RAPPORT D'ACTIVITÉ

Dmitry Grigoryev

Curriculum Vitæ

Né le 10 Mai, 1954 à Léningrad, URSS,

Adresse actuelle:
Mathématiques, Université Lille-1
Villeneuve d'Ascq 59655, France
dmitry.grigoryev@math.univ-lille1.fr
http://logic.pdmi.ras.ru/~grigorev

Diplômes

1976 Diplôme de mathématicien obtenu avec les félicitations du jury, Université d'Etat de Léningrad

1979 Doctorat en mathématiques (Candidat ès Sciences, diplôme d'état); thèse soutenue à l'Institut Steklov des mathématiques de l'Académie des Sciences d'URSS, Léningrad, titre de la thèse :
Complexité algébrique d'une famille de formes bilinéaires

1985 Doctorat supérieur (Docteur ès Sciences) diplôme d'état délivré par la Commission de Haute Attestation du Conseil des Ministres d'URSS:
Complexité des calculs en algébre polynomiale

Postes

l'Institut Steklov des mathématiques de l'Académie des Sciences d'URSS, Léningrad:

1976-1984 Chargé de recherche,

1984-1988 Chercheur supérieur,

1988-1992 Directeur du laboratoire de recherche en méthodes algorithmiques

1989-1990 Professeur (mi-temps) à l'Institut Polytechnique de Léningrad (URSS)

1992-1998 Professeur aux Départements des Mathématiques et de l'Informatique de Pennsylvania State University (USA)

1998-2008 Directeur de recherche du CNRS, Université de Rennes
2008-à présent Directeur de recherche du CNRS, Université de Lille

Visites

1981, 1985 Institut des Mathématiques de l'Académie des Sciences d'Hongrie, Budapest

1982, 1995 ETH, Zürich (Suisse)

1986 Département d'Informatique de l'Université de Saarbruck (Allemange)

1988 Institut de Calcul Formel, Linz (Autriche)

1989, 1990, 2001, 2005-2009 Max-Planck Institut des Mathématiques, Bonn (Allemagne)

1991-1998 Département d'Informatique de l'Université de Bonn

1992 Département des Mathématiques de l'Université de Nice

1994 l'Université Pompeu Fabra, Barcelona

1995, 1996 Département d'Informatique de l'Université Paris-12

1998 l'Institut de Recherche des Mathématiques, Berkeley (USA)
1995-1999, 2001- 2004, 2006 l'IHES, Bures-sur-Yvette
1999 Département des Mathématiques de l'Université de Bath (Angleterre)
2003-2005 Fraunhofer Institut, Bonn

Exposés en conférences

1976, 1979, 1986, 1988 Conférence en Logique Mathématique, Kichinev, Novosibirsk, Moscou, Leningrad (URSS)

1977, 1979 Workshop en Théorie de Graphes, Odessa (URSS)

1981, 1983, 1985 Workshop International en Complexité des Calculs, Leningrad, Grodno, Kuldiga

1981 Conférence Internationale en Théorie de Calcul, Szeged (Hongrie)

1983 Workshop en Méthodes de Calcul, Zvenigorod (URSS)

1984 Conférence Internationale MFCS, Praga

1985 Workshop International en Calcul Formel, Dubna (URSS)

1986 Congrès International de Mathématiciens, Berkeley (USA) (**45 minutes**)

1986, 1988, 1990, 1994, 1996, 1998, 2000, 2003 Workshop International en Théorie de Complexité, Oberwolfach (Allemagne)

1987 Conférence Européenne en Calcul Formel, Leipzig (Allemagne)

1988 Workshop en Problèmes de Calcul, Riga (URSS)

1989 Workshop en Calcul dans la Théorie des Nombres, Minsk (URSS)

1990, 1992, 1994, 1996, 1998, 2000, 2003 Workshop International International en Méthodes Efficaces en Géométrie Algébrique, Pise (Italie), Nice, Santander (Espagne), Eindhoven (Pays-Bas), Saint-Malo, Bath (Angleterre), Kaiserslautern (Allemagne)

1990 Workshop International en Géométrie Algébrique Réelle, Oberwolfach

1990, 1991, 1996 Conférence Internationale en Calcul Algébrique, Tokyo (Japon), Bonn (Allemagne), Zurich (Suisse)

1991 Workshop International en Complexité Continue, Dagstuhl (Allemagne)

1991 Workshop International en Complexité d'Interpolation, Dagstuhl

1993 Conférence Internationale en Algorithmes Algébriques, Porto-Rico

1993, 1997 Réunion de AMS, session spéciale, Syracuse, College Park (USA)

1993 Workshop International en Algorithmes et Complexité Continue, Barcelona

1994 Workshop International en Complexité Algorithmique de Modèles Algébriques et Géométriques, Paris

1995 Conférence Internationale en Méthodes Numériques, Utah (USA)

1995, 1998 IEEE Conférence en Informatique, Milwaukee, Palo Alto (USA)

1995, 1998, 2004 Workshop International en Calcul Réel et Complexité, Dagstuhl

1995 Workshop en Arithmétique Faible, Paris

1996 Workshop International en Théorie de Complexité, Yorktown Heights, IBM (USA)

1997 Workshop International en Corps Finis, Oberwolfach

1997 ACM Conférence en Théorie de Calcul, El Paso (USA)

1997 Workshop International en Complexité et Géométrie, Toronto

1998 Congrès Pacifique de Mathématiciens, Hong-Kong

1998 Workshop International en Bornes Inférieures de Complexité, Toronto

1998 Workshop International en Calcul Quanta, Dagstuhl

1999 Conférence de la Fondation de Volkswagen, Berlin

2000 Workshop International en Modèles Finis et Logique, Paris

2000 Conférence Internationale en Applications de Calcul, St.Petersbourg (Russie)

2000 Workshop International en Modèles de Calcul, Edimbourg, Ecosse

2001 Congrès International en Géométrie Réelle et Analytique, Rennes

2001 Workshop International en Géométrie Modérée, Luminy

2001 Conférence en Logique et Informatique, Paris

2002 Symposium International sur Algorithmes et Informatique Théorique, Sophie-Antipolis

2002 Conférence Internationale sur Automates, Langues, Programmation, Malaga (Espagne)

2003 Colloque International sur Codes et Cryptographie, INRIA, Rocquencourt

2004 Colloque International sur Calcul Formel et Algoithmes certifiés, Luminy

2004 Atelier International sur Méthodes Algébriques en Cryptographie, Dortmund (Allemagne)
2005 l'Ecole de Printemps d'Informatique, Montagnac
2005 Atelier International sur Méthodes Algébriques en Cryptographie, Bochum (Allemagne)
2007 Atelier International sur Groupes Combinatoires, Dortmund
2007 Atelier International sur Théorie de Preuves, Bonn
2009 Atelier Logique et Informatique, Université Paris-12

Exposés aux séminaires

1976-1988 Université de Moscou (URSS)

1979 l'Institut de Méthodes Mathématiques dans l'économie, Moscou

1981-1985 l'Institut des Mathématiques de l'Academie des Sciences, Minsk

1979-1988 Université de Riga

1981 l'Institut de Problèmes de Transmission d'Informations, Moscou

1985 l'Institut des Mathématiques de l'Academie des Sciences d'Hongrie, Budapest

1986 Réunion de la Société des Mathématiques de Moscou

1986 Université de Saarbruck (Allemagne)

1986, 1988, 1990, 2001 Université de Bonn

1986, 1989 Université d'Aix-la-Chapelle (Allemagne)

1986, 1991 Université d'Augsburg (Allemagne)

1986, 1988, 1991, 1995, 2003 Université de Francfort (Allemagne)

1988 l'Institut de Systèmes de Programmes, Pereslavl-Zalesskii (URSS)

1989 Université de Bielefeld (Allemagne)

1988, 1989, 2001 Université de Dortmund (Allemagne)

1989, 1995 Université de Konstanz (Allemagne)

1989 Université de Passau (Allemagne)

1988 Université de Linz (Autriche)

1988 Université de Vienna (Autriche)

1988 Université de Salzbourg (Autriche)

1989 IBM centre de recherche, Heidelberg (Allemagne)

1990 Université de Karlsruhe (Allemagne)

1990 Université de Regensbourg (Allemagne)

1991 Université de Darmstadt (Allemagne)

1991, 2001 Société de Mathématiques et de Traitement de Données, St.Augustin (Allemagne)

1991 Université de Wurzbourg (Allemagne)

1991, 1992 Université de Pennsylvanie (USA)

1991, 1993 Université de Cornell (USA)

1991 Bell Labs, ATT (USA)

1991, 1993 Université de Carolina Nord (USA)

1991 Duke Université (USA)

1991 Université de Delaware (USA)

1992 Harvard Université (USA)

1992 Université de Strasbourg

1992 Université de Nice

1992 IBM centre de recherche, Yorktown Heights (USA)

1992 Université de New-York, Albany (USA)

1993 Université de Maryland (USA)

1992, 1993, 1994, 1997 Université de Princeton (USA)

1993 Rutgers Université (USA)

1993 l'Institut de Advanced Study (USA)

1994 Université de Toronto (Canada)

1994 Drexel Université, Philadelphie (USA)

1994 Université de Barcelona (Espagne)

1995 Université de Mannheim (Allemagne)

1995, 2003 Université de Paderborn (Allemagne)

1995, 1996, 1998, 1999 l'IHES, Bures-sur-Yvette

1982, 1995, 1996 ETH, Zurich (Suisse)

1996 Université de Bochum (Allemagne)

1996 Université de Courant, New-York

1997 Université d'Ilmenau (Allemagne)

1999 Université de Bath (Angleterre)

1999 Université d'Oxford (Angleterre)

1999 Université d'Angers

1999 Université de Dijon

1999 l'ENS de Lyon

1995, 1996, 1998, 1999 Université Paris-12

2001 Université de Lille

2001 Université de Trier (Allemagne)

2001 Université de Delft (Pays-Bas)

1995, 1999-2008 Université de Rennes

1999-2002, 2004 Institut des Mathématiques de l'Académie des Sciences de la Russie

2003 l'ENS, Paris

2004 Université de Louvain (Belgique)

2005 Université de Tilburg (Pays-Bas)

2005 Université Paris-7

2006-2009 Max-Planck Institut des Mathématiques, Bonn

2006-2008 Université de Technologie d'Informatique, Bonn

Autres activités

Rédactionnelles:

rédacteur des ouvrages sur la théorie de complexité de l'Institut Steklov des Mathématiques de l'Académie des Sciences de l'URSS, Léningrad, 1980-1992,

rédacteur du recueil "Méthodes algébriques en cryptographie", Springer, 2005,

rédacteur des recueils sur Informatique, Springer, 2007, 2008.

membre des comités de rédaction des revues:

Computational Complexity, Birkhauser, à partir de 1990,

Applicable Algebra in Engineering, Communications and Computations, Springer, à partir de 2003,

Groups, Complexity, Cryptology, Heldermann, à partir de 2007.

Membre de comité de programme:

Workshop en Complexité de Calculs, Léningrad (URSS) 1981, Grodno (URSS) 1983, Riga (URSS) 1985

Conférence en Logique Mathématique, Léningrad (URSS) 1988, 1999

Conférence Internationale en Calcul Algébrique, Tokyo (Japon) 1990, Kiev (Ukraine) 1993, Londre (Canada) 2001

Méthodes Efficaces en Géométrie Algébrique, Pise (Italie) 1990, Nice 1992, Santander (Espagne) 1994, Eindhoven (Hollande) 1996, Saint Malo 1998, Bath (Angleterre) 2000, Kaiserslautern (Allemagne) 2003, Alghero (Italie) 2005, Linz (Autriche) 2007, Barcelona (Espagne) 2009.

Conférence Internationale en Informatique, St.Petersbourg 2006 (Président du comité de programme), Yecaterinbourg 2007, Moscou 2008, Novosibirsk 2009, Kazan 2010

Congrès International de Mathématiciens (section d'informatique théorique), Berlin 1998, Inde 2010.

Codirecteur du programme Russie-Allemagne en théorie de complexité au sein de la Volkswagen-Fondation, 1993-1999

Rapporteur dans les revues et compte-rendus:

J. American Math. Society, Computational Complexity, SIAM J. Comput., J.Comput. and System Sci., J. Association Comput. Machinery, Theoret. Comput. Sci., J. Symbolic Comput., Foundations of Comput. Math., Discrete and Comput. Geometry, J. Complexity, J. Algorithms, Information and Comput., Information Processing Letters, Applicable Algebra in Engineering Communic. and Comput., IEEE Transactions on Information Theory, Finite Fields and Applications, Monatshefte, DIMACS series in Discrete Math. and Comput.

Proceedings: ACM Symp. Theory of Comput., IEEE Symposium Foundations of Comput. Sci., Symp. on Comput. Complexity, Symp. on Foundations of Comput. Technology and Theoretical Comput. Sci.

Directions des thèses

I.Ponomarenko (1983-1986)
A.Ayad (2003-2006)
A.Kojevnikov (2005-2007)
S.Nikolenko (2006-2008)

Enseignement

J'ai donné un cours de DEA "Bornes inférieures en complexité algébrique", l'Université de Rennes, 2000

Prix

1984 Prix de la Société Mathématique de Léningrad (URSS)
1994 Prix International de Max-Planck de la Société de Max-Planck
1995 Bourse de NSF (USA)
2002 Prix International d'Alexandre von Humboldt de la Fondation d'Humboldt

Travaux et objectifs

Abstract

Mon domaine de recherche est la théorie de complexité. Le but de celle-ci est à borner la complexité de problèmes algorithmiques et alors, des problèmes divers exigent d'approches de parties différentes des mathématiques.

Contents

1 Cryptographie à clef publique et la théorie de groupes 11

1.1 Un invariant d'une représentation d'un groupe comme une clef secrète . . . 11

1.2 Cryptoschémas homomorphes . . . 15

1.3 Simulations de calculs de façon codée . . . 20

1.4 Cryptoschéma universel et puissance d'adversaires . . . 22

2 Complexité en calcul formel 22

2.1 Factorisation de polynômes à plusieurs variables . . . 22

2.2 Résolution d'un système algébrique et élimination de quantificateurs . . . 23

2.3 Résolution d'un système d'inégalités polynomiales et décidabilité dans algèbre de Tarski . . . 24

2.4 Calcul formel avec polynômes creux . . . 25

3 Construction de stratification de Thom-Whitney-a universelle et théorème de type de Sard pour variétés singulières 27

4 Complexité dans la robotique 28

5 Complexité de calculs avec équations différentielles 29

5.1 Equations différentielles ordinaires . . . 29

5.2 Equations différentielles partielles . . . 30

6 Complexité quantique et probabiliste 32

7 Approximations et complexité **33**

8 Bornes inférieures sur la complexité **35**
8.1 Arbres de calculs et la courbure 35
8.2 Arbres probabilistes de calculs et singularités 36
8.3 Complexité des arbres analytiques, topologiques et parallèles . 38
8.4 Complexité additive, fonctions algébriques et de Pfaff 40
8.5 Complexité multiplicative et le rang d'un tenseur 40
8.6 Complexité de fonctions booléiennes 41

9 Un caractère topologique de la classe de complexité P **42**

10 Complexité de machines de Blum-Shub-Smale **43**

11 Complexité de preuves algébriques et semi-algébriques **43**

12 Complexité d'isomorphisme de graphes et d'algèbres **47**

13 Complexité de machines de Turing et de Kolmogorov **47**

14 Problèmes de complexité en biologie **47**

1 Cryptographie à clef publique et la théorie de groupes

1.1 Un invariant d'une représentation d'un groupe comme une clef secrète

La théorie de groupes convient bien aux buts de la cryptographie à clef publique, quand même jusqu'ici des groupes, en particulier *non-abéliens* sont impliqués très peu dans la cryptographie. Je propose une idée d'utilisation possible de groupes pour un codage à clef publique.

Soit $G : E \to E$ une action d'un groupe G sur un ensemble E qui joue le rôle de l'espace des *messages codés*. Un sous-ensemble $M \subset E$ consiste en tous les *messages propres* étant transversal par rapport aux orbites de l'action de G, c'est-à-dire pour n'importe quels messages $m_1, m_2 \in M, m_1 \neq m_2$ propres différents leurs orbites Gm_1 et Gm_2 sont disjointes. Alors, E, G, M

constituent une *clef publique* créée par le destinataire Alice. Si l'expéditeur Bob veut envoyer un message $m \in M$ par un canal de communication public, il choisit (de façon aléatoire) un élément g du groupe G et Bob transmet un message codé $gm \in E$. Par cela même il s'agit d'un codage *probabiliste*. Dans le cas le plus simple M consiste en deux éléments $M = \{m_0, m_1\}$ qui correspondent aux bits.

