per}(M_n)$.*
- *There exists a family (M_n) of matrices with polynomial sizes and bounded pathwidth such that the entries of M_n are 0, 1, constants or variables of f_n and $f_n = \mathrm{ham}(M_n)$.*
- *There exists a family (M_n) of symmetric matrices with polynomial sizes and bounded pathwidth such that the entries of M_n are 0, 1, constants or variables of f_n and $f_n = \sum_{P \in \mathcal{P}(M_n)} W(P)$, where $\mathcal{P}(M_n)$ is the set of perfect matchings of G_{M_n}.*[3]

This theorem shows that somehow the permanent and hamiltonian can not distinguish the intrinsic complexity of bounded pathwidth and bounded treewidth graphs. A natural question is the existence of a graph cover able to do this distinction. Let us remark that bounded width circuits with polynomial sizes are not even in VP since the family (X^{2^n}) can be computed by such circuits. We mention also the famous analog result in boolean complexity due to Barrington [1].

[3] The last 8 properties are our results.

2.3 Expressiveness of graph covers on graphs of bounded weighted cliquewidth

In Chapter 5, we want to study the expressiveness of graph covers on graphs of bounded cliquewidth. But since every weighted graph can be considered as a clique (which has cliquewidth 2) where non-edges are of weight 0, by taking weighted cliquewidth as the cliquewidth of the underlying unweighted graph, we would add no restriction and obtain VNP-complete problems. For this reason, we define weighted cliquewidth (Definition 2.2.4), weighted NLC-width (Definition 2.2.5), and weighted MC-width (Definition 2.2.6) in a different way. Since, in all three universal algebras used to define these widths, the edges are added by "blocks of complete bipartite graphs", we add the uniformity condition that all edges added by the same operation between all vertices with label a and all vertices with label b must have the same weight. We show that the three obtained weighted widths are still equivalent as in the unweighted case (Theorem 2.2.8).

Contrary to what we achieved for the bounded pathwidth or treewidth graphs, we only obtain distinct lower bounds and upper bounds to the complexity of graph covers on bounded weighted cliquewidth graphs.

For the lower bound, we show that every arithmetic formula can be evaluated as the sum of directed cycle covers of a graph with weighted cliquewidth 13 (Proposition 5.1.1), the sum of Hamiltonian circuits of a graph with weighted cliquewidth 34 (Proposition 5.1.2), and the sum of perfect matchings of a graph with weighted cliquewidth 26 (Proposition 5.1.3). For these results, we have to modify the graphs constructed in the similar proofs for graphs of bounded treewidth. We can not reuse the results for the bounded treewidth case.

Due to our restrictions on how weights are assigned in our definition of weighted cliquewidth it is not true that *weighted* graphs of bounded treewidth have bounded W-cliquewidth. In fact, if one tries to follow the proofs in [11, 6] that show that graphs of bounded treewidth have bounded cliquewidth, then one obtains that a weighted graph G of treewidth k has weighted cliquewidth at most $3 \cdot (|W_G| + 1)^{k-1}$ or $3 \cdot (\Delta + 1)^{k-1}$. W_G denotes the set of weights on the edges of G and Δ is the maximum degree of G. Weighted trees still have bounded weighted cliquewidth (the bound is 3), but we show in collaboration with Ioan Todinca that there exists a family of weighted planar graphs with treewidth 2 and unbounded W-cliquewidth (Corollary B.1.4).

Alternatively, we can prove similar results (Propositions B.2.2, B.2.3, and B.2.4) with bigger constants (22/45/44 instead of 13/34/26) bounding the weighted cliquewidth using the results for bounded pathwidth graphs and the Lemma B.2.1 proving that every graph of pathwidth k has weighted cliquewidth at most $k + 2$.

Another consequence of Lemma B.2.1 combined with the following Theorem of Bodlaender [4]:

Theorem 4. *Every graph G of treewidth k has pathwidth at most $O(k \log(n))$, where n is the number of vertices of G.*

is that every graph G of treewidth k has a weighted cliquewidth at most $O(k \log(n))$, where n is the number of vertices of G. The family of graphs obtained in Corollary B.1.4 proves that this bound is tight.

