Habilitation à diriger des recherches
soutenue par

Pascal KOIRAN

devant
l'UNIVERSITÉ CLAUDE BERNARD-LYON I
le 10 décembre 1999.

Complexité et décidabilité pour les modèles de calcul algébriques et analogiques

Membres du jury :

Michel COSNARD
Angus MACINTYRE
Christian MICHAUX
Bruno POIZAT
Marie-Françoise ROY (présidente)
Miklos SANTHA

Rapporteurs :

Lenore BLUM
Bruno POIZAT
Eduardo SONTAG

Habilitation préparée au Laboratoire de l'Informatique du Parallélisme
(unité mixte CNRS, INRIA, ENS Lyon, No 5688)

Résumé

Les travaux décrits dans ce document sont regroupés en deux grands thèmes : calculabilité et connexionisme d'une part, complexité algorithmique en algèbre et en géométrie d'autre part. J'ai développé surtout ce second thème car il englobe la plupart de mes travaux récents. Dans l'ensemble de ce rapport j'insiste autant sur les problèmes ouverts que sur le résultats obtenus, ce qui je l'espère en rendra la lecture plus intéressante.

La plupart des articles sont disponibles, parfois dans une version préliminaire, sur ma page web : http ://www.ens-lyon.fr/~koiran.

Chapitre 1

Calculabilité et connexionisme

Les recherches en connexionisme ont connu un développement important depuis une quinzaine d'années. Des résultats encourageants ont été obtenus dans des domaines d'application variés (par exemple en traitement du signal, reconnaissance de formes, robotique, prédiction de séries temporelles, optimisation). Mes travaux dans ce domaine contribuent à une théorie mathématique des modèles connexionistes, notamment par des comparaisons avec les modèles classiques.

1.1 Apprentissage

Après son introduction en 1984 par Valiant [Val84] le modèle d'apprentissage PAC (probablement approximativement correct) est devenu rapidement le formalisme le plus populaire dans les études mathématiques de l'apprentissage inductif (par l'exemple). Il existe plusieurs livres sur ce sujet ; on pourra consulter en particulier [KV94]. Les deux questions principales qui se posent dans ce modèle (et souvent dans la pratique) portent sur la complexité en échantillon (combien d'exemples faut-il pour apprendre correctement ?) et sur la complexité algorithmique proprement dite (quelles ressources, par exemple en temps ou en espace, sont-elles nécessaires pour traiter ces exemples ?). Je me suis surtout intéressé à la première de ces deux questions.

Dans la version de base du modèle PAC, on cherche à apprendre les fonctions d'une classe $\mathcal{C}$ de fonctions booléennes sur un domaine X. L'algorithme d'apprentissage prend en entrée un paramètre de précision ϵ et un paramètre de confiance δ. Il a accès à un oracle qui lui fournit des "exemples" de la forme $(x_i, c(x_i))$, où $c \in \mathcal{C}$ est la fonction à apprendre. Les x_i sont tirés indépendamment suivant une distribution de probabilité P sur X (P est arbitraire et inconnue de l'algorithme). En sortie l'algorithme produit une hypothèse $h \in \mathcal{C}$ qu'on souhaite être une bonne approximation de c dans le sens où $P(c(x) \neq h(x)) \leq \epsilon$. L'algorithme est dit "probablement approximativement correct" si pour toute fonction $c \in \mathcal{C}$ et toute distribution de probabilité P, il produit une bonne approximation de c avec probabilité au moins $1-\delta$. Ce modèle peut être généralisé dans plusieurs directions ; en particulier, on peut autoriser l'algorithme à produire des hypothèses n'appartenant pas à $\mathcal{C}$, et chercher à apprendre des fonctions à valeurs réelles.

1.1.1 VC dimension et aspects statistiques

On démontre que dans le modèle PAC la complexité en échantillon est *grosso modo* proportionnelle à un certain paramètre combinatoire (la dimension de Vapnik-Chervonenkis, ou *VC dimension*) de la famille de représentations utilisée par l'algorithme d'apprentissage. La VC dimension d'une famille de fonctions booléennes (ou de manière équivalente d'une famille de parties d'un domaine X) est définie de la manière suivante.

Définition 1 *Soit $\mathcal{F}$ une famille de parties d'un ensemble X. On dit que $\mathcal{F}$ pulvérise un sous-ensemble $A \subseteq X$ si pour tout $E \subseteq A$ il existe $S \in \mathcal{F}$ tel que $E = S \cap A$. La VC dimension de $\mathcal{F}$ est la cardinalité du plus grand sous-ensemble de X pulvérisé par $\mathcal{F}$.*

La VC dimension mesure donc en quelque sorte la richesse d'une classe de fonctions ; en particulier, si $\mathcal{F}$ est finie sa VC dimension est bornée par $\log_2 |\mathcal{F}|$. Cette notion intervient dans diverses branches des mathématiques : combinatoire, logique (théorie des modèles), probabilités et statistiques, analyse fonctionnelle (voir par exemple le livre de Ledoux et Talagrand [LT91]). Son importance est due en particulier à l'existence de lois uniformes des grands nombres pour les classes de VC dimension finie [VC71, Vap82]. La version suivante est tirée d'un article qui a popularisé ces résultats dans la communauté informatique [BEHW89].

Théorème 1 *Soit P une distribution de probabilité sur X et $\mathcal{F}$ une famille*[1] *de parties de X de VC dimension finie d. Si on tire m points $x_1, \ldots, x_m$ indépendamment suivant P alors*

$$\Pr\{(x_1, \ldots, x_m); \exists S \in \mathcal{F} \mid P(S) \geq \epsilon \text{ et } x_1 \notin S, \ldots, x_m \notin S\} \leq \delta$$

dès que

$$m \geq \max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}; \frac{8d}{\epsilon}\log\frac{13}{\epsilon}\right).$$

En terme d'apprentissage ce résultat signifie qu'avec probabilité au moins $1-\delta$, si on réussit à trouver une hypothèse qui classifie correctement tous les exemples alors le taux d'erreur en généralisation de cette hypothèse est au plus ϵ.

Le théorème 1 et ses généralisations ne résolvent que partiellement le problème de la complexité en échantillon, car pour des systèmes complexes tels que les réseaux de neurones artificiels, le calcul de la VC dimension est assez délicat.

[1]Pour être tout-à-fait rigoureux il faudrait ajouter une hypothèse de mesurabilité, voir [BEHW89].

En fait, un calcul exact est hors de question, et il faut se contenter de bornes inférieures et supérieures. Ces bornes sont souvent exprimées en fonction du nombre W de paramètres ajustables (ou *poids*) du réseau. Pour les réseaux en couche les plus simples, la VC dimension est de l'ordre de W, ou un peu plus ($W \log W$ pour les rśeaux d'unités à seuil). J'ai montré en collaboration avec Eduardo Sontag [26] que pour la plupart des réseaux en couche utilisés en pratique,[2] la VC dimension est en fait nettement plus élevée (elle est au moins de l'ordre de W^2). Nous avons ensuite poursuivi cette étude pour les réseaux récurrents [25]. Ces réseaux sont souvent utilisés dans les applications où le paramètre temps est important, par exemple : prédiction de séries temporelles, traitement du signal, reconnaissance de la parole, problèmes de contrôle...

A la suite des travaux de Goldberg et Jerrum [GJ95], puis de Karpinski et Macintyre [KM97c], on peut dire qu'il est maintenant possible d'obtenir des bornes supérieures "raisonnables" pour à-peu-près n'importe quelle classe de réseaux. Ces bornes ne sont pas toujours optimales, et leur amélioration est un problème mathématique parfois difficile (on pense en particulier aux réseaux sigmoidaux, pour lesquels la borne supérieure est $O(W^4)$ [KM97c] et la borne inférieure $\Omega(W^2)$ [26]). Il faut bien reconnaître que ces questions n'ont qu'une importance limitée pour la pratique de l'apprentissage. En effet, les bornes de VC dimension sont quasiment toujours trop pessimistes pour être utilisables. Ceci pour plusieurs raisons : d'abord, le calcul exact de la VC dimension est en général impraticable ; ensuite, les bornes sur la complexité en échantillon en fonction de la VC dimension ne sont pas optimales non plus ; et enfin ces bornes portent sur le pire cas (en fait à la fois sur la pire fonction cible et la pire distribution de probabilité). En conclusion, on peut dire que jusqu'à présent la théorie de la VC dimension a plus réussi du point de vue du physicien que du point de vue de l'ingénieur (c'est-à-dire qu'elle explique *pourquoi* les algorithmes d'apprentissage utilisés en pratique fonctionnent, mais elle n'a apporté qu'une aide limitée à ceux qui cherchent *comment* les améliorer).

La théorie de la VC dimension ne s'appliquait originellement qu'aux problèmes de classification. Une extension pleinement satisfaisante aux problèmes de régression a été proposée il y a quelques années [ABDCBH93]. Cependant, au départ on ne connaissait que très peu d'exemples concrets pour lesquels la dimension correspondante est finie (les fonctions d'une variable réelle de variation bornée constituant le principal exemple). J'ai proposé en collaboration avec Leonid Gurvits [27] une méthode qui permet d'obtenir de nombreux exemples de cette sorte. On retrouve comme cas particuliers les fonctions de variation bornée, et certaines familles de réseaux de neurones. Ce travail a été poursuivi par d'autres chercheurs, l'idée générale étant que "la taille des poids est plus importante que la taille du réseau" [Bar97]. Des idées similaires, développées de

[2]Il s'agit des réseaux d'unités continues, pour lesquels les techniques de "rétropropagation du gradient" ont été développées.

manière indépendante, apparaissent dans la théorie des *support vector machines* de Vapnik [Vap95].