Pour décoder Alice doit connaître un invariant $w : E \to A$ de l'action de G, c'est-à-dire, w est constant sur chaque orbite de G. L'invariant w constitue une *clef secrète* d'Alice et afin de décoder un message $e = gm$ Alice calcule $w(e)$ et cherche $m \in M$ tel que $w(e) = w(m)$, donc m a été le message propre de Bob. Pour un décodage univoque il faut que $w(m_2) \neq w(m_3)$ lorsque $m_2 \neq m_3 \in M$. Dans le cas $M = \{m_0, m_1\}$ ça signifie que $w(m_0) \neq w(m_1)$ et Alice compare $w(e)$ avec $w(m_0)$ (ou bien avec $w(m_1)$). D'autre part, sans connaître de w il devrait être difficile à un adversaire Charle à décoder le message e.

Alors, on a esquissé une idée en gros et afin de produire un cryptoschéma particulier on a besoin de spécifier G, E, M, w, A.

La première suggestion est à utiliser la théorie de représentations de groupes. Alors, $G \subset GL_n(F)$ soit un groupe de matrices qui agit sur l'espace $E = F^n$. Dans la théorie de représentations un corps F est d'habitude algébriquement clos, cependant pour des calculs on traite des sous-corps de F, en particulier, les corps finis ou bien le corps de nombres rationnels.

Afin de coder il faut avoir publiquement le groupe G au moyen d'une famille de ces génératrices $\{g_i\} \subset GL_n(F)$ étant par cela même des matrices. Un invariant $w \in F[X_1, \dots, X_n]$ est un polynôme des coordonnées de vecteurs $X_1 e_1 + \dots + X_n e_n \in E$ où $e_1, \dots, e_n$ est une base de l'espace E. Malheuresement, seulement pour peu de séries de groupes G leurs invariants (ou bien l'anneau des invariants) sont connus explicitement. Et bien entendu, ces invariants sont connus publiquement. Or nous avons besoin des cacher, comment on peut le faire?

Alors, je propose de faire intervenir une conjugaison secrète $a^{-1}Ga$ d'un groupe G connu publiquement, où $a \in GL_n(F)$ est une matrice secrète. Donc, $\hat{w}(x) = w(ax)$ est un invariant de la conjugaison $a^{-1}Ga$. On prend $M = \{v_0, v_1\}$ pourvu que les vecteurs $v_0, v_1 \in F^n$ satisfassent la condition $w(av_0) \neq w(av_1)$.

Maintenant on décrit un cryptoschéma suivant (voir [100]):

La clef publique: génératrices $\{\hat{g}_i\}$ du groupe $a^{-1}Ga$ et les vecteurs v_0, v_1

La clef secrète: a

Le codage: afin de coder le message propre 0 (respectivement, 1) Bob choisit de façon aléatoire un élément $g \in a^{-1}Ga$ sous la forme de $g = \hat{g_{i_1}} \cdots \hat{g_{i_l}}$ et transmet l'élément $e = gv_0 \in E$ (respectivement, $e = gv_1 \in E$) par un canal de communication public.

Le décodage: afin de décoder un message $e \in E$ Alice calcule $w(ae)$ et le compare avec $w(av_0)$ (ou bien avec $w(av_1)$). Si $w(ae) = w(av_0)$, alors le message propre de Bob a été 0, sinon $w(ae) = w(av_1)$ et le message propre a été 1.

Je vais donner quelques exemples de groupes et leurs invariants appropriés.

Exemple 1. Le groupe $G \subset GL_n(F)$ est engendré par le groupe S_n symétrique qui permute les éléments de la base $e_1, \dots, e_n$ et par les matrices t diagonales telles que $te_i = c_i e_i$ où $c_i^m = 1, 1 \leq i \leq n$ pour un certain entier m. En tant qu'un invariant w on peut prendre $x_1^m + \cdots + x_n^m$.

Exemple 2. Le groupe $G = SL_n(F)$ agit sur le produit symétrique $E = S^2(F^n)$, en d'autres termes l'espace de matrices symétriques (ou c'est la même chose, de formes quadratiques). L'action de G est décrite par la formule $v \to mvm^T$ où $v \in S^2(F^n)$ et $m \in SL_n(F)$. Alors, le déterminant $w = \det(v)$ est un invariant.

Exemple 3. Soit le groupe $G = GL_n(F)$ qui agit sur la somme directe $F^{2n^2} = F^n \oplus \ldots \oplus F^n$ par la formule $m(p_1, \dots, p_{2n}) = (mp_1, \dots, mp_{2n})$. Pour n'importe quelle paire de partitions $I_1 \cup J_1 = I_2 \cup J_2 = \{1, \dots, 2n\}$ où toutes les puissances $|I_1| = |I_2| = |J_1| = |J_2| = n$ sont égales à n on fait intervenir un invariant rationnel

$w = (\det_{I_1} \det_{J_1})/(\det_{I_2} \det_{J_2})$

où $\det_I$ note le déterminant de tous les vecteurs p_i avec $i \in I$.

Jusqu'au présent pour aucun cryptoschéma sa sécurité *absolue* n'est prouvée. On ne connaît que les théorèmes de façon relatives: notamment, on fixe un certain procédé d'attaque et démontre qu'au moyen d'une telle attaque c'est impossible à casser un cryptoschéma donné. Ou bien plus précisément, si on cassait le cryptoschéma au moyen d'une telle attaque, alors une certaine conséquence incroyable en résulterait.

C'est pourquoi dans les articles sur la cryptographie on considère nombreuse attaques que l'auteurs comptent être raisonnables. Nous tenons également 3 attaques suivantes.

Dans celle première un adversaire Charle cherche un invariant $\hat{w}$ du groupe $a^{-1}Ga$ sous la forme de $\hat{w}(v) = w(bv)$ où w est un invariant connu d'un degré d du groupe G et b est une matrice inconnue. Donc, $\hat{w}$ doit remplir les équations

$$w(bv) = \hat{w}(v) = \hat{w}(\hat{g}_i v) = w(b\hat{g}_i v)$$

quel que soit i. Charle obtient un système d'équations polynomiales du degré d de n^2 variables $b_{i,j}$ étant les éléments de la matrice $b = (b_{i,j})$. Ce système comporte $\binom{n+d-1}{d}$ équations.

Autrement, Charle peut chercher $\hat{w}$ sous la forme d'un polynôme du degré d avec des coefficients inconnus, alors chaque équation $\hat{w}(v) = \hat{w}(\hat{g}_i v)$ pour un vecteur v paramétrique rend $\binom{n+d-1}{d}$ équations linéaires des coefficients inconnus de $\hat{w}$.

Si le degré d est assez grand, en d'autres termes G ne posséde aucun invariant de petit degré, alors Charle a besoin de résoudre un système de $\binom{n+d-1}{d}$ équations. Dans trois exemples numérotés les groupes ne possédent que des invariants de degrés supérieurs à n, donc Charle a besoin de traiter des systèmes de tailles exponentielles. Et bien entendu, d'autre part Alice doit savoir calculer l'invariant $w(v)$ (ou bien $\hat{w}(v) = w(av)$) en temps polynomial.

Au cours de la deuxième attaque Charle cherche une matrice b telle que $bHb^{-1} = G$ où H est le groupe $H = \langle \hat{g}_i \rangle$ engendré par les éléments $\{\hat{g}_i\}$. En général, celui-ci s'appelle *le problème de conjugaison* et en particulier, le problème d'isomorphisme de graphes s'y raméne.

En train de la troisième attaque étant donné un message codé $e \in E$ Charle cherche un élément $g \in H = \langle \hat{g}_i \rangle$ tel que soit $gv_0 = e$ soit $gv_1 = e$, autrement dit, il cherche un élément g avec l'aide duquel Bob a codé son message propre. Donc, on est amené au *problème d'orbites*: si deux éléments appartiennent à la même orbite? Celui-ci est NP-complet en général.

Dans [117] on a fabriqué un cryptoschéma dont la clef secrète est un invariant d'un groupe G et on produit G en utilisant la construction de [110] en donnant G comme un arbre de groupes. Donc, afin de casser ce cryptoschéma il suffirait trouver l'arbre ou bien un invariant de G directement.

1.2 Cryptoschémas homomorphes

Maintenant on passe aux cryptoschémas homomorphes. Soient G, H des groupes et $f : G \to H$ un épimorphisme. En outre on suppose qu'un sous-ensemble $R \subset G$ est donné qui est isomorphe à H, notamment, $|R| = |H|, f(R) = H$ et de plus, on a un algorithme de la complexité polynomiale qui assure cet isomorphisme. Ça veut dire que pour n'importe quel élément $r \in R$ le dernier algorithme rend un élément $h \in H$ tel que $f(r) = h$ et inversement, pour chaque $h \in H$ l'algorithme rend un élément (univoque) $r = f^{-1}(h) \in R$.

Un élément $h \in H$ est traité comme une lettre d'un message propre. Pour le coder on suppose qu'on connaît publiquement un algorithme qui engendre des éléments du noyau $\ker(f) \subset G$. Alors, pour coder h Bob choisit de façon aléatoire un élément $k \in \ker(f)$ et transmet l'élément $e = rk = f^{-1}(h)k \in G$ par un canal de comminication public. C'est-à-dire, un cryptoschéma homomorphe est celui *probabiliste*.

Par cela même, afin de décoder e Charle doit vérifier pour chaque $h_1 \in H$ si $(f^{-1}(h_1))^{-1}e \in \ker(f) \iff f(e) = h_1$, autrement dit Charle doit savoir reconnaître le noyau $\ker(f)$, on tient que ceci soit difficile.

On dit que (G, H, f, R) est un *cryptoschéma homomorphe* lorsque Alice posséde une *clef secrète* pour calculer f. Au lieu de groupes G, H on pourrait considérer des anneaux, algèbres, corps ou bien d'autres structures algébriques, leurs homomorphismes et des cryptoschémas homomorphes correspondants.

Quel est un motif afin d'étudier des cryptoschémas homomorphes? Un d'eux est un protocole qui s'appelle

Calcul avec données celées.

Soit $\Gamma(x_1, \ldots, x_n) : H^n \to H$ une fonction sur H qui est donnée au moyen d'un *circuit*. On désigne par $\odot$ (respectivement, par $\otimes$) l'opération dans le groupe H (respectivement, dans le groupe G). Un circuit C avec une entrée $x_1, \ldots, x_n$ est une suite d'opérations dont i-ème ($1 \le i \le m$) a la forme

$y_i =$ soit

1) $y_j \odot y_k$ où $j, k < i$ (autrement dit, y_j, y_k ont été calculés plus tôt dans le circuit), soit

2) $y_j \odot h$ où $h \in H$, soit $h \odot y_k$, soit

3) $x_{l_1} \odot x_{l_2}$ où $1 \leq l_1, l_2 \leq n$, soit $x_{l_1} \odot h$, soit $h \odot x_{l_2}$, soit

4) $x_{l_1} \odot y_k$, soit $y_j \odot x_{l_2}$.

Alors, y_i rend une fonction $Y_i(x_1, \ldots, x_n) : H^n \to H$ par récurrence sur i. On traite la dernière fonction $Y_m(x_1, \ldots, x_n)$ à titre du résultat du circuit.

Afin de *simuler le circuit C de façon codée* dans le groupe G on code d'abord l'entrée $(x_1, \ldots, x_n)$ par $(\overline{x_1}, \ldots, \overline{x_n})$ en employant un cryptoschéma homomorphe et ensuite on code respectivement, toutes les opérations de C en obtenant le circuit $\overline{C}$:

$\overline{y_i}$ = soit

1) $\overline{y_j} \otimes \overline{y_k}$, soit

2) $\overline{y_j} \otimes g$ où $g \in G$ tel que $f(g) = h$, soit $g \otimes \overline{y_k}$, soit

3) $\overline{x_{l_1}} \otimes \overline{x_{l_2}}$, soit $\overline{x_{l_1}} \otimes g$, soit $g \otimes \overline{x_{l_2}}$, soit

4) $\overline{x_{l_1}} \otimes \overline{y_k}$, soit $\overline{y_j} \otimes \overline{x_{l_2}}$.

De même par récurrence sur i on définit les fonctions $\overline{Y_i}(\overline{x_1}, \ldots, \overline{x_n}) : G^n \to G$. Puisque $F : G \to H$ est un homomorphisme on conclut que $f(\overline{Y_i}(\overline{x_1}, \ldots, \overline{x_n})) = Y_i(x_1, \ldots, x_n)$. Par conséquent, si on ait le circuit $\overline{C}$ dans le groupe G et par cela même la valeur du résultat du circuit $\overline{C}$ pour l'entrée $(\overline{x_1}, \ldots, \overline{x_n})$, alors il paraît difficile à établir la valeur du résultat du circuit C pour l'entrée $(x_1, \ldots, x_n)$ parce que à cet effet il faudrait calculer l'homomorphisme f. C'est pourquoi on dénomme $\overline{C}$ une *simulation codée* de C.

Enfin, nous sommes en état d'énoncer le problème de calcul avec données celées. Bob possède un circuit $C(x)$ dans le groupe H qui calcule la fonction $\Gamma(x)$ et ne veut pas découvrir C. D'autre part, Alice veut apprendre la valeur $\Gamma(x^{(0)})$ pour une valeur particulière $x^{(0)} \in H^n$ de l'entrée sans révéler $x^{(0)}$. Afin d'effectuer ces enjeux ils utilisent un cryptoschéma homomorphe et une simulation codée comme suit.

1) Alice transmet l'entrée codée $\overline{x^{(0)}} \in G^n$ à Bob;

2) Bob calcule $g^{(0)} = \overline{C}(\overline{x^{(0)}}) \in G$ et transmet $g^{(0)}$ à Alice;

3) Alice calcule $f(g^{(0)}) = \Gamma(x^{(0)}) \in H$ (en tenant compte que f est un homomorphisme).

Une situation duale est

Calculs avec fonctions celées.

Maintenant c'est Alice qui connaît un circuit C pour calculer la fonction Γ. Bob veut calculer $\Gamma(x^{(0)})$ sans découvrir $x^{(0)}$.

1) Alice transmet le circuit codé $\overline{C}$ à Bob;

2) Bob calcule $g^{(0)} = \overline{C}(\overline{x^{(0)}}) \in G$ et transmet $g^{(0)}$ à Alice;

3) Alice calcule $f(g^{(0)}) = \Gamma(x^{(0)}) \in H$ et transmet $\Gamma(x^{(0)})$ à Bob.

On a considéré un cryptoschéma homomorphe qui s'appelle

Cryptoschéma homomorphe de résidus quadratiques.

Soit $n = pq$ où p, q sont deux nombres premiers impairs secrets. Alors, le groupe d'éléments inversibles modulo n qu'on désigne par $Z_n^* \simeq Z_p^* \times Z_q^*$ est isomorphe au produit direct de deux groupes (quand même cet isomorphisme est celé!). Le sous-groupe de carrés $(Z_n^*)^2 \simeq (Z_p^*)^2 \times (Z_q^*)^2$ a l'indice égale à 4 dans le groupe Z_n^* parce que l'indice de $(Z_p^*)^2$ dans Z_p^* égale à 2, ceci du même porte sur Z_q^*. Le symbole de Jacobi

$x \to \left(\frac{x}{p}\right)\left(\frac{x}{q}\right) = J_n(x)$

est calculable en temps polynomial et on prend comme le groupe $G = \{x : J_n(x) = 1\}$ dans lequel le groupe des carrés $(Z_n^*)^2$ a l'indice égale à 2. Donc, l'épimorphisme $f : G \to H = Z_2$ dont le noyau égale à $\ker(f) = (Z_n^*)^2$, est celui de résidus quadratiques.

D'ailleurs, f procure un cryptoschéma homomorphe. Vraiment, Alice en utilisant sa clef secrète p, q, produit un non-carré $a \in Z_n^*$ quelconque, alors on prend $R = \{1, a\}$ et $f(1) = 1, f(a) = -1 \in Z_2$. Bob peut engendrer des éléments du $\ker(f)$: il choisit un élément $b \in Z_n^*$ de façon aléatoire, alors $b^2 \in \ker(f) = (Z_n^*)^2$.

