# Contraintes, incertitudes et modèles graphiques

Thomas Schiex

AfIA
Association française
pour l'Intelligence Artificielle

# Thomas SCHIEX : Contraintes, incertitudes et modèles graphiques

*UMIAT / Statistiques et Algorithmiques pour la Biologie*
*Université de Toulouse, INRA, France*
*mia.toulouse.inra.fr*

**Thomas SCHIEX**
*EurAI Fellow en 2017*
*Thomas.Schiex@inra.fr*

## L'article qui m'a marqué

Difficile de choisir... Après plus de 30 années de recherche, fallait-il se focaliser sur les réseaux de contraintes et leurs algorithmes ? ou sur l'intégration de préférences/coûts/probabilités à ces modèles ?

J'ai donc pensé à cet article de RM Stallman et GJ Sussman intitulé *Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis* paru dans *Artificial Intelligence* en 1977. En retournant le voir, j'y ai trouvé les phrases suivantes « The system notes a contradiction [...] The antecedents of the contradictory facts are scanned [...]. A short list of relevant choice-points eliminates from consideration a large number of combinations... ». Les ingrédients du cocktail moderne du solveur SAT ou CP, combinant analyse de conflit, recherche arborescente et propagation de contraintes sont déjà là. Nous avons mis 25 ans à transformer ces idées en des outils redoutables, ayant un impact significatif sur le monde réel capables de résoudre tant d'instances significatives de problèmes pourtant NP-difficiles : transformation de conjectures mathématiques en théorèmes [9], vérification des processeurs que nous concevons [13], conception d'objets moléculaires au cœur du vivant [14]...

Qui aurait parié là-dessus il y a 41 ans, si ce n'est le petit cercle de spécialistes du domaine ? Tout comme pour la rétro-propagation du gradient (qui remonte aux années 80), il aura fallu des machines beaucoup plus puissantes, de nombreux travaux sur l'optimisation des algorithmes, des architectures et deux ingrédients cruciaux : une confrontation intense avec le monde réel (oui, là-aussi, des « data ») et des compétitions ouvertes de logiciels Open Source [1] sur ces dernières pour aboutir à des progrès marquant durablement.

Dans ce modèle très vertueux, toute bonne idée est immédiatement récupérable et améliorable par d'autres. Cela rend la gloire du vainqueur naturellement éphémère mais permet de globalement progresser bien plus rapidement.

## Mais où en sommes nous ?

Programmation par contraintes ? Algorithmique des modèles graphiques discrets ? Bioinformatique ? Les applications de l'INRA nous forcent à sortir d'un domaine de recherche spécialisé, avec un corpus de connaissances, de résultats et de critères d'évaluations partagés.

Un réseau de contraintes, ou une formule SAT permet de décrire de façon factorisée une fonction booléenne comme la conjonction de propriétés d'intérêt impliquant un petit nombre de variables. Cette fonction sépare solutions et non-solutions. Charge au solveur (CP/SAT) de trouver un extrêmum de cette fonction (une solution) et de rendre un rêve réalité : énoncer les propriétés de ce que vous cherchez et si tout se passe bien (n'oublions pas le théorème de Cook [5]), on vous donne la solution (ou on vous prouve qu'il n'y en a pas [2]).

1. Open source si cher au premier auteur de l'article mentionné au dessus (RM Stallman est un initiateur du projet GNU et de la licence GPL).
2. Oui, elle peut faire 200 To.





Comment citer ce document :
Schiex, T. (2018). Contraintes, incertitudes et modèles graphiques. Bulletin de l'AFIA (100), 40-42.

Hélas, la biologie est le royaume de l'exception, de l'incertitude et de l'information incomplète. La logique pure et les réseaux de contraintes y ont la vie dure. Mais il est possible de faire des ponts entre la logique et l'incertain.