Pour terminer cette section, mentionnons des résultats qui anticipent sur le chapitre 2 : la VC dimension est également la notion clé pour l'étude des algorithmes probabilistes sur les réels ou d'autres structures algébriques. Elle permet d'obtenir des analogues des résultats "BPP $\subseteq$ P/poly" [18] et "BPP $\subseteq \Sigma_2$" [17] (pour le cas classique, voir par exemple le livre de Balcazar, Diaz et Gabarro [BDG88]). Des résultats similaires ont été obtenus indépendamment par Karpinski et Macintyre [KM97a, KM97b] (ces deux articles traitent en particulier des cas p-adiques et pfaffiens). Tous ces résultat sont eux aussi basés sur la loi uniforme des grands nombres. Par contre les bornes inférieures en complexité de circuit de [40] utilisent uniquement des propriétés combinatoires de la VC dimension. Rappelons enfin que la VC dimension permet de concevoir des algorithmes géométriques simples et efficaces [HW87] (un tel algorithme pour le problème de localisation de point dans un arrangement d'hyperplans intervient dans la preuve du théorème de transfert de [11], qui est présenté dans la section 2.2.6).

1.1.2 Aspects algorithmiques

La théorie algorithmique de l'apprentissage est dans un état moins satisfaisant que la théorie statistique dans la mesure où l'on n'a réussi à trouver aucune classe de fonctions "raisonnablement expressive" qui admette un algorithme d'apprentissage polynomial. Peut-être ceci reflète t-il simplement le fait qu'apprendre est réellement une tâche difficile ; mais cette situation est bien peu satisfaisante car aucun résultat ne suggère qu'une telle classe de fonctions n'existe pas. De plus le statut de certaines candidates naturelles (par exemple la classe des formules booléennes sous forme normale disjonctive) reste ouvert. Cependant, on sait montrer sous des hypothèses cryptographiques que la classe de toutes les formules booléennes n'est pas apprenable en temps polynomial [KV94], et ceci quelque soit la classe d'hypothèses utilisée.[3]

Afin d'obtenir des résultats positifs, on est donc conduit à imposer toutes sortes de restrictions sur les fonctions à apprendre. J'ai ainsi proposé un algorithme permettant d'apprendre en temps polynomial une certaine classe de réseaux neuronaux acycliques à entrées et sorties réelles [32], quand la dimension de l'espace d'entrée est constante. L'algorithme proposé est itératif, c'est-à-dire qu'il détermine les paramètres de chaque unité à tour de rôle (plutôt que de tenter de les optimiser simultanément). Des heuristiques dérivées de cet algorithme ont été programmées et testées [30, 28].

[3]On doit en fait se limiter à des hypothèses évaluables en temps polynomial, ce qui n'est guère restrictif.

1.2 Calcul par systèmes dynamiques

Le caractère itératif du fonctionnement des réseaux de neurones récurrents fait qu'il est naturel de les comparer aux modèles de calculs étudiés en informatique théorique, tels que : automates d'états finis, automates à piles, machines de Turing,.... Hava Siegelmann et Eduardo Sontag ont montré qu'une machine de Turing quelconque peut-être simulée par un réseau fini d'unités continues (voir [Sie99] pour un exposé des travaux de Siegelmann). Leur preuve s'applique uniquement à des systèmes "linéaires seuillés." J'ai généralisé ce résultat à une classe de réseaux nettement plus grande [29].

Ces modèles connexionistes peuvent être vus comme des cas particuliers de calcul par système dynamique. La dynamique en question s'obtient par itération d'une certaine fonction $f : \mathbb{R}^n \to \mathbb{R}^n$ (les automates cellulaires constituent un autre exemple de calcul par système dynamique). Je me suis alors intéressé plus largement au calcul par itération de fonction, notamment en basse dimension. On montre ainsi qu'une machine de Turing quelconque peut être simulée par une fonction continue affine par morceaux itérée sur un carré [33].

Ce résultat implique que certains problèmes naturels sont indécidables. C'est le cas du problème d'accessibilité : étant donné f, un point de départ x_0 et un point d'arrivée x_1, peut-on atteindre x_1 en itérant f sur le point de départ x_0 ?

Il est nécessaire de recourir à des systèmes plus complexes pour simuler les machines de Turing en dimension un. Cependant, si on travaille sur toute la droite réelle au lieu d'un intervalle, la simulation est encore possible par itération de fonctions "élémentaires" [24].

Les travaux décrits ci-dessus dans la section 1.2 sont pour l'essentiel contenus dans ma thèse. J'ai repris depuis peu ces recherches en m'intéressant à des problèmes de stabilité. La motivation vient d'une question de Sontag [Son95] : la stabilité des systèmes linéaires seuillés est-elle décidable ? L'importance de ce problème vient du rôle fondamental joué par la stabilité en théorie des systèmes et du contrôle, et de l'utilisation croissante dans ce domaine de techniques connexionistes, ou plus généralement de techniques non linéaires. En étudiant à nouveau les systèmes dynamiques en basses dimensions, j'ai obtenu avec Vincent Blondel, Olivier Bournez, Christos Papadimitriou et John Tsitsiklis des résultats d'indécidabilité en dimension deux, et de décidabilité en dimension un [23]. Cette étude a été conduite pour les systèmes affines par morceaux. Des modèles de ce type sont devenus assez populaires dans le domaine des "systèmes hybrides", aussi bien en temps discret [Son96] qu'en temps continu [AMP95].

Dans un article en préparation [22] nous démontrons enfin la conjecture de Sontag : la stabilité asymptotique globale des systèmes linéaires seuillés est indécidable (ce résultat apparaît aussi dans la thèse d'Olivier Bournez [Bou99]). Nous montrons également l'indécidabilité de trois autres propriétés de ces systèmes : la convergence globale, la mortalité (est-ce que toute trajectoire atteint l'origine ?) et la nilpotence. Notons que ces résultats ont des conséquences "purement mathématiques". Par exemple, pour une infinité de valeurs de n il existe un système linéaire saturé nilpotent f de dimension n tels que $f^{2^n} \neq 0$ (en fait, 2^n peut être remplacé par n'importe quelle fonction récursive de n). Ceci illustre bien la différence avec les systèmes linéaires : dans ce cas toutes les propriétés mentionnées sont décidables, et si une fonction linéaire f de n variables est nilpotente alors $f^n = 0$ (cette dernière propriété est en fait vraie pour toutes les fonctions polynomiales, et même pour les fonctions analytiques).

Notons finalement que les résultats algorithmiques sur la stabilité semblent extrêmement rares dès qu'on sort du cadre des systèmes linéaires, et que des questions anciennes restent sans réponse (on pense en particulier au problème d'Arnold sur la stabilité des équations différentielles, pour lequel on ne dispose que de résultats très partiels [dCD93]). A cause de l'importance pratique de ce problème, il existe par contre une abondante littérature sur des conditions suffisantes de stabilité (voir par exemple [LM95] sur les systèmes linéaires seuillés). Au vu de nos résultats d'indécidabilité, on peut même dire que ces études de cas particuliers sont promises à un grand avenir !

Chapitre 2

Complexité algorithmique en algèbre et en géométrie

2.1 Contexte de mes travaux

2.1.1 Qu'est-ce que la complexité algorithmique ?

La complexité algorithmique, c'est l'étude des ressources nécessaires et suffisantes pour résoudre un problème donné. La principale ressource étudiée est le temps de calcul, mais ce n'est pas la seule. On peut s'intéresser par exemple à l'espace mémoire, au nombre de processeurs (en calcul parallèle)... Cette étude se divise naturellement en deux parties.

- Les bornes supérieures : il s'agit d'analyser et de concevoir des algorithmes efficaces.
- Les bornes inférieures : il s'agit de montrer qu'aucun algorithme ne peut résoudre un problème donné sans consommer un minimum de ressources.

Pour pouvoir étudier ces questions mathématiquement, on se livre à une étude asymptotique, en considérant que la taille des données n est un paramètre qui tend vers l'infini. Par exemple, il est bien connu qu'un tableau de n éléments d'un ensemble totalement ordonné peut être trié en $O(n \log n)$ comparaisons. Ce résultat est optimal car tout algorithme de tri dans un modèle de calcul à base de comparaisons doit en effectuer au moins $\Omega(n \log n)$ dans le pire des cas (et aussi en moyenne). La méthode de Gauss pour la résolution des systèmes linéaires est un autre exemple d'algorithme polynomial, c'est-à-dire dont le temps d'exécution est majoré par un polynôme en n. On dit que les problèmes correspondants sont dans P. Par contre, on ne connaît pas d'algorithme polynomial pour des problèmes NP-complets tels que le voyageur de commerce ou le sac-à-dos (voir la section 2.1.3).

Quand il s'agit de comparer les performances pratiques de deux algorithmes, ce n'est pas toujours le plus efficace asymptotiquement qui l'emporte car n peut ne pas être assez grand. Cependant les progrès technologiques finissent souvent par donner raison à la théorie. En effet, l'augmentation de la puissance de calcul et des capacités mémoires des ordinateurs permet de traiter des problèmes de taille toujours plus grande : on se déplace donc inexorablement vers la zone où l'analyse asymptotique devient pertinente. C'est ainsi que certains algorithmes jadis considérés d'intérêt "purement théorique" sont devenus efficaces en pratique.

2.1.2 Algorithmique en algèbre et en géométrie

Aux débuts de l'algorithmique on s'est intéressé surtout à des problèmes de nature combinatoire (portant sur les arbres, les graphes, les chaînes de caractères...). Ce domaine de recherche est toujours très vivant, mais on assiste à l'heure actuelle à un développement du point de vue algorithmique en dehors de ce terrain traditionnel des mathématiques discrètes. Ce mouvement concerne presque toutes les branches des mathématiques : logique, théorie des nombres, algèbre, géométrie, topologie, analyse...

Cette algorithmisation des mathématiques est particulièrement avancée en algèbre et en géométrie algébrique (réelle ou complexe). Comme en algorithmique discrète, les activités correspondantes couvrent un large spectre, de la théorie (complexité structurelle, bornes inférieures, conception d'algorithmes efficaces) aux applications (en robotique [Can88, CLO92] ou en démonstration automatique [CLO92]) en passant par le développement d'outils logiciels appropriés (systèmes et bibliothèques de calcul formel). M'étant particulièrement intéressé à la complexité structurelle, je tente de présenter ce point de vue dans la section 2.1.3.