We show conversely that VP is an upper bound of the complexity of directed cycle covers (Proposition 5.2.3), Hamiltonian circuits (Proposition 5.2.1) and perfect matchings (Proposition 5.2.2) of bounded weighted cliquewidth graphs. While these results seem incomplete, we don't have any evidence that the classes of complexity defined by these covers on graphs of bounded weighted cliquewidth should coincide with already known classes such as $\mathrm{VP_e}$, VDET, or VP.

2.4 Expressiveness of graph covers on planar graphs

We study the expressiveness of graph covers on planar graphs in Chapter 6. Here most of the work has already been done since Bürgisser proves in [5] that the hamiltonian is still VNP-complete on planar graphs when the characteristic of the field is distinct of 2. We remark that the complexity of the hamiltonian on planar graphs when the characteristic is 2 is still an open problem. It is most probably still VNP-complete since Valiant shows in [30] that counting the number of Hamiltonian circuits of a planar graph modulo 2 is $\oplus$P-complete. But the parsimonious reduction used by Valiant from #3−SAT does not give a proof of VNP-completeness in characteristic 2.

The permanent is also still VNP-complete on planar graphs by a result of Datta, Kulkarni, Limaye, and Mahajan [12] proving the #P-completeness of the planar permanent (their proof is easily extended to a proof of VNP-completeness). The same proof yields the VDET-completeness of the planar permanent in characteristic 2.

The perfect matchings are here more singular since it is well-known by a result of Fisher, Kasteleyn, and Temperley [14, 17, 26] that they can efficiently be counted by the use of Pfaffians. In fact, the Pfaffian can still be used to evaluate the sum of weights of perfect matchings of a planar graph.

We show the equivalence between families of polynomials generated by perfect matchings of planar graphs and families of polynomials computed by (weakly) skew circuits of polynomial sizes (Theorem 6.2.3). The equivalence between skew and weakly skew circuits of polynomial sizes is proven in [27] (another proof is given in [22]). Recall that the determinant is complete for this class of complexity, hence its name VDET.

We prove first that every skew circuit can be simulated by the sum of perfect matchings of a planar bipartite graph of quadratic size in the size of the circuit (Proposition 6.2.4). The proof consists to draw the circuit on the plane, then use a gadget to remove crossings and last we show how to construct a planar bipartite graph which sum of perfect matchings is equal to the value of the planar skew circuit. We remark that this proof can be modified to work for weakly skew circuits as well, hence potentially giving another proof of equivalence between skew and (weakly skew) circuits if combined with the next result.

We then show that every sum of planar perfect matchings can be computed by skew circuits of polynomial size (Proposition 6.2.5). In fact, we prove more since we prove that every Pfaffian can be evaluated by skew circuits of polynomial size. Then we use the result of Fisher, Kasteleyn, and Temperley to conclude.

These results show that Pfaffian, "bipartite Pfaffian" (determinant), and "planar bipartite Pfaffian" have the same algebraic complexity. The computational equivalence between Pfaffian and determinant was already known since Mahajan, Subramanya, and Vinay [21] show that their boolean complexity is in NC, more precisely they are both GapL-complete. These results qualify[4] Knuth's affirmation [19]: "[...]Pfaffians are more fundamental than determinants, in the sense that determinants are merely the bipartite special case of a general sum over matchings[...]". While we can easily compute the determinant by Pfaffians using the identity

$$\det(M) = (-1)^{n(n-1)/2}\ \mathrm{Pf}\begin{pmatrix} 0 & M \\ -M^T & 0 \end{pmatrix},$$

[4] "Qualify" is the correct translation of "nuancer", but this meaning is rarely used. Alternatively but less precisely, one could say "slightly change", "slightly modify", "subtly transform", or "refine".

I don't know if there exists an identity giving easy computation of the Pfaffian by determinants. We do have the following identity that gives us the absolute value of the Pfaffian

$$(\mathrm{Pf}(A))^2 = \det(A).$$

3 Hypergraph partitioning

In the second part of this thesis, we study a variant of hypergraph partitioning. It is reasonable to say that every "How to divide efficiently for conquering" problem can be formulated as a hypergraph partitioning problem. Two well-known variants of this problem are the MinCutBipartition Problem and the MinCutBisection Problem. The former asks to minimize the weights of hyperedges crossing a bipartition of the vertices of the hypergraph. It is well-known that this problem can be solved in polynomial time. The later asks to minimize the weights of hyperedges crossing a bisection (bipartition in two sets of same size) of the vertices of the hypergraph. This problem is NP-hard. Almost all variants of hypergraph partitioning used in applications are hence NP-hard since they give constraints or objective functions related to the size of the sets in the partition.