Afin de décoder ce cryptoschéma, en d'autres termes de calculer f il faut reconnaître pour un élément $c \in Z_n^*$ si $c \in (Z_n^*)^2$? Pour ça Alice calcule les symboles de Legendre $\left(\frac{c}{p}\right)$, $\left(\frac{c}{q}\right)$ et on a

$c \in (Z_n^*)^2 \iff \left(\frac{c}{p}\right) = \left(\frac{c}{q}\right) = 1.$

Quant à la sécurité du cryptoschéma homomorphe de résidus quadratiques, c'est une question ouverte ainsi que pour n'importe quel cryptoschéma (quel que soit celui-ci homomorphe ou non). Nous savons déjà que le décodage du cryptoschéma de résidus quadratiques est équivalent à ce que reconnaître si un élément $b \in Z_n^*$ soit un carré? Une seule affirmation à ce sujet qui est connue, dit que si on exigait non seulement la reconnaissance si $b \in Z_n^*$ soit un carré mais encore à produire un élément $c \in Z_n^*$ tel que $c^2 = b$, alors on pourrait trouver les diviseurs p, q, c'est-à-dire, factoriser l'entier n, mais on croit que le problème de factorisation est dur. C'est une question ouverte si on peut réduire le problème de factorisation directement au problème de reconnaissance de l'ensemble de carrés $(Z_n^*)^2$ et par cela même démontrer la sécurité du cryptoschéma de résidus quadratiques.

En fait, on puisse considérer c tel que $c^2 = b$ en tant qu'une preuve d'appartenance de $b \in (Z_n^*)^2$ ou en d'autres termes, une *preuve du calcul correct de* f. Nous avons formalisé ce concept d'un cryptoschéma homomorphe y compris une preuve du calcul correct de f [99]. A cet effet il faut assurer une *suite exacte* d'homomorphismes de groupes

$$A \xrightarrow{P} G \xrightarrow{f} H \to \{1\}$$

où un élément $a \in A$ tel que $P(a) = g$ joue le rôle d'une preuve d'appartenance $g \in \ker(f)$ parce que $\mathrm{im}(P) = \ker(f)$. Dans le cryptoschéma de résidus quadratiques on a

$$Z_n^* \xrightarrow{P} G \xrightarrow{f} Z_2 \to \{1\}$$

où $P(a) = a^2$, $\mathrm{im}(P) = (Z_n^*)^2 = \ker(f)$ et le groupe G a été introduit cidessus.

On dit qu'on a *cassé un cryptoschéma homomorphe avec une preuve* lorsque on posséde un algorithme de la complexité polynomiale qui inverse l'application P, c'est-à-dire, pour tout élément $g \in G$ l'algorithme vérifie si $g \in \ker(f)$ et dans ce cas il rend un élément $a \in A$ tel que $P(a) = g$. Donc, si on cassait le cryptoschéma de résidus quadratiques *avec une preuve*, on pourrait factoriser l'entier n. Par ailleurs, dans la définition d'un cryptoschéma homomorphe avec une preuve on demande qu'Alice puisse inverser l'application P (à l'aide d'une clef secrète).

En fait, un groupe G peut être infini. Or, c'est pas grave pour le codage et le décodage parce qu'on peut écrire chaque élément codé $g \in G$ d'un message propre $h \in H$ sous la forme $g = rk$ où $r = f^{-1}(h) \in R$ et l'élément $k \in \ker(f)$ égale à un certain produit des génératrices de $\ker(f)$ données publiquement, autrement dit égale à un mot fini d'une longueur polynomiale. De plus, on a des algorithmes qui codent (et décodent à l'aide d'une clef secrète) en temps polynomial.

Théorème ([99]). Pour chaque groupe fini H on peut construire un cryptoschéma homomorphe sur H.

C'est une question ouverte s'il existe un cryptoschéma homomorphe sur H avec un groupe G *fini*.

Le cryptoschéma dans le théorème a un défaut que la construction du groupe G repose sur le produit libre de certains groupes; et des manipulations avec des mots du produit libre sont pénible. C'est pourquoi on a proposé dans [103] un autre cryptoschéma qui invoque le groupe modulaire $SL_2(\mathbf{Z})$.

Plus précisement, pour n'importe quel groupe fini H on a construit dans [103] un groupe $G_1 \subset SL_2(\mathbf{Z})$ donné publiquement par une famille B des génératrices et un assignement $B \to H$ qui engendre un epimorphisme. D'autre part, Alice connaît une autre famille A des génératrices de G_1 telle que pour un élément $g \in G_1$ quelconque Alice peut trouver une (unique) représentation de g comme un mot en matrices de A. Alors, afin de casser ce cryptoschéma Charle pourrait prévoir deux attaques possibles : soit trouver la famille A soit trouver une représentation de g comme un mot en matrices de B.

Dans [110] on a produit un *protocole d'accord de clefs* sur un groupe soluble quelconque, ce protocole généralise celui de Diffie-Hellman (qui est défini sur le groupe $\mathbf{Z}_p^*$). En outre, on a proposé la construction d'une famille de groupes de matrices sur un anneau fini commutatif. Cette construction rend un fondament pour des cryptoschémas homomorphes et pour des protocoles d'accord de clefs (dont casse se ramène au problème du conjugaison dans les groupes de la famille construite). La construction étend celle des groupes $\mathbf{Z}_n^*$ qu'on employe dans cryptographie classique.

On a considéré le problème de signature dans [118] et fabriqué un schéma très simple de signature qui repose sur la difficulté (en fait, NP-complétude) du problème d'homomorphisme de graphes, de groupes ou bien d'algèbres. Plus précisement, la clef secrète est un homomorphisme f, la clef publique est une paire u, v telle que $f(u) = v$. Afin de signer Alice choisit (de façon secrète) un isomorphisme g et en tant qu'une signature w tel que $g(v) = w$. Pour vérifier Bob exige (de façon aléatoire) de présenter soit g (et alors vérifie que $g(v) = w$) soit gf (et alors vérifie que $(gf)(u) = w$). L'idée est ce que

Bob apprend peu des informations sur la clef secrète f.

On propose dans [122] un schéma de signature en employant l'anneau A de matrices sur l'anneau de polynômes avec coefficients d'un anneau fini. La difficulté de casse de ce schéma repose sur le problème de conjugaison dans l'anneau A.

1.3 Simulations de calculs de façon codée

Comme nous avons déjà vu, le dernier théorème permet de simuler de façon codée un circuit dans le groupe H par un circuit dans un groupe G. Immédiatement, ce n'est pas très attirant puisque des calculs authentiques s'effectuent dans des anneaux qui comprennent deux opérations, alors que un groupe comprend une seule opération. C'est un objet de recherche à construire un cryptoschéma homomorphe sur un anneau.

Quand même on peut contourner cet obstacle grâce à la construction suivante dû à Barrington qui permet de simuler n'importe quel circuit booléien $B(x_1, \ldots, x_n) : \{0,1\}^n \to \{0,1\}$ dans un groupe H quelconque *irrésoluble*. A son tour, comme d'habitude, on peut simuler n'importe quel calcul par un circuit booléien.

Alors, on dit qu'un mot sous la forme de

$$M = h_1^{x_{i_1}} \cdots h_m^{x_{i_m}}$$

où $h_1, \ldots, h_m \in H, 1 \leq i_1, \ldots, i_m \leq m$ *simule* le circuit booléien B s'il existe un élément $1 \neq h \in H$ approprié tel que pour n'importe quelle entrée booléienne $(x_1, \ldots, x_n) \in \{0,1\}^n$ on a

$$B(x_1, \ldots, x_n) = 0 \iff M = 1,\, B(x_1, \ldots, x_n) = 1 \iff M = h$$

Alors, M est nommé une *h-simulation* du circuit B dans le groupe H.

Théorème (Barrington). Pour n'importe quel groupe H irrésoluble, un élément $1 \neq h \in H$ quelconque et chaque circuit booléien $B(x_1, \ldots, x_n)$ avec la *profondeur* d (en d'autres termes, la profondeur est la complexité parallèle), on peut construire une h-simulation de B dans H avec la complexité $m \leq \exp(O(d))$.

On note que dans le cas du groupe symétrique $H = S_5$ on a $m \leq 4^d$. D'habitude, on employe ce théorème à circuits booléiens avec la profondeur logarithmique $d \leq O(\log n)$.

Une combinaison de ce théorème avec notre résultat (voir la fin du chapître précédent) rend un mot sous la forme de

$$\overline{M} = g_1^{x_{i_1}} \cdots g_m^{x_{i_m}}$$

qui *simule de façon codée* le circuit booléien B. Ça signifie que $f(g_1) = h_1, \ldots, f(g_m) = h_m$ et pour n'importe quelle entrée booléienne $(x_1, \ldots, x_n) \in \{0,1\}^n$ on a

$$B(x_1, \ldots, x_n) = 0 \iff f(g_1^{x_{i_1}} \cdots g_m^{x_{i_m}}) = 1,$$

$$B(x_1, \ldots, x_n) = 1 \iff f(g_1^{x_{i_1}} \cdots g_m^{x_{i_m}}) = h$$

pour un certain h fixé et d'ailleurs, c'est difficile à calculer $B(x_1, \ldots, x_n)$ parce que à cet effet on aurait besoin de savoir calculer f.

Alors, on résume avec le corollaire suivant

Corollaire ([99]). On peut simuler de façon codée n'importe quel circuit booléien ayant la profondeur d dans le groupe G avec la complexité $\exp(O(d))$.

Par conséquent, on peut accomplir des calculs avec fonctions celées (voir le chapître précédent) dans le groupe G avec la complexité $\exp(O(d))$.

Une autre solution du problème de simulation de calculs de façon codée était possible si on aurait un cryptoschéma sur des anneaux $K \to R \to 0$ tel que la taille des éléments de K soit polynomiale. C'est une question ouverte. Pour n'importe quel anneau R fini commutatif on a produit dans [103] un anneau K fini commutatif dont éléments ont la taille exponentielle. Par ailleurs, on a démontré dans [103] que si la taille des éléments de K était polynomiale on ne devrait pas donner K avec sa base explicite, sinon Charle pourrait casser ce cryptoschéma sur les anneaux.

1.4 Cryptoschéma universel et puissance d'adversaires

C'est une question encore ouverte, si un adversaire qui possède plus grande puissance de calculs que l'autre, pourrait casser plus de cryptoschémas? On a répondu cette question dans [111] pour des adversaires qui peuvent utiliser un *bit de conseil* (rendu par un oracle). Plus précisement, pour n'importe quel k on a construit une fonction f

1) qu'on peut calculer en temps linéaire avec un bit de conseil;

2) il y a un adversaire qui peut inverser f en temps polynomial avec une certaine probabilité p;

3) aucun adversaire même avec usage d'un nombre logarithmique de bits de conseil ne peut inverser f en temps $O(n^k)$ avec la probabilité p.

Dans [113] on a résolu le vieux problème, s'il existe un cryptoschéma universel? C'est-à-dire, un cryptoschéma tel que si un adversaire pouvait le casser, on pourrait casser un cryptoschéma quelconque.

2 Complexité en calcul formel

Les problèmes de factorisation de polynômes, de résolution de systèmes d'équations (sur un corps *algébriquement clos*) ou ceux d'inégalités (sur un corps *réellement clos*) étaient explorés depuis longtemps. On connaît les algorithmes pour la factorisation dûs à Newton, Gauss, Kronecker, Hilbert, Hensel, Zassenhaus, Berlekamp, pour des systèmes d'équations dûs à Cayley, Kronecker, Macauley, Tarski, Seidenberg, Heintz et pour des systèmes d'inégalités dûs à Sturm, Tarski, Seidenberg, P.Cohen, Collins. Mais tous les algorithmes mentionnés ont la complexité très grande, notamment exponentielle pour la factorisation et au moins double-exponentielle pour la résolution des systèmes. On a construit des algorithmes avec la complexité polynomiale pour la factorisation ou proche à celle polynomiale pour la résolution des systèmes.

2.1 Factorisation de polynômes à plusieurs variables

A.Lenstra, H.Lenstra, L.Lovasz ont inventé un algorithme qui factorise (avec la complexité polynomiale) un polynôme $f \in \mathbf{Q}[X]$ univariable.

Dans [18, 20] (voir aussi [21, 25, 35]) on a construit un algorithme qui factorise (également avec la complexité polynomiale) un polynôme $f \in F[X_1, \ldots, X_n]$

à plusieurs variables sur un corps F engendré fini soit sur $\mathbf{Q}$ dans le cas de la caractéristique zéro soit sur un corps fini. En particulier, en tant que F on peut prendre n'importe quel corps algébrique.

2.2 Résolution d'un système algébrique et élimination de quantificateurs

Soient $f_1, \dots, f_k \in F[X_1, \dots, X_n]$ des polynômes et $\overline{F}$ désigne la clôture algébrique de F. D.Lazard avait produit un algorithme (avec la complexité polynomiale) qui trouve toutes les solutions du système $f_1 = \dots = f_k$ algébrique dans hypothèse que le système n'a qu'un nombre *fini* des solutions dans l'espace *projectif* $\mathbf{P}\overline{F}^n$.

Quand un système a un nombre infini de solutions il faut tout d'abord comprendre qu'est-ce que veut dire "résoudre le système"? Dans [19, 20] (voir aussi [21, 25, 35]) on résout le système en trouvant les *composantes irréductibles* V_i de la variété projective

$$V = \cup V_i = \{x \in \mathbf{P}\overline{F}^n : f_1(x) = \dots = f_k(x) = 0\}.$$

A son tour, chaque V_i est rendu au moyen de son corps $\overline{F}(V_i)$ des *fonctions rationnelles* sur V_i. Le langage des composantes irréductibles permet de répondre aux questions principales sur le système, par exemple, quelle est la dimension de la variété V ou bien celle de la variété affine $V \cap \overline{F}^n$? La complexité de l'algorithme de [19, 20] est majorée par d^{n^2} où les degrés $deg(f_i) \leq d$.

Une généralisation du problème de la résolubilité d'un système algébrique est celui de l'élimination de quantificateurs dans la théorie de corps *algébriquement clos*. Etant donnée une formule

$$\exists X_{1,1} \cdots \exists X_{1,s_1} \forall X_{2,1} \cdots \forall X_{2,s_2} \cdots \exists X_{a,1} \cdots \exists X_{a,s_a} \Phi(X_{1,1}, \dots, X_{a,s_a}, Y_1, \dots, Y_n)$$

où Φ est une combinaison booléienne de formules du type $(f = 0)$ où $f \in F[X_{1,1}, \dots, X_{a,s_a}, Y_1, \dots, Y_n]$ est un polynôme. Le théorème dû à Tarski-Seidenberg dit que la formule ci-dessus est équivalente à une formule $\Psi(Y_1, \dots, Y_n)$ appropriée sans quantificateurs. Le problème de la résolubilité d'un système correspond au cas quand le nombre des blocs des quantificateurs $a = 1$ égale à 1 et le nombre $n = 0$ des variables libres égale à 0.

Dans [22, 24] (voir aussi [25, 35]) on a produit un algorithme qui élimine des quantificateurs et rend une certaine formule $\Psi(Y_1, \ldots, Y_n)$ avec la complexité d^{s^a} où $s = s_1 + \cdots + s_a + n$ est le nombre de toutes les variables dans la formule. La construction des formules due à Fischer-Rabin montre que cette borne est proche à celle exacte. Alors, ça signifie que le nombre a est le paramètre principal de la complexité de la formule. Auparavant on avait connu des algorithmes pour l'élimination avec la complexité d^{s^s}. Donc, notre borne est meilleure lorsque a est plus petit essentiellement que s. Il convient de noter que dans les propositions habituelles des mathématiques a est assez petit.

2.3 Résolution d'un système d'inégalités polynomiales et décidabilité dans algèbre de Tarski

Etant donnés des polynômes $f_1, \ldots, f_k \in \mathbf{Q}[X_1, \ldots, X_n]$. Dans [27] (voir aussi [23, 25, 35]) on a produit un algorithme qui résout le système d'inégalités $f_1 \geq 0, \ldots, f_k \geq 0$ et rend au moins un point (si le système est résoluble) dans chaque composante connexe de l'ensemble semi-algébrique

$$M = \{x \in \mathbf{R}^n : f_1 \geq 0, \ldots, f_k \geq 0\}.$$

La complexité de l'algorithme est d^{n^2}. Auparavant on avait connu des algorithmes pour la résolubilité des systèmes d'inégalités avec la complexité d^{2^n}.