En algèbre et en géométrie, le problème le plus fondamental (et le plus lié au problème "P=NP ?" dans les structures algébriques) est celui de l'élimination des quantificateurs. Ce problème a une histoire longue et riche (qui commence avec Tarksi dans les années 30) et je ne prétends pas lui rendre justice ici. On pourra trouver un petit historique dans [Ren92], et des algorithmes "modernes" dans [Ren92, HRS90, BPR96] (on pourra aussi consulter l'article de synthèse de van den Dries [vdD88], qui donne plusieurs preuves du théorème de Tarski et en présente quelques applications). Ces algorithmes éliminent les quantificateurs sur $\mathbb{R}$, en utilisant en particulier des idées de Grigoriev et Vorobjov [Gri88b, Gri88a]. Pour le cas complexe, on pourra consulter [FGM90]. On trouvera dans [BPR00] des résultats et des références supplémentaires sur les algorithmes de géométrie réelle utiles en robotique.

2.1.3 Complexité structurelle

En complexité structurelle on essaye de regrouper des problèmes de "difficulté semblable" dans des "classes de complexité", et d'étudier les relations entre ces classes (relations d'inclusion, d'inclusion stricte...). Par exemple, deux des plus importantes classes de complexité sont P et NP. La première est incluse dans la seconde, mais on ne sait pas démontrer que cette inclusion est stricte : c'est le fameux problème "P $\neq$ NP ?". Pour toutes les notions de complexité classique (booléenne) je renvoie à [BDG88, BDG90, Pap94].

Il s'agit donc d'une branche très fondamentale de la recherche en algorithmique, qui se donne pour but de mettre de l'ordre dans le foisonnement des problèmes rencontrés. Les résultats obtenus ne sont cependant pas sans importance pour le concepteur d'algorithme puisqu'ils lui indique quand il est inutile de s'acharner à trouver une solution efficace à un problème donné (c'est le cas en particulier si le problème en question est NP-complet). Ainsi, on ose à peine imaginer l'état de confusion épouvantable dans lequel se trouveraient des domaines tels que l'optimisation combinatoire si la notion de NP-complétude n'avait pas été inventée. Il y a mieux : en cas de réponse négative la théorie de la complexité peut proposer des approches permettant de contourner la difficulté. Par exemple, on peut abandonner la recherche d'une solution exacte pour se contenter d'une solution approchée. La théorie peut indiquer pour quels problèmes cette approche à une chance de se révéler fructueuse, et quelle qualité d'approximation on est en droit d'espérer. Dans une autre approche, on se contente d'un algorithme efficace en moyenne plutôt que dans tous les cas.

Le développement d'une complexité structurelle pour les problèmes algébriques est beaucoup plus récent, et est motivé par l'espoir de bénéfices similaires pour l'algorithmique algébrique. Le mouvement a été lancé par Blum, Shub et Smale[1] dans l'article fondamental qui introduit la machine de Turing réelle [BSS89]. La principale contribution de cet article est probablement la définition de la classe NP sur les réels, et la mise en évidence de problèmes NP-complets naturels. Comme en complexité classique, le problème "P = NP ?" est le principal problème ouvert dans le modèle de Blum-Shub-Smale et la plupart de ses variantes. On peut se poser la même question pour les nombres complexes au lieu des nombres réels, et même pour des structures arbitraires comme le fait Poizat [Goo94, Poi95]. Outre son degré maximum de généralité, on peut retenir deux points saillants dans l'approche de Poizat :

(i) la proposition d'un modèle de calcul à base de circuits qui peut remplacer les machines de [BSS89] ;

(ii) la mise en évidence de manière particulièrement claire du lien entre le problème "P=NP ?" et l'élimination des quantificateurs : pour toute structure M, $\mathrm{P}_M = \mathrm{NP}_M$ si et seulement si il est possible d'éliminer un bloc de quantificateurs existentiels en temps polynomial (en représentant la formule éliminante par un circuit au sens de M). Cette équivalence est vraie quand l'algorithme d'élimination est un algorithme au sens de la structure

[1] Il faut quand-même mentionner le modèle algébrique de Valiant [Val79a, Val82], qui est antérieur de plusieurs années. Mais il est techniquement très différent, et surtout il est conçu pour l'étude de problèmes de nature plus combinatoire (le calcul du permanent d'une matrice est un problème complet typique dans ce modèle, alors que Blum, Shub et Smale s'intéressent plutôt à des problèmes d'élimination).

[Poi95] aussi bien que quand il s'agit d'un algorithme classique [1].

Pour les réels ($M = (\mathbb{R}, +, -, \times, <, 0, 1)$) et les complexes ($M = (\mathbb{C}, +, -, \times, =, 0, 1)$), les algorithmes "modernes" cités dans la section 2.1.2 n'éliminent un bloc de quantificateurs existentiels qu'en temps simplement exponentiel. Il semble que les algorithmes réels n'utilisent en général que la représentation usuelle des formules (qui sont des circuits d'un genre particulier). L'intérêt d'un codage plus compact des formules (ou au moins des polynômes) a été plus exploré dans le cas complexe, voir par exemple [GHM+98, HM93]. Les bornes obtenues restent cependant exponentielles (le contraire serait étonnant).

Le problème "$\mathrm{P}_M = \mathrm{NP}_M$?" est ouvert dans la plupart des cas intéressants, et en particulier pour les réels et les complexes. Dans le cas général on a souvent $\mathrm{P}_N \neq \mathrm{NP}_M$ pour une raison triviale (absence d'élimination des quantificateurs). Le résultat $\mathrm{P}_{\mathbb{R}_{vs}} \neq \mathrm{NP}_{\mathbb{R}_{vs}}$ sur la structure $\mathbb{R}_{vs} = (\mathbb{R}, +, =, 0, 1)$ des réels avec addition et ordre, dû à Klaus Meer [Mee92] (voir aussi [4] pour une preuve très simple), est déjà plus intéressant. Par contre, on ne connaît aucun exemple de structure M telle que $\mathrm{P}_M = \mathrm{NP}_M$, ce qui pourrait donner aux audacieux l'envie de formuler une "hypothèse de Cook généralisée". L'inégalité $\mathrm{P} \neq \mathrm{NP}$ du cas classique, qui semble déjà si difficile à établir, n'est peut-être que la pointe de l'iceberg...

Il n'est peut-être pas inutile de préciser ici que ni Blum-Shub-Smale, ni les autres chercheurs ayant contribué à la complexité algébriques durant ces quarante dernières années ne suggèrent de stocker effectivement des nombres réels arbitraires dans la mémoire d'un ordinateur. Ils proposent simplement une *modélisation* de certaines situations, qui a des chances d'être pertinente si les opérations importantes sont les comparaisons et les opérations arithmétiques. Une telle démarche est familière aux physiciens, qui savent bien que tout modèle n'est qu'une approximation de la réalité. De la même manière, un algorithme de tri travaillant sur un ensemble totalement ordonné quelconque ne constitue qu'une abstraction commode (mais qui semble présenter moins de difficultés psychologiques). Remarquons finalement à l'attention de ceux qui restent réfractaires aux calculs dans les structures que le problème "$\mathrm{P}_M = \mathrm{NP}_M$?" peut-être vu comme un problème d'algorithmique classique. En effet, comme on l'a déjà remarqué en (ii), il est possible de se contenter d'un algorithme classique pour éliminer les quantificateurs.

J'ai consacré une grande partie de mes travaux à la complexité structurelle dans le modèle de Blum-Shub-Smale, contribuant ainsi à développer cet aspect trop longtemps négligé de la complexité algébrique. Les résultats obtenus sont décrits dans la section suivante.

2.2 Résultats obtenus

2.2.1 Le rôle des paramètres

Une machine de Turing sur une structure M peut utiliser comme paramètres (ou "constantes") un uple quelconque d'éléments de M. Ce choix est parfois controversé, mais semble très naturel si l'on souhaite développer une théorie générale de la calculabilité et de la complexité dans les structures algébriques. En effet, si on se donne une fonction réelle f calculable à l'aide d'un dispositif prenant en entrée une suite de réels, il est naturel d'exiger que la fonction obtenue à partir de f en fixant la valeur d'une des entrées reste calculable. Remarquons d'ailleurs qu'en théorie des modèles on travaille souvent avec des formules à paramètres, peut-être pour des raisons similaires.[2]

On peut se demander si l'utilisation de paramètres permet d'accélérer la résolution de certains problèmes, ou de décider des problèmes supplémentaires. Nous allons d'abord étudier cette question dans le cas des problèmes booléens, puis nous allons discuter de sa relation avec le problème "P = NP ?" sur M. Certains des résultats obtenus ont des applications à des problèmes de nature à priori très différente. Par exemple, dans le cas complexe, éliminer un uple de paramètres algébriquement indépendants (ce que font Blum, Cucker, Shub et Smale grâce à leur théorème du témoin [BCSS96]) revient à se demander quel est le comportement d'une machine pour une entrée générique ; ceci est bien utile dans les calcul de dimension (voir la section 2.2.2).

Parties booléennes

J'ai commencé mes travaux de complexité algébrique par l'étude des "parties booléennes" des classes de complexité réelle : on considère des machines réelles avec entrées booléennes, ce qui permet de comparer les classes de complexité réelles et booléennes. J'ai d'abord mené cette étude dans le cadre d'une version affaiblie du modèle de Blum-Shub-Smale dans laquelle seul un usage modéré des multiplications est possible. Ce travail a obtenu le *Machtey Award* à FOCS'93 [15]. On trouve que dans ce modèle faible, la partie booléenne du temps polynomial est P/poly (et on écrit : $\mathrm{BP}(\mathrm{P}_W) = \mathrm{P/poly}$). Ici, P/poly désigne la classe des problèmes qui peuvent être résolus en temps polynomial à l'aide d'une "fonction de conseil" de taille polynomiale (ou de manière équivalente par une famille non uniforme de circuits booléens de taille polynomiale [BDG88]). Ce résultat peut être vu comme une généralisation des travaux de Siegelmann et Sontag sur la puissance de calcul des réseaux de neurones à poids réels [SS94]. L'outil principal est un lemme, amélioré par la suite [BPR96], sur l'existence de "petits" points

[2] On apprend dans un premier cours de logique qu'un énoncé de la forme $\forall x\ F(x)$ est vrai dans une structure M si pour tout élément a de M, l'énoncé $F(a)$ est vrai dans M ; ce qui amène bien à considérer des énoncés à paramètres même si l'énoncé de départ est sans paramètre.

rationnels dans les ensembles semi-algébriques. L'étude du modèle faible a été poursuivie par Cucker, Shub et Smale [CSS94], qui montrent notamment que $\mathrm{P}_W \neq \mathrm{NP}_W$.