We study here a purely constraint driven variant of hypergraph partitioning defined as follows.

Definition 2 (P_k^l problem). *Given two parameters k and l, $1 \le l < k$, the problem is: Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph ($|V| = n$ and $|\mathcal{E}| = m$); let $t_1, \ldots, t_k$ be positive integers such that $n = \sum_{i=1}^{k} t_i$. Does-there exist a coloring (partition) of V in k subsets of sizes $t_1, \ldots, t_k$ such that the vertices of a hyperedge in $\mathcal{E}$ have at most l distinct colors ? We note P_k^l this decision problem.*

We first study these problems on arbitrary hypergraphs. We show that when $l = 1$ any instance of problem P_k^l can be translated into an instance of k-Subset-Sum [16]. Hence P_k^1 is solvable in time $O(nm+n^k)$ (Theorem 7.2.1). Next we prove that when $l > 1$, P_k^l is NP-complete. We use the following result of Kloks, Kratochvíl, and Müller.

Theorem 5. *P_3^2 is* NP*-hard on instances with colors of equal sizes.*

The problem P_3^2 is strongly related to the branchwidth problem on graphs (see [18], [20]). In [18], Kloks *et al.* show that P_3^2 is NP-complete, proving in particular that the branchwidth problem is NP-complete on splitgraphs and bipartite graphs.

We extend their NP-hardness result using two lemmas proving that:

- if P_k^2 is NP-hard, then P_{k+1}^2 is also NP-hard (Lemma 7.2.3);
- if P_k^l is NP-hard, then P_{k+1}^{l+1} is also NP-hard (Lemma 7.2.5).

We conclude by the NP-completeness of the problem when $l > 1$ (Theorem 7.2.6).

We then study the problem on hypergraphs with disjoint hyperedges. Here we obtain a surprising result since the complexities are inversed. A major difference lies in the size of an input. For general hypergraphs an input has size $O(mn)$. Since hypergraphs with disjoint hyperedges can be seen as interval hypergraphs, an input now has size $O(m \log(n))$. This difference yields the NP-completeness of the problems P_k^1 on this class of hypergraphs (Theorem 7.3.1) since they contain the 2-partition problem.

In fact the true results of this chapter are that we prove that, when $l > 1$, the problems P_k^l are solvable in linear time. We first prove that when $k - 2l < 0$ the problem can be solved in linear time (Theorem 7.3.7) using continuous colorings (Definition 7.3.4). Then we study the structure of the solutions (Theorems 7.3.11 and 7.3.12), proving that either any continuous coloring gives a solution, or if there is a solution, there is one where the biggest hyperedge exhausts at least one color. We can now do an exhaustive search on this second type of solutions and exhaust colors until we obtain an instance such that a continuous coloring is a solution or such that $k - 2l < 0$ (Algorithm 1). We prove that this algorithm has complexity $O((k-2l)m + l((l-1)k^l)^k)$ linear in the size of an instance but exponential in parameters k and l. The complexity relatively to n is neglected since it's only the cost of additions and subtractions (we consider these operations to be taking constant time, otherwise it would be a multiplicative factor of $l((l-1)k^l)^k)$).

This result yields a polynomial time absolute approximation (+1) algorithm for minimizing the maximum number of colors seen by a hyperedge provided that k is fixed (we do a dichotomic search on l and the smallest $l \geq 2$ such that P_k^l has a solution is either the minimum that we search, or this minimum is 1 and we obtain 2).

We give further linear time results for interval hypergraphs with maximum degree 2 when $l = k - 1$ and $k \geq 4$ (Proposition 7.4.6). Another result of linearity is proven in [20] for P_3^2 on this class of hypergraph with instances where all colors have the same size. We conjecture that on the whole class of interval hypergraphs all problems P_k^l are NP-complete.