Le problème qui généralise celui de la résolubilité d'un système d'inégalités est le problème de la décidabilité dans la théorie de corps *réellement clos* (ou bref, *l'algèbre de Tarski*). De même que dans le cas de corps algébriquement clos, étant donnée une formule

$$\exists X_{1,1} \cdots \exists X_{1,s_1} \forall X_{2,1} \cdots \forall X_{2,s_2} \cdots \exists X_{a,1} \cdots \exists X_{a,s_a} \Phi(X_{1,1}, \ldots, X_{a,s_a})$$

où maintenant Φ est une combinaison booléienne de formules du type $(f \geq 0)$ où $f \in \mathbf{Q}[X_{1,1}, \ldots, X_{a,s_a}]$ est un polynôme. Dans [28] (voir aussi [25, 35]) j'ai produit un algorithme qui décide si la formule donnée soit vraie avec la complexité d^{s^a}. Comme dans le cas de corps algébriquement clos la construction de Fischer-Rabin montre que cette borne est proche à celle exacte. Auparavant on avait connu des algorithmes pour la décidabilité avec la complexité d^{s^s}.

Dans [44, 45] (voir aussi [38, 42, 43]) on a produit un algorithme qui trouve les composantes connexes d'un ensemble semi-algébrique M avec la complexité $d^{n^{O(1)}}$.

On a construit dans [107] un algorithme qui résout un système d'équations *quadratiques* $f_1 = \cdots = f_k = 0$ où $f_1, \ldots, f_k \in \mathbf{Z}[X_1, \ldots, X_n]$ en temps $n^{O(k)}$. De plus, l'algorithme produit au moins un point dans chaque composante connexe de l'ensemble semi-algébrique $\{f_1 = \cdots = f_k = 0\} \subset \mathbf{R}^n$. Alors, on obtient une façon de dualité entre le nombre k d'équations et la dimension n par rapport à l'algorithme de [27] qui rend la complexité $k^{O(n)}$ pour le même problème.

2.4 Calcul formel avec polynômes creux

Calcul formel habituel traite des polynômes dans l'écriture *dense*, autrement dit on écrit tous les monômes d'un polynôme jusqu'aux ceux d'un certain degré. Lorsqu'un polynôme posséde peu de monômes, alors c'est raisonnable à énumérer seulement ses monômes dont coefficients ne sont pas zéro. Une telle écriture de polynômes s'appelle *creuse*. On désigne par t_f le nombre des monômes dans un polynôme $f \in F[X_1, \ldots, X_n]$.

On considère le problème *d'interpolation* d'un polynôme $f \in F[X_1, \ldots, X_n]$ creux en supposant que seul nombre t_f est donné, avec cela son degré est inconnu a priori. On tient que f est donné au moyen d'une *boîte noire* pour rendre ces valeurs $f(x)$.

Dans le premier résultat dans ce domaine on a fabriqué [26] un algorithme qui interpole le déterminant $f = \det(x_{i,j}), 1 \leq i, j \leq n$ avec la complexité polynomiale en t_f, n. Ensuite Ben-Or, Tiwari ont étendu cet algorithme pour un polynôme f arbitraire sur n'importe quel corps F de la *caractéristique zéro*. Dans [32] on a construit un algorithme d'interpolation de polynômes creux sur un corps F *fini*. Dans [41] on a mis ces algorithmes dans le cadre plus général lorsque f est une somme de t_f fonctions propres d'un certain opérateur L linéaire. Dans ce cadre l'algorithme de [41] permet de trouver pour une fonction quelconque sa décomposition (creuse) soit monomiale soit de Fourier etc. Un outil crucial pour cet algorithme est un critère introduit dans [41] de ce qu'une fonction est creuse en termes du wronskien par rapport de l'opérateur L.

De façon plus générale on considère des fonctions *rationnelles creuses*, celles-ci signifient qu'une fonction est une fraction (peut-être réductible) de deux polynômes creux. Dans [40, 51] on a construit un algorithme qui interpole des fonctions rationnelles creuses. De nouveau, cet algorithme emploie le critère engageant le wronskien. Dans [49] cette construction a été étendue sur des fonctions algébriques réelles.

Dans [47] on a démontré que le problème de la divisibilité d'une paire de polynômes creux appartient à la classe de complexité **coNP** dans l'hypothèse généralisée de Riemann. C'est une question ouverte si on puisse tester la divisibilité d'une paire de polynômes creux avec la complexité polynomiale? H.Lenstra a fabriqué un algorithme qui teste la divisibilité d'un polynôme *creux* sur un polynôme d'un *petit degré* avec la complexité polynomiale.

Dans [48] on a donné une borne supérieure sur le nombre N des zéros d'un polynôme $f \in GF(q)[X_1, \ldots, X_n]$ sur un corps fini en termes de t_f et en reposant sur cette borne on a fabriqué un algorithme qui approxime N en temps polynomial. Il convient de noter que le problème de calculer N est **NP**-dur.

Dans [46] on a fait intervenir une autre définition d'une fonction rationnelle *creuse* par rapport à la décomposition en *fractions partielles* et on a produit un algorithme avec la complexité polynomiale afin de l'interpoler.

Il arrive parfois en calcul formel qu'un polynôme $f \in F[X_1, \ldots, X_n]$ n'est pas creux lui-même, mais celui-ci devient creux après une transformations linéaire appropriée des coordonnées, c'est-à-dire $f(AX + C)$ est creux pour une matrice A et un vecteur C. Dans [50] on a construit un algorithme qui rend une telle transformation si celle-ci existe, autrement l'algorithme rend la réponse qu'aucune transformation n'existe. Dans [62, 63] on a fabriqué un algorithme pour ce problème avec une meilleure complexité.

Afin de développer calcul formel pour des polynômes *creux* il faudrait avoir des algoritmes pour les problèmes de base de calculs: diviser avec reste, calculer le PGCD, factoriser des polynômes, résoudre des systèmes d'équations algébriques etc. en supposant que tant l'entrée que la sortie d'un problème donné sont creuses.

3 Construction de stratification de Thom-Whitney-a universelle et théorème de type de Sard pour variétés singulières

Soit $F : K^n \to K^l$ une application polynomiale ayant sa valeur critique 0 isolée. On étudie des stratifications $\{S_i\}_i$ de l'ensemble $Sing(F) = \cup_i S_i \subset K^n$ des points critiques de F dans $F^{-1}(0)$ telles que $\{S_i\}_i$ vérifie les conditions de Thom et de Whitney-a.

On dit qu'un ensemble $S \subset Sing(F)$ est *universel* si pour n'importe quelle stratification $\{S'_j\}_j$ il existe (et alors unique) un strate S'_j tel que l'intersection $S \cap S'_j$ est ouverte et dense dans les deux S et S'_j. Une stratification $\{S_i\}_i$ s'appele *universelle* si chaque son strate S_i est universel.

On dit qu'un ensemble S est *régulier de type de Gauss* si on peut étendre l'application de Gauss $x \to T_x(S)$ des points réguliers $x \in S$ sur tous les points de S de façon continue (et avec cela unique). On considère des stratifications $\{S_i\}_i$ avec tout S_i étant régulier de type de Gauss.

On peut employer la construction due à Glaeser à n'importe quel faisceau $R \subset M \times W$ des espaces vectoriels où M étant une variété et W un espace vectoriel. On note par $R^{(1)}$ le faisceau dont fibre $(R^{(1)})_x$ (pour $x \in M$ quelconque) est l'enveloppe linéaire du fibre $(\overline{R})_x$. De façon pareile on produit les faisceaux $R^{(i)}$, $i \geq 1$ jusqu'à $R^{(m)}$ tel que $R^{(m)} = R^{(m+1)}$ et on désigne $Gl(R) := R^{(m)}$. On appele *m l'indice de stabilisation*. On a $m \leq 2 \cdot \dim(W)$.

On considère le sous-faisceau $\mathcal{T} := \{(x, dF(x))\}_x$ du faisceau $T^*(K^n) = K^n \times (K^n)^*$ cotangentiel et y applique la construction de Glaeser en obtenant le faisceau $G := G_F := Gl(\mathcal{T})$.

On note un ensemble constructible $\mathcal{G}_r := \{x \in Sing(F) : \dim(G_x) = r\}$. On dit qu'une composante irréductible $\mathcal{G} \subset \mathcal{G}_r$ est *Lagrangienne* si G_x est le complement orthogonal à l'espace tangentiel $T_x(\mathcal{G})$ pour n'importe quel point régulier $x \in \mathcal{G}$.

Proposition [120]. Si chaque composante irréductible de $\{\mathcal{G}_r\}_r$ est Lagrangienne alors $\{\mathcal{G}_r\}_r$ assure une stratification universelle.

Théorème [120]. S'il existe une stratification universelle alors chaque composante irréductible de $\{\mathcal{G}_r\}_r$ est Lagrangienne.

La démonstration du Théorème est beaucoup plus difficile que celle de la Proposition et utilise notre version du théorème de type de Sard pour variétés singulières [120].

Par ailleurs, on a fabriqué dans [120] un exemple d'application $F : K^5 \to K$ où $F = ax^2+2b^2xy+cy^2$ n'admettant aucune stratification universelle. En reposant sur [22] on déduit que la complexité de construction de G (et par cela même d'une stratification universelle si celle-ci existe) est bornée par $d^{2^{O(m)}} \leq d^{2^{O(n)}}$. On a produit un exemple d'application F avec l'indice de stabilisation m croissant de façon linéaire en n. C'est une question ouverte si la complexité d'une stratification universelle puisse augmenter comme (double-exponentiel) $d^{2^{O(n)}}$?

4 Complexité dans la robotique

On étudie un problème de la robotique du traçage d'un chemin entre deux points fixés en évitant d'un obstacle donné. Comme un obstacle on considère un ensemble semi-algébrique représenté par un système d'inégalités polynomiales.

S'il s'agit d'un chemin sur le plan, alors on peut tracer un chemin optimal en temps polynomial (en taille booléienne du système d'inégalités). De plus, on peut considérer l'optimalité à certains sens différents. On a construit un algorithme [77] qui rend le plus court chemin qui correspond la classe d'homotopie donnée de l'espace libre, le dernier signifie le complément de l'obstacle. Avec cela l'algorithme représente l'ensemble des plus courts chemins qui correspondent toutes les classes d'homotopies (constituant le groupe libre) en forme d'un graphe convenable.

Un autre algorithme [74] produit en temps polynomial le chemin (entre deux points) qui est linéaire par morceaux en possédant le nombre minimal possible de chaînons dans une classe d'homotopie donnée. A la fois ce chemin produit est optimal par rapport à la somme des angles de rotation.

Dans les deux algorithmes mentionnés le rôle important joue un outil développé dans [74, 77] pour des calculs efficaces dans le groupe fondamental.

Les problèmes pareils à tracer un chemin optimal dans l'espace 3-dimensionnel sont **NP**-complets déjà pour un obstacle d'autant simple comme étant une réunion de polyèdres convexes. Donc, c'est une question importante à distinguer des familles d'obstacles pour lesquelles on pourrait tracer un chemin optimal en temps polynomial. Dans [94] on a construit un algorithme qui rend le plus court chemin lorsqu'un obstacle est une réunion de droites, d'ailleurs une classe d'homotopie d'un chemin est donnée. C'est une question ouverte à tracer le plus court chemin en temps polynomial si aucune classe d'homotopie n'est donnée.

On a fait intervenir dans [94] une autre famille d'obstacles, notamment celles a-séparables pour lesquelles on peut tracer une approximation du plus court chemin en temps polynomial de $1/a$.

5 Complexité de calculs avec équations différentielles

Le *calcul formel différentiel*, en d'autres termes des calculs avec équations différentielles, n'est pas si bien développé comme le calcul formel, autrement dit des calculs avec polynômes (voir le chapître 2). Pour les problèmes de base du calcul formel différentiel leurs bornes de la complexité ne sont pas connues, et ça se fait l'objet de recherche.

5.1 Equations différentielles ordinaires

La plus générale pose du problème de la résolubilité de systèmes d'équations différentielles est l'élimination de quantificateurs dans la théorie de corps *différentiellement clos*. Ceux-ci jouent le rôle semblable pour des équations différentielles au celui de corps algébriquement clos pour des équations algébriques. C'était Seidenberg qui a offert la première fois un algorithme pour l'élimination de quantificateurs, mais sa complexité a été trop grande (celle-ci a augmenté comme une itération des fonctions exponentielles et avec cela le nombre des itérations dépend du nombre de fonctions inconnues, telles bornes sont dénommées *non-élémentaires*). Dans [29] j'ai proposé un procédé pour l'élimination de quantificateurs dans la théorie de corps différentiellement clos *ordinaires* dont complexité augmente comme une itération de trois fonctions exponentielles. C'est une question ouverte à construire un procédé avec la complexité élémentaire pour des corps différentiellement clos *partiels*.

Une généralisation de l'algèbre de polynômes est l'algèbre (non-commutative) D d'opérateurs linéaires différentiels $\sum a_i(X)\frac{d^i}{dX^i}$ avec les coefficients $a_i(X) \in \mathbf{C}(X)$ rationnels. Le problème de la factorisation d'un opérateur généralise celui pour des polynômes. Quant au problème de la factorisation on connaît les algorithmes dûs à Beke, Schlesinger, Ore, F.Schwarz, mais la complexité de ces algorithmes est au moins triple-exponentielle. Dans [30, 31] j'ai produit un algorithme pour la factorisation en D avec la complexité double-exponentielle. Une extension de ce problème est celui de la factorisation d'un système linéaire différentiel du premier ordre $u^{'} = Au$ où A est une matrice avec des coefficients rationnels. Pour ce problème dans [37] j'ai construit un algorithme également avec la complexité double-exponentielle.

En outre dans [31] j'ai fabriqué un algorithme avec la complexité polynomiale qui calcule le plus grand commun diviseur *gauche* d'une famille d'opérateurs. D'ailleurs, dans [57] j'ai proposé un algorithme avec la complexité polynomiale afin de résoudre un système linéaire différentiel à plusieurs inconnues.

Dans [33] on a produit un procédé afin de résoudre un système d'équations non-linéaires différentielles ordinaires en séries avec *exposants réels*. Celles-ci étendent des séries de Newton-Puiseux qui ont des exposants rationnels et qui constituent un corps algébriquement clos. C'est une question ouverte, quelles classes d'équations différentielles possédent des solutions en séries avec exposants réels?

5.2 Equations différentielles partielles

Le problème de la factorisation d'un opérateur linéaire différentiel *partiel* est encore ouvert. On a produit dans [105] un algorithme qui factorise un opérateur lorsque son *symbole* est séparable à l'aide d'un procédé nommé la *descente de Hensel*. En l'appliquant on a effectué dans [105] l'analyse complète de la factorisation d'opérateurs du deuxième ordre.

Dans [108] on fait intervenir la décomposition de Loewy d'un D-module. Auparavant, Loewy a construit la décomposition d'un opérateur linéaire différentiel ordinaire. Dans le cas d'un D-module holonome on a produit dans [108] un algorithme qui trouve sa décomposition de Loewy. En reposant sur [108] on a donné dans [119] une description complète de types possibles de décompositions de Loewy pour des opérateurs linéaires différentiels partiels de l'ordre 3 (pour l'ordre 2 c'est la conséquence de [105]).

Dans [39] j'ai étudié le problème de la résolution d'un système linéaire sur l'algèbre

$$D_n = F(X_1, \ldots, X_n)[\frac{\partial}{\partial X_1}, \ldots, \frac{\partial}{\partial X_n}]$$

des opérateurs différentiels partiels linéaires (ou bien sur l'algèbre de Weil

$$W_n = F[X_1, \ldots, X_n, \frac{\partial}{\partial X_1}, \ldots, \frac{\partial}{\partial X_n}],$$

respectivement). Dans [39] j'ai produit un algorithme qui résout ce problème avec la complexité d^{2^n} où d majore les ordres des coefficients du système sur D_n (ou bien les ordres et les degrés de ceux sur W_n, respectivement). Un problème particulier de la résolution en question est celui de l'appartenance à un idéal (gauche) de l'algèbre D_n ou bien W_n, autrement dit, étant donnés $f, f_1, \ldots, f_k \in D_n$ il faut trouver $g_1, \ldots g_k \in D_n$ tels que $f = g_1 f_1 + \cdots + g_k f_k$ ou établir que f n'appartient pas à l'idéal gauche engendré par $f_1, \ldots, f_k$ (on pose le même problème pour W_n). Dans [39] j'ai montré que la borne d^{2^n} est exacte en fabriquant en exemple d'un idéal qui généralise pour les algèbres D_n et W_n la construction de Mayr-Meyer destinńee à l'algèbre des polynômes. On peut énoncer la conjecture que dans le cas de *l'idéal unité*, c'est-à-dire $1 = g_1 f_1 + \cdots + g_k f_k$ la borne doit être meilleure considérablement, notamment d^n. Ça serait une généralisation pour les algèbres D_n et W_n du *Nullstellensatz efficace* dû à Brownawell-Heintz-Galligo-Kollar.