Un modèle graphique permet de décrire une fonction comme la combinaison (la somme par exemple) de fonctions impliquant peu de variables. Il peut être stochastique (champs de Markov, réseau Bayésien) en décrivant une distribution de probabilités ou non (réseau de contraintes, valuées ou non, formules SAT pondérées...). Enoncez (ou estimez) un modèle graphique, et nous vous chercherons une solution très vraisemblable, vraiment préférée. Si tout se passe bien (Cooke est là aussi), cela sera la plus vraisemblable (et cela aura été prouvé). Un réseau de contraintes est ainsi un cas particulier de modèle graphique, booléen. Le premier est un dépositaire de l'IA symbolique, logique. Le second, souvent farci de probabilités, marginales ou jointes, est plutôt originaire du "Machine Learning" : en particulier, il s'estime à partir de données. Ces deux domaines, qui partagent tant de principes algorithmiques, sont pourtant largement isolés dans leurs conférences respectives (SAT/CP vs. UAI/CVPR par exemple).

Massivement utilisés en analyse d'image, du langage naturel ou en bioinformatique, les modèles graphiques font partie de l'attirail de base de l'apprentissage automatique [3]. Il devient possible, parfois à coût algorithmique très raisonnable (ceux liés à la résolution de problèmes d'optimisation convexe [11, 12]), d'estimer leurs paramètres et leur structure à partir de données complètes. Puis de raisonner dessus : optimisation (NP-difficile), comptage (#P-difficile).

Je vois finalement mon domaine de recherche à la jointure : généraliser aux modèles graphiques (stochastiques ou non), les capacités de raisonnement garanties développées pour les modèles booléens (logiques). Sur ce terrain, nous avons observé des progrès aussi importants que pour les prouveurs SAT. Le solveur *toulbar2* [6] peut résoudre et prouver l'optimalité d'instances (issues de problèmes d'analyse d'images) de 500 000 variables avec des domaines de taille 4 (mais aussi buter sur des instances aléatoires de taille bien plus réduite). Il est maintenant utilisé par d'autres pour résoudre des problèmes variés [3].

## Vers l'infini et au delà

En général, les prévisions sont fausses car entâchées de biais, dont ceux liés à des évènements récents, jugés cruciaux, alors qu'ils ne l'étaient finalement pas.

Il me semble néanmoins, qu'en sus des éternelles quêtes pour une meilleure efficacité, pour une compréhension de celle-ci et pour une plus grande facilité d'utilisation, il y a quatre grandes directions qui sont plus particulièrement attirantes en ce moment pour les spécialistes de l'exploration des espaces combinatoires (CSP, SAT ou modèles graphiques entre autres).

1. *Beyond NP* : la théorie de la complexité nous a longtemps invité à penser que les problèmes NP-complets définissaient des instances majoritairement intraitables. Les progrès de ces 30 dernières années montrent que le comportement exponentiel non maîtrisable peut être contenu suffisamment longtemps pour permettre de traiter bon nombre d'instances significatives de ces problèmes intraitables. Il n'est donc pas incongru de s'attaquer à la résolution d'instances significatives de problèmes placés plus haut, dans la hiérarchie polynomiale ou autour de son apex : problèmes PSPACE ou #P-complet [16]. Il est aussi raisonnable, dans ces cas, de viser des garanties plus légères

3. Cf. /www.inra.fr/mia/T/toulbar2/publication.html.





Comment citer ce document :
Schiex, T. (2018). Contraintes, incertitudes et modèles graphiques. Bulletin de l'AFIA (100), 40-42.

(approximations [17], garanties PAC – *Probably Approximately Correct* [4]), pas forcément polynomiales et en progressant sur la base d'instances dont la résolution a du sens. Il faut donc aussi collectionner des instances qui ne soient pas purement artificielles.