4 Perspectives

On the hypergraph partitioning problems P_k^l, there are several ways which should be investigated. First, it would be interesting to know if P_k^l problems are always linear (polynomial) when $l > 1$ on intervals hypergraphs with maximum degree d (they will always be NP-complete when $l = 1$), or if the problems P_k^l when $l \neq 1$ become NP-complete as it is the case on general hypergraphs.

In the long term, this problem could be studied on circular intervals hypergraphs or other classes of hypergraphs with a compact encoding with logarithmic size in the number of vertices. It would be interesting to observe the same phenomenon of complexity inversion when restricting the maximum degree with respect to the class of general hypergraphs.

Concerning the algebraic complexity, the short-run perspectives are clear. First it is to improve either the lower bound ($\mathrm{VP_e}$), or the upper bound (VP) on the complexity of graph covers of bounded cliquewidth graphs. The second perspective is to study other graph covers and graph polynomials (partial matchings or Tutte's colored polynomial, they are both known to be VNP-complete on general graphs) on the same graph classes.

Arriving at the end of this document, I must note that I have only partly defended the thesis presented in introduction: The most significant classes of graphs capture an intrinsic part of the complexity of graphs and by extension an intrinsic part of the algorithmic and algebraic complexities. Indeed, our links between classes of complexity and classes of graphs are only obtained relatively to a few kinds of graph covers (even if they are the most fundamentals of the domain). The long-term perspectives of this work are to bound or even measure this bias. I will attempt to suggest some types of results that would constitute a much more general and stronger defense for this thesis.

This thesis is somewhat true for "coarse-grained complexity", i.e. decidability, since we know that a class of graphs has decidable monadic second order logic:

– on vocabulary τ_2 only if it has bounded treewidth (Seese [25]);
– on vocabulary τ_1 with even cardinality predicates only if it has bounded cliquewidth (Courcelle and Oum [9]).[5]

The same question is still open for first order logic.

In his habilitationschrift, Bürgisser presents the unifying concept of generating function of a graph property. He defines a graph property as a set $\mathcal{E}$ of finite graphs closed by isomorphism. The generating function of a graph property on the graph $G = (V, E)$ is defined by

$$\mathrm{GF}(G, \mathcal{E}) = \sum_{E' \subseteq E} w(E'),$$

where the sum is on all subsets of edges E' such that $(V, E') \in \mathcal{E}$. Here again the weight $w(E')$ of a set of edges is defined as the product of the weights of the edges.

Thus one can define the permanent as the generating function of the graph property "to be a disjoint union of circuits" taken on the family of weighted cliques G_n^X where the arc from i to j has weight $X_{i,j}$. In the same way the hamiltonian can be defined as the generating function of the graph property "to be a circuit" taken on the family of weighted cliques G_n^X.

We can extend this definition to consider a generating function on a matrix over $\mathbb{K}$ by replacing a weighted graph by its adjacency matrix but more importantly we can consider the generating function of a logical formula ϕ on vocabulary τ_1 or τ_2 :

$$\mathrm{GF}(G, \phi) = \sum_{E' \subseteq E} w(E'),$$

where the sum is on all subsets of edges E' such that (V, E') is a model of ϕ. This possibility enables us to define the permanent as a first order generating function over τ_1. Indeed the property "to be a disjoint union of circuits" is expressed by the local condition "every vertex has a unique inner neighbor and a unique outer neighbor" translated in the following formula

$$\begin{aligned}\phi_1 = \forall x, [(\exists y, \mathrm{Adj}(x,y) \wedge (\forall z, \mathrm{Adj}(x,z) \Rightarrow y = z))\\ \wedge \quad (\exists u, \mathrm{Adj}(u,x) \wedge (\forall v, \mathrm{Adj}(v,x) \Rightarrow u = v))].\end{aligned}$$

One can also define the permanent with perfect matchings taken on complete bipartite graphs. This is also a first order generating function over τ_1. The formula is even simpler because we only need to say that each vertex has a unique neighbor:

$$\phi_2 = \forall x, [\exists y, \mathrm{Adj}(x,y) \wedge (\forall z, \mathrm{Adj}(x,z) \Rightarrow y = z)].$$

In the undirected case, we suppose that the relation Adj is symmetric.