L'étude des parties booléennes est poursuivie dans [20] et [19] pour les machines sans multiplication. Le principal problème ouvert dans ce domaine est la détermination de la partie booléenne du temps polynomial dans le modèle BSS "complet" (les réels avec toutes les opérations arithmétiques). Le meilleur résultat dans cette direction a été obtenu par Cucker et Grigoriev [CG97], qui montrent que $\mathrm{BP}(\mathrm{P}_{\mathbb{R}}) \subseteq \mathrm{PSPACE/poly}$. En fait, ils montrent même que $\mathrm{BP}(\mathrm{PAR}_{\mathbb{R}}) = \mathrm{PSPACE/poly}$, ou PAR désigne le temps parallèle polynomial. La classe PAR joue un peu le même rôle pour les réels que PSPACE en complexité classique (pour les réels, se donner une borne sur l'espace ne sert à rien car tout calcul peut s'effectuer en espace de travail constant [Mic89]).

Il n'est pas inutile de préciser que dans ces résultats la non-uniformité provient de la possibilité de coder (et de décoder) une fonction de conseil dans les décimales d'un paramètre réel. On a parfois reproché à cette technique son manque de réalisme. Je conviens volontiers qu'en pratique on ne s'amuse effectivement jamais à extraire les décimales d'un nombre réel. L'intérêt de ces résultats est plutôt de montrer que même un algorithme "aberrant" ne tirera rien de plus qu'une non-uniformité polynomiale de paramètres réels. Ce qui ne représente somme toute qu'une ressource assez limitée et bien étudiée en complexité classique (par exemple on considère généralement que les problèmes "P = NP ?" et "P/poly = NP/poly ?" ne sont pas de nature très différente, et on sait montrer que si P/poly = NP/poly alors la hiérarchie polynomiale s'effondre [KL82] ; voir [9] pour une généralisation de ce résultat à une structure arbitraire).

Cette possibilité de codage (ou plutôt de décodage) disparaît si on supprime la relation d'ordre. Par exemple [15], la partie booléenne des réels avec addition, multiplication et égalité est incluse dans BPP (le temps polynomial probabiliste). La même preuve montre que $\mathrm{BP}(\mathrm{P}_{\mathbb{C}}) \subseteq \mathrm{BPP}$ (voir [Sma98]). Il en résulte que si $\mathrm{P}_{\mathbb{C}} = \mathrm{NP}_{\mathbb{C}}$ alors $\mathrm{NP} \subseteq \mathrm{BPP}$. L'égalité $\mathrm{P}_{\mathbb{C}} = \mathrm{NP}_{\mathbb{C}}$ serait donc très surprenante puisqu'on admet généralement en complexité booléenne que les problèmes NP-complets ne peuvent pas être résolus efficacement par des algorithmes probabilistes (d'après le paragraphe précédent, si c'était possible la hiérarchie polynomiale s'effondrerait puisque $\mathrm{BPP} \subseteq \mathrm{P/poly}$). D'autres théorèmes de transfert établissant des liens entre la complexité structurelle de différentes structures sont présentés dans le paragraphe suivant et dans la section 2.2.6. Remarquons enfin que les parties booléennes interviennent dans le "sens facile" des théorèmes de transfert entre complexité algébrique et complexité booléenne (voir la section 2.2.6).

Transfert entre structures élémentairement équivalentes

Comme on n'arrive pas à résoudre le problème "P = NP ?" pour la plupart des structures "intéressantes", on peut au moins essayer d'établir des relations entre ces différents problèmes. Dans une avancée importante, Blum, Cucker, Shub et Smale ([BCSS96], voir aussi [BCSS98]) ont établit que ce problème a la même réponse dans tous les corps algébriquement clos de caractéristique nulle.

Essayons d'abord d'esquisser les grandes lignes de la preuve. Soit K et $\mathbb{K}$ deux corps algébriquement clos de caractéristique nulle, avec $K \subseteq \mathbb{K}$. Supposons que $\mathrm{P}_K = \mathrm{NP}_K$. Alors il existe un algorithme polynomial qui résout le problème de satisfaction des formules de K (qui est NP-complet pour toute structure). Il n'est pas difficile de voir alors que le même algorithme résout le problème de satisfaction des formules de $\mathbb{K}$. Ceci est dû au fait que $\mathbb{K}$ est une extension élémentaire de K, c'est-à-dire qu'une formule du premier ordre à paramètres dans K est vraie dans $\mathbb{K}$ si et seulement si elle est vraie dans K. Nous concluons donc que $\mathrm{P}_{\mathbb{K}} = \mathrm{NP}_{\mathbb{K}}$. Réciproquement, supposons que $\mathrm{P}_{\mathbb{K}} = \mathrm{NP}_{\mathbb{K}}$: il existe alors un algorithme polynomial qui résout le problème de satisfaction des formules de $\mathbb{K}$. Si cet algorithme n'utilise que des paramètres de K, on déduit du même argument d'équivalence élémentaire que $\mathrm{P}_K = \mathrm{NP}_K$. Mais dans le cas général on ne peut pas conclure aussi directement. Cette difficulté est résolue grâce à un théorème d'élimination des paramètres, qui constitue le coeur de la preuve du théorème de transfert : la restriction à des entrées dans K d'un problème $\mathrm{P}_{\mathbb{K}}$ est toujours un problème P_K.

J'ai donné dans [13] une preuve "directe" du théorème de transfert qui évite le théorème d'élimination des paramètres. Dans cette preuve les bornes effectives d'élimination des quantificateurs [FGM90] remplacent la théorie des nombres. J'ai également donné une preuve du théorème d'élimination des paramètres basée sur les mêmes techniques, qui se sont aussi révélées utiles dans l'étude des problèmes de dimension (voir la section 2.2.2). Dans cet article les mêmes résultats sont obtenus aussi pour les corps de caractéristique positive au niveau non-uniforme, ce qui répond partiellement à une question de [BCSS96]. J'ai ensuite généralisé tous ces résultats à la hiérarchie polynomiale [2] (rappelons que P et NP sont les premiers niveaux de cette hiérarchie). Il n'est pas clair que cela puisse se faire par des méthodes de théorie des nombres.

On peut tenter de reproduire le schéma de la preuve de [BCSS96] dans un cadre beaucoup plus général, en fait dès qu'on dispose de l'élimination des quantificateurs (comme on l'a vu en 2.1.3, c'est le cadre le plus général dans lequel le problème "P = NP?" se pose). Ce point de vue est largement développé dans [9]. Soit donc T une théorie (consistante, complète) admettant l'élimination des quantificateurs. Soient M et N deux modèles de T, avec $M \subseteq N$. Comme N est une extension élémentaire de M, on voit comme dans le cas complexe que $\mathrm{P}_M = \mathrm{NP}_M$ implique $\mathrm{P}_N = \mathrm{NP}_N$. Pour la réciproque, ce sont encore les paramètres dans le "grand modèle" N qui posent problème. Nous nous sommes particulièrement intéressé à la théorie des corps réels clos. Seuls des résultats partiels ont pu être obtenus; par exemple quand R est un corps réels clos contenant $\mathbb{R}$, on ne sait pas si $\mathrm{P}_R = \mathrm{NP}_R$ implique $\mathrm{P}_\mathbb{R} = \mathrm{NP}_\mathbb{R}$. Nous avons observé que contrairement au cas complexe, il n'y a pas d'élimination des paramètres dans le cas réel (suivant la terminologie de [9], $\mathbb{R}$ n'est pas "P-stable"). Heureusement il existe une autre stratégie d'attaque, proposée par Michaux [Mic94] : il suffirait de montrer que $\mathrm{P}_\mathbb{R} = \mathrm{P}_\mathbb{R}/\mathrm{const}$ (on dirait alors suivant [9] que $\mathbb{R}$ est "P-saturé") pour obtenir le transfert de $\mathrm{P}_R = \mathrm{NP}_R$ à $\mathrm{P}_\mathbb{R} = \mathrm{NP}_\mathbb{R}$. Un problème est dans $\mathrm{P}_\mathbb{R}/\mathrm{const}$ s'il peut-être résolu en temps polynomial en utilisant pour toutes les entrées de taille n un conseil de taille k, avec k une constante dépendant uniquement du problème; de plus, le même conseil doit fonctionner également pour toutes les entrées de taille inférieure à n. Dans cet article, Michaux demande si les problèmes de $\mathrm{P}_\mathbb{R}/\mathrm{const}$ sont décidables. Nous avons répondu par l'affirmative dans [9], en montrant même que $\mathrm{PAR}_\mathbb{R}/\mathrm{const} = \mathrm{PAR}_\mathbb{R}$. Mais la question $\mathrm{P}_\mathbb{R} = \mathrm{P}_\mathbb{R}/\mathrm{const}$ reste ouverte. Notons qu'une réponse positive permettrait aussi d'obtenir un analogue du théorème de Ladner [Lad75] dans le cas réel (ce théorème affirme l'existence de problèmes de NP qui ne sont ni NP-complets ni polynomiaux, sous l'hypothèse $\mathrm{P} \neq \mathrm{NP}$). Par contre, il n'est pas difficile de voir que $\mathbb{C}$ est P-saturé en utilisant la propriété de saturation de cette structure (au sens habituel de la logique). Il en résulte un théorème de Ladner sur $\mathbb{C}$ [BDMM, Poi96, MM] ainsi qu'une preuve simple du théorème de transfert de [BCSS96] dans un cas particulier (si K est un corps algébriquement clos contenant $\mathbb{C}$, $\mathrm{P}_\mathbb{C} = \mathrm{NP}_\mathbb{C}$ est équivalent à $\mathrm{P}_K = \mathrm{NP}_K$). On peut voir également que $\mathbb{R}$ muni de sa structure d'espace vectoriel ordonné $(+, -, <, 0, 1)$ est P-stable, et donc P-saturé [9].