J'ai démontré dans [104] une inégalité faible de Bézout pour un D-module L sur l'algèbre D_n. Plus précisement, si L est engendré par des opérateurs de l'ordre inférieur à d alors le coefficient *dominant* du polynôme de Hilbert-Kolchin p_L de L est plus petit que $d^{4^{n-t}}$ où t désigne le degré de p_L. On peut poser la conjecture que le coefficient dominant doit être majoré par une fonction exponentielle (comme dans le cas classique d'algèbre des polynômes).

Dans [115] on a prouvé la borne supérieure (et exacte) double exponentielle $d^{2^{O(n)}}$ sur le degré et la complexité de construction d'une base de Janet d'un D-module (pour l'algèbre D_n ou bien W_n). Auparavant, une borne pareille a été connue pour une base de Groebner pour un module sur l'algèbre des polynômes. Le dernier utilise essentiellement certains concepts de géométrie algébrique et c'est pourquoi pour la démonstration dans le cas différentiel il faudrait inventer une d'autre méthode.

Le *degré de liberté* d'un système d'équations différentielles partielles *extérieures* est décrit par son *genre* d'après E.Cartan. D'un autre côté la dimension de l'espace des solutions du système s'accroît comme le polynôme de Hilbert-Kolchin. Le degré de ce polynôme juste égale au genre du système. Un procédé habituel pour calculer ce polynôme est à rendre la base de Janet, mais sa complexité est double-exponentielle. Dans [34, 36] j'ai fabriqué un algorithme dont complexité est exponentielle qui calcule le genre. C'est un objet de recherche à construire un algorithme avec la complexité exponentielle qui calcule le polynôme de Hilbert-Kolchin, au moins son coefficient dominant (étant le plus important).

Soit K un corps différentiellement clos et d_1, d_2 deux dérivés sur K. On fait intervenir dans [116] $K[d_1, d_2]$-module $K[d_1, d_2][\{G^{(s)}\}_{s\in Q}]$ tel que

$$dG^{(s)} = \sum_{1\leq i\leq k} (df_i)G^{(s+s_i)}$$

pour $f_i \in K$ et nombres rationels $s_i > 0$ appropriés où $d = d_1, d_2$. On introduit un analogue de séries de Newton-Puiseux

$$\sum_{j\geq 0} g_j G^{(s-j/q)}$$

($q > 0$ étant un nombre entier) et démontre dans [116] que n'importe quel opérateur $L \in K[d_1, d_2]$ a une telle série comme sa solution.

En reposant sur ce rśultat j'ai fabriqué un algorithme qui trouve tous les diviseurs du premier ordre (droits ou gauches) de L. En tant qu'une d'autres conséquence de la méthode de [116] on démontre dans [123] que n'importe quel opérateur $L \in K[d_1, d_2]$ (ou bien un D-module) séparable n'a qu'un nombre fini de D-modules maximales qui le contient (on appele tels D-modules comme *facteurs maximales*). Cela donne une déscription d'opérateurs dans $K[d_1, d_2]$ du troisième ordre avec un nombre fini de facteurs maximales (pour des opérateurs du deuxième ordre une telle déscription a été établi dans [105]).

6 Complexité quantique et probabiliste

Après que Shor a inventé son algorithme *quantique* fameux pour factoriser d'entiers avec la complexité polynomiale, la question s'est posée, quelle est la puissance de machines quantiques en général? Dans [65, 73] j'ai montré que des machines quantiques, étant donné un groupe abelien G qui agit sur un ensemble M, ont pu trouver avec la complexité polynomiale le sous-groupe de G stabilisant, c'est-à-dire, tous les éléments du sous-groupe gardent un point $m \in M$ fixé. Comme conséquence ça donne un algorithme [65, 73] qui trouve le groupe de tous les déplacements $(c_1, \dots, c_n)$ qui gardent un polynôme $f \in F[X_1, \ldots, X_n]$ donné, c'est-à-dire, $f(X_1+c_1, \ldots, X_n+c_n) = f$ sur un corps *fini* F.

Lorsqu'on considère des algorithmes *probabilistes* qui forment une sous-classe d'algorithmes quantiques, on n'a réussi construire un algorithme probabiliste [65, 73] que sur un corps $F = GF(p)$ pour un nombre *premier* p dont la complexité étant polynomiale par ailleurs, a été pire que celle de l'algorithme quantique mentionné. Il convient de noter qu'un algorithme *déterministe* avec la complexité polynomiale aussi pour le problème de déplacements en question n'est connu que dans le cas de corps F de la caractéristique zéro [65, 73].

Dans [109] on a fabriqué un modèle de calculs quantique optique qui permet d'accélérer la programmation dynamique comparativement des calculs déterministes.

7 Approximations et complexité

Il y a un rapport intuitif d'approximations et leurs complexités: mieux approximation on désire atteindre, plus du temps cela exigera. C'est le théorème de Liouville qui offre un tel rapport pour approximations de nombres algébriques au moyen de nombres rationnels. A savoir, ce théorème donne une borne inférieure sur approximations en termes de la complexité des équations algébriques auxquelles satisfont des nombres algébriques (et rationnels).

On pose la question, si un rapport pareil soit possible pour des solutions d'équations différentielles (ordinaires)? Un contre-exemple est connu qui montre que pour des solutions d'équations différentielles non-linéaires du deuxième ordre aucun tel rapport n'est possible.

Donc, on peut considérer des solutions d'équations soit linéaires soit du premier ordre. D'ailleurs, on étudie approximations, premièrement, dans un voisinage de l'infinité, et deuxièmement, dans un intervalle borné. Alors, j'ai prouvé quatre résultats à ce sujet.

Dans le premier résultat [54, 56] on considère une fonction f comme une composition de solutions d'équations différentielles linéaires. Plus précisément, par récurrence sur i on produit une série des corps F_i (pour la base on admet $F_0 = \mathbf{R}(X)$). Puis, on suppose que chaque solution d'une équation différentielle avec ses coefficients du corps F_i appartient à F_{i+1}, et enfin le corps F_{i+1} est engendré par toutes telles solutions.

Le premier résultat [54] dit que n'importe quelle fonction f du corps F_i est inférieure à $\exp_i$ et supérieure à $(\exp_i)^{-1}$, où $\exp_i$ désigne l'itération i fois de la fonction exponentielle. Plus précisément, la dernière affirmation est remplie partout dans la droite réelle sauf un ensemble d'une mesure finie.

En outre, le nombre des zéros de la fonction f dans un intervalle $[-x, x]$ est petit que $\exp_{i-1}(x)$. De plus, toutes les bornes mentionnées sont exactes.

Le deuxième résultat [55, 56] traite le comportement des compositions des solutions d'équations du premier ordre dans un voisinage de l'infinité. Plus précisément, on définit par récurrence sur i la série des corps P_i (pour la base on admet de nouveau $P_0 = \mathbf{R}(X)$) des fonctions de Pfaff. Si une fonction f satisfait une équation différentielle $f' = p(f)$ où un polynôme $p(Z) \in P_i[Z]$ a ses coefficients dans le corps P_i, alors la fonction f appartient à P_{i+1}. Enfin le corps P_{i+1} est engendré par toutes telles fonctions f.

Le deuxième résultat [55] dit que n'importe quelle fonction f du corps P_i remplit les bornes

$$(\exp_i)^{-1} < |f| < \exp_i$$

dans un certain voisinage de l'infinité.

Alors, dans les deux résultats susmentionnés [56] la complexité parallèle i détermine le taux d'approximations possibles dans un voisinage de l'infinité.

Maintenant, on considère des fonctions f qui sont définies dans un intervalle I. D'abord on suppose qu'une fonction f_1 satisfait une équation différentielle linéaire $L_1 f_1 = 0$ où $L_1 = \sum a_j \frac{d^j}{dX^j}$ est un opérateur avec les coefficients polynomials $a_j \in \mathbf{Z}[X]$. Soient une fonction f_2 et une équation $L_2 f_2 = 0$ similaires. Le troisième résultat [92] fournit une borne inférieure sur la norme $||f_1 - f_2||_I = \max_{x \in I} |(f_1 - f_2)(x)|$ en termes des complexités des opérateurs L_1, L_2.

Ce serait intéressant à établir une borne inférieure sur la norme $||f_1 - f_2||_I$ pour des fonctions f_1, f_2 qui sont des compositions de solutions d'équations différentielles linéaires avec des coefficients polynomials, c'est-à-dire, $f_1, f_2 \in F_i$. A son tour, pour ça il faudrait obtenir une borne inférieure sur le wronskien d'éléments de F_i.

Finalement, dans le quatrième résultat [96] on étudie un sous-anneau M_i de la classe P_i des fonctions de Pfaff. Notamment, par récurrence sur i (pour la base on admet $M_0 = P_0 = \mathbf{R}(X)$) on définit M_{i+1} comme un anneau engendré par toutes les intégrales d'éléments de M_i, autrement dit, par toutes les fonctions f telles que $f' \in M_i$. Le quatrième résultat donne une borne inférieure sur la norme $||f||_I$ pour $f \in M_i$ en termes de la complexité d'une construction de f.

C'est une question ouverte à établir une borne inférieure sur $||f||_I$ pour des fonctions $f \in P_i$ de Pfaff.

Un autre objet de recherche est à établir des rapports entre approximation et complexité pour des fonctions à plusieurs variables.

8 Bornes inférieures sur la complexité

L'obtention de bornes inférieures est le défi le plus difficile de la théorie de complexité. Le plus célèbre d'eux est le problème **P-NP**. Jusqu'à présent aucune borne inférieure au-delà de polynomiale n'est établie. Le mieux qu'on puisse attendre actuellement, c'est d'obtenir des bornes inférieures pour des modèles de calcul modiques.

8.1 Arbres de calculs et la courbure

Soit F un corps. Un *arbre de tests* sur F est un modèle de calculs qui pour n'importe quelle entrée $x \in F^n$ peut se ramifier dans chaque son pas selon ce que si $f(x) = 0$ ou $f(x) \neq 0$ pour un polynôme $f \in F[X_1, \dots, X_n]$ approprié (un tel polynôme s'appelle un polynôme de test). On suppose que le degré $\deg f$ est petit suffisamment. Alors, le calcul pour l'entrée x marche le long d'un des chemins de l'arbre de tests et le bout de chaque chemin est muni par une des réponses "oui" ou "non". On dit que x est acceptée par l'arbre si la réponse est "oui", sinon x est rejetée. Toutes les entrées $x \in F^n$ qui sont acceptées constituent l'ensemble S reconnu par l'arbre, évidemment, l'ensemble S est constructible. Lorsque le corps $F = \mathbf{R}$ est réel, on considère ramifications en trois voies selon le signe $f(x)$. Dans ce cas l'ensemble S reconnu est semi-algébrique.

Un renforcement de ce concept est un *arbre de calculs*. La différence avec un arbre de tests est ce qu'un arbre de calculs peut calculer lui-même des polynômes de test. Par conséquent, le degré d'un polynôme de test puisse s'accroître exponentiellement avec la complexité de l'arbre de calculs. Donc, l'arbre de calculs est un modèle assez puissant et en particulier, une de paraphrases du problème **P-NP** est ce que s'il existe un arbre de calculs de la complexité polynomiale qui reconnaît l'ensemble des entiers $\{1, \ldots, 2^n\}$ sur le corps complexe **C**? C'est pourquoi, on s'intéresse à bornes inférieures sur la complexité des arbres de calculs.

Yao, Montana-Morais-Pardo ont démontré une telle borne inférieure pour la reconnaissance d'un ensemble S en forme de $\log(\sum_i b_i(S))$ où $b_i(S)$ désigne le i-ième nombre de Betti (c'est-à-dire, le rang du i-ième groupe d'homologie). Il en résulte des bornes inférieures pour nombreux problèmes algébriques et combinatoires. Par exemple, la borne inférieure quadratique n^2 pour le problème du *sac-à-dos* (ça signifie, s'il existe une sous-somme $\sum_{j \in J} a_j = 1$ égale 1, $J \subset \{1, \ldots, n\}$ dans un ensemble $\{a_1, \ldots, a_n\}$ donné), en outre la borne inférieure $n \log n$ pour le problème de la *coïncidence* d'ensembles (ça signifie, si deux ensembles $\{a_1, \ldots, a_n\}$ et $\{c_1, \ldots, c_n\}$ coïncide, et la même borne $n \log n$ pour le problème de la *distinction* d'ensemble (ça signifie, si tous les éléments d'un ensemble $\{a_1, \ldots, a_n\}$ sont différents deux-à-deux). On observe que deux dernières bornes sont exactes.

Quand même cette borne inférieure n'est pas utilisable pour des ensembles S topologiquement trivials. en particulier, lorsque S est un polyèdre, comme par exemple dans le problème **NP**-complet du *commis voyageur*. Pour ça on a développé une méthode en engageant les courbures principales [58, 61] qui implique la borne inférieure sur la complexité de la reconnaissance d'un polyèdre S, étant logarithmique du nombre des facettes de toutes les dimensions de S. Dans [71] cette borne a été généralisée pour des arbres de calculs.

8.2 Arbres probabilistes de calculs et singularités

Un arbre probabiliste de tests (ou de calculs, respectivement) admet des ramifications aléatoires, outre des opérations qui paraissent dans des arbres de tests (ou de calculs, respectivement). On définit qu'une entrée x est acceptée (ou rejetée, respectivement) par un arbre si la probabilité de ce que x est acceptée (ou rejetée, respectivement) est supérieure à $2/3$. Dans cette définition on puisse remplacer $2/3$ par n'importe quelle constante entre $1/2$ et 1.

On a construit un arbre probabiliste de calculs qui résout le problème de la *coïncidence* d'ensembles avec la complexité linéaire n. Par cela même un arbre probabiliste de calculs est un modèle plus puissant que celui déterministe pour lequel une borne inférieure $n \log n$ est établie (voir le sous-chapître précédent). De plus, toutes les méthodes pour obtention les bornes inférieures sur la complexité des arbres déterministes (mentionnées dans le sous-chapître précédent) ratent pour ceux probabilistes parce qu'elles engagent les caractères soit topologiques (les groupes d'homologie) soit différentielle-géométriques (les courbures principales). D'autre part ces caractères ne sont pas invariants par rapport à changements d'un ensemble à une petite mesure près, ce qui se passe aux arbres probabilistes.

C'est pourquoi on a développé une autre méthode afin de démontrer des bornes inférieures sur la complexité des arbres probabilistes sur le corps réel $\mathbf{R}$. Cette méthode repose sur des *points de triche* soi-disants, dont coordonnées sont infinitesimales. Elle entraîne la borne inférieure sur la complexité pour des arbres probabilistes de tests [66, 69] qui reconnaissent soit des arrangements (c'est-à-dire, des réunions d'hyperplans) soit des polyèdres. Dans les deux cas la borne est logarithmique du nombre des facettes de toutes les dimensions. En particulier il suit de là la borne inférieure quadratique n^2 pour le problème du *sac-à-dos* et la borne $n \log n$ pour le problème de la *distinction* pour des arbres probabilistes de tests.

En engageant le concept de la *complexité limite* de Strassen, on a généralisé la borne inférieure quadratique n^2 pour le problème du *sac-à-dos* pour des arbres probabilistes de calculs [71]. En estimant le degré de l'application du gradient on a établi [76] la borne inférieure $n \log n$ pour le problème de la *distinction* pour des arbres probabilistes de calculs. En fait, dans [76] la borne est démontrée qui est logarithmique du nombre des facettes de toutes les dimensions pour reconnaissance soit d'un arrangement soit d'un polyèdre. Cela signifie que la complexité probabiliste de la reconnaissance d'un ensemble dépend de la géométrie de cet ensemble. Quand on reconnaît une réunion de sous-espaces de codimensions supérieures à 1 (plutôt que hyperplans), la borne de [76] ne marche plus comme le montre la borne supérieure linéaire susmentionnée pour le problème de la *coïncidence* d'ensembles. Il reste ouverte la question sur la complexité probabiliste d'autres problèmes qui se représentent des réunions de sous-espaces comme par exemple, celui *d'inclusion* d'ensembles, autrement dit si $\{a_1, \dots, a_n\} \subset \{b_1, \dots, b_m\}$.