The hamiltonian is more complex from a logical point of view since connectivity between two vertices can only be checked using a path (i.e. a set of arcs or vertices). Hence we need to use monadic second order logic on the vocabulary τ_2 (or τ_1).[6]

With all this concepts, we can now give the idea of results being real evidences of my thesis. These are of the following type.

[5] In both cases, it is "if and only if" when considering certain "regular" classes of graphs defined with HR or VR grammars.

[6] On τ_2 one obtains the formula:

$$\phi_3 = \forall x \in V, [(\exists y \in V, \mathrm{Adj}(x,y) \wedge (\forall z \in V, \mathrm{Adj}(x,z) \Rightarrow y = z))$$

Imaginary theorem 1 *Every generating function of logic XO satisfying the conditions C is* VX*-complete on graphs with bounded XL width.*

These kinds of theorems with holes are a bit confusing. Here is a more concrete example, one can choose for logic XO the monadic second order logic on vocabulary τ_2, a set of conditions C equal to "being VNP-complete on general graphs", $\mathsf{VX} = \mathrm{VP_e}$ and XL = treewidth. One obtains:

Imaginary theorem 2 *Every generating function of monadic second order logic on vocabulary τ_2,* VNP*-complete on general graphs is* $\mathrm{VP_e}$*-complete on graphs with bounded treewidth.*

We note that this kind of theorem seems unlikely for planar graphs, even for first order logic. Indeed, the results of chapter 6 show that the combinatorial interpretations of the permanent are one VNP-complete, the other VDET-complete on planar graphs. Hence, except if VDET = VNP or if we find a pertinent set of conditions C able to distinguish formula ϕ_1 and formula ϕ_2, we must renounce this way for planar graphs. (It is possible that a set of conditions C which distinguish whether relation Adj is symmetric or not works, but it needs to be verified.)

For these imaginary theorems, one of the difficulties will certainly reside in the variation of complexity of a generating function according to the characteristic of field $\mathbb{K}$, as for the permanent which is VDET-complete in characteristic 2 and VNP-complete otherwise. It will probably be easier to restrict to fields with characteristic 0.

Of course restraining to algebraic complexity is not sufficient; one will need similar results for the boolean complexity. These are of the following type.

Imaginary theorem 3 *Every decision problem expressible in logic XO satisfying the conditions C is* X*-complete on graphs with bounded XL width.*

Once again we take as an example, XO equals to the monadic second order logic on vocabulary τ_2, a set of conditions C equals to "being NP-complete on general graphs", $\mathsf{X} = \mathrm{NC}^1$ and XL = treewidth. One obtains:

Imaginary theorem 4 *Every decision problem expressible in monadic second order logic on vocabulary τ_2 and* NP*-complete on general graphs is* NC^1*-complete on graphs with bounded treewidth.*

The choices of $\mathrm{VP_e} = \mathrm{VNC}^1$ and NC^1 in both examples are a consequence of Bodlaender's theorem on tree decompositions of logarithmic depth.

I will close this session of imaginary theorems by a dichotomy theorem for generating functions of first order, which would give an algebraic analog to Schaeffer's theorem.

Imaginary theorem 5 *Every first order generating function is* VNP*-complete (on a field of characteristic 0?) on general graphs if and only if it satisfies the conditions C. Otherwise, it is in* VP *(*$\mathrm{VP_e}$ *?).*

I hope all this theorems will be the subject of my future research.

$$
\begin{aligned}
&\wedge \quad (\exists u \in V, \mathrm{Adj}(u,x) \wedge (\forall v \in V, \mathrm{Adj}(v,x) \Rightarrow u = v))]\\
\wedge \quad &\forall x, y \in V, [\exists C \subseteq E, \forall e \in C,\\
&((\mathrm{Inc}(x,e) \vee (\exists f \in C, z \in V, \mathrm{Inc}(z,e) \wedge \mathrm{Inc}(f,z)))\\
&\wedge((\mathrm{Inc}(e,y) \vee (\exists f \in C, z \in V, \mathrm{Inc}(e,z) \wedge \mathrm{Inc}(z,f))))],
\end{aligned}
$$

where the adjacency relation Adj is defined as a "macro" using the incidence relation Inc.