Remarquons finalement que pour une théorie seulement supposée modèle-complète (cela signifie que toute formule est équivalente à une formule existentielle) le problème "NP = co-NP ?" a un sens (il s'agit essentiellement de savoir si toute formule universelle peut être transformée en une formule existentielle par un algorithme polynomial). C'est le cas par exemple de la théorie des réels avec exponentielle, qui n'admet pas l'élimination des quantificateurs. L'étude des propriétés de transfert dans ce cadre est largement ouverte.

2.2.2 Complexité de problèmes géométriques

L'importance de la notion de NP-complétude vient en grande partie de ce qu'on connaît des centaines voire des milliers de problèmes NP-complets (un livre classique de la fin des années 1970 en recense déjà plus de 300 [GJ79]), qui apparaissent dans presque tous les domaines de la science et de la technologie. Pour le modèle de Blum-Shub-Smale la situation n'est pas aussi réjouissante. Il y a encore peu de temps, on ne connaissait qu'essentiellement un unique problème NP-complet sur $\mathbb{C}$. Il s'agit du problème des zéros ("Hilbert's Nullstellensatz as a decision problem", connu aussi sous le nom de "théorie existentielle des complexes"), qui consiste à décider si un système d'équations polynomiales en plusieurs variables complexes admet une solution.

J'ai exhibé dans [12] un deuxième problème NP-complet sur $\mathbb{C}$: il s'agit du calcul de la dimension des variétés algébriques. Puis j'ai montré que pour chaque entier fixé d, décider si une variété a une composante irréductible de codimension au moins d est un problème $\mathrm{NP}_{\mathbb{C}}$-complet [5] (ce résultat se généralise d'ailleurs aux ensembles constructibles, comme l'indique le titre de l'article). Par contre, si d n'est plus fixé mais fait partie des données du problème, on peut montrer que ce problème de codimension n'est probablement pas dans $\mathrm{NP}_{\mathbb{C}}$ ni dans co-$\mathrm{NP}_{\mathbb{C}}$ [5] (car décider dans $\mathrm{NP}_{\mathbb{C}}$ ou dans co-$\mathrm{NP}_{\mathbb{C}}$ si une variété a un point isolé entraînerait l'effondrement de la hiérarchie polynomiale au deuxième niveau ou la réfutation de l'hypothèse de Riemann généralisée). L'hypothèse de Riemann généralisée (GRH) intervient ici à cause du résultat suivant [2] : sous GRH, tout problème booléen de $\mathrm{NP}_{\mathbb{C}}$ est dans la classe "Arthur-Merlin" AM (avec les notations de la section 2.2.1, $\mathrm{BP}(\mathrm{NP}_{\mathbb{C}}) \subseteq \mathrm{AM}$). Ce résultat est presque optimal car bien sûr $\mathrm{BP}(\mathrm{NP}_{\mathbb{C}})$ contient NP, et on considère généralement que AM est "juste au dessus" de NP (c'est en fait une version probabiliste de NP ; on a en particulier les inclusions $\mathrm{NP} \subseteq \mathrm{AM} \subseteq \Pi_2$). Il en résulte que dans le modèle de calcul standard (booléen), les problèmes de dimension et de codimension de [12] et [5] sont dans AM.

La preuve que $\mathrm{BP}(\mathrm{NP}_{\mathbb{C}}) \subseteq \mathrm{AM}$ se fait par réduction au problème des zéros. La borne AM pour ce problème est obtenue par une méthode tout-à-fait différente des solutions proposées auparavant : il s'agit de relier l'existence de solutions dans $\mathbb{C}$ à l'existence de solutions dans $\mathbb{Z}/p\mathbb{Z}$ (un système a des solutions dans $\mathbb{C}$ si et seulement il a des solutions dans $\mathbb{Z}/p\mathbb{Z}$ pour "beaucoup" de "petits" p). Cependant la *preuve* de la correction de l'algorithme repose en partie sur des techniques classiques en théorie de l'élimination. Ce travail a obtenu le prix du meilleur article de l'année 1996 dans *Journal of Complexity* [16]. L'étude de telles bornes effectives a été continuée par plusieurs auteurs, en vue d'applications au modèle de calcul de Valiant [Bür98], à la complexité des équations diophantiennes [Roj99], voire d'un point de vue de géométrie arithmétique "pure" [HMPS]. Avant [16] la meilleure borne connue pour le problème des zéros (ainsi que pour les problèmes de dimension et de codimension) était PSPACE. La borne AM pour ces problèmes n'est pas trop loin de l'optimum puisqu'ils sont tous trois NP-durs. Il reste de nombreux problèmes d'élimination dont la complexité n'est connue que de manière approximative. Par exemple, on ne connaît pas de meilleure borne que PSPACE pour décider de la validité des formules prénexes sur $\mathbb{C}$ dès qu'il y a plus d'un bloc de quantificateurs (voir aussi en 2.2.4 pour des exemples de problèmes ouverts en géométrie complexe).

Dans le cas réel la situation est en partie similaire dans la mesure ou on ne connaît que peu de problèmes NP-complets. Le problème $\mathrm{NP}_{\mathbb{R}}$-complet "canonique" est toujours (comme dans n'importe quelle structure) le problème de décision des formules existentielles, ou si l'on préfère le problème des zéros (réels), ou encore le problème 4FEAS d'existence de racines réelles pour un polynôme de degré au plus quatre, problème auquel le problème des zéros réels se réduit facilement ; voir [CR92] pour quelques autres exemples. J'ai montré dans [10] que le calcul de la dimension des ensembles semi-algébriques est $\mathrm{NP}_{\mathbb{R}}$-complet, ce qui constitue donc l'analogue réel du résultat de [12] sur la dimension complexe. L'analogie avec le cas complexe s'arrête là car je n'ai pas grand chose à dire sur la complexité booléenne des problèmes de géométrie réelle. En effet, on ne voit pas du tout par quoi remplacer les réductions modulo p. La complexité booléenne de 4FEAS ou du calcul de la dimension réelle reste donc à déterminer précisément : la meilleure borne supérieure est PSPACE, alors que la meilleure borne inférieure n'est que NP. Notons cependant qu'en utilisant un algorithme "efficace" d'élimination des quantificateurs [Ren92, HRS90, BPR96], on peut calculer la dimension d'un semi-algébrique de $\mathbb{R}^n$ défini par k inéquations de degré d en temps $(kd)^{O(n^2)}$. L'exposant $O(n^2)$ vient de la réduction qui montre que le problème de la dimension réelle est dans $\mathrm{NP}_{\mathbb{R}}$: cette réduction produit une formule existentielle avec $O(n^2)$ quantificateurs (la borne $(kd)^{O(n^2)}$ peut être obtenue de manière plus directe, voir [Vor98], section 3). Des considérations similaires permettent naturellement d'obtenir des bornes sur la complexité déterministe des problèmes de $\mathrm{NP}_{\mathbb{C}}$. Par exemple j'ai donné dans [12] un algorithme déterministe de calcul de la dimension complexe de complexité algébrique $(kd)^{O(n)}$. On doit à Chistov [Chi96] un algorithme de complexité comparable pour le modèle de calcul booléen. Cet auteur a ensuite étudié la complexité du calcul des dimensions composantes irréductibles [Chi97].

Pour calculer la dimension réelle en temps $(kd)^{O(n)}$ il suffirait de trouver une réduction produisant une formule existentielle avec $O(n)$ quantificateurs (mais rien ne prouve que ce soit la bonne approche...). Une telle borne pourrait être considérée comme optimale en l'état actuel de nos connaissances puisque c'est la meilleure borne connue pour le problème (plus simple) de décision des formules existentielles. Bien sûr on peut aussi se demander si la dimension peut être calculée en temps polynomial ; d'après ce qui précède, cela est possible si et seulement si $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$.

2.2.3 Polynômes creux

Dans une direction assez différente (plus algébrique que géométrique) j'ai montré avec Cucker et Smale que les racines entières d'un polynôme creux en une variable à coefficients entiers peuvent être calculées en temps polynomial [3]. Dans ce résultat comme dans le reste de la section 2.2.3, il n'est donc question que de complexité classique (et non pas algébrique). L'outil principal est un algorithme de détermination du signe d'un polynôme creux en un point entier. Dans cet article nous posions le problème du calcul des racines rationnelles d'un polynôme creux. Notre travail a inspiré deux articles de H. W. Lenstra, Jr. [Len98a, Len98b]. Dans le premier article, Lenstra obtient un analogue pour les corps valués (comme par exemple le corps $\mathbb{Q}_p$ des nombres p-adiques) de la règle de Descartes (un polynôme de $k+1$ termes a au plus $2k$ racines réelles non nulles). Dans le second article, il répond finalement à notre question en montrant que pour un polynôme creux toutes les racines dans $\mathbb{Q}$ (ou même dans une extension finie de $\mathbb{Q}$) peuvent être calculées en temps polynomial. Bien que [Len98b] utilise des techniques tout-à-fait différentes de [Len98a], on est conduit naturellement à se poser la question du calcul en temps polynomial (ou au moins de l'existence) des racines dans $\mathbb{Q}_p$ d'un polynôme creux à coefficients entiers. Cette question peut être rapprochée du problème ouvert bien connu sur l'existence de racines réelles pour un polynôme creux, et il semble bien qu'il doive être considéré comme ouvert également. Dans tout ce domaine les problèmes ouverts sont d'ailleurs surabondants. Par exemple, on ne sait pas s'il est possible de calculer en temps polynomial le signe en un point rationnel d'un polynôme creux à coefficients entiers en une variable ; et on ne sait même pas décider si un polynôme à coefficients entiers en deux variables a une racine entière (quand le nombre de variables est suffisamment grand, le problème devient naturellement indécidable [Mat93]).

2.2.4 D'autres problèmes de géométrie ?

Les résultats obtenus dans la section 2.2.2 portent tous peu ou prou sur des problèmes de dimension, et sont tous obtenus par réduction au problème des zéros. Il serait intéressant d'obtenir des résultats similaires pour des problèmes de nature différente, en particulier pour des problèmes qui ne semblent pas pouvoir se réduire au problème des zéros. Je souhaite ici attirer l'attention sur deux problèmes de géométrie complexe : le calcul du degré d'une variété, et le problème du point isolé. Comme la quasi-totalité des problèmes de géométrie réelle ou complexe, ces deux problèmes peuvent être résolus en espace polynomial (ou en temps parallèle polynomial si on travaille avec un modèle de calcul algébrique), mais on ne sait pas les mettre dans une classe de complexité plus petite.