2. *Intégrer les avancées du Machine Learning dans nos outils* : certaines classifications réalisées par les outils de résolution sont cruciales pour l'efficacité des solveurs. On peut penser par exemple à la notion de *glue clause* utilisée dans le solveur SAT GLUCOSE [1]. Cette classe a été identifiée par tâtonnements. Peut-on utiliser l'apprentissage automatique pour identifier des classes plus pertinentes, sur la base des nombreux benchmarks accumulés. Idem pour nos heuristiques variées, nos choix de schéma de branchement, de niveau de propagation, de preprocessing... Il devrait même être possible de les intégrer comme une heuristique de choix de valeur, entraînable sur une « famille d'instances » [8].
3. *Construire sur les épaules du machine learning* : les outils du Machine Learning produisent des modèles qui peuvent décrire une fonction (régressions variées) ou relation (classification) ou d'autres objets mathématiques qu'il serait possible d'exploiter directement comme une fonction participant à la définition du problème à résoudre [2]. Mais le respect de contraintes dures n'est typiquement pas aisé à garantir dans ces modèles. En les couplant à des outils avec garanties, il deviendrait possible d'aller des données vers des décisions structurées complexes avec des garanties partielles.
4. *Faire progresser le machine Learning certifiable, explicable, équitable* : une des limitations non négligeables des réseaux de neurones convolutionnels profonds (par exemple) est leur relative incapacité à apprendre des modèles qui résistent à des manipulations. Il n'est pas difficile de les tromper en perturbant de façon minimale leur entrée [15]. Les méthodes de preuve de l'IA avec garantie devraient pouvoir fournir des outils d'analyse de ces modèles [10, 7] permettant de garantir qu'ils satisfont certaines propriétés d'intérêt.

## Références

[1] Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern sat solvers. In *International Joint Conference in AI*, volume 9, pages 399–404, 2009.

[2] Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. Neuron constraints to model complex real-world problems. In *International Conference on Principles and Practice of Constraint Programming*, pages 115–129. Springer, 2011.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[4] Supratik Chakraborty, Kuldeep S Meel, and Moshe Y Vardi. A scalable approximate model counter. In *International Conference on Principles and Practice of Constraint Programming*, pages 200–216. Springer, 2013.

[5] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[6] Barry Hurley, Barry O'Sullivan, David Allouche, George Katsirelos, Thomas Schiex, Matthias Zytnicki, and Simon De Givry. Multi-language evaluation of exact solvers in graphical model discrete





Comment citer ce document :
Schiex, T. (2018). Contraintes, incertitudes et modèles graphiques. Bulletin de l'AFIA (100), 40-42.

optimization. *Constraints*, 21(3) :413–434, 2016.

[7] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex : An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[8] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, pages 6351–6361, 2017.

[9] Oliver Kullmann. The science of brute force. *Communications of the ACM*, 2017.

[10] Nina Narodytska, Shiva Prasad Kasiviswanathan, Leonid Ryzhyk, Mooly Sagiv, and Toby Walsh. Verifying properties of binarized deep neural networks. *arXiv preprint arXiv :1709.06662*, 2017.

[11] Youngsuk Park, David Hallac, Stephen Boyd, and Jure Leskovec. Learning the network structure of heterogeneous data via pairwise exponential markov random fields. In *Artificial Intelligence and Statistics*, pages 1302–1310, 2017.

[12] Stefan Seemayer, Markus Gruber, and Johannes Söding. Ccmpred—fast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, 30(21) :3128–3130, 2014.

[13] Xujie Si, Xin Zhang, Radu Grigore, and Mayur Naik. Maximum satisfiability in software analysis : Applications and techniques. In *International Conference on Computer Aided Verification*, pages 68–94. Springer, 2017.

[14] David Simoncini, David Allouche, Simon de Givry, Ceéline Delmas, Sophie Barbe, and Thomas Schiex. Guaranteed discrete energy optimization on large protein design problems. *Journal of chemical theory and computation*, 11(12) :5980–5989, 2015.

[15] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv :1312.6199*, 2013.

[16] Seinosuke Toda. On the computational power of pp and $\oplus$p. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 514–519. IEEE, 1989.

[17] Clément Viricel, David Simoncini, Sophie Barbe, and Thomas Schiex. Guaranteed weighted counting for affinity computation : Beyond determinism and structure. In *International Conference on Principles and Practice of Constraint Programming*, pages 733–750. Springer, 2016.