Dans [75] on a obtenu la borne inférieure pareile sur la complexité des arbres probabilistes de calculs qui reconnaissent un arrangement sur un corps algébriquement clos de la caratéristique zéro. Cette borne est logarithmique du nombre de facettes de toutes les dimensions.

D'autre part, aucune borne inférieure n'est connue sur la complexité des arbres probabilistes sur les corps de la caractéristique positive.

Alors, on a démontré qu'une accélération est possible au facteur $\log n$ pour des arbres probabilistes par rapport aux arbres déterministes. Ce serait intéressant à éclaircir la question, à quel point une meilleure accélération soit possible?

Dans [114] on fait intervenir le concept de la *complexité de communication* sur le corps réel. On a démontré la borne inférieure $n/2$ (qui est proche de celle optimale n) sur la complexité *probabiliste* de communication pour le problème du sac-à-dos et pour celui d'intersection d'ensembles.

8.3 Complexité des arbres analytiques, topologiques et parallèles

Dans la théorie de complexité on considère également des arbres plus puissants que ceux de calculs. Rabin a fait intervenir un modèle d'arbres *analytiques* qui puissent se ramifier dans chaque pas selon le signe d'une fonction analytique réelle. Autrement dit, cette fonction joue le rôle d'une fonction de test. Rabin a démontré la borne inférieure (exacte) n sur la complexité d'un arbre analytique quelconque qui reconnaît l'octant $\mathbf{R}_+^n = \{(x_1, \ldots, x_n) \in \mathbf{R}^n : x_1 \geq 0, \ldots, x_n \geq 0\}$.

Dans [68] on a fabriqué un arbre analytique *probabiliste* qui reconnaît $\mathbf{R}_+^n$ avec la complexité $O(\log^5 n)$. Ceci montre encore une opposition entre des arbres deterministes et ceux probabilistes. Par ailleurs, dans [68] on a prouvé une borne inférieure $\sqrt{n}$ sur la complexité des arbres analytiques probabilites qui reconnaissent une réunion appropriée parmi de tous les 2^n octants.

Jusqu'ici on considérait la profondeur d'un arbre en tant que sa complexité. C'est plus difficile à minorer la *taille* d'un arbre et dans cette direction on a obtenu deux résultats suivants. Dans [67] on a démontré une borne inférieure exponentielle sur la taille de n'importe quel arbre de calculs qui reconnaît l'octant. Dans [68] on a prouvé une borne inférieure exponentielle sur la taille de n'importe quel arbre *analytique* qui reconnaît une réunion appropriée des octants.

Dans [59, 64] on a fait intervenir des *arbres de Pfaff*. Les fonctions de tests le long de chaque chemin d'un arbre de Pfaff constituent une chaîne de Pfaff. Par conséquent, un arbre de Pfaff T est celui analytique. Supposons que T reconnaît un ensemble S étant soit semi-pfaffien ayant c composantes connexes soit un polyèdre ayant c facettes de toutes les dimensions. Dans [59, 64] on a démontré une borne inférieure $\sqrt{\log c}$ sur la complexité de T.

Un autre modèle de calculs est un arbre *parallèle* qui puisse accomplir simultanément quelques opérations. En d'autres termes, un arbre parallèle comporte plusieurs processeurs. D'habitude on suppose qu'un processeur quelconque puisse mettre en marche deux ceux neufs et par cela même après t pas parallèles un arbre puisse utiliser 2^t processeurs.

A.Yao a démontré une borne inférieure $\sqrt{\log c}$ sur la complexité d'un arbre parallèle qui reconnaît un ensemble semi-algébrique S ayant c composantes connexes. Dans [70] j'ai montré que cette borne $\sqrt{\log c}$ est proche à celle exacte lorsque S est un ensemble *semi-linéaire*, en particulier une réunion de polyèdres. Ça signifie qu'en comparaison des arbres *consécutifs*, quand on considère ceux parallèles on ne pourrait qu'attendre une accéleration des calculs à un carré de la complexité.

Lorsqu'on ne compte dans un arbre de calculs que les ramifications on est amené au concept d'un arbre *topologique*. Autrement dit, un arbre topologique est un cas particulier d'un arbre analytique qui n'admet que des polynômes comme des fonctions de tests. Etant donnés des polynômes $f_1, \dots, f_k \in \mathbf{R}[X_1, \dots, X_n]$, on nomme une *cellule* un ensemble semi-algébrique M qui est connexe maximal remplissant la propriété que les signes des polynômes $f_1, \dots, f_k$ sont constants sur M. On désigne par c le nombre de toutes les cellules. Dans [84] j'ai fabriqué un arbre topologique T dont complexité est $\log c$ qui partage toutes les cellules, c'est-à-dire, pour n'importe quel chemin de T l'ensemble de tous les points de $\mathbf{R}^n$ qui satisfont les tests le long de ce chemin est contenu dans une certaine cellule. Evidemment, cette borne supérieure sur la complexité est exacte.

Si un arbre calcule une fonction g (au lieu de reconnaître un ensemble) on peut considérer un arbre *approximant* qui calcule g de façon approximative. Dans [83] on a étendu sur des arbres approximants les bornes inférieure établies dans [69, 71, 72, 75, 76] (et mentionnées dans les chapîtres précédents) sur la complexité en termes du nombre des composantes connexes et du nombre des facettes.

8.4 Complexité additive, fonctions algébriques et de Pfaff

La complexité *additive* $C_+(f)$ d'un polynôme f égale au nombre minimal d'additions nécessaire afin de calculer f (avec cela on ne compte pas ni multiplications ni divisions). C'est une question encore ouverte si la complexité additive est calculable? Dans [52] on a démontré qu'elle devient calculable si on admet des opérations de prises de racines dans des calculs.

Dans [53] on a considéré des circuits avec de prises de racines pour calculer des fonctions algébriques. On a prouvé dans [53] que si on ajoute aux opérations de circuits de plus des prises du logarithme et de la fonction exponentielle, alors la complexité d'une fonction algébrique ne change pas. Autrement dit, les fonctions exponentielles et les logarithmes n'aident pas à accélérer des calculs de fonctions algébriques.

Dans [17] j'ai prouvé une borne inférieure $C_+(f) > \sqrt{\log r}$ où r désigne le nombre des racines *réelles* de f. La preuve employe la borne supérieure sur le nombre des racines *réelles* des fonctions de Pfaff due à Khovanski.

Dans [16] j'ai fait intervenir des calculs *orientés* d'une familles A de formes linéaires et j'ai réussi calculer précisement la complexité additive $C_+(A)$. A cet effet on traite A comme une matrice et dans [12] j'ai établi que A possède une *décomposition de Bruhat généralisée* $A = u_1 w_A u_2$ où w_A est une matrice de permutation *unique* et u_1, u_2 sont les matrices triangulaires supérieures. De plus, si $A = v_1 w v_2$ est une autre décomposition pareile, alors $w_A \prec w$ où $\prec$ désigne la relation de l'ordre sur le groupe des permutations d'après Weil. Il convient de noter que la décomposition de Bruhat a été connue auparavant pour des matrices A seulement *régulières* et avec cela on a $w = w_A$. Alors, dans [16] j'ai démontré que la complexité additive $C_+(A)$ est égale au nombre d'inversions de la permutation w_A.

8.5 Complexité multiplicative et le rang d'un tenseur

Strassen a développé la théorie qui lie la complexité multiplicative (c'est-à-dire, le nombre de multiplications et divisions n'en comptant pas d'additions) d'une famille de formes bilinéaires avec le rang d'un tenseur. Plusieurs problèmes de calcul, en particulier, la multiplication de polynômes ou bien de matrices se ramènent aux calculs des familles des formes bilinéaires.

Le concept du rang d'un tenseur généralise celui d'une matrice, mais à l'opposé du dernier on ne connaît pas d'algorithme raisonnable pour calculer le rang d'un tenseur. Dans [5, 6, 8] j'ai décrit le rang d'une paire de formes bilinéaires. Il est bien connu (grâce à la transformation de Fourier) que la complexité multiplicative de la multiplication de polynômes sur n'importe quel corps *infini* est linéaire et celle-ci est majorée par $n \log n$ sur un corps fini. Dans [5, 6, 8] j'ai montré que cette complexité multiplicative sur un corps *fini* est proche à linéaire.

Pour une seule forme bilinéaire A sur un corps quelconque sa complexité multiplicative $C_m(A)$ coïncide avec son rang $rg(A)$. Mais cela devient faux en général pour une forme A sur un anneau K. D'autre part, l'inégalité $rg(A) \leq C_m(A)$ est toujours vraie. Dans [7, 15] j'ai démontré que lorsque $K = F[X_1, \ldots, X_n]$ est une algèbre des polynômes sur n'importe quel corps F on a l'égalité $rg(A) = C_m(A)$ dans le cas $n = 2$ de deux variables (plus général, cette égalité est remplie pour un anneau K de la dimension homologique globale inférieure ou égale à 2). Pour le nombre n de variables arbitraire l'inégalité $C_m(A) < 2rg(A)$ est prouvée dans [7, 15] et de plus le quotient $C_m(A)/rg(A)$ puisse être aussi proche à 2 que l'on désire quand n tend vers l'infini.

C'est un problème ouvert à établir des bornes inférieures *non-linéaires* sur la complexité multiplicative (en d'autres termes le rang d'un tenseur). Pour arriver à cette fin il faudrait inventer une généralisation de la notion du déterminant (d'une matrice) pour d'un tenseur qui serait égale à zéro sur des tenseurs avec le rang assez petit.

8.6 Complexité de fonctions booléiennes

On ne connaît aucunes bornes inférieures *absolues* sur la complexité de fonctions booléiennes. C'est pourquoi on s'intéresse à bornes dans quelques suppositions raisonnables. Une telle borne a été démontré dans [2]. Une autre borne inférieure pour des circuits booléiens *monotones* a été obtenue dans [4].

Outre la taille T (qui joue le rôle du temps) d'un circuit on considère aussi sa *mémoire* S et dans [2] j'ai établi les bornes inférieures sur le produit TS. En particulier, on a $TS \geq n^2$ pour le problème de la multiplication de polynômes du degré n et on a $TS \geq n^3$ pour le problème de la multiplication de matrices à n lignes et n colonnes.

Dans [78, 80, 81] on a considéré des circuits de la *profondeur* 3 sur un coprs *fini* et on a obtenu une borne inférieure exponentielle sur la complexité (la taille) d'un circuit qui calcule le déterminant. Il convient de noter que la question pareile pour des corps *infinis* est encore ouverte et du même pour des corps *finis* lorsque l'on considère des circuits de la profondeur 4.

9 Un caractère topologique de la classe de complexité P

Afin d'aborder le problème **P-NP** et en général d'obtenir des bornes inférieures sur la complexité, ce serait utile à trouver certains caractères inhérents de la classe **P** et d'autres classes de complexité. Dans cette direction il n'a été connu qu'un seul théorème de Strassen qui décrit la famille de tous les circuits algébriques qui calculent polynômes avec la complexité polynomiale, c'est-à-dire, la classe **P**.

Nous proposons [88] un autre caractère d'une façon topologique de la classe **P**. D'abord, pour simplifier on considère le cas des polynômes univariables. On désigne par $M_{\mathbf{P}}$ l'ensemble des vecteurs des multiplicités des zéros de tous les polynômes de la classe **P**. Nous avons démontré que le nombre des éléments de $M_{\mathbf{P}}$ est inférieure à la fonction exponentielle de la complexité d'un polynôme, alors que le nombre de tous les vecteurs des multiplicités possibles s'accroît comme une fonction exponentielle du degré d'un polynôme, autrement dit double-exponentielle de la complexité.

Dans le cas de polynômes multivariables $M_{\mathbf{P}}$ signifie l'ensemble des vecteurs des multiplicités des zéros de tous les systèmes de polynômes de la classe **P**, sous réserve que le système ait un nombre fini des zéros. On a établi [88] une borne supérieure pareile sur le nombre des éléments de $M_{\mathbf{P}}$ dans le cas de polynômes multivariables. Pour la démonstration on a obtenu [88] une borne supérieure sur la complexité des bases de Groebner *paramétriques*. En outre, on a construit [91] des exemples de systèmes de polynômes qui montrent que la borne susmentionnée sur le nombre des éléments de $M_{\mathbf{P}}$ est exacte.

Alors, on a établi un rapport de la complexité avec un caractère d'une façon topologique, notamment, le nombre des vecteurs de l'ensemble $M_{\mathbf{P}}$. Afin d'obtenir une borne inférieure sur la complexité il faudrait produire un polynôme (ou bien un système de polynômes) dont vecteur des multiplicités des zéros n'appartient pas à $M_{\mathbf{P}}$. Mais c'est la difficulté que l'ensemble $M_{\mathbf{P}}$

n'est pas décrit explicitement, on sait seulement que le taux de $M_{\mathbf{P}}$ est très petit dans l'ensemble de tous les vecteurs des multiplicités possibles.

10 Complexité de machines de Blum-Shub-Smale

Une machine de Blum-Shub-Smale (bref, BSS) est un modèle de calculs qui différe des modèles habituels (par exemple, machines de Turing, machines d'accès arbitraire etc.) en ce que machine de BSS traite des données réelles (plutôt que celles binaires). En particulier, une machine de BSS peut effectuer les opérations arithmétiques avec des nombres réels et se ramifier conformément au signe d'un nombre réel.

Pour des machines de BSS on considère les classes de complexité similaires aux celles pour les modèles de calculs habituels. Le plus grand défi dans ce domaine est l'analogue du problème **P-NP** pour des machines de BSS. Nous avons obtenu deux résultats sur les classes de complexité de machines de BSS.

Le premier d'eux [60] décrit la classe **PPar** du temps parallèle polynomial de machines de BSS lorsqu'on se restreint à l'entrée binaire. A savoir, cette classe coïncide avec la classe **P/poly** du temps polynomial avec des "conseils" de la taille polynomiale.

Dans le deuxième résultat [93] on étudie des ensembles A **NP**-complets qui sont creux. Ça veut dire que la composante n-ième $A_n \subset \mathbf{R}^n$ de A a sa dimension $\dim A_n < n$. Cette notion étend celle des modèles de calculs habituels pour lesquels il est connu qu'aucun ensemble **NP**-complet creu n'existe à moins que $\mathbf{P} \neq \mathbf{NP}$. Pour des machines de BSS nous avons démontré qu'aucun ensemble $\mathbf{NP_W}$-complet creu n'existe où $\mathbf{NP_W}$ désigne la classe du temps non-deterministe polynomial faible.

11 Complexité de preuves algébriques et semi-algébriques

Des preuves algébriques sont une approche au problème $\mathbf{P} - \mathbf{NP}$ qui est un des plus grands défis mathématiques (il est mis en la première place dans la liste bien connue des problèmes du millénaire). L'essence de cette approche est comme suit.

Un de problèmes **NP**-complets typiques est celui de solubilité d'un système d'équations algébriques $f_1 = \ldots = f_k = 0$. Et pour ça on a fait intervenir un système logique qui s'appelle "calcul de polynômes". En tant que ses axiomes on utilise les polynômes $f_1, \ldots, f_k \in F[X_1, \ldots, X_n]$ et il y a deux règles qui permettent premierèment, de produire n'importe quelle combinaison linéaire de polynômes déjà déduits et deuxiemèment, de multiplier par une variable quelconque. Alors, le calcul permet de produire éléments de l'idéal engendré par $f_1, \ldots, f_k$. A titre de la mesure de complexité on considére le degré maximal de tous les polynômes intermédiaires déduits.

Si le degré était constante alors la classe de complexité **P** coïnciderait avec la classe **NP**. C'est pourquoi on s'intéresse à l'obtention de bornes inférieures sur le degré surtout pour des systèmes d'équations booléiennes, c'est-à-dire, lorsque les polynômes $f_1, \ldots, f_k$ contiennent forcement les polynômes $X_i^2 - X_i$ quelque soit $1 \leq i \leq n$. Pour des systèmes booléiennes on a une borne supérieure évidente n sur le degré et donc l'intention est à démontrer une borne inférieure linéaire.