Il existe plusieurs notions de degré. On utilisera ici la définition suivante : le degré d'une variété $V \subseteq \mathbb{C}^n$ de dimension d est le cardinal (fini) de $V \cap E$, où E est un espace affine "générique" de dimension $n-d$. Le calcul du degré est donc essentiellement un problème de comptage. Des classes de complexité spécifiques ont été introduites pour capturer la complexité de tels problèmes. Dans ce contexte, la classe #P de Valiant [Val79b, Val79c] (encore lui !) joue le même rôle que NP dans l'étude des problèmes de décision. Il existe aussi un analogue de la hiérarchie polynomiale : c'est la hiérarchie de comptage de Wagner [Wag86]. Les problèmes de cette hiérarchie peuvent tous être résolus en espace polynomial. Il est donc naturel de se demander si le degré est dans la hiérarchie de comptage. Comme pour le problème des zéros on s'intéressait à l'existence de solutions dans $\mathbb{Z}/p\mathbb{Z}$, il serait tentant de compter ici le nombre de solutions. Malheureusement, il ne semble pas possible d'obtenir des bornes effectives menant à des résultats intéressants du point de vue de la complexité. En fait, je ne sais même pas compter le nombre de racines complexes distinctes d'un polynôme creux à coefficients entiers.

Le problème du point isolé est le suivant : décider si 0 est un point isolé d'une variété $V \subseteq \mathbb{C}^n$. Comme pour le degré, on ne voit pas comment se ramener au problème des zéros. A la recherche d'une nouvelle réduction à un problème de théorie des nombres, j'ai montré dans un manuscript non publié que les deux propriétés suivantes sont équivalentes (on suppose que $0 \in V$) :

(i) 0 est un point isolé de V ;

(ii) l'ensemble des nombres premiers p tels que V a un point dans $(p\mathbb{Z}_p)^n - \{0\}$ est fini.

Ici $\mathbb{Z}_p$ est l'anneau des entiers p-adiques. On est cependant très loin d'un résultat algorithmique. D'abord, il faudrait obtenir des bornes sur la taille des p à utiliser. Ensuite, étant donné un nombre premier p, il faudrait être capable de décider si V a un point dans $(p\mathbb{Z}_p)^n - \{0\}$. Il existe bien quelques travaux sur la complexité algorithmique des systèmes d'équations sur les p-adiques (voir par exemple [Bis91, MW96, MW97]). Cependant les algorithmes connus ne conduisent qu'à des bornes simplement exponentielles sur le temps de calcul (cela reste vrai même pour des polynômes creux en une variable, voir la section 2.2.3). Par contre, dans le cas du problème des zéros il suffit de "deviner" une solution dans $(\mathbb{Z}/p\mathbb{Z})^n$.

Je termine cette section 2.2.4 par une digression sur la complexité algébrique du problème du point isolé (alors qu'il n'était question jusqu'à présent que de complexité booléenne). Si on travaille sur $\mathbb{R}$ au lieu de $\mathbb{C}$, le fait que 0 est isolé dans V peut s'exprimer par la formule évidente :

$$\exists \epsilon > 0 \ \forall x_1, \ldots, x_n \ \left(\sum_{i=1}^{n} x_i^2 \leq \epsilon \wedge (x_1, \ldots, x_n) \in V \rightarrow \bigwedge_{i=1}^{n} x_i = 0 \right). \tag{2.1}$$

Ceci montre que sur les réels, le problème du point isolé est dans le deuxième niveau de la hiérarchie polynomiale (Σ_2) ; de plus, on en déduit un algorithme simplement exponentiel par élimination des quantificateurs. Par contre, sur les complexes on ne sait pas si ce problème est dans la hiérarchie polynomiale, et les algorithmes "efficaces" pour le résoudre sont beaucoup moins directs [GH91, GH93] (il est à noter que l'algorithme "polynomial" de [GH93] doit en fait être considéré comme exponentiel suivant nos conventions ; il est juste un peu moins exponentiel que celui de [GH91]). Tout ça parce qu'il n'existe pas de formule telle que (2.1) sur les complexes... Mais est-on bien sûr qu'une telle formule n'existe pas ? Ceci n'a en fait rien d'évident, et il semble qu'il s'agisse d'un problème ouvert, probablement assez difficile (dernière minute : ce problème est maintenant résolu, voir en section 2.2.5). Remarquons que 0 est isolé dans V si et seulement si $V - \{0\}$ est fermé ; il suffirait donc de disposer d'une formule qui reconnaisse les ensembles fermés parmi les constructibles.

2.2.5 Définissabilité

Nous avons conjecturé dans [8] qu'être fermé n'est pas une propriété définissable des ensembles définissable de $\mathbb{C}^n$, et ceci dès que $n \geq 2$; c'est-à-dire qu'il n'existe pas d'énoncé F dans le langage des anneaux $(+, -, \times, =, 0, 1)$ augmenté d'un prédicat n-aire I_n qui vérifie la propriété suivante : pour tout ensemble définissable $X \subseteq \mathbb{C}^n$, X est fermé si et seulement si l'énoncé F est vrai quand on interprète I_n par la fonction caractéristique de X (rappelons que par élimination des quantificateurs les ensembles définissables de $\mathbb{C}^n$ sont exactement les ensembles constructibles, c'est-à-dire les ensembles définissables par une combinaison booléenne d'équations polynomiales). Par contre nous avons vu avec (2.1) que la propriété "0 est un point isolé" est définissable pour les ensembles définissables de $\mathbb{R}^n$, et de même il est immédiat que les fermés de $\mathbb{R}^n$ sont définissables. Dans [8] nous donnons quelques exemples de propriétés définissables dans $\mathbb{C}^n$ (par exemple la dimension), et nous montrons que la connexité des ensembles définissables de $\mathbb{C}^2$ n'est pas définissable. Nous donnons trois preuves de ce résultat, toutes basées sur une réduction de la la parité des sous-ensembles finis de $\mathbb{C}$ à la connexité. La première preuve est basée sur une réduction du cas réel au cas complexe. Dans le cas réel, on savait déjà que la parité n'est pas définissable [BDLW98]. Depuis quelques années, les questions de ce type sont l'objet d'investigations intensives à cause de leur intérêt pour les bases de données spatiales (ou plus généralement pour les bases de données "contraintes", voir aussi par exemple [BL96, BL97, GS97]). La deuxième preuve est basée sur un résultat important de complexité[3] : la fonction booléenne Parité n'est pas dans AC^0 (voir par exemple [Hås87] ou [LV97]). Notre troisième preuve est une preuve "directe" de théorie des modèles.

Enfin, nous montrons que la technique classique dans ce domaine, qui consiste à se ramener à un problème sur les structures finies (voir par exemple notre réduction de la parité à la connexité) ne permet pas de montrer que les fermés de $\mathbb{C}^n$ ne sont pas définissables. Notre conjecture vient finalement d'être démontrée dans [6]. La démonstration est assez proche dans l'esprit de la troisième preuve de [8] dans la mesure où toutes deux reposent sur la construction d'une théorie complète "judicieuse". Dans le cas de [6] il s'agit de la "limite des théories de courbes génériques", et cette théorie (ainsi que sa construction) pourraient être d'intérêt plus général. On obtient aussi en même temps (toujours dès que $n \geq 2$) la non-définissabilité parmi les variétés de $\mathbb{C}^n$ des variétés ayant un point isolé, ainsi que de celles dont l'origine est un point isolé. Il en va de même pour les variétés irréductibles. Ce dernier résultat reste vrai pour les réels, mais la preuve [7] est beaucoup plus simple. En effet, dans le cas réel il suffit de se servir dans le catalogue des théories complètes déjà répertoriées (la théorie adéquate est celle

[3]Certaines recherches contemporaines en logique (en particulier en théorie des modèles finis [EF95]) sont motivées par des applications possibles à la complexité ; ici on voit donc qu'inversement, la complexité peut venir au secours de la logique !

des réels avec exponentielle).

2.2.6 Théorèmes de transfert : de la complexité booléenne à la complexité algébrique

Dans sa préface au livre de Blum, Cucker, Shub et Smale [BCSS98], Richard Karp écrit : "It is interesting to speculate as to whether the questions of whether $P_{\mathbb{R}} = NP_{\mathbb{R}}$ and whether $P_{\mathbb{C}} = NP_{\mathbb{C}}$ are related to each other and to the classical P versus NP question, which in the present setting asks whether $P_{\mathbb{Z}_2} = NP_{\mathbb{Z}_2}$. I am inclined to think that the three questions are very different and need to be attacked independently." On voudrait presque croire que Karp à raison, ce qui, faute de pouvoir résoudre la version classique du problème "P=NP ?", laisserait un espoir d'en résoudre les versions algébriques (quitte à en tirer ensuite des enseignements permettant de progresser sur le front classique, comme Valiant l'avait suggéré à propos de son modèle de calcul algébrique... mais nous n'en sommes pas encore là). Cependant, j'ai montré avec mon étudiant Hervé Fournier que si l'on calcule sur les réels sans multiplication (avec addition et ordre uniquement), les problèmes "P=PSPACE ?" [11] et "P=NP ?" [4] sont en fait équivalents à leurs versions classiques. Les premiers résultats de ce type apparaissent dans [Poi95], où on montre par exemple que le problème "P=NP ?" sur les réels munis uniquement de leur relation d'ordre est équivalent au problème classique.

Ces développements attirent au bout du compte l'attention sur le cas complexe, car on peut espérer montrer $P_{\mathbb{C}} \neq NP_{\mathbb{C}}$ sans avoir à montrer au passage que $P \neq NP$ ou $P \neq PSPACE$ (ou du moins, on ne voit pas comment les techniques de [4] et [11], qui utilisent l'ordre de manière cruciale, s'étendraient au cas complexe). Des stratégies d'attaque du problème "$P_{\mathbb{C}} = NP_{\mathbb{C}}$?" ont été proposées par Shub et Smale [SS96], mais elles n'ont pas permis d'aboutir jusqu'à présent.