References

1. David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In *STOC*, pages 1–5. ACM, 1986.
2. Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992.
3. Hans L. Bodlaender. NC-algorithms for graphs with small treewidth. In Jan van Leeuwen, editor, *WG*, volume 344 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 1988.
4. Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998.
5. Peter Bürgisser. Completeness and reduction in algebraic complexity theory. Habilitationsschrift Universitt Zrich, August 1998.
6. Derek G. Corneil and Udi Rotics. On the relationship between clique-width and treewidth. *SIAM J. Comput.*, 34(4):825–847, 2005.
7. Bruno Courcelle. The monadic second-order logic of graphs I : Recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.
8. Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Context-free handle-rewriting hypergraph grammars. In Hartmut Ehrig, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *Graph-Grammars and Their Application to Computer Science*, volume 532 of *Lecture Notes in Computer Science*, pages 253–268. Springer, 1990.
9. Bruno Courcelle and Sang il Oum. Vertex-minors, monadic second-order logic, and a conjecture by seese. *J. Comb. Theory, Ser. B*, 97(1):91–126, 2007.
10. Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics*, 108(1-2):23–52, 2001.
11. Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.
12. Samir Datta, Raghav Kulkarni, Nutan Limaye, and Meena Mahajan. Planarity, determinants, permanents, and (unique) perfect matchings. In *Proceedings of 2nd International Computer Science Symposium in Russia CSR*, volume 4649 of *Lecture Notes in Computer Science*. Springer-Verlag, 2007.
13. R. Fagin. Generalized first-order spectra and polynomial time recognizable sets. In R. Karp, editor, *Complexity of Computations*, pages 43–73, Providence, RI, 1974. American Mathematical Society.
14. M. E. Fisher. Statistical mechanics of dimers on a plane lattice. *Phys. Rev.*, 124:1664–1672, 1961.
15. R. Halin. S-functions for graphs. *Journal Geometry*, 8:171–186, 1976.
16. Dorit S. Hochbaum and Anu Pathria. The bottleneck graph partition problem. *Networks*, 28(4):221–225, 1996.
17. P. W. Kasteleyn. The statistics of dimers on a lattice. *Physica*, 27:1209–1225, 1961.
18. T. Kloks, J. Kratochvíl, and H. Müller. New branchwidth territories. In *Proceedings 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS'99)*, volume 1563 of *Lecture Notes in Computer Science*, pages 173–183. Springer-Verlag, 1999.
19. Donald E. Knuth. Overlapping pfaffians. *Electr. J. Comb.*, 3(2), 1996.
20. L. Lyaudet. Largeur de branche des graphes de cordes. Master's thesis, ENSLyon, LIP, 2004. http://www.ens-lyon.fr/LIP/Pub/DEA2004.php.
21. Meena Mahajan, P. R. Subramanya, and V. Vinay. The combinatorial approach yields an NC algorithm for computing pfaffians. *Discrete Applied Mathematics*, 143(1-3):1–16, 2004.
22. Guillaume Malod and Natacha Portier. Characterizing Valiant's algebraic complexity classes. In *Mathematical Foundations of Computer Science*, volume 4162 of *Lecture Notes in Computer Science*, pages 704–716. Springer-Verlag, 2006.
23. Neil Robertson and Paul D. Seymour. Graph minors I: Excluding a forest. *J. Comb. Theory, Ser. B*, 35(1):39–61, 1983.

24. Neil Robertson and Paul D. Seymour. Graph minors III: Planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.
25. Detlef Seese. The structure of models of decidable monadic theories of graphs. *Ann. Pure Appl. Logic*, 53(2):169–195, 1991.
26. H. N. V. Temperley and M. E. Fisher. Dimer problems in statistical mechanics – an exact result. *Philosophical Magazine*, 6:1061–1063, 1961.
27. Seinosuke Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Transactions on Information and Systems*, E75-D(1):116–124, 1992.
28. Leslie G. Valiant. Completeness classes in algebra. In *Proc. 11th ACM Symposium on Theory of Computing*, pages 249–261, 1979.
29. Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
30. Leslie G. Valiant. Completeness for parity problems. In Lusheng Wang, editor, *COCOON*, volume 3595 of *Lecture Notes in Computer Science*, pages 1–8. Springer, 2005.