Il était connu une borne inférieure sous-linéaire (en particulier, la racine carrée) pour certains principes logiques, par exemple, pour celui de "pigeons-trous" d'après Dirichlet. Ou également on connaissait une borne inférieure linéaire pour le problème du sac-à-dos (étant **NP**-complet comme on le sait bien) au-dessus d'un corps F de la caractéristique zéro. Enfin dans [85,86] on a établi une borne inférieure linéaire pour le système de tautologies de Tseitin et aussi pour le principe de parité au-dessus d'un n'importe quel corps F. Pour ça on a développé la théorie de "presque-groupes" soi-disants et on l'a employé pour des idéals binomiaux (cela étend la théorie pareile produite plus tôt dans [82] pour le calcul du Nullstellensatz). C'est un résultat definitif et la meilleure possible borne inférieure dans le calcul de polynômes.

La borne établie signifie que ce ne soit pas possible à prouver de coïncidence des classes **P** et **NP** au sein du calcul de polynômes, autrement dit n'en reposant que sur le Nullstellensatz. C'est pourquoi on a fait intervenir dans [87] un système logique nommé "le calcul du Positivstellensatz" qui étend le calcul de polynômes et bénéficie d'inégalités polynomiaux.

A savoir, on dit qu'on a une réfutation dans le calcul du Positivstellensatz si on déduit un certain élément $\sum_{1 \leq i \leq k} f_i g_i$ dans le calcul de polynômes et en outre on calcule au moyen d'un circuit algébrique quelques polynômes

$h_1, \ldots, h_m$ qui remplissent l'égalité

$$1 + \sum_{1 \le j \le m} h_j^2 = \sum_{1 \le i \le k} f_i g_i$$

On définit le degré de la réfutation comme le degré de la déduction de l'élément $\sum_{1 \le i \le k} f_i g_i$ dans le calcul de polynômes. La complétude du calcul du Positivstellensatz est basée sur le Positivstellensatz.

Dans [87] on a démontré une borne inférieure exponentielle sur le degré dans le calcul du Positivstellensatz pour le système polynomial "telescopique", soi-disant. Mais celui-ci n'est pas booléien et dans [89] j'ai établi une borne inférieure linéaire pour le système de tautologies de Tseitin et de plus pour le principe de parité. Finallement, dans [90] j'ai démontré une borne inférieure linéaire pour le problème du sac-à-dos.

Dans [89, 90] pour la démonstration de la borne inférieure d sur le degré il faut fabriquer une application linéaire sur l'anneau des polynômes dont la forme quadratique induite soit non-negative étant restreinte sur le sousespace des polynômes du degré plus petit que $d/2$.

De nouveau ces bornes établies signifient que au sein du calcul du Positivstellensatz ce n'est pas possible à démontrer coïncidence **NP** avec **coNP**. Et ensuite afin de traiter des systèmes d'inégalités $f_1 \ge 0, \ldots, f_k \ge 0$ (au lieu de systèmes d'équations) il faut regarder des systèmes logiques plus forts dans lesquels on peut déduire pas à pas des éléments quelconques du cône $c(f_1, \ldots, f_k)$ engendré par les polynômes $f_1, \ldots, f_k$. A savoir, on définit $c(f_1, \ldots, f_k)$ par récursion selon les règles suivantes:

0. $f_1, \ldots, f_k \in c(f_1, \ldots, f_k)$
1. $f^2 \in c(f_1, \ldots, f_k)$ quelque soit polynôme $f \in \mathbf{R}[X_1, \ldots, X_n]$;
2. si $g_1, g_2 \in c(f_1, \ldots, f_k)$ alors
$g_1 + g_2, g_1 g_2 \in c(f_1, \ldots, f_k)$.

Des pas du calcul généralisé du Positivstellensatz suivent ces règles et on définit le degré d'une déduction comme le degré maximal de tous les polynômes intermédiaires. Ça serait intéressant à obtenir une borne inférieure sur le degré pour n'importe quel problème naturel, par exemple, pour celui de parité.

Un cas particulier de tels systèmes est le calcul d'après Lovasz-Schrijver qui permet de produire des éléments du cône de degrés supérieurs ou égal à 2. Ce système-là est complet pour le problème de la programmation entière. C'est un problème ouvert à démontrer une borne inférieure sur la complexité de déductions dans le calcul d'après Lovasz-Schrijver, par exemple, pour le problème du sac-à-dos. Lorsqu'on considère [97, 98] une généralisation LS^4 du calcul de Lovasz-Schrijver qui admet de produire des polynômes du degré 4 (plutôt que 2), on peut construire [97] des preuves de complexités polynomiales pour les problèmes **NP**-complets de cliques et du sac-à-dos. C'est pourquoi le problème d'obtenir des bornes inférieures pour LS^d avec $d \geq 4$ est difficile. D'autre part, pour une version faible (dénommée, statique) de LS^d on a établi [98] une borne inférieure exponentielle sur la complexité de preuves pour le problème du sac-à-dos. Dans la version statique une preuve est représentée par une formule plutôt que par un calcul comme dans les systémes de preuves habituels.

Une autre direction d'aborder le $\mathbf{P} - \mathbf{NP}$ problème est la considération du calcul de polynômes qui admet d'introduire de nouvelles variables, ceci est le trait de calculs qui permet de les renforcer considérablement. Je suppose d'essayer de démontrer une borne inférieure sur le degré pour ce calcul étendu. Pour ça je vais regarder des déformations d'algèbres quotients par rapport à l'idéal engendré par le système de polynômes.

Mais ce calcul étendu est difficile à considérer et c'est pourquoi on va regarder une version intermédiaire où on déduit des formules polynomiales et au bout d'une déduction à partir d'un système de polynômes on tente de prouver qu'on peut déduire l'unité, autrement dit on est amené à la contradiction conformément aux certaines règles. Je vais obtenir des bornes inférieures sur la complexité de tels calculs logiques.

Il convient de noter que le calcul de polynômes lorsqu'on déduit des formules polynomiales [95] est plus fort que des systèmes de Frege et par cela même des bornes inférieures sur la complexité de ce calcul impliquerait celles des systèmes de Frege. D'autre part, ce calcul est assez fort et permet de réduire de déductions de certains principes logiques, en particulier, pour celui de "pigeon-trous" d'après Dirichlet et pour les tautologies de Tseitin [95]. C'est une question ouverte d'obtenir des bornes inférieures sur la complexité dans un systéme de preuves qui déduit des formules polynomiales.

A la fois, les bornes obtenues fournissent les bornes inférieures sur le degré pour le Positivstellensatz. Avec cela, elles font la paire les bornes supérieures pour le Positivstellensatz que s'occupent M.-F.Roy de notre équipe avec H.Lombardi (Besançon).

12 Complexité d'isomorphisme de graphes et d'algèbres

Le problème de la complexité d'isomorphisme de graphes est encore ouvert. C'est pourquoi on s'intéresse à classes de graphes et des algorithmes qui reconnaissent l'isomorphisme de graphes de telles classes avec la complexité polynomiale. Une de telles classes qui consiste en graphes ayant la *multiplicité de leur spectre* restreinte on a produit dans [14].

L'isomorphisme de graphes est lié étroitement avec l'isomorphisme d'algèbres. Dans [13] j'ai fabriqué une classe d'algèbres dont le problème d'isomorphisme est équivalent à celui de graphes. Autres liaisons avec certains problèmes de polynômes provenants des invariants de graphes sont établies dans [10].

13 Complexité de machines de Turing et de Kolmogorov

Dans ce chapître il s'agit de comparaison de la complexité de deux modèles de calculs. Schoenhage avait réussi mimer une machine de Turing *multidimensionnelle* T au moyen d'un algorithme de Kolmogorov K en *temps réel*, c'est-à-dire, chaque pas de T est mimé par un certain nombre constant de pas de K. Par contre, dans [1] j'ai construit un ensemble E et un algorithme de Kolmogorov qui reconnaît E en temps réel et démontré qu'aucune machine de Turing miltidimensionnelle ne puisse le reconnaître en temps réel.

Dans [3] j'ai établi la complexité de simulation d'un algorithme de Kolmogorov au moyen d'une machine de Turing d'une dimension m et de plus la complexité de simulation d'une machine de Turing d'une dimension n au moyen de celle d'une dimension $m < n$ via m, n. Dans [9] ces résultats étaient étendus sur des machines *non-deterministes.*

14 Problèmes de complexité en biologie

On a étudié dans [101] le problème de la complexité de transformations avec des séquences génétiques en l'alphabet $\{A, C, G, T\}$. Le nombre minimal d'opérations nécessaire afin de transformer une séquence à une autre est décrit par la complexité d'après Kolmogorov qui est uncalculable en général.

D'autre part, pour certaines familles d'opérations appropriées la version correspondante de la complexité d'après Kolmogorov devient calculable. On a démontré dans [101] que si les opérations admettent soit un ajouté d'une lettre soit une répétition d'une sous-séquence à la fin de la séquence, alors on peut calculer cette version de la compexité d'après Kolmogorov en temps polynomial. Si on admet de plus, les opérations de modification d'une seule lettre d'une séquence, on ne peut que majorer la version correspondante de la complexité d'aprés Kolmogorov en temps polynomial.

On considère dans [102, 106] un modèle de réseaux de gènes. On peut traiter ce modèle comme un système dynamique. Nous démontrons dans [102] que ces réseaux peuvent engendrer toutes les structures spatio-temporelles. En outre on produit un algorithme qui nous permet de déterminer les paramètres d'un réseau qui rend la structure donnée. Enfin, nous obtenons dans [102] des bornes inférieures sur le nombre de gènes du réseau qui engendrent la structure. Les dernières bornes reposent sur la méthode de Khovanski.

Dans [112] on considère la stabilité et l'évolution de réseaus génétiques. On a démontré qu'un réseau est instable (lorsque ses paramètres sont fixés), en d'autres mots la probabilité de maintenir l'homéostasie du réseau tend vers 0 quand le temps tend à l'infini. D'autre part, si les paramètres d'un réseau peuvent évoluer, la probabilité de sa survie peut devenir positive.

Dans [121] on considère le comportement d'une espèce biologique en cours d'évolution comme un système dynamique avec un ensemble de paramètres et qui est subi une agitation à l'extérieur. On démontre que si les paramètres ne changent pas alors le système ne va survivre qu'avec une probabilité tendant vers 0 (ça signifie qu'il va quitter le domaine dans lequel on peut subsister). D'autre part, il existe un algorithme probabiliste tel que si les paramètres changent selon cet algorithme, la probabilité de ce que le système va survivre reste positive. On argumente que si l'agitation est assez forte alors les paramètres doivent être discrets. Cela permet de lier l'évolution aux problèmes NP-complets, c'est-à-dire le problème de survivance peut être NP-complet.

On fait introduire dans [124] une classe de systèmes dynamiques aléatoires qui contient réseaux neurals en particulier. On montre que presque tous les systèmes de la dernière classe sont instables. Cela entraîne que si une suite de systèmes simule un procédé d'évolution stable alors la complexité de Kolmogorov des systèmes de cette suite doît accroître.

LISTE DE PUBLICATIONS

1. Kolmogorov's algorithms are stronger than Turing machines. — Notes of Scientific Seminars of LOMI, vol. 60, 1976, p. 29–37 (traduction en anglais dans J. Soviet Math., vol. 14, N 5, 1980).

2. An application of separability and independence notions for proving lower bounds of circuit complexity. — ibid., p. 38–48 (traduction en anglais dans J. Soviet Math., vol. 14, N 5, 1980, p. 1450–1456).

3. Imbedding theorems for Turing machines of different dimensions and Kolmogorov's algorithms. — Soviet Math. Dokl., 1977, vol. 18, N 3, p. 588–592.

4. On a nonlinear lower bound for circuit complexity of a set of disjunctions in monotone boolean basis. — Notes of Scientific Seminars of LOMI, vol. 68, 1977, p. 19–25 (traduction en anglais dans J. Soviet Math., vol. 15, N 1, 1981, p. 11–13).

5. Some new bounds on tensor rank. — Preprint LOMI, E-2-78, 1978, 12 p.

6. Mutiplicative complexity of a pair of bilinear forms and of the polynomial multiplication. — Lecture Notes Computer Science, 1978, vol. 64, p. 250–256.

7. Relation between the rank and the multiplicative complexity of a bilinear form over a noetherian commutative ring. — Notes of Scientific Seminars of LOMI, vol. 86, 1979, p. 66–81 (traduction en anglais dans J. Soviet Math., vol. 17, 1981, p. 1987–1998).

8. The algebraic computational complexity of a set of bilinear forms. — Journal of Computational Math. and Mathematical Physics, 1979, vol. 19, N 3, p. 563–580 (traduction en anglais dans USSR Comput. Math. and Math. Physics, vol. 19, 1980, p. 1-20).

9. Time bounds of multidimensional Turing machines. — Notes of Scientific Seminars of LOMI, vol. 88, 1979, p. 47–55 (traduction en anglais dans J. Soviet Math., vol. 20, 1982, p. 2290–2295).

10. Two reductions of graph isomorphism to problems for polynomials. — ibid., p. 56–61 (traduction en anglais dans J. Soviet Math., vol. 20, 1982, p. 2296–2298).

11. On the Eisenbud–Levine formula over a perfect field (avec N.Ivanov). — Soviet Math. Dokl., 1980, vol. 21, N 3, p. 662–664.

12. Analogy of Bruhat decomposition for the closure of a cone of Chevalley group of a classical serie. — Soviet Math. Dokl., 1981, vol. 23, N 2, p. 393–397.

13. On the complexity of the "wild" matrix problems, of the isomorphism of algebras and graphs. — Notes of Scientific Seminars of LOMI, 1981, vol. 105, p. 10–17 (traduction en anglais dans J. Soviet Math., vol. 22, 1983, p. 1285–1289).

14. Isomorphism of graphs with bounded eigenvalue multiplicity. — Proc. 14 ACM Symp. Th. Comput., 1982, p. 310–324 (avec L. Babai, D. Mount).

15. Multiplicative complexity of a bilinear form over a commutative ring. — Lecture Notes Computer Science, 1981, vol. 118, p. 281–286.

16. Additive complexity in directed computations. — Theoretical Computer Science, 1982, vol. 19, p. 39–67.

17. Lower bounds in algebraic complexity. — Notes of Scientific Seminars of LOMI, vol. 118, 1982, p. 25–82 (traduction en anglais dans J. Soviet Math., vol. 29, 1985, p. 1388–1425).

18. Polynomial–time factoring of the multivariable polynomials over a global field. — Preprint LOMI E-5-82, 1982, 39 p. (avec A. Chistov).

19. Subexponential–time solving systems of algebraic equations. I, II. — Preprint LOMI E-9-83, E-10-83, 119 p. (avec A. Chistov).

20. Polynomial factoring over a finite field and solving systems of algebraic equations. — Notes of Scientific Seminars of LOMI, vol. 137, 1984, p. 20–79 (traduction en anglais dans J. Soviet Math., 1986, vol. 34, p. 1762–1803).

21. Fast decomposition of polynomials into irreducible ones and the solution of systems of algebraic equations. — Soviet Math. Dokl., vol. 29, 1984, p. 380–383 (avec A. Chistov).

22. Complexity of quantifier elimination in the theory of algebraically closed fields. — Lecture Notes Computer Science, 1984, vol. 176, p. 17–31 (avec A. Chistov).

23. Finding real solutions of systems of algebraic inequalities in subexponential time. — Soviet Math. Dokl., 1985, vol. 32, N 1, p. 316–320 (avec N. Vorobjov(jr.)).

24. Complexity of deciding the first–order theory of algebraically closed fields. — Izvestia of Academy of Sciences of the USSR, 1986, vol. 50, N 5, p. 1106–1120 (traduction en anglais dans Math. USSR Izvestia, vol. 29, N 2, 1987, p. 459–475).

25. Computational complexity in polynomial algebra. — Proc. of International Congress of Mathematicians, 1986, Berkeley, vol. 2, p. 1452–1460.

26. The matching problem for bipartite graphs with polynomially bounded permanents is in NC. — Proc. 28 Symp. Found. Comput. Sci., IEEE, 1987, p. 166–172 (avec M. Karpinski).

27. Solving systems of polynomial inequalities in subexponential time. — J. Symp. Comput., 1988, vol. 5, p. 37–64 (avec N. Vorobjov (jr.)).

28. Complexity of deciding Tarski algebra. — J. Symp. Comput., 1988, vol. 5, p. 65–108.

29. Complexity of quantifier elimination in the theory of ordinary differential equations. — Lecture Notes Computer Science, 1989, vol. 378, p. 11–25.

30. Complexity of factoring an ordinary linear differential operator. — Soviet Math. Dokl., 1989, vol. 38, N 3, p. 452–457.

31. Complexity of factoring and GCD calculating of ordinary linear differential operators. — J. Symp. Comput., 1990, vol. 10, N 1, p. 7–37.

32. Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields — SIAM J. Comput., 1990, vol. 19, N 6, p. 1059–1063 (avec M. Karpinski, M. Singer).

33. Solving ordinary differential equations in the series with real exponents. — Trans. AMS, 1991, vol. 327, N 1, p. 329–351 (avec M. Singer).

34. Complexity of calculating characteristics and genre of a system of exterior differential equations. — Soviet Math. Dokl., 1989, vol. 39, N 3, p. 432–436.

35. Computational complexity in commutative algebra. — Math. Notes, 1989, vol. 46, N 1, p. 563–568.

36. Complexity of computing the characteristics and the genre of a system of exterior differential equations. — Lecture Notes Computer Science, 1989, vol. 358, p. 534–543.

37. Complexity of irreducibility testing for a system of linear ordinary differential equations. — Proc. Int. Symp. on Symb. Algebr. Comput., ACM, 1990, Japan, p. 225–230.

38. How to test in subexponential time whether two points can be connected by a curve in a semialgebraic set. — Ibid., p. 104–105.

39. Complexity of solving systems of linear equations over the rings of differential operators. — Proc. Int. Symp. Effective Methods in Algebraic Geometry, 1990, Italy, Birkhäuser, Progr. in Math., vol. 94, p. 195–202.

40. Interpolating of sparse rational functions without knowing bounds on exponents. — Proc. 31 IEEE Symp. FOCS, 1990, p. 840–846 (avec M.Karpinski, M.Singer).

41. The interpolation problem for k–sparse sums of eigenfunctions of operators. — Adv. Appl. Math., 1991, vol. 12, p. 76–81 (avec M. Karpinski, M. Singer).

42. Comptage des composantes connexes d'un ensemble semi–algébrique en temps simplement exponentiel. — Compte–Rendus de l'Acad. des Sci. Paris, 1990, t. 311, Serié I, p. 879–882 (avec J. Heintz, M. F. Roy, P. Solerno, N. N. Vorobjov (jr.)).

43. Determination of the number of connected components of a semialgebraic set in subexponential time. — Soviet Math. Dokl., 1991, vol. 42, N 2, p. 563–566 (avec N. N. Vorobjov(jr.)).

44. Counting connected components of a semialgebraic set in subexponential time. — Computational Complexity, 1992, vol. 2, N 2, p. 133-186 (avec N. N. Vorobjov (jr.)).

45. Finding connected components of a semialgebraic set in subexponential time. — Appl. Algebra in Engineering, Communication and Computing, 1992, v. 2, p. 217-238 (avec J. Canny, N. N. Vorobjov (jr.)).

46. Algorithms for sparse rational interpolation. — Proc. Int. Symp. on Symb. Algebr. Comput., ACM, 1991, Bonn, p. 7–13 (avec M. Karpinski).

47. Existence of short proofs for nondivisibility of sparse polynomials under the Extended Riemann Hypothesis. — Proc. ACM Int. Symp. Symb. Alg. Comput., Berkeley, 1992, p. 117–122 (avec M. Karpinski, A. Odlyzko).

48. An approximation algorithm for the number of zeroes of arbitrary polynomials over $GF[q]$ — Proc. 32 IEEE Symp. FOCS, 1991, p. 662–669 (avec M. Karpinski).

49. Computational complexity of sparse real algebraic function interpolation. — "Computational Algebraic Geometry," édité par A. Galligo. — Progress in Mathematics, Birkhauser, 1993, vol. 109, p. 91-104 (avec M. Karpinski, M. Singer).

50. A zero-test and an interpolation algorithm for the shifted sparse polynomials. — Proc. Int. Conf. Appl. Algebra and Error Correcting Codes, Puerto Rico, May 1993, LNCS 673, p. 162-169 (avec M. Karpinski).

51. Computational complexity of sparse rational interpolation. — SIAM J. Comput., 1994, vol. 23, N 1, p. 1-11 (avec M. Karpinski, M. Singer).

52. Computability of the additive complexity for algebraic circuits with root extracting. — Theor.Comput.Sci., 1996, vol. 157, 1, p. 91-99 (avec M.Karpinski)

53. On computing algebraic functions using logarithms and exponentials. — SIAM J.Comput., 1995, 2, p.242-246 (avec M.Singer, A.Yao).

54. Deviation theorems for solutions of linear ordinary differential equations and applications to parallel complexity of sigmoids. — St.Petersburg Math. J., 1995, 6, N 1, p. 89-106.

55. Deviation theorems for pfaffian sigmoids. — St.Petersburg Math. J., 1995, 6, N 1, p. 107-112.

56. Deviation theorems for solutions of differential equations and applications to lower bounds on parallel complexity of sigmoids. — Theor.Comp. Sci., 1994, vol. 133, 1, p. 23-33.

57. NC solving of a system of linear differential equations in several unknowns. — Theor.Comput.Sci., 1996, vol. 157, 1, p. 79-90.

58. Complexity lower bounds on testing membership to a polyhedron by algebraic decision trees. – Proc.ACM Symp. Th. Comput., Montreal, 1994, p. 635-644 (avec M.Karpinski, N.Vorobjov).

59. Complexity lower bounds for computation trees with elementary transcendental function gates. – Proc.IEEE Symp. Found. Comput. Sci., Santa Fe, 1994, p. 548-552 (avec N.Vorobjov).

60. On the power of real Turing machines over binary inputs. – SIAM J. Comput., 1997, 1, p. 243-254 (avec F.Cucker).

61. Improved lower bound on testing membership to a polyhedron by algebraic decision trees. – Proc.IEEE Symp. Found. Comput.Sci., Milwaukee, 1995, p. 258-265 (avec M.Karpinski, N.Vorobjov).

62. Algorithms for computing sparse shifts for multivariate polynomials. – Proc.Intern.Symp. Symbol.Algebr.Comput., ACM, 1995, Montreal, p. 96-103 (avec Y.Lakshman).

63. Algorithms for computing sparse shifts for multivariate polynomials. – Appl.Algebra in Eng., Communic., Comput., 2000, 11, 1, p. 43–67 (avec Y.Lakshman).

64. Complexity lower bounds for computation trees with elementary transcendental function gates. – Theor.Comput.Sci., 1996, vol. 157, 2, p. 185-214 (avec N.Vorobjov).

65. Testing shift-equivalence of polynomials using quantum machines. – Proc.Intern.Symp. on Symb.Algebr.Comput., ACM, 1996, Zürich, p. 49-54

66. A lower bound for randomized algebraic decision trees. – Proc.ACM Symp. Th.Comput., Philadelphia, 1996, p. 612-619 (avec M.Karpinski, F.Meyer auf der Heide, R.Smolensky)

67. An exponential lower bound on the size of algebraic decision trees for MAX problem. – Computational Complexity, 1998, vol.7, p. 193-203 (avec M.Karpinski, A.Yao)

68. Randomized versus deterministic analytic decision trees – Computational Complexity, 1997, vol.4, 6, p. 376-388 (avec M.Karpinski, R.Smolensky)

69. A lower bound for randomized algebraic decision trees – ibid., p. 357-375 (avec M.Karpinski, F.Meyer auf der Heide, R.Smolensky)

70. Nearly sharp complexity bounds for multiprocessor algebraic computations. – J. Complexity, 1997, vol.13, 1, p.50-64.

71. Quadratic complexity lower bound for randomized computation trees solving the knapsack problem. – Proc.ACM Symp. Th.Comput., El Paso, 1997, p. 76-85 (avec M.Karpinski)

72. Lower bound on testing membership to a polyhedron by algebraic decision and computation trees. – Discrete and Computational Geometry, 1997, vol.17, 2, p.191-215 (avec M.Karpinski, N.Vorobjov)

73. Testing shift-equivalence of polynomials by deterministic, probabilistic and quantum machines. – Theor. Comput.Sci., 1997, vol. 180, p. 217-228

74. Computing Minimum-Link Path in a Homotopy Class amidst Semi-Algebraic Obstacles in the Plane. – Lect.Notes Comput.Sci., 1997, vol. 1255, p. 114-129 (avec A.Slissenko)

75. Complexity lower bounds for randomized computation trees over zero characteristic fields. – Computational Complexity, 1999, 8, 4, p. 316–329

76. Randomized complexity lower bound for arrangements and polyhedra. – Discrete and Computational Geometry, 1999, vol. 21, p. 329-344

77. Polytime Algorithm for the Shortest Path in a Homotopy Class amidst Semi-Algebraic Obstacles in the Plane. – Proc. ACM Intern.Conf.Symbolic and Algebraic Computations, Rostock, Germany 1998, p. 17-24 (avec A.Slis senko)

78. Exponential Lower Bound on the Size of Depth 3 Arithmetic Circuit Computing Determinant over a Finite Field. – Proc. ACM Symp. Th.Comput., Dallas 1998, p. 577-582 (avec M.Karpinski)

79. Randomized Complexity Lower Bounds. – Proc. ACM Symp. Th.Comput., Dallas 1998, p. 219-223

80. Exponential Complexity Lower Bounds for Depth 3 Arithmetic Circuits in Algebras of Functions over Finite Fields. – Proc. IEEE Symp. Found.Comput.Sci.., Palo Alto, 1998, p. 269-278 (avec A.Razborov)

81. Exponential Complexity Lower Bounds for Depth 3 Arithmetic Circuits in Algebras of Functions over Finite Fields. – Appl.Algebra in Eng., Communic.,Comput, 2000, 10, 6, p. 465–487 (avec A.Razborov)

82. Tseitin's Tautologies and Lower Bounds for Nullstellensatz Proofs. – Proc. IEEE Symp. Found.Comput.Sci. Palo Alto, 1998, p. 648-652

83. Complexity Lower Bounds for Approximation Algebraic Computation Trees. – J.Complexity, 1999, vol. 15, 4, p. 499-512 (avec F.Cucker)

84. Topological Complexity of the Range Searching. – J.Complexity, 2000, vol. 16, p.50-53.

85. Linear Gaps Between Degrees for the Polynomial Calculus Modulo Distinct Primes. – Proc. ACM Symp. Th. Comput., 1999, p. 547–556 (avec S.Buss, R.Impagliazzo, T.Pitassi)

86. Linear Gaps Between Degrees for the Polynomial Calculus Modulo Distinct Primes. – J. Comput. Syst. Sci., 2001, vol. 62, p. 267–289 (avec S.Buss, R.Impagliazzo, T.Pitassi)

87. Complexity of Null- and Positivstellensatz proofs. – Ann. Pure and Appl. Logic, 2001, 113/1-3, p. 153–160 (avec N.Vorobjov)

88. Bounds on numbers of vectors of multiplicities for polynomials which are easy to compute. – Proc. ACM Intern.Conf.Symbolic and Algebraic Computations, Scotland, 2000, p. 137–145 (avec N.Vorobjov)

89. Linear Lower Bound on Degrees of Positivstellensatz Calculus Proofs for the Parity. – Theor. Comput. Sci., 2001, vol. 259, p. 613–622

90. Complexity of Positivstellensatz proofs for the knapsack. – Computational Complexity, 2001, 10, p. 139–154

91. Constructing double-exponential number of vectors of multiplicities of solutions of polynomial systems. – dans Contemporary Mathematics, AMS, 2001, vol. 286, p. 115–120

92. Approximation and complexity: Liouvillean type theorems for linear differential equations on an interval. – Foundations of Computational Mathematics, 2001, vol. 1, p. 289–295

93. There are no sparse NP_W-hard sets. – SIAM J.Computing, 2001, vol. 31, p. 193–198 (avec F.Cucker).

94. Approximating shortest path for the skew lines problem in time doubly logarithmic in 1/epsilon. – Theor. Comput. Sci. 2004, vol. 315, p. 371-404 (avec D.Burago, A.Slissenko)

95. Algebraic proof systems over formulas. – Theor. Comput. Sci., 2003, vol. 303, p. 83–102 (avec E.Hirsch).

96. Approximation and complexity II: iterated integration. – Foundations of Computational Mathematics, 2002, vol. 2, p. 295–304

97. Complexity of semialgebraic proofs. – Lect. Notes Comput. Sci., 2002, vol. 2285, p. 419–430 (Proc. STACS, 2002, Sophie-Antipolis) (avec E.Hirsch, D.Pasechnik)

98. Exponential lower bound for static semi-algebraic proofs. – Lect. Notes Comput.Sci., 2002, vol. 2380, p. 257–268 (Proc. ICALP, 2002, Malaga) (avec E.Hirsch, D.Pasechnik)

99. Homomorphic public-key cryptosystems and encrypting boolean circuits. – Appl. Algebra in Eng., Communic., Comput., 2006, vol. 17, p. 239–255 (avec I.Ponomarenko)

100. Public-key cryptography and theory of invariants. – J. Math. Sci., 2005, vol. 126, issue 3, p. 1152-1157.

101. Algorithmic aspects of genetic sequences and relative Kolmogorov complexity. – Intern. J. Pure Appl. Math., 2004, vol. 11, 3, p. 283–292 (avec A.Grigorieva)

102. Complexity of patterns generated by genetic circuits and Pfaffian functions. – Ann. Pure Appl. Logic, 2006, vol. 141, p. 412–428 (avec S.Vakoulenko)

103. Homomorphic public-key cryptosystems over groups and rings. – Quaderni di Matematica, 2004, vol. 13, p. 305–325 (avec I.Ponomarenko)

104. Weak Bézout inequality for D-modules. – J.Complexity, 2005, vol. 21, p. 532–542

105. Factoring and solving linear partial differential equations. – Computing, 2004, vol. 73, p. 179–197 (avec F.Schwarz)

106. Complexity of gene circuits, Pfaffian functions and morphogenesis problem. – Comptes-Rendus Acad. Sci., sér. 1, 2003, vol. 337, issue 11, p. 721–724 (avec S.Vakulenko)

107. Polynomial-time computing over quadratic maps I: sampling in real algebraic sets. – Computational Complexity, 2005, vol. 14 p. 20–52 (avec D.Pasechnik)

108. Loewy- and primary-decompositions of D-modules. – Adv. Appl. Math., 2007, vol. 38, p. 526–541 (avec F.Schwarz)

109. Optical device accelerating dynamic programming. – Physics of Particles and Nuclei Letters, 2007, vol. 4, p. 141–142 (avec A.Kazakov, S.Vakulenko)

110. Constructions in public-key cryptography over matrix groups. – Contemporary Math., AMS, 2006, vol. 418, p. 103–119 (avec I.Ponomarenko)

111. Time hierarchies for cryptographic function inversion with advice. – J. Math. Sci., 2009, vol. 158, p. 633-644 (avec E.Hirsch, K.Pervyshev)

112. Evolution in random environment and structural instability. – J. Math. Sci., 2006, vol. 138, p. 5644-5662 (avec S,Vakulenko)

113. A complete public-key cryptosystem. – Groups, Complexity, Cryptology, 2008, vol. 1, p. 1-12 (avec E.Hirsch, K.Pervyshev)

114. Probabilistic communication complexity over the reals. – Computational Complexity, 2008, vol. 17, p. 536–548.

115. Complexity of a standard basis of a D-module. – à paraître dans St.Petersburg Math. J. (avec A.Chistov)

116. Analogue of Newton-Puiseux series for non-holonomic D-mofules and factoring. - arxiv.org:0811.1367

117. Invariant-based cryptosystems and their security against provable break à paraître dans St.Petersburg Math. J. (avec A.Kojevnikov, S.Nikolenko)

118. Zero-knowledge autentification schemes from actions on graphs, groups, or rings (avec V.Shpilrain)

119. Loewy decomposition of linear third-order PDE's in the plane. – Proc. Intern. Symp. Symbol. Algebr. Comput., ACM, Austria, 2008, p. 277–286 (avec F.Schwarz)

120. Construction of universal Thom-Whitney-a stratifications, their functoriality and Sard-type Theorem for singular varieties. – Preprint MPI fuer Mathematik 2008-32, Bonn (avec P.Milman)

121. Instability, evolution and morphogenesis. - dans Progress in Math. Biology Research, Nova publ., 2008, p. 55–100 (avec S.Vakulenko)

122. Authentication from matrix conjugation. - Preprint MPI fuer Mathematik, Bonn, 2008-78 (avec V.Shpilrain)

123. Non-holonomic ideals in the plane and absolute factoring. - arxiv.org:0811.1368 (avec F.Schwarz)

124. Instability, complexity and evolution. à paraître dans J. Math. Sci (avec S.Vakulenko)