Il n'y a pas de tel théorème de transfert pour le problème "P = NP ?" sur les réels avec addition et égalité car, comme on l'a remarqué dans la section 2.1.3, il est possible de répondre très simplement à cette question (par la négative). Par contre, nous avons obtenu dans [4] plusieurs théorèmes de transfert sur la hiérarchie polynomiale de cette structure. Nous avons par exemple montré que l'effondrement de la hiérarchie polynomiale sur $\mathbb{R}_{vs}$ est équivalent à son effondrement dans le cas classique. Il semble possible d'obtenir des théorèmes de transfert pour la hiérarchie polynomiale d'autres structures. Ce sera je l'espère l'objet d'une future publication.

Bibliographie

[ABDCBH93] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. In *Proc. 34th IEEE Symp. on Foundations of Computer Science*, pages 292–301, 1993.

[AMP95] E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise constant derivatives. *Theoretical Computer Science*, 138 :35–66, 1995.

[Bar97] P. Bartlett. For valid generalization, the size of the weights is more important than the size of the network. In *Advances in Neural Information Processing Systems 9*, 1997.

[BCSS96] L. Blum, F. Cucker, M. Shub, and S. Smale. Algebraic settings for the problem "P≠NP ?". In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis*, volume 32 of *Lectures in Applied Mathematics*, pages 125–144. American Mathematical Society, 1996.

[BCSS98] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

[BDG88] J.L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1988.

[BDG90] J.L. Balcázar, J. Diáz, and J. Gabarró. *Structural Complexity II*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1990.

[BDLW98] M. Benedikt, G. Dong, L. Libkin, and L. Wong. Relational expressive power of constraint query languages. *Journal of the ACM*, pages 1–34, 1998.

[BDMM] S. Ben-David, K. Meer, and C. Michaux. A note on non-complete problems in $NP_{\mathbb{R}}$. To appear in *Journal of Complexity*.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4) :929–965, 1989.

[Bis91] E. E. Bishop. *Machines and complexity over the p-adic numbers*. PhD thesis, University of California at Berkeley, 1991.

[BL96] M. Benedikt and L. Libkin. On the structure of queries in constraint query languages. In *Proc. 11th IEEE Symposium on Logic in Computer Science*, pages 25–34, 1996.

[BL97] M. Benedikt and L. Libkin. Languages for relational databases over interpreted structures. In *Proc. 16th ACM Symposium on Principles of Database Systems*, pages 87–98, 1997.

[Bou99] O. Bournez. *Complexité algorithmique des systèmes dynamiques continus et hybrides*. PhD thesis, Ecole Normale Supérieure de Lyon, 1999.

[BPR96] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43(6) :1002–1045, 1996.

[BPR00] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets on a variety. *Journal of the AMS*, 13 :55–82, 2000.

[BSS89] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers : NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21(1) :1–46, July 1989.

[Bür98] P. Bürgisser. Cook's versus Valiant's hypothesis. To appear in *Theoretical Computer Science*, 1998.

[Can88] J. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, Massachusetts, 1988.

[CG97] F. Cucker and D. Grigoriev. On the power of real Turing machines with binary inputs. *SIAM Journal on Computing*, 26(1) :243–254, 1997.

[Chi96] A. Chistov. Polynomial-time computation of the dimension of algebraic varieties in zero-characteristic. *Journal of Symbolic Computation*, 22 :1–25, 1996.

[Chi97] A. Chistov. Polynomial-time computation of the dimensions of components of algebraic varieties in zero-characteristic. *Journal of Pure and Applied Algebra*, 117/118 :145–175, 1997.

[CLO92] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer, New York, 1992.

[CR92] F. Cucker and F. Rosselló. On the complexity of some problems for the Blum, Shub & Smale model. In *Proceedings of Latin'92*, LNCS 583, pages 117–129. Springer-Verlag, 1992.

[CSS94] F. Cucker, M. Shub, and S. Smale. Separations of complexity classes in Koiran's weak model. *Theoretical Computer Science*, 133(1) :3–14, 1994.

[dCD93] N. C. A. da Costa and F. A. Doria. On Arnold's Hilbert symposium problems. In *Computational Logic and Proof Theory (Brno, 1993)*, LNCS 713, pages 152–158, Berlin, 1993. Springer.

[EF95] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer, 1995.

[FGM90] N. Fichtas, A. Galligo, and J. Morgenstern. Precise sequential and parallel complexity bounds for quantifier elimination over algebraically closed fields. *Journal of Pure and Applied Algebra*, 67 :1–14, 1990.

[GH91] M. Giusti and J. Heintz. Algorithmes – disons rapides – pour la décomposition d'une variété algébrique en composantes irréductibles et équidimensionnelles. In T. Mora and C. Traverso, editors, *Effective Methods in Algebraic Geometry (MEGA'90)*, Progress in Mathematics 94, pages 169–194. Birkhäuser, 1991.

[GH93] M. Giusti and J. Heintz. La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial. In *Computational Algebraic Geometry and Commutative Algebra (Cortona, 1991)*, pages 216–256. Sympos. Math. XXXIV, Cambridge University Press, 1993.

[GHM+98] M. Giusti, J. Heintz, J.E. Morais, J. Morgenstern, and L.M. Pardo. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124 :101–146, 1998.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness.* Freeman, 1979.

[GJ95] P. Goldberg and M. Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18 :131–148, 1995.

[Goo94] J. B. Goode. Accessible telephone directories. *Journal of Symbolic Logic*, 59(1) :92–105, 1994.

[Gri88a] D. Grigoriev. Complexity of deciding Tarksi algebra. *Journal of Symbolic Computation*, 5 :65–208, 1988.

[Gri88b] N. Grigoriev, D.and Vorobjov. Solving systems of polynomial inequalities in subexponential time. *Journal of Symbolic Computation*, 5 :37–64, 1988.

[GS97] S. Grumbach and J. Su. Queries with arithmetical constraints. *Theoretical Computer Science*, 173 :151–181, 1997.

[Hås87] J. Håstad. *Computational limitations for small depth circuits.* MIT Press, 1987.

[HM93] J. Heintz and J. Morgenstern. On the intrinsic complexity of elimination theory. *Journal of Complexity*, 9 :471–498, 1993. preprint : Technical Report 93-17, Laboratoire I3S, Université de Nice – Sophia Antipolis.

[HMPS] K. Hägele, J. E. Morais, L. M. Pardo, and M. Sombra. On the intrinsic complexity of the arithmetic Nullstellensatz. To appear in *Journal of Pure and Applied Algebra.*

[HRS90] J. Heintz, M.-F. Roy, and P. Solerno. Sur la complexité du principe de Tarski-Seidenberg. *Bulletin de la Société Mathématique de France*, 118 :101–126, 1990.

[HW87] D. Haussler and E. Welzl. ϵ-nets and simplex range queries. *Discrete and Computational Geometry*, 2(2) :127–151, 1987.

[KL82] R. Karp and R. Lipton. Turing machines that take advice. *L'Enseignement Mathématique*, 28 :191–209, 1982.

[KM97a] M. Karpinski and A. Macintyre. Approximating the volume of general Pfaffian bodies. In *Structures in logic and computer science*, LNCS 1261, pages 162–173, Berlin, 1997. Springer.

[KM97b] M. Karpinski and A. Macintyre. Approximating volumes and integrals in o-minimal and p-minimal theories, 1997. preprint.

[KM97c] M. Karpinski and A. Macintyre. Polynomial bounds for VC dimension of sigmoidal and general Pfaffian neural networks. *Journal of Computer and System Sciences*, 54 :169–176, 1997.

[KV94] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

[Lad75] R. Ladner. On the structure of polynomial time reducibility. *Journal of the ACM*, 22 :155–171, 1975.

[Len98a] H. W. Lenstra. Finding small degree factors of lacunary polynomials. preprint, 1998.

[Len98b] H. W. Lenstra. On the factorization of lacunary polynomials. preprint, 1998.

[LM95] D. Liu and A. N. Michel. *Dynamical systems with saturation nonlinearities : Analysis and design*. Lecture Notes in Control and Information Sciences 195. Springer, London, 1995.

[LT91] M. Ledoux and M. Talagrand. *Probability in Banach Spaces*. Springer Verlag, 1991.

[LV97] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications (second edition)*. Graduate Texts in Computer Science. Springer, 1997.

[Mat93] Y. Matiyasevich. *Hilbert's tenth problem*. The MIT Press, 1993.

[Mee92] K. Meer. A note on a $P{\neq}NP$ result for a restricted class of real machines. *Journal of Complexity*, 8 :451–453, 1992.

[Mic89] C Michaux. Une remarque à propos des machines sur $\mathbb{R}$ introduites par Blum, Shub et Smale. *C. R. Acad. Sci. Paris*, 309, Série I :435–437, 1989.

[Mic94] C. Michaux. P$\neq$NP over the nonstandard reals implies P$\neq$NP over $\mathbb{R}$. *Theoretical Computer Science*, 133 :95–104, 1994.

[MM] G. Malajovich and K. Meer. On the structure of $\mathrm{NP}_{\mathbb{C}}$. To appear in the *SIAM Journal on Computing*.

[MW96] M. Maller and J. Whitehead. Computational complexity over the 2-adic numbers. In J. Renegar, M. Shub, and S. Smale, editors, *The Mathematics of Numerical Analysis (Park City, Utah, 1995)*, pages 513–521. American Mathematical Society, 1996.

[MW97] M. Maller and J. Whitehead. Computational complexity over the p-adic numbers. *Journal of Complexity*, 13(2) :195–207, 1997.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Poi95] B. Poizat. *Les Petits Cailloux*. Nur Al-Mantiq Wal-Ma'rifah **3**. Aléas, Lyon, 1995.

[Poi96] B. Poizat. Gonflette dans les modèles ω-saturés. Exposé au groupe de travail LIP-IGD "Complexité Algébrique", Lyon, 1996.

[Ren92] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. parts I, II, III. *Journal of Symbolic Computation*, 13(3) :255–352, March 1992.

[Roj99] J. Maurice Rojas. On the complexity of diophantine geometry in low dimensions. In *Proc. 31st ACM Symposium on Theory of Computing*, 1999.

[Sie99] H. Siegelmann. *Neural Networks and Analog Computation. Beyond the Turing Limit.* Progress in Theoretical Computer Science. Birkhäuser, 1999.

[Sma98] S. Smale. On the P=NP problem over the complex numbers. Lecture given at the MSRI workshop on Complexity of Continuous and Algebraic Mathematics, November 1998. Lecture on video at www.msri.org.

[Son95] E. D. Sontag. From linear to nonlinear : some complexity comparisons. In *Proc. IEEE Conference on Decision and Control*, pages 2916–2920, 1995.

[Son96] E. D. Sontag. Interconnected automata and linear systems : A theoretical framework in discrete time. In *Hybrid Systems III : Verification and Control*, pages 436–448. Springer, 1996.

[SS94] H. T. Siegelmann and E. D. Sontag. Analog computation via neural networks. *Theoretical Computer Science*, 131(2) :331–360, 1994.

[SS96] M. Shub and S. Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "P=NP". *Duke Mathematical Journal*, 81(1) :47–54, 1996.

[Val79a] L. G. Valiant. Completeness classes in algebra. In *Proc. 11th ACM Symposium on Theory of Computing*, pages 249–261, 1979.

[Val79b] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8 :181–201, 1979.

[Val79c] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8 :410–421, 1979.

[Val82] L. G. Valiant. Reducibility by algebraic projections. In *Logic and Algorithmic (an International Symposium held in honour of Ernst Specker)*, pages 365–380. Monographie n^o 30 de L'Enseignement Mathématique, 1982.

[Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27 :1134–1142, 1984.

[Vap82] V. Vapnik. *Estimation of dependencies based on empirical data*. Springer, New-York, 1982.

[Vap95] V. N. Vapnik. *The nature of statistical learning theory.* Springer, New York, 1995.

[VC71] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies to their probabilities. *Theory Probab. Apl.*, 16 :264–280, 1971.

[vdD88] L. van den Dries. Alfred Tarski's elimation theory for real closed fields. *Journal of Symbolic Logic*, 53(1) :7–19, 1988.

[Vor98] N. Vorobjov. Computing local dimension of a semialgebraic set. In *Proc. 30th ACM Symposium on Theory of Computing*, pages 483–487, 1998.

[Wag86] K. W. Wagner. Some observations on the connection between counting and recursion. *Theoretical Computer Science*, 47 :131–147, 1986.

Publications personnelles

Complexité Algébrique

[1] P. Koiran. Circuits versus trees in algebraic comlexity. To appear in *Proceedings of STACS 2000.*

[2] P. Koiran. Elimination of parameters in the polynomial hierarchy. *Theoretical Computer Science*, 215 :289-304, 1999.

[3] F. Cucker, P. Koiran and S. Smale. A polynomial-time algorithm for diophantine equations in one variable. *Journal of Symbolic Computation*, 27 :21-29, 1999.

[4] H. Fournier and P. Koiran. Lower bounds are not easier over the reals : inside PH. LIP Research Report 99-05, Ecole Normale Supérieure de Lyon, 1999.

[5] P. Koiran. The complexity of local dimensions for constructible sets. LIP Research Report 99-05, Ecole Normale Supérieure de Lyon, 1999. To appear in *Journal of Complexity.*

[6] O. Chapuis, E. Hrushovski, P. Koiran, et B. Poizat. La limite des théories de courbes génériques. Prépublication, 1999.

[7] P. Koiran. On defining irreducibility. e-Print Logic Archive, http ://xxx.lanl.gov, July 1999.

[8] O. Chapuis and P. Koiran. Definability of geometric properties in algebraically closed fields. *Mathematical Logic Quarterly*, 45(4) :533-550, 1999.

[9] O. Chapuis and P. Koiran. Saturation and stability in the theory of computation over the reals. *Annals of Pure and Applied Logic*, 99 :1-49, 1999.

[10] P. Koiran. The real dimension problem is $\mathrm{NP}_{\mathbb{R}}$-complete. *Journal of Complexity*, 15 :227-238, 1999.

[11] H. Fournier and P. Koiran. Are lower bounds easier over the reals ? In *Proc. 30th ACM Symposium on Theory of Computing*, pages 507-513, 1998.

[12] P. Koiran. Randomized and deterministic algorithms for the dimension of algebraic varieties. In *Proc. 38th IEEE Symposium on Foundations of Computer Science*, pages 36–45, 1997.

[13] P. Koiran. Elimination of constants from machines over algebraically closed fields. *Journal of Complexity*, 13(1) :65–82, 1997.

[14] F. Cucker, P. Koiran, and M. Matamala. Complexity and dimension. *Information Processing Letters*, 62 :209–212, 1997.

[15] P. Koiran. A weak version of the Blum, Shub & Smale model. *Journal of Computer and System Sciences*, 54 :177–189, 1997. Version préliminaire : *Proc. 34th IEEE Symposium on Foundations of Computer Science* (Machtey Award), 1993.

[16] P. Koiran. Hilbert's Nullstellensatz is in the polynomial hierarchy. *Journal of Complexity*, 12(4) :273–286, 1996. Prix du meilleur article de l'année. Version longue : DIMACS report 96-27 (http ://dimacs.rutgers.edu/).

[17] P. Koiran. Approximating the volume of definable sets. In *Proc. 36th IEEE Symposium on Foundations of Computer Science*, pages 134–141, 1995.

[18] F. Cucker, M. Karpinski, P. Koiran, T. Lickteig, and K. Werther. On real Turing machines that toss coins. In *Proc. 27th ACM Symposium on Theory of Computing*, pages 335–342, 1995.

[19] F. Cucker and P. Koiran. Computing over the reals with addition and order : Higher complexity classes. *Journal of Complexity*, 11 :358–376, 1995.

[20] P. Koiran. Computing over the reals with addition and order. *Theoretical Computer Science*, 133(1) :35–48, 1994.

[21] M. Cosnard and P. Koiran. Relations between models of parallel abstract machines. In *Proceedings of the Heinz Nixdorf Symposium,, Paderborn, November 1992*, Lecture Notes in Computer Science, pages 37–46. Springer-Verlag, 1993.

Calculabilité et Connexionisme

[22] V. Blondel, O. Bournez, P. Koiran and J. Tsitsiklis. The stability of saturated linear dynamical systems is undecidable. To appear in *Proceedings of STACS 2000.*

[23] V. Blondel, O. Bournez, P. Koiran, C. Papadimitriou and J. Tsitsiklis. Deciding stability and mortality of piecewise affine dynamical systems. LIP research report 1999-05.

[24] P. Koiran and C. Moore. Closed-form analytic maps in one and two dimensions can simulate Turing machines. *Theoretical Computer Science*, 210(1) :217-223, 1999.

[25] P. Koiran and E. D. Sontag. Vapnik-Chervonenkis dimension of recurrent neural networks. *Discrete Applied Mathematics*, 86(1) :63-80, 1998. Version préliminaire dans *Proc. 3rd European Workshop on Computational Learning Theory*, LNCS 1208, pages 223–237. Springer, 1997.

[26] P. Koiran and E. D. Sontag. Neural networks with quadratic VC dimension. *Journal of Computer and System Sciences*, 54 :190–198, 1997. Version préliminaire dans *Advances in Neural Information Processing Systems 8* (NIPS'95), MIT Press, 1996.

[27] L. Gurvits and P. Koiran. Approximation and learning of convex superpositions. *Journal of Computer and System Sciences*, 55(1) :161–170, 1997. Version préliminaire dans *Proc. 2nd European Workshop on Computational Learning Theory,* LNCS 904, pages 222-236, Springer, 1995.

[28] N. Dunkin, P. Koiran, and J. Shawe-Taylor. A new incremental learning technique. In *Proc. 8th Italian Workshop on Neural Nets (WIRN'96)*, pages 112–118. Springer, 1997.

[29] P. Koiran. A family of universal recurrent networks. *Theoretical Computer Science*, 168(2) :473–480, 1996.

[30] P. Koiran and J. Shawe-Taylor. Techniques in neural learning. Technical Report 95-36, NeuroCOLT, 1995.

[31] P. Koiran. Dynamics of discrete time, continuous state Hopfield networks. *Neural Computation*, 6 :459–468, 1994.

[32] P. Koiran. Efficient learning of continuous neural networks. In *Proc. 7th ACM Conference on Computational Learning Theory*, pages 348–355, 1994.

[33] P. Koiran, M. Cosnard, and M. Garzon. Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132 :113–128, 1994.

[34] M. Cosnard, P. Koiran, and H. Paugam-Moisy. A step towards the frontier between one-hidden-layer and two-hidden layer neural networks. In *Proc. IJCNN Nagoya*, volume 3, pages 2292–2295, 1993.

[35] M. Cosnard, P. Koiran, and H. Paugam-Moisy. Complexity issues in neural network computations. In *Proceedings of Latin'92*, volume 583 of *Lecture Notes in Computer Science*, pages 530–545. Springer Verlag, 1992.

[36] M. Cosnard, P. Koiran, and H. Paugam-Moisy. Bounds on the number of units for computing arbitrary dichotomies by multilayer perceptrons. *Journal of Complexity*, 10(1), March 1994.

[37] P. Koiran. On the complexity of approximating mappings using feedforward networks. *Neural Networks*, 6 :649–653, 1993.

[38] P. Koiran. Function approximation using a partition of the input space. In *Proc. IJCNN Baltimore*, volume I, pages 883–887, 1992.

[39] P. Koiran. Approximation of mappings and application to translation invariant networks. In *Proc. IJCNN Singapore*, pages 2292–2298, 1991.

Divers

[40] P. Koiran. VC dimension in circuit complexity. In *Proc. 11th IEEE Conference on Computational Complexity*, pages 81–85, 1996.

[41] J. Cohen, V. Deschamps, and P. Koiran. Meta-level interpretation of CLP(Linear lists, Integers) enhanced by lazy evaluation and enumeration. In F. Benhamou and ed. A. Colmerauer, editors, *Constraint Logic Programming – Selected Research*, pages 457–481. The MIT Press, 1993.

[42] B. Durand, P. Koiran, and C. Mazenc. Writing a compiler with Eiffel. In *Proc. 5th International Eiffel Conference*, 1990.

Thèse

[43] P. Koiran. Puissance de calcul des réseaux de neurones artificiels. Thèse de doctorat, Ecole Normale Supérieure de Lyon, 